# Towards Optimum Query Segmentation: In Doubt Without

Matthias Hagen Martin Potthast Anna Beyer Benno Stein

Bauhaus-Universität Weimar
99421 Weimar, Germany
<first name>.<last name>@uni-weimar.de

## ABSTRACT

Query segmentation is the problem of identifying those keywords in a query, which together form compound concepts or phrases like `new york times`. Such segments can help a search engine to better interpret a user's intents and to tailor the search results more appropriately. Our contributions to this problem are threefold. (1) We conduct the first large-scale study of human segmentation behavior based on more than 500 000 segmentations. (2) We show that the traditionally applied segmentation accuracy measures are not appropriate for such large-scale corpora and introduce new, more robust measures. (3) We develop a new query segmentation approach with the basic idea that, in cases of doubt, it is often better to (partially) leave queries without any segmentation.

This new *in-doubt-without approach* chooses different segmentation strategies depending on query types. A large-scale evaluation shows substantial improvement upon the state of the art in terms of segmentation accuracy. To draw a complete picture, we also evaluate the impact of segmentation strategies on retrieval performance in a TREC setting. It turns out that more accurate segmentation not necessarily yields better retrieval performance. Based on this insight, we propose an in-doubt-without variant which achieves the best retrieval performance despite leaving many queries unsegmented. But there is still room for improvement: the optimum segmentation strategy which always chooses the segmentation that maximizes retrieval performance, significantly outperforms all other tested approaches.

**Categories and Subject Descriptors**: H.3.3 [Information Search and Retrieval]: Query formulation

**General Terms**: Algorithms, Experimentation

**Keywords**: Query Segmentation

## 1. INTRODUCTION

Keyword queries are the predominant way of expressing information needs on the web. All major search engines offer a search box where users enter the keywords which they believe describe the information sought. Once submitted, the task of a search engine is to use the query to second-guess the user's actual information need, and then to retrieve web documents most relevant to it. The user expects nothing less than to "be understood" by the search engine, regardless of the validity of the clues given in her query. Hence, search engines nowadays rely on predictors that help them to interpret, correct, classify, and reformulate every submitted query in a split second before the actual document retrieval begins. In this paper we study one such predictor for query reformulation which identifies indivisible sequences of keywords in a query; the corresponding task is called query segmentation.

Almost all search engines offer search operators allowing their users to clearly specify which parts of a query are indivisible (e.g., via double quotes). However, less than 1.12% of queries contain quotes or other operators [19], which means that the vast majority of searchers seem to be unaware of this option. On the other hand, several studies have shown that the retrieval performance can be improved when treating important phrases and compound concepts like `new york times` as indivisible segments of a query [2, 9, 14]. The search engine can exploit such hints to increase result precision since documents that do not contain a segment's words in the exact same order can be discarded. Apparently, the majority of users not using quotes unwittingly miss out on this additional performance, which calls for an automatic solution to query segmentation.

Our contributions include the first large-scale analysis of human query quoting, new and more robust accuracy measures, and a new approach to query segmentation. Our in-depth analysis of the segmentations in the recently published Webis Query Segmentation Corpus[1] sheds light on how humans quote queries. One of the insights is that the traditional accuracy measures for query segmentation algorithms are unqualified for corpora of this size: they appear to function on the previous standard corpus from Bergsma and Wang,[2] but their weaknesses become apparent on the larger corpus. To address this issue, we propose robust accuracy measures that take into account annotator consensus.

The algorithmic contribution of this paper also derives from our corpus analysis. Current state-of-the-art segmentation methods typically segment all queries with the following single strategy: "Phrases with a high web occurrence frequency are good segments." In contrast, our study shows that different strategies are required to handle different types of queries. Most notably, a good strategy often is to refrain from segmenting too many keywords—an approach that we call in-doubt-without segmentation. A large-scale evaluation shows tailored variants of this approach to perform best in most situations. The evaluation includes accuracy comparisons, as well as a carefully set up retrieval experiment within the TREC framework. In the TREC experiment, our new approach comes closest to the optimum segmentation algorithm, which always chooses the segmentation of a query that maximizes retrieval performance.

