Web Scale NLP: A Case Study on URL Word Breaking

Kuansan Wang Christopher Thrasher Bo-June "Paul" Hsu
ISRC, Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA

ABSTRACT

This paper uses the URL word breaking task as an example to elaborate what we identify as crucialin designingstatistical natural language processing (NLP) algorithmsfor Web scale applications: (1) rudimentary multilingual capabilities to cope with the global nature of the Web, (2) multi-style modeling to handle diverse language styles seen in the Web contents, (3) fast adaptation to keep pace with the dynamic changes of the Web, (4) minimal heuristic assumptions for generalizability and robustness, and (5) possibilities of efficient implementations and minimal manual efforts for processing massive amount of data at a reasonable cost. We first show that the state-of-the-art word breaking techniquescan be unified and generalized under the Bayesian minimum risk (BMR) framework that, using a Web scale N-gram, can meet the first three requirements. We discuss how the existing techniques can be viewed asintroducing additional assumptions to the basic BMR framework, and describe a generic yet efficient implementation called word synchronous beam search.Testing the framework and its implementation on a series of large scale experiments reveals the following. First, the language style used to build the modelplays a critical role in the word breaking task, and the most suitable for the URL word breaking task appears to be that ofthe document title where the best performance is obtained. Models created from other language styles, such as from document body, anchor text, and even queries, exhibit varying degrees of mismatch. Although all styles benefit from increasing modeling power which, in our experiments, corresponds tothe use of a higher order N-gram, the gain is most recognizable for the title model. The heuristics proposed by the prior arts do contribute to the word breaking performance for mismatched or less powerful models, but are less effective and, in many cases,lead to poorer performance than the matched model with minimal assumptions. For the matched model based on document titles, an accuracy rate of 97.18% can already be achieved using simple trigram without any heuristics.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Language models

General Terms

Algorithms, Experimentations.

Keywords

Compound splitting, multi-style language model, URL segmentation, Web scale word breaking, word segmentation.

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-3/11/03.

1. INTRODUCTION

Ever since Banko and Brill's work a decade ago [2], it has been widely observed that many natural language processing (NLP) algorithms that excel at small data sets are quickly eclipsed by their less sophisticated counterparts as the size of training data grows larger. Ample empirical evidences reported over the past decade have led to the belief that the size of training data eventually becomes more critical than the sophistication of the algorithms themselves, especially for the scale as enormous as the World Wide Web [19]. Our own deployment experience is congruent to the belief, yet we regularly find that the data size is merely one of the several important factors that contribute to the successful deployments of Web scale NLP technologies. In this paper, we further elaborate the considerations towards developing Web scale NLP techniques through an apparently straightforward NLP task, namely, URL word breaking.

Word breaking is an extensively studied topic and has become such an essential component that many commercial products, such as Microsoft Office and SQL server, have all included the functionality as part of thesoftware. Being able to break the words at proper boundaries has been critical for NLP tasks such as entity recognition[6][8], information retrieval[1][7][22], and machine translation[5][16].Aside from these traditional roots, wide adoption of many modern technologies has catapulted the need of word breaking to a new height. In particular, the specifications for Internet domain names, UNIX file/path names and URL have forced authors to name their resources without spaces, and the constraints on the short messages for mobile and micro-blogging services have prompted the inventions of many compound words and numeronyms beyond the conventional means.For example, in our word breaker case, we would like to be able to split "247moms" as "24 7 moms" rather than "247 moms" and "w84u" as "w8 4 u" ("wait for you"). To the best of our knowledge, none of the existing systems have been able to cope with these new yet frequent Web instigated languages with ease and at scale.

In this paper, we focus the discussion of the word breakingproblem on URL because URL plays a significant role in many Web applications,includingestimating the document prior for Web search [7][22] and classifying the document topics[6][14][21].Though the problemappearsstraightforward at the first glance, the design considerations for a URL word breaker share many common traits that we deem necessary for techniques to be Web-scale ready. First, thanks to the global reach of the Web, more and more text on the Web is multilingual in nature. Within a URL, it is common to find an internet domain name or the path/filename to be composed in multiple languages. The phenomenon is also observed in the search queriesand accounts for the majority of the compound word splittingerrors [1]. While it is possible for many NLP applications to be preceded by a language identification component, it is not the case for URL or query processing because the text is often too short for a language identifier to be effective. As a result, a more practical approach is to

design the algorithm to be multilingual in the first place.This is not to say, however, that one could simply build a conglomerate language model that indiscriminately includes all the text materials in one training pool. Recent studies on the language usages of Web documents have indicated that various portions of Web documents often exhibit unique language styles that their respective language models have significantly different statistics from one another [11][25]. The notion is indeed quite intuitive: the body of the document tends to use a style for formal and detail descriptions, whereas the title of a document is more of a summary style. The anchor text, embedded in the web document body, appears to be closer to the title language style, perhaps because it is often used to summarize the document it is linked to. Huang *et al*[11] show that it is critical to choose the "matched" style of language to create models for query processing, while Wang *et al*[25] demonstrate the importance of combining the various language styles for information retrieval. For this work, we are indeed surprised by the experimental results that indicate which style of the language model performs best for the URL word breaking. This is only to highlight the importance of using a matched language style in statistical modeling for Web scale NLP. Third, the Web is known for its dynamic pace in content updates, a phenomenon that requires the underlying NLP techniques to be able to adapt quickly and at large scale. For word breaker, the need of fast adaptation is patently clear as URLs that include newly created words (e.g. brand names, events, etc.) are being added at an incredible pace. Furthermore, Web is also known to contain large varieties of materials that individually seem atypical yet jointly account for the bulk of the contents. Such a "long tail" phenomenon inevitable brings lots of exceptions to any rule one would hope to characterize the Web. It is therefore not surprising that, when designing for Web scale processing, we have found systems that have fewer heuristic assumptions tend to fare better in terms of robustness and generalizability. This is indeed an observation from Banko and Brill's work [2], and seems to be confirmed again in our word breaking experiments. Finally, due to the enormous size and the dynamic nature of the Web, techniques requiring significant manual interventions or labeling efforts have questionable feasibility. As a result, studies towards minimizing manual intervention (e.g. [1]) or using entirely data-driven unsupervised approaches (e.g.[19]) have gathered strong interests as Web scale applications become more prevalent. Many unsupervised algorithms have been proposed, including those based on non-parametric Bayesian learning that have become an actively research area in machine learning. Examples for applying Bayesian learning to word breaking also abound (e.g. [3][10]), all of which utilize elaborated learning techniques that are still computationally expensive. It is not uncommon that a Web crawler has discovered millions of new URLs during the time it takes to run a single iteration of the model training. It is therefore critical for any algorithm to have some efficient implementations for Web scale deployments.

In this paper, we describe a word breaker that satisfiesall the five considerations. At the core is the formulation of the word breaking task as a Bayesian minimum risk (BMR) problem that can be tackled with language modeling techniques. We show that BMR formulation is general enoughthat can unify known effective approaches to word breaking under a single framework. Furthermore, we show an optimal solution to the BMR problem can be implemented using an efficient algorithm called word synchronous beam search, adapted from real-time speech recognition. As for the language models, we choose Microsoft Web N-gram, which is shown[26] to be multilingual, multi-style and fast adaptive to Web languages, to address the first three of the scalability criteria. Unlike the previous offers (e.g. [3]), Microsoft N-gram is particularly useful for URL word breaking because it takes a non-traditional approach in tokenization that preserves numbers in its lexicon so that the word breaker can successful detect word boundaries amidst numerals. Most importantly, we show that the N-gram based solution to the BMR problem can be realized in an unsupervised fashion without manual intervention orany compromise on the word breaking performance, making the approach amenable to Web scale deployment.

The rest of the paper is organized as follows. In Sec. 2, we first review the related efforts for word breaking. In addition to the historically text based approaches, we include the work from speech recognition in which the word boundaries are buried in the acoustic signals that explicit word segmentation is needed for producing readable transcripts. These speech oriented techniques, at the mathematical level, can be neatly unified with other text based methods into the BMR framework, as formally described in Sec. 3. A challenging aspect of implementing the BMR framework is an efficient algorithm to search for the optimal hypothesis. In Sec. 3, we also describe the beam search algorithm used in this work, which is an adaptation from the word synchronous decoder used in the modern real-time continuous speech recognizers. In Sec. 4, we present a series of large scale experiments. The results indicate the importance of the language style plays a critical role, and the heuristics or supervised learning methods that are previously shown effective remain so only for models with mismatched styles.

2. RELATED WORK

Historically, the word breaking problem manifests itself as the word segmentation problem for Asian language processing (e.g. Chinese, Japanese) where the writing systems do not include a white space to delimit characters at the word boundaries, or as the compound splitting problem for many European languages (e.g. German, Dutch) where words can be freely joined to form compound words. The early work in this area embraced the rule based approach. For example, Suzuki *et al*[23] employed a broad coverage parser for Japanese word breaking. The approach utilizes many specific features in Japanese that do not exist in other languages. Chi *et al*[6], motivated by the observation that the title and the URL of a web document play a key role in understanding the document contents, proposed a heuristic based bidirectional matching algorithm that uses the document contents to resolve ambiguities in URL word breaking. The heuristics employed in this approach, however, require a delicate training process in order to achieve acceptable accuracy. As a result, it is not clear how robust the method is and how manageable the rules it employs are for large scale applications where the training materials are often very noisy and the training process cannot be tightly controlled.However, the intuition to use document title as a main source for URL breaking is consistent with our finding which, as discussed later, suggests that the language style used to author document titles seems to be the most effective one for URL breaking.

Monz and de Rijke[18] used a lexicon based approach for compound word splitting for monolingual information retrieval. Together with rule-based stemming, they demonstrated that a simple algorithm can lead tobig gains in retrieval accuracy. Brown [5] demonstrated a similar lexicon based approach for compound

splitting for machine translation, where the lexicon is obtained from the parallel corpus used to train the translation system. Essentially, these lexicon-based approaches are a special case of the statistical method using only word unigrams described in Sec. 3.1.Another generalization of the lexicon approaches by Koehn and Knight [16]uses the geometric mean of the unigram probabilities as a measurement to assess whether to split compound words. The geometric mean heuristic remains a popular technique used in statistical machine translation. However, we show analytically in Sec. 3.1 why the heuristic does not appear mathematically optimal and in Sec. 4 that it performs poorly for Web scale URL word breaking.

As in many areas of NLP, statistical rather than rule-based approaches are receiving more attention recently. One pertinent area is to recover word boundaries from the phonemic transcription of continuous speech.Brent [4] proposed a sophisticated Bayesian framework, called modular linguistic model, which treats the word segmentation problem as a five step generation model. The idea of employing a five step generation process is, at least in theory, to allow each step to be modified independently for investigating various model assumptions. However, with the complexity of the framework come the practical issues of making the model tractable in an efficient manner. Brent introduced quite a few statistical independence assumptions, together with a dynamic programming based method to approximate the search process. Goldwater *et al*[10]have found that the word segmentation outcomes are sosensitiveto the implementations of the dynamic programming that they proposed an alternative approach where the generative model is treated as a much simpler and standard Dirichlet process and the well-known Gibbs sampling[9] is used in lieu of the approximations needed in Brent's dynamic programming. One key contributionin this work is to extend the Dirichlet process from unigram to bigram probabilities and show that the contextual dependencies play a critical role in word segmentation. This observation is largely supported by our empirical data. As noted, however, Gibbs sampling is a very slow algorithm to converge, and the complexity of extending the framework to higher order N-gram is also significant.Venkataraman[24]proposed yet another alternative in which unigram through trigram language model with the Viterbi algorithm are used to incrementally infer word boundaries. Motivated by similar works in speech recognition [17], Venkataraman showed his approach, though much simpler in formulation as well as in implementation, achieved comparable results as Brent's modular linguistic model with dynamic programming approximation. As will be elaborated in the next section, Venkataraman's approach is a special case of the BMR framework. More recently, Khaitan*et al*[15]proposed an algorithm that combines heuristic rules with data-driven methods for URL word breaking. The main insight is to treat stop word, known lexicon, or common N-gram with their respective scoring mechanisms. Specifically, instead of using the geometric mean of word frequencies as suggested by Koehn and Knight, they proposed to use the word length distribution as the cues for score normalization for words in the lexicon. Inevitably, the system involves a few free parameters, which Khaitan*et al* tuned on the English Wikipedia data. In the following sections, we show the work can be unified under the BMR framework, and the efficacy of their heuristics can be reproduced in our experiments but only for cases where the mismatched models are used.

3. BAYESIAN MINIMUM RISK FRAMEWORK FOR WORD BREAKING

We formulate our Web scale word breaker in the Bayesian minimum risk (BMR) framework: given a character string u and a risk function R, the task of a word breaker is to find

$$\hat{s} = \arg\min_s E[R(u,s)] \quad (1)$$

where$s = (w_1, w_2, w_3, \ldots w_{|s|})$ is a segmentation of $u$and $|s|$ denotes the number of words in the segmentation s. If we let $|u|$ represent the string length, i.e., number of characters in u, then the total number of possible segmentations is $2^{|u|-1}$.

3.1 MAP Decision Rule

In this work, we further adopt a uniform risk function for the framework, namely, $R(u, s) = 1$ if$s$ is not a correct split of$u$, 0 otherwise. Under the uniform risk function, it is widely known that the optimal solution to (1) is the maximum *a posteriori* decision (MAP) decision rule:

$$\begin{aligned}\hat{s} &= \arg\max_{s\in\Omega} P(s \mid u) = \arg\max_{s\in\Omega} P(u \mid s)P(s) \\ &= \arg\max_{s\in\Omega} \log P(u \mid s)P(s)\end{aligned} \quad (2)$$

where$P(u \mid s)$ and $P(s)$ are called the transformation and the segmentation prior model, respectively.Ostensibly, the search space Ω is the set of all word sequences that can produce u when all the spaces between words are removed.

In practice, however, the probability distributions for the transformation and the segmentation prior models are not known and have to be estimated from data. When we can only work with the estimated models $\widetilde{P}(s \mid u)$ and $\widetilde{P}(s)$ that bear unknown estimation errors from the real distribute $P(s \mid u)$ and $P(s)$, respectively, directly applying the MAP decision rule in the form of (2) is no longer guaranteed to be optimal. This is known as the plugged-in MAP problem and is often addressed by adjusting (2) with log-linear interpolation as

$$\hat{s} = \arg\max_s \alpha \log \widetilde{P}(u \mid s) + \log \widetilde{P}(s) \quad (3)$$

whereα, called the transformation model weight, is a parameter that can be manually tuned or automatically learned from a development data set. The idea of log-linear adjustment can be repeatedly applied when more assumptions, or "features", are included in the model. Indeed, (3) can be viewed as the simplest form of using only the estimated transformation model score $\widetilde{P}(s \mid u)$ and the estimated prior model score $\widetilde{P}(s)$ as two features for the machine learning based approach to this problem (e.g. [1]). This paper, however, does not consider machine learning based approach as they typically require labeled data, an effort that is not easily scaled for Web applications.