*CIKM'12,* October 29–November 2, 2012, Maui, HI, USA.

[1] 50 000 queries, each segmented by 10 annotators [11].

[2] 500 queries, each segmented by 3 annotators [5].

Finally, we introduce new and simple baseline segmentation algorithms, which choose only segments that are titles of Wikipedia articles or so-called *strict noun phrases*. Though many query segmentation algorithms employ Wikipedia titles as one feature among several others, the segmentation power of these titles alone has not been analyzed yet. It turns out that Wikipedia titles form a strong baseline, performing better than many state-of-the-art algorithms.

The following Section 2 presents the basic notation for query segmentation and briefly surveys the related work. Section 3 presents a large-scale corpus analysis which reveals human strategies on how to segment web queries. As a result of this analysis, new, more robust segmentation accuracy measures are proposed in Section 4. Section 5 introduces two new Wikipedia baselines, and Section 6 discusses our in-doubt-without framework. An empirical evaluation in Section 7 shows that a variant of our new approach achieves the best retrieval performance improvement. The paper closes with concluding remarks in Section 8.

## 2. NOTATION AND RELATED WORK

Almost all query segmentation studies comply with the following notation. A query $q$ is viewed as a sequence $(w_1, \ldots, w_k)$ of $k$ keywords. Every contiguous subsequence of $q$ forms a potential segment. This includes one-word segments, which are usually left unquoted in practice. A valid segmentation $S$ for $q$ consists of disjunct segments $s$ whose concatenation yields $q$ again (i.e., valid segmentations leave the order of keywords untouched). The problem of query segmentation is typically defined as the automatic identification of the “best” valid segmentation for a query $q$, where “best” refers to segmentations that humans would choose or that maximize retrieval performance. Note that a valid segmentation determines for each pair $\langle w, w' \rangle$ of consecutive keywords in $q$ whether or not there should be a segment break between $w$ and $w'$. Hence, there are $2^{k-1}$ valid segmentations for a $k$-keyword query and $k(k-1)/2$ potential segments with at least two keywords.

Risvik et al. [17] were the first to propose an algorithm for query segmentation. Their approach scores potential segments by pointwise mutual information and query log frequency. Later on, Jones et al. [13] propose an approach that is based on mutual information alone, favoring segments with high mutual information. Huang et al. [12] construct a tree of concepts from mutual information scored segments and then use this tree to decide upon the final segmentation. However, note that in most segmentation studies, mutual information segmentation forms the baseline, often performing worse compared to more sophisticated approaches.

The supervised learning method by Bergsma and Wang [5] combines many features (web and query log frequencies, POS tags, etc.) but is focused solely on noun phrase queries. Bergsma and Wang also introduced what was to become a standard evaluation corpus for query segmentation. Note, however, that the few queries used for training and testing were segmented by the same annotator. This leaves some doubts about the generalizability of the reported good accuracy. Bendersky et al. [2] later employ a variant of Bergsma and Wang’s method as a subroutine in a two-stage segmentation process. Instead of a supervised approach requiring expensive training instances, Tan and Peng [18] and Zhang et al. [20] propose unsupervised methods. Zhang et al. compute segment scores from the eigenvalues of a correlation matrix corresponding to a given query. Tan and Peng use web scale language models and boost a segment’s probability if it is used prominently in Wikipedia.

Two other approaches are based on pseudo-relevance feedback. Brenes et al. [8] segment phrases found in search result snippets of the unquoted query. Bendersky et al. [3, 4] insert a break between two keywords whenever a likelihood ratio is below some threshold. The likelihood ratios are obtained from the top-ranked documents for the unquoted query. On the upside, these approaches achieve promising results; on the downside, however, all queries have to pass a search engine’s retrieval pipeline twice before any results are returned, once unquoted and once more after segmentation.

Hagen et al. [10, 11] avoid such runtime problems by means of a basic, yet effective scoring approach. All valid segmentations of a query are scored by a weighted sum of normalized web phrase frequencies. The normalization makes segments of different lengths comparable in terms of their frequencies. In our hybrid framework, we use one of these normalization schemes as a subroutine.