3.1.1 Binomial Transformation and N-gram Prior Model

The MAP decision rule can be viewed as the generalizedversionof the work by Venkataraman[24]in which the transformation model assumes the binomial distribution with the parameter $P_\$$, i.e.,

$$\begin{aligned}\log \widetilde{P}(u \mid s) &= \log \widetilde{P}(u \mid w_1, \ldots, w_{|s|}) \\ &= (|u| - |s| - 1)\log(1 - P_\$) + |s| \log P_\$\end{aligned} \quad (4)$$

and the segmentation prior model is a Markov N-gram:

$$\log \tilde{P}(s) = \log P(w_1)...P(w_{|s|} \mid w_{|s|-N+1},...,w_{|s|-1}) \quad (5)$$

One can interpret $P_\$$ as the probability that adjacent characters in u should be inserted a space in between. If one further follows the maximum entropy principle [12] and picks $P_\$ = 0.5$, then the transformation model becomes a constant for all s and does not play a role in (3), leading the MAP decision rule to degenerate into an N-gram model entirely based on (4). A popular special case of (5) is the unigram model, leading the MAP decision rule to degenerate into

$$\hat{s} = \arg\max_s \sum_{i=1}^{|s|} \log P(w_i) \quad (6)$$

Similarly, one can view the work by Goldwater et al [10] is a special case where the maximum entropy principle is followed in choosing the translation model, and the segmentation prior $\tilde{P}(s)$ is modeled as a Dirichlet process. This paper, however, does not consider nonparametric Bayesian approach because the training of the model is computationally intensive and the speed it can adapt to rapidly changing data is an open research question.

3.1.2 Geometric Mean Model

A close variation of (6) is the geometric mean heuristic proposed by Koehn and Knight[16]in which (6) is further normalized by the number of words in the hypothesized segmentation:

$$\hat{s} = \arg\max_s \frac{1}{|s|} \sum_{i=1}^{|s|} \log P(w_i) \quad (7)$$

The heuristic, though reasonably motivated and widely adopted, remains difficult to reconcile with MAP decision rule in the form of either (3) or (2).

In this work, we also generalize the classic geometric mean method to higher order N-gram so that the comparison of this method to other techniques is not handicapped because lexical contexts are not used. The generalization is straightforward by just rewriting (7) with (5) as

$$\hat{s} = \arg\max_s \frac{1}{|s|} \sum_{i=1}^{|s|} \log P(w_i \mid w_{i-N+1,...,i-1})$$

which is used in the evaluation reported in Sec. 4.

3.1.3 Word length adjustment model

Instead of assuming a binomial distribution for the translation model that implies any two adjacent characters have a uniform probability $P_\$$ as a word boundary regardless of their context,Kaitan*et al*[15] proposeda heuristic that suggests word boundary probability follow the distribution of the word length. The heuristic is quite intuitive: if the probably of a randomly drawn word to have a length m is larger than length n, then the probability of encountering a word boundary after consecutive $m$should be higher than after$n$ characters. This heuristic can be realized by choosing the translation model as

$$\log \tilde{P}(u \mid s) = \sum_{i=1}^{|s|} \log \tilde{P}(|\, w_i \,|) \quad (8)$$

where $\tilde{P}(|\, w_i \,|)$ is the empirical word length distribution that can be estimated from a unigram model, as Kaintan*et al* did on the EN-US Wikipedia corpus. In this work, we use the Microsoft N-gram service [26]that contains more than 1.3 billion words. Weobtain an empirical word length histogram and probability mass distribution, shown in Figure 1, for the top 100,000 words from the Web snapshot taken in June 2009[1]. These words altogether account for 96.5% of the entire unigram probability mass. As can be seen, the word length distribution is skewed towards short words, especially for those shorter than five characters long. After five, the word length distribution roughly tracks the word length histogram until when the length reaches 14 and beyond where we see such long words are taking less and less proportional probability mass. The fact that the distribution curve is not flat makes the probability adjustment based on word length a well-motivated heuristic.

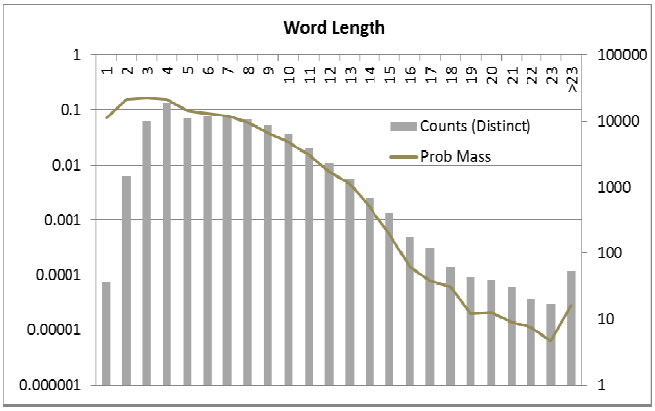


Figure 1: Histogram (bar, scale at right) and unigram probability distribution with respect to the length of the word from Microsoft Web-Ngram. The results illustrate that short words (length < 5 characters), though small in number, dominate the probability mass.

3.2 Word Synchronous Beam Search

A critical element of applying BMR for Web scale application is an efficient implementation of the search process that solves the $\arg\max_s$ in(2). As pointed out by [10], the performance of a powerful model can be largely compromised by suboptimal realizations of the search algorithm. On the other hand, the search space has an exponential complexity that cannot be attacked with brute force. In this work,we use the word synchronous beam search algorithm, a common technique forreal-time speech recognition, to iterate through the segmentation hypotheses efficiently while minimizing the search errors.

As the name implies, the search algorithm operates on the word boundaries hypothesized by each segmentation s. We note that we can express both the transformation and the segmentation prior in an iterative form based on word boundaries. For example, we can rewrite (4) into

$$\log P(u \mid s) = (|\, u \,| - 1)\log(1 - P_\$) + \sum_i \log \frac{P_\$}{1 - P_\$}$$

and (5) into

$$\log P(s) = \sum_i \log P(w_i \mid w_{i-N+1},...,w_{i-1})$$

[1] The list is published by the Microsoft Web N-gram service at http://web-ngram.research.microsoft.com/info/.

and see the total score for each segmentation hypothesis can be accumulated by advancing through the partial word sequences = (w_1,w_2,…w_i…) of the hypothesis, with the score incrementon each word w_i being $\log P_\$ /(1-P_\$)$ and $\log P(w_i \mid ..., w_{i-1})$ from the transformation and the segmentation prior models, respectively.Similarly, the word length adjustment model of (8) has already in a formulation amenable to word synchronous search, with the score increment on word boundary as log $P(|\ w_i|)$. With word length adjustment model or the binomial model when$P_\$ \le 0.5$, we are guaranteed that the partial score of a hypothesis will be greater than the full score because log probabilities are always negative. Accordingly, if the full score of hypothesis *s* is already greater than the partial scores of the competing hypotheses, we can safely conclude that these competing hypotheses will have lower full scores than*s*that pruning them from the search process will not incur search errors. The only condition where the partial hypothesis score does no longer decrease monotonically with word boundaries is when$P_\$ > 0.5$ as the increment from the transformation model $\log P_\$ /(1-P_\$)$ is positive. To minimize the pruning errors, we use the beam search approach, namely, we do not prune hypotheses unless their partial scores are lower than the leading score by more than a threshold (commonly called the beam width). It is well known from its speech recognition applications that word synchronous beam search algorithm reduces the complexity effectively and plays the pivotal role in real-time continuous speech recognition[2].

Among all the methods unified by the BMR framework described in Sec. 3.1, the geometric mean heuristics as in (6) poses some challenge to be implemented in the word synchronous fashion. In our experiments, we test the geometric mean heuristicusinga two-stage*N*-best rescoring technique: in the first stage, the beam search is carried out to identify the top *N* candidates whose scores are consequently normalized by their word sequence lengths in the second stage. The beam search algorithm can be easily generalized for *N*-best search by setting the pruning at the threshold below the N^{th} highest score rather than the top one.

4. EXPERIMENTS

As an ongoing process of monitoring a commercial search engine performance, a regularly sample of 100,000 URLs is maintained and manually annotated. Although the Internet standard specification has been recently amended to allow non-ASCII characters in URLs, we have found their usages in URLs remain a small portion, often atlow single digit percentage point, in our ongoing samples. As a result, we exclude them from our data set in this work. After discarding the non-ASCII URLs and perform a first round of splitting based on punctuation marks, we obtain a total of 266,535 unique tokens from a 100,000 URLs sample that are solely composed of alphanumeric characters. We use this set of tokens to assess the impacts of various configurations of BMR framework on work breaking performance.

4.1 Measurements

A key challenge in determining the "correct" word breaking outcome is there might be many answers that are all plausible. It has beenwidely known that there exist style disagreements among the high quality editorial contents (e.g.on "homepage" versus "home page") and the inherent ambiguities (e.g. "911stories" can be "nine-one-one stories" or "nine-eleven stories" depending on whether the stories are about emergency phone calls or September 11 events). The situation is particularly challenging when internet domain names or social network monikers, many of which are artificial concatenations of common words (e.g., "CheapTickets"), have been promoted as brand names. When these name owners use these compound words heavily in the literature as if they are inseparable words drawn from a convention lexicon, requiring these words to be always split becomes a questionable mandate, especially one of our main objectives for word breaking is to facilitate Web search where the query terms corresponding to brand names are better left intact. As a result, our annotation guidelines allow human experts to assign multiple word breaking results with various degrees of "plausibility".

Due to the multiplicity of word breaking outcomes, we regard an ideal word breaker as to produce a list of possible results ranked bytheir plausibility. As such, we apply metrics for evaluating rank list quality for word breaking. More specifically, we consider Precision at K (P@K), NDCG at K[13] and pSkip[27] in this work. Briefly, P@K measures the percentage of results at top K positions of the ranked list that are not annotated as outright implausible. For NDCG, we treat the plausibility judgments as the "relevance gain" and compute the cumulated, discounted, and normalized gains for top K positions as proposed in [13]. In this paper, we report both P@K and NDCG@K for K up to 3. Finally, because our word breaker implementation (Sec. 3.3) often produces variable length rank lists, we include the pSkip metric, which quantifies how often an implausible result is mistakenly ranked in front of a plausible result regardless of the list length, to measure how effective a word breaker can avoid ranking errors. Opposite to its two counterparts, a smaller pSkip corresponds to a better word breaker. For presentational simplicity, we show the 1-pSkip results in the following so that all the metrics are on the same scale and their downward/upward trends have the same quality implications. Our results show these metrics, although with various design nuances, largely agree with each other.

4.2 Result Analysis

We mix and match various modeling techniques for the translation model $P(u \mid s)$ and the segmentation prior model $P(s)$. For the transformation model, we evaluate the binomial (denoted as BI/$P_\$$/α in Fig. 2) and the word length adjustment (denoted as WL/α) techniques as described in Sec. 3.1. We note that both techniques degenerate to the maximum entropy (ME) model for BI/0.5/αor WL/0.0. For the segmentation model we evaluate the descriptive language style as in Web document body (B), the summary style as in the document title by the author themselves (T), the summary style as in the anchor text by the referring authors (A), and the query style as used by search engine users to retrieve the document (Q). All these language models assume Markov N-gram with N = 1 through 3 and are obtained from the June 2009 snapshot of the Microsoft Web N-gram service. In addition, we include the geometric method (GMean, Sec.3.1.2) and extend it for N-gram, and consider a multi-style language model (M) for comparison. Following the work of [25], the multi-style language model is a mixture distribution of individual style language models as $P(s) = \sum_i c_i P_i(s)$, where i = B, T, A, Q. The mixture coefficients c_i are dynamically computed using the EM

[2] The implementation can be found as a Web application at http://research.microsoft.com/en-us/um/people/kuansanw/wordbreaker/ or as a Web service at http://web-ngram.research.microsoft.com/info/break.html.

algorithm to maximize $P(s)$. Variants of the segmentation prior model are denoted as S/N where S is the language style and N is the Markov order (e.g. T/2 means the title bigram model).

4.2.1 Language Style of the Segmentation Prior Model

Our results[3]as shown in Fig. 2 indicate the style of the language plays a very critical role in the effectiveness of the segmentation prior model$P(s)$. Given the significant portion of the search queries are navigational and basically are composed of word-split versions of the domain or file path names, we originally expected the query language to be the most suitable style to create the language model for URL word breaking. Instead, the results show the title language model performs best in URL word breaking, suggesting document authors use the same style of language to compose titles and URLs. The query language, anchor text, and the document body seem to be on par with one another.

Unlike the retrieval task, the multi-style language model does not further improve the word breaking results from the individual models. This suggests that the strengths for each individual model are not complementary, or the language in composing URL is indeed not as versatile in style as queries.

4.2.2 Impacts of the Modeling Power of the Segmentation Prior Model

The higher order N-gram models have weaker assumptions on the underlying probabilistic distributions and hence, in theory,should perform better.Our experimental results agree with the theoretical prediction, especially for results from unigram to bigram. In the meantime, we also observe that the amount of gains for mismatched models (B, A, Q, M) seem to taper off after bigrams, a phenomenon also frequently reported in the literature. The matched language model built from document titles, on the other hand, seems to be an exception in that substantial accuracy gain can still be observed with higher order models. For example, the accuracy as measured by P@1 grows from 95.23% for bigram to 97.18% for trigram for the ME model. This finding is consistent with the work on query processing [11] that highlights the importance of language model style: since the increasing power accompanied by the larger Markov order can amplify the model mismatch, one can expect the more powerful model to lead to more proportional gains when the data used to create the model and the underlying NLP task are matched.

4.2.3 Effectiveness of the Translation Model and Plugged-in MAP Adjustments

By comparing against the ME model, the experimental results confirm that the various heuristics proposed for the translation models do improve the word breaking performance, with word length adjustment (WL) appearing more robust to free parameter tunings than the binomial models. For mismatched models, the data suggests the WL with a translation model weight at 0.5 seem to outperform the ME model, although the degree of improvements is modest. This rule of thumb does no longer hold when matched and reasonably powerful segmentation prior model are used. Specifically, the P@1 accuracy for the ME model using title trigram is 97.18%, in contrast to 96.55% for WL/0.5. The results seem to reinforce the postulated maximum entropy principle proposed almost half a century ago [12] that the best probability distribution, after subjecting to known constraints, is the one with the largest entropy. The results also reinforce the observations made by [2][19] that a simple NLP algorithm can outperform its more sophisticated counterparts by training with large amount of data. The experimental results seem to suggest the effect is more pronounced with large amount of training data *matched* to the language style of the underlying NLP task.

An interesting observation is that, although binomial model has more free parameters to tune, its performance is worse than word length adjustment model. This is consistent with Banko and Brill's observation [2] that more features do not necessarily translate into better performance for large data. After all, all these features are all heuristics and can amplify the over fitting problems when more data are used to train the model.

4.2.4 Efficacy of Geometric Mean Methods

The experimental data show the geometric mean heuristic, though well motivated, shown useful and widely adopted, does not scale up for the Web application. Based on the experimental data, its performance is significantly worse (P@1 often less than 50%) than others that can be better justified in the BMR framework.

5. SUMMARY

This paper first outlines the five design considerations that, based on our deployment experience, characterize the properties for NLP techniques to be scalable for Web applications. Although none of the known methods can meet these criteria simultaneously, the essence of these approaches can be unified in a mathematically sound manner under the proposed framework based on Bayesian minimum risk reduction. Together with a large and dynamically adaptive N-gram, the proposed framework can have an efficient implementation that requires no human supervision and achieve desirable performance at Web scale. The claim appears to be supported by the results of a series of large scale URL word breaker experiments measured by five metrics.