Two recent approaches are based on query log analyses. Mishra et al. [15] score segmentations based on phrase frequencies in a query log, while Li et al. [14] exploit click-through information. A potential problem with the use of query logs (besides the non-availability of up-to-date query logs to academia) is the fact that co-occurring keywords in queries may not exist as a phrase in any web document. Using such sequences of keywords as segments would yield no search results at all.

Besides query segmentation algorithms, only two papers specifically address evaluation methodology: Bergsma and Wang [5] have introduced the first query segmentation corpus consisting of 500 queries which have been segmented by three annotators each. Hagen et al. [11] recently published the Webis-QSeC-10 corpus which is two orders of magnitude larger. Many of the aforementioned algorithms have already been evaluated on the Bergsma-Wang corpus, while the Webis-QSeC-10 has not yet spread widely. To ensure comparability, we use both in our evaluations.

## 3. HOW HUMANS QUOTE

This section reports on the first large-scale study of human quoting behavior in web search queries. Our study is based on a corpus analysis of the recently published Webis Query Segmentation Corpus [11]. The size of this corpus forms a unique opportunity to study human query quoting on a representative scale. The previous corpora of Bergsma and Wang [5] and Bendersky et al. [3] are too small to draw meaningful conclusions from them. Our study divides into two parts: Section 3.1 briefly introduces the corpus and verifies its validity with regard to two important corpus parameters. Sections 3.2 and 3.3 then focus on characteristics of human query quoting that have repercussions on segmentation algorithm design.

### 3.1 Corpus Verification

The Webis Query Segmentation Corpus (Webis-QSeC-10) consists of 53 437 web queries and at least 10 segmentations per query (1072 queries have more than 10). The segmentations were obtained by crowdsourcing via Amazon’s Mechanical Turk (AMT). Altogether, 1795 different workers from AMT each segmented about 300 queries on average. Although the workers worked independently, they naturally often chose identical segmentations on the same query. The frequency of submission of certain segmentations is reflected in the corpus: for example, the query `new york times` has been segmented nine times as `"new york times"` and once without any segments as `new york times`. The submission frequency of a segmentation may thus be interpreted as if workers voted on it, and the example already shows that there are segmentations which got a majority of votes and others which did not. The second segmentation in the above example also shows that even nonsense segmentations got votes, say, the corpus contains some noise. The amount of votes collected per query, however, and a carefully set up manual review and rejection policy of bad workers made sure that such noise segmentations do not dominate [11].

**Table 1: Query lengths in the Webis-QSeC-10 and the AOL log.**

| Query length | Webis-QSeC-10 | | AOL query log | |
|---|---|---|---|---|
| | Queries | Ratio | Queries | Ratio |
| 3 | 23 833 | 44.60% | 2 750 697 | 45.64% |
| 4 | 14 571 | 27.27% | 1 620 818 | 26.89% |
| 5 | 7 678 | 14.37% | 846 449 | 14.04% |
| 6 | 3 803 | 7.12% | 418 621 | 6.95% |
| 7 | 1 864 | 3.49% | 202 275 | 3.36% |
| 8 | 947 | 1.77% | 102 792 | 1.70% |
| 9 | 481 | 0.90% | 55 525 | 0.92% |
| 10 | 260 | 0.49% | 30 423 | 0.50% |
| Σ | 53 437 | 100.00% | 6 027 600 | 100.00% |

*Query Length.*

The Webis-QSeC-10 has been sampled from the subset of the AOL query log [16] which consists of only queries with 3–10 keywords. Queries with just 1 or 2 keywords are not included as 1-word queries cannot contain any phrases and 2-word queries are typically handled very well by proximity features. The sampling maintains the query length distribution and the query frequency distribution of the entire query log [11]. This means there are more short queries than long ones, and that queries submitted more frequently are more likely to be part of the corpus. The Webis-QSeC-10 reflects well the query length distribution of the AOL query log (see Table 1). The slight difference in the length ratios is due to a post-processing of the query sample including a semi-automatic spelling correction and query filtering [11].