Key findings of this study are as follows. First, contextual information is critical as higher order N-gram for segment prior performs better. Secondly, language style is important, and surprisingly title language model is better than models built with query, anchor text or document body, suggesting the style of the title has the least amount of mismatch to that of URL. When language style mismatch is larger, the plugged-in MAP problem is more pronounced and additional heuristics that are proven useful empirically before can contribute to accuracy. However, when the language style has little mismatch, the plugged-in MAP problem becomes less severe, no additional heuristics seem necessary and the model parameters can be chosen in congruent to maximum entropy principle. Furthermore, the performance indeed improves as predicted by the theory when the model becomes more powerful. This suggests that many empirical results that find high order models do not help may be due to the model mismatch.

6. ACKNOWLEDGMENTS

The work has been previously demonstrated in WWW-2010 and NAACL-2010 with the assistance of Dr. Evelyne Viegas. The authors would also like to thank Dr. Guihong Cao for making available the data sets for evaluations reported in Sec. 4. Mr. Yuzhe Jin conducted the investigation work (not included in this paper) that demonstrates the character level language models are as effective as the word level models while being an intern at Microsoft Research.

[3] Detailed experimental results are available for download at http://research.microsoft.com/apps/pubs/?id=144355.

7. REFERENCES

[1] Alfonseca, E., Bilac S., and Pharies, S. 2008. Decompounding query keywords from compounding languages. In *Proc. ACL/HLT-2008*, Columbus, OH, 253-256.

[2] Banko, M. and Brill, E. 2001. Mitigating the paucity-of-data problem: exploiting the effect of training corpus size on classifier performance for natural language processing. In *Proc. 1st International Conference on Human Language Technology Research*, San Diego, CA, 1-5.

[3] Brants, T. and Franz, A. 2006. Web 1T 5-gram version 1. Linguistic Data Consortium, ISBN1-58563-397-6, Philadelphia, PA.

[4] Brent, M. 1999. An efficient, probabilistically sound algorithm for segmentation and word discovery. Machine Learning, 34(1), 71-105.

[5] Brown, R.D. 2002. Corpus driven splitting of compound words. In *Proc. 9th International Conference on Theoretical and Methodological Issues in Machine Translations (TMI-2002)*, Keihanna, Japan.

[6] Chi, C.-H., Ding, C., and Lim, A. 1999. Word segmentation and recognition for web document framework. In *Proc. CIKM-1999*, Kansan City, MO, 458-465.

[7] Craswell, N., Robertson, S. E., Zaragoza, H., and Taylor, M. 2005. Relevance weight for query independent evidence. In *Proc. SIGIR-2005*, Salvador, Brazil, 416-423.

[8] Gao, J., Wang, H., Ren, D., and Li, G. 2006. Discriminative pruning of language models for Chinese word segmentation. In *Proc. ACL-2006*, Sydney, Australia, 1001-1008.

[9] Gelfand, A. E. 1995. Gibbs sampling. Journal of the American Statistical Association, 452, 1300-1304.

[10] Goldwater, S., Griffiths, T., and Johnson, M. 2006. Contextual dependencies in unsupervised word segmentation. In Proc. *ACL-2006*, Sydney, Australia, 673-680.

[11] Huang, J., Gao, J., Miao, J., Li, X., Wang, K., Behr, F., and Giles, C. 2010. Exploring web scale language models for search query processing. In *Proc. WWW-2010*, Raleigh, NC, 451-460.

[12] Jaynes, E. T. 1957. Information theory and statistical mechanics. Physics Review Series II, America Physical Society, 106(4), 620-630.

[13] K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, pages 422-446, 20(4), 2002

[14] Kan, M.-Y. andThi, H. 2005. Fast webpage classification using URL features. In *Proc. CIKM-2005*, Bremen, Germany, 325-326.

[15] Khaitan, S., Das A., Gain, S., and Sampath, A. 2009. Data-driven compound splitting method for English compounds in domain names. In *Proc. CIKM-2009*, Hong Kong, China, 207-213.

[16] Koehn, P. and Knight, K. 2003. Empirical methods for compounding splitting. In *Proc. EACL-2003*, Budapest, Hungary, 187-193.

[17] Larson, M., Willett, D., Köhler, J., and Rigoll, G. 2000. Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speech. In *Proc. ICSLP-2000*, Beijing, China.

[18] Monz, C. and de Rijke, M. 2001. Shallow morphological analysis in monolingual information retrieval for Dutch, German, and Italian. Evaluation of Cross-Language Information Retrieval Systems, volume 2406 of Lecture Notes in Computer Science, Springer, 262-277.

[19] Norvig, P. 2008. Statistical learning as the ultimate agile development tool. ACM 17th Conference on Information and Knowledge Management Industry Event (CIKM-2008), Napa Valley, CA.

[20] Och, F. J. and Ney, H. 2003. A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1), 19-51.

[21] Salvetti, F. and Nicolov N. 2006. Weblog classification for fast splog filtering: a URL language model segmentation approach. In *Proc. NAACL-2006*, New York, NY, 137-140.

[22] Song, R., Xin, G., Shi, S., Wen, J.-R., Ma., and W.-Y. 2006. Exploring URL hit priors for web search. In *Proc. ECIR-2006*, London, UK, 277-288.

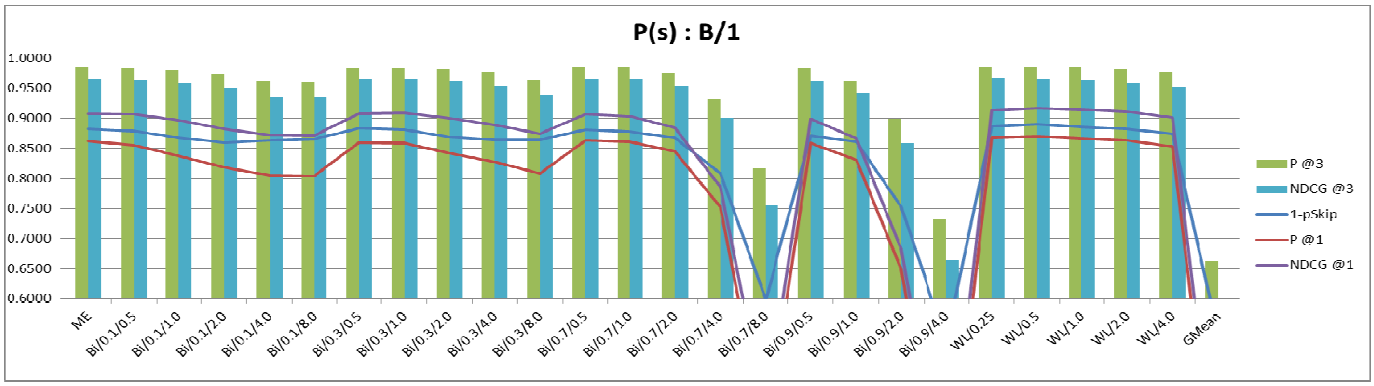
[23] Suzuki, H., Brockett, C., and Kacmarcik, G. 2000. Using broad-coverage parser for word-breaking in Japanese. In *Proc. 18th Conference on Computational Linguistics*, Saarbrüchen, Germany, 822-828.

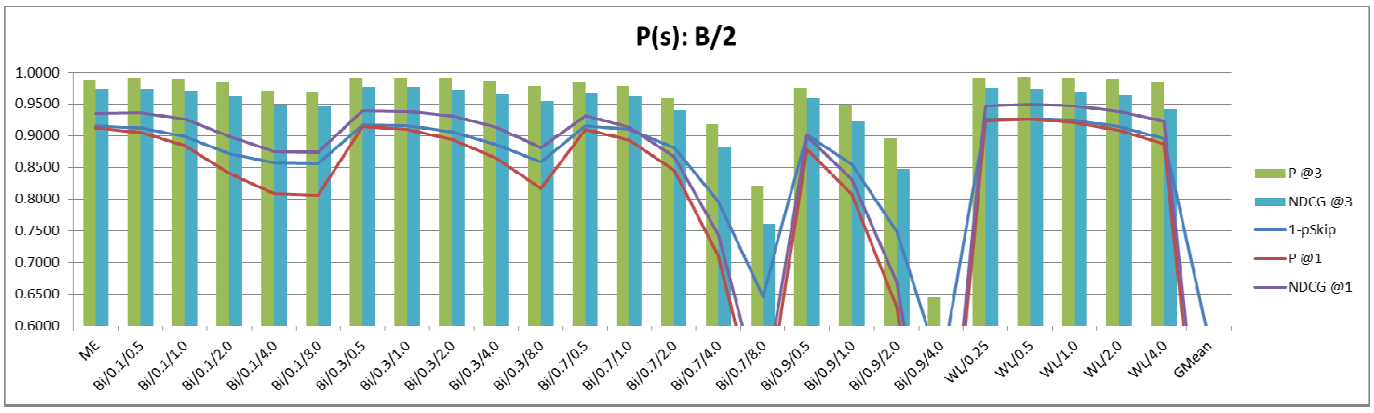
[24] Venkataraman, V. 2001. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3), 351-372.

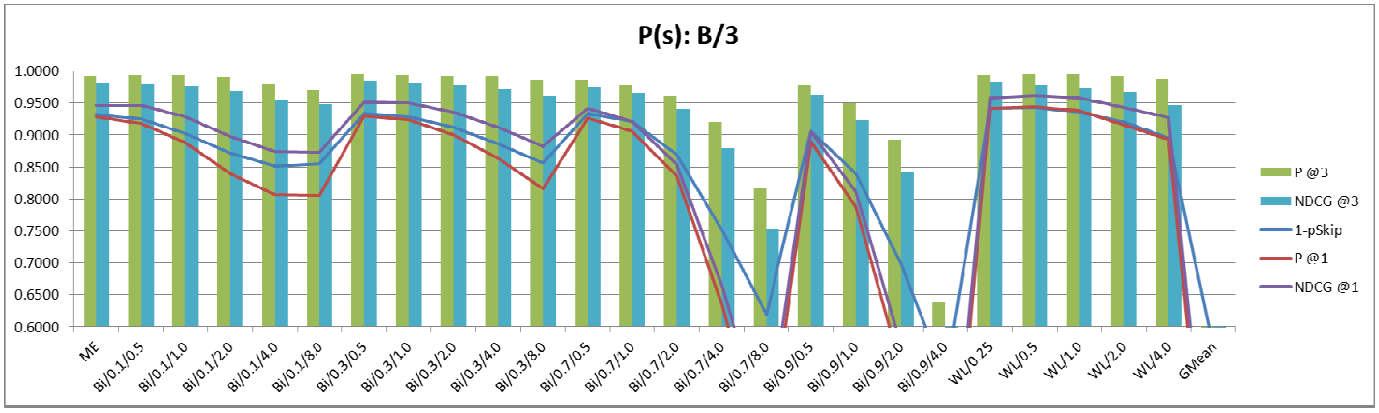
[25] Wang, K., Li, X., and Gao, J. 2010. Multi-style language model for web scale information retrieval. In *Proc. SIGIR-2010*, Geneva, Switzerland, 467-474.

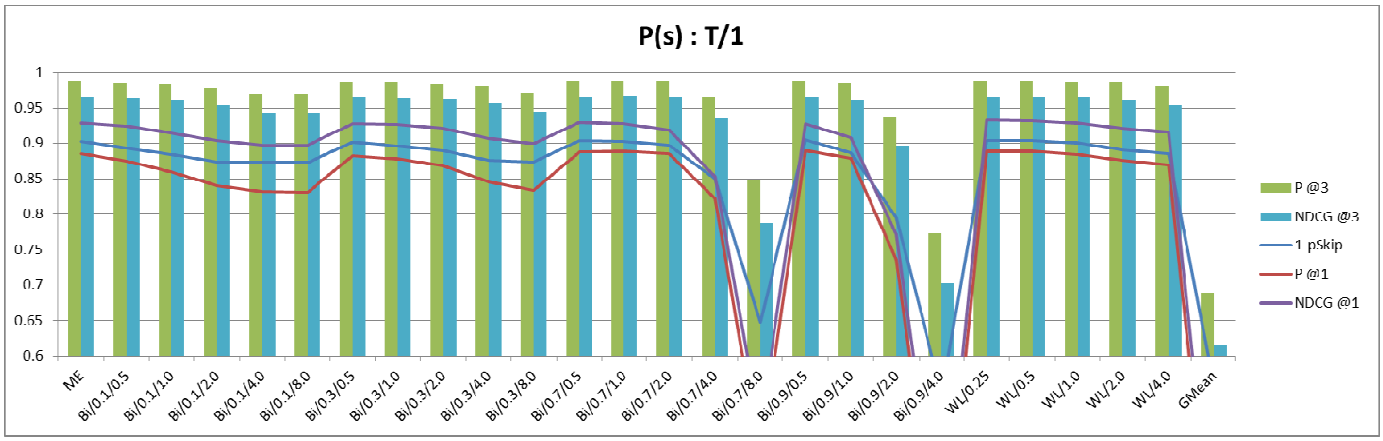
[26] Wang, K., Thrasher, C., Viegas, E., Li, X., and Hsu, B.-J. 2010. An overview of Microsoft web n-gram corpus and applications. In *Proc. NAACL/HLT-2010*, Los Angeles, CA.

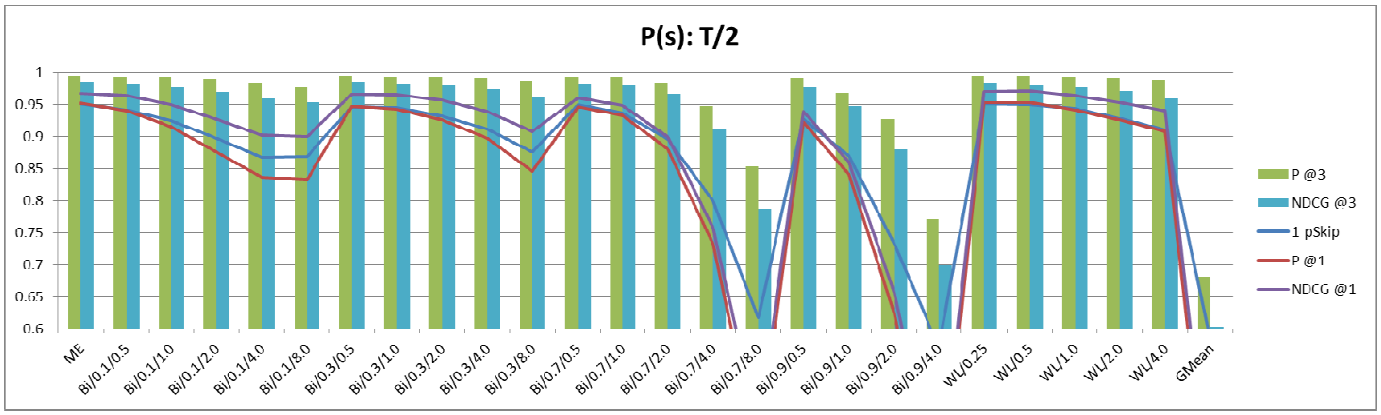
[27] Wang, K., Walker, T., and Zheng, Z. 2009. PSkip: estimating web search ranking quality from clickthrough data. In *Proc. KDD-2009*, Paris, France.

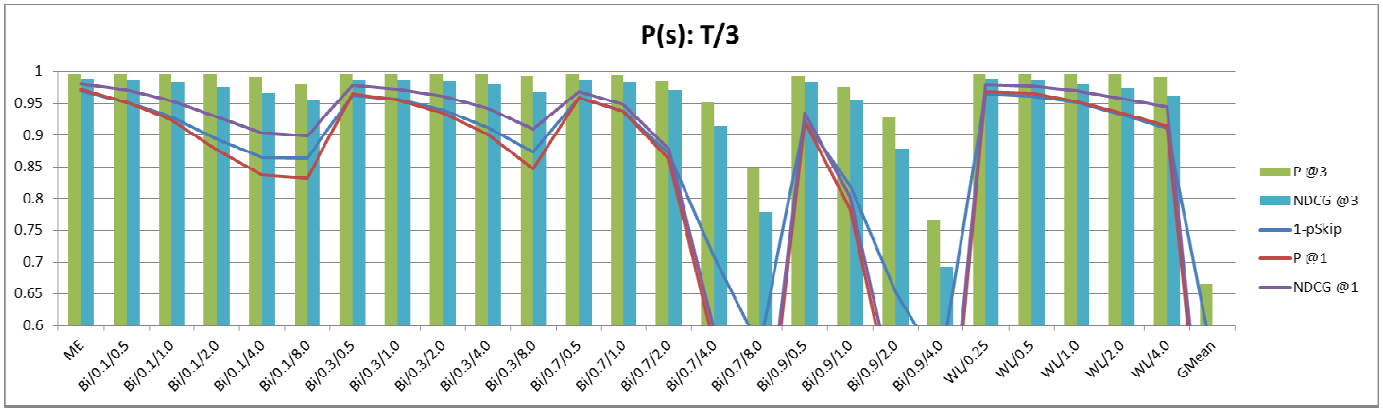

P(s) : B/1
1.0000
0.9500
0.9000
0.8500
0.8000
0.7500
0.7000
0.6500
0.6000
ME
Bi/0.1/0.5
Bi/0.1/1.0
Bi/0.1/2.0
Bi/0.1/4.0
Bi/0.1/8.0
Bi/0.3/0.5
Bi/0.3/1.0
Bi/0.3/2.0
Bi/0.3/4.0
Bi/0.3/8.0
Bi/0.7/0.5
Bi/0.7/1.0
Bi/0.7/2.0
Bi/0.7/4.0
Bi/0.7/8.0
Bi/0.9/0.5
Bi/0.9/1.0
Bi/0.9/2.0
Bi/0.9/4.0
WL/0.25
WL/0.5
WL/1.0
WL/2.0
WL/4.0
GMean
P @3
NDCG @3
1-pSkip
P @1
NDCG @1

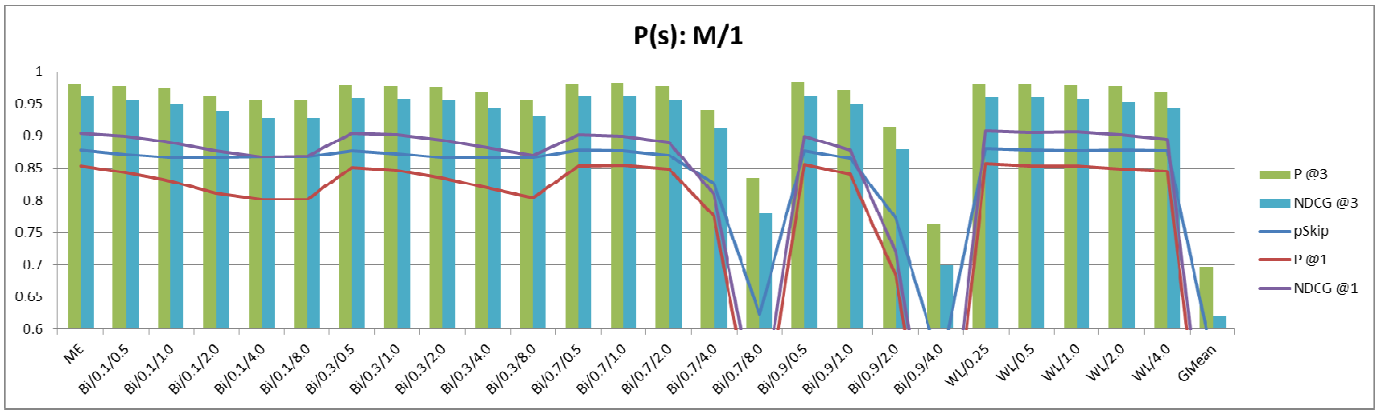

P(s): B/2
1.0000
0.9500
0.9000
0.8500
0.8000
0.7500
0.7000
0.6500
0.6000
ME
Bi/0.1/0.5
Bi/0.1/1.0
Bi/0.1/2.0
Bi/0.1/4.0
Bi/0.1/8.0
Bi/0.3/0.5
Bi/0.3/1.0
Bi/0.3/2.0
Bi/0.3/4.0
Bi/0.3/8.0
Bi/0.7/0.5
Bi/0.7/1.0
Bi/0.7/2.0
Bi/0.7/4.0
Bi/0.7/8.0
Bi/0.9/0.5
Bi/0.9/1.0
Bi/0.9/2.0
Bi/0.9/4.0
WL/0.25
WL/0.5
WL/1.0
WL/2.0
WL/4.0
GMean
P @3
NDCG @3
1-pSkip
P @1
NDCG @1

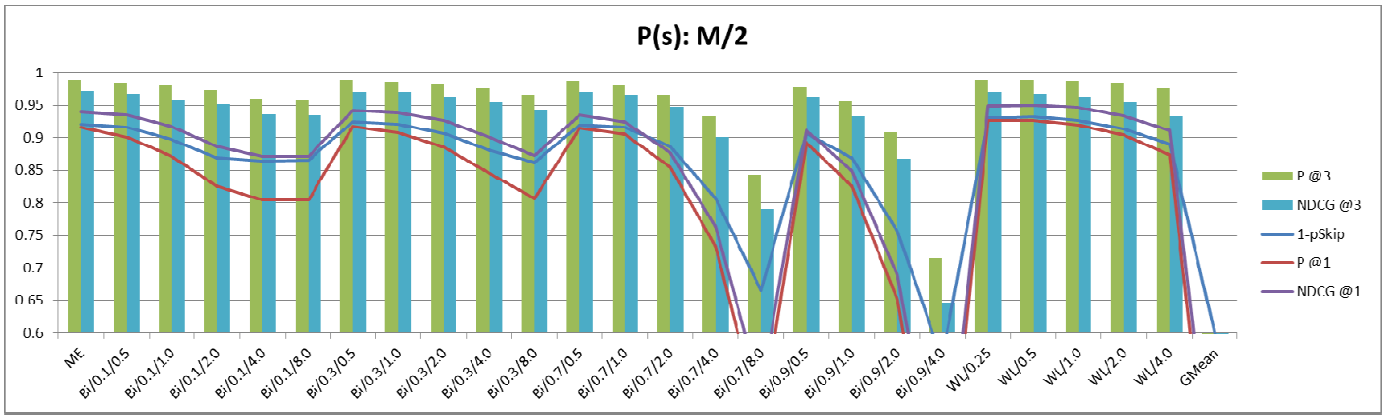

P(s): B/3
1.0000
0.9500
0.9000
0.8500
0.8000
0.7500
0.7000
0.6500
0.6000
ME
Bi/0.1/0.5
Bi/0.1/1.0
Bi/0.1/2.0
Bi/0.1/4.0
Bi/0.1/8.0
Bi/0.3/0.5
Bi/0.3/1.0
Bi/0.3/2.0
Bi/0.3/4.0
Bi/0.3/8.0
Bi/0.7/0.5
Bi/0.7/1.0
Bi/0.7/2.0
Bi/0.7/4.0
Bi/0.7/8.0
Bi/0.9/0.5
Bi/0.9/1.0
Bi/0.9/2.0
Bi/0.9/4.0
WL/0.25
WL/0.5
WL/1.0
WL/2.0
WL/4.0
GMean
P @3
NDCG @3
1-pSkip
P @1
NDCG @1


P(s) : T/1
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
ME
Bi/0.1/0.5
Bi/0.1/1.0
Bi/0.1/2.0
Bi/0.1/4.0
Bi/0.1/8.0
Bi/0.3/0.5
Bi/0.3/1.0
Bi/0.3/2.0
Bi/0.3/4.0
Bi/0.3/8.0
Bi/0.7/0.5
Bi/0.7/1.0
Bi/0.7/2.0
Bi/0.7/4.0
Bi/0.7/8.0
Bi/0.9/0.5
Bi/0.9/1.0
Bi/0.9/2.0
Bi/0.9/4.0
WL/0.25
WL/0.5
WL/1.0
WL/2.0
WL/4.0
GMean
P @3
NDCG @3
1-pSkip
P @1
NDCG @1


P(s): T/2
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
ME
Bi/0.1/0.5
Bi/0.1/1.0
Bi/0.1/2.0
Bi/0.1/4.0
Bi/0.1/8.0
Bi/0.3/0.5
Bi/0.3/1.0
Bi/0.3/2.0
Bi/0.3/4.0
Bi/0.3/8.0
Bi/0.7/0.5
Bi/0.7/1.0
Bi/0.7/2.0
Bi/0.7/4.0
Bi/0.7/8.0
Bi/0.9/0.5
Bi/0.9/1.0
Bi/0.9/2.0
Bi/0.9/4.0
WL/0.25
WL/0.5
WL/1.0
WL/2.0
WL/4.0
GMean
P @3
NDCG @3
1 pSkip
P @1
NDCG @1


P(s): T/3
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
ME
Bi/0.1/0.5
Bi/0.1/1.0
Bi/0.1/2.0
Bi/0.1/4.0
Bi/0.1/8.0
Bi/0.3/0.5
Bi/0.3/1.0
Bi/0.3/2.0
Bi/0.3/4.0
Bi/0.3/8.0
Bi/0.7/0.5
Bi/0.7/1.0
Bi/0.7/2.0
Bi/0.7/4.0
Bi/0.7/8.0
Bi/0.9/0.5
Bi/0.9/1.0
Bi/0.9/2.0
Bi/0.9/4.0
WL/0.25
WL/0.5
WL/1.0
WL/2.0
WL/4.0
GMean
P @3
NDCG @3
1-pSkip
P @1
NDCG @1


P(s): M/1
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
ME
Bi/0.1/0.5
Bi/0.1/1.0
Bi/0.1/2.0
Bi/0.1/4.0
Bi/0.1/8.0
Bi/0.3/0.5
Bi/0.3/1.0
Bi/0.3/2.0
Bi/0.3/4.0
Bi/0.3/8.0
Bi/0.7/0.5
Bi/0.7/1.0
Bi/0.7/2.0
Bi/0.7/4.0
Bi/0.7/8.0
Bi/0.9/0.5
Bi/0.9/1.0
Bi/0.9/2.0
Bi/0.9/4.0
WL/0.25
WL/0.5
WL/1.0
WL/2.0
WL/4.0
GMean
P @3
NDCG @3
pSkip
P @1
NDCG @1


P(s): M/2
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
ME
Bi/0.1/0.5
Bi/0.1/1.0
Bi/0.1/2.0
Bi/0.1/4.0
Bi/0.1/8.0
Bi/0.3/0.5
Bi/0.3/1.0
Bi/0.3/2.0
Bi/0.3/4.0
Bi/0.3/8.0
Bi/0.7/0.5
Bi/0.7/1.0
Bi/0.7/2.0
Bi/0.7/4.0
Bi/0.7/8.0
Bi/0.9/0.5
Bi/0.9/1.0
Bi/0.9/2.0
Bi/0.9/4.0
WL/0.25
WL/0.5
WL/1.0
WL/2.0
WL/4.0
GMean
P @3
NDCG @3
1-pSkip
P @1
NDCG @1

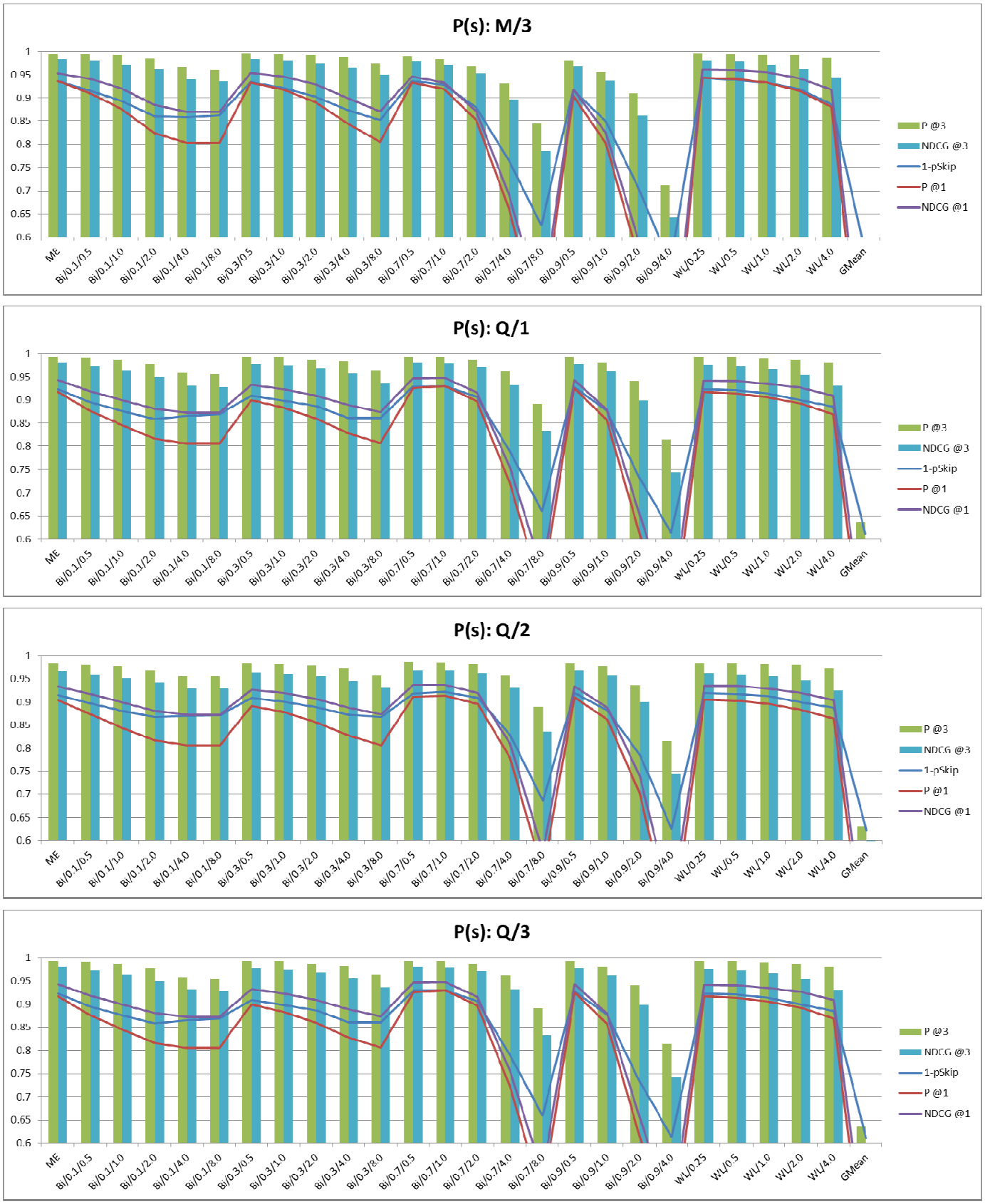


Figure 2: URL word breaker results as measured by five metrics.