*Noun Phrase Queries.*

One point of criticism about the previously used Bergsma-Wang corpus was that it consists of only noun phrase queries and for instance Barr et al. find, based on a Yahoo query log, that many queries are not noun phrase queries [1]. The amount of noun phrase queries in the Webis-QSeC-10 has not been analyzed yet. Since not all the 53 437 queries can be checked manually in a reasonable amount of time, we check only the 4850 queries marked as training set. The left part of Table 2 (column group "Manual Detection") contains the results of our analysis dependent on the length of a query. Interestingly, there seem to be three different "groups" with respect to query length. About 90% of the short queries (3–5 words) are noun phrase queries. From the mid-length queries (6–8 words), about 60% are noun phrase queries. Then, for long queries (9–10 words), the picture turns upside down: only a minority of them are noun phrases. One reason for this is that the longer a query, the more likely are questions, song titles, and lyrics.

## 3.2 Automatic Noun Phrase Query Detection

Having verified that the Webis-QSeC-10 represents well the properties found in larger query logs, we now turn our attention to the huge amount of human segmentations found in the corpus. One of our intentions is to compare human quoting on noun phrase queries with that on other queries. However, the Webis-QSeC-10 does not feature annotations which tell the two types of queries apart. Hence, we resort to using an automatic parts-of-speech tagger to label queries as noun phrase queries or other queries. Automatic POS tagging of queries brings about the problem that traditional POS taggers are not trained for use on queries [1]. Since training a POS tagger for queries, again, requires manual intervention first, we instead carry out a restrictive POS tagging strategy in order to identify a specific subset of all noun phrase queries. We restrict our analysis to noun phrases composed of only nouns, numbers, adjectives, and articles, say, *strict noun phrase* queries (SNP queries). These parts-of-speech can be identified reliably using the available POS taggers.

**Table 2: Distribution of noun phrase (NP) queries and strict noun phrase (SNP) queries in (a subset of) the Webis-QSeC-10.**

| Query length | Manual detection | | | Automatic detection | | |
|---|---|---|---|---|---|---|
| | Queries | NP queries | Ratio | Queries | SNP queries | Ratio |
| 3 | 2 164 | 2 074 | 95.84% | 23 833 | 15 969 | 67.00% |
| 4 | 1 320 | 1 213 | 91.89% | 14 571 | 6 660 | 45.71% |
| 5 | 692 | 604 | 87.28% | 7 678 | 1 934 | 25.19% |
| 6 | 344 | 240 | 69.77% | 3 803 | 486 | 12.78% |
| 7 | 180 | 114 | 63.33% | 1 864 | 144 | 7.73% |
| 8 | 81 | 44 | 54.32% | 947 | 40 | 4.22% |
| 9 | 46 | 14 | 30.43% | 481 | 9 | 1.87% |
| 10 | 23 | 8 | 34.78% | 260 | 3 | 1.15% |
| Σ | 4 850 | 4 311 | 88.89% | 53 437 | 25 245 | 47.24% |

As a pilot study to verify this approach, we have employed Qtag[3] and the Stanford tagger, checking their performance on 1000 queries chosen at random from the Webis-QSeC-10 training set. Qtag identifies 447 queries as SNP queries and the Stanford tagger 455 queries. From these, only about 1% were false positives while only few SNP queries were missed. Altogether, both taggers achieve a precision of 99% at a recall of about 90% in our pilot study. In this connection, we decide to use Qtag as it is about five times faster than the Stanford tagger. When run on the entire Webis-QSeC-10, the SNP query distribution shown in the right part of Table 2 (column group "Automatic Detection") is obtained. About 47% of the queries in the Webis-QSeC-10 are tagged as SNP queries. Note that a direct comparison between manually detected noun phrase queries and automatically detected strict noun phrase queries is rendered difficult due to the fact that the latter are only a subset of all noun phrase queries. However, the automatic SNP query detection now allows for an analysis of human quoting behavior on SNP vs. other queries on the entire Webis-QSeC-10.

## 3.3 Characteristics of Human Query Quoting

In our analysis of how humans quote queries, we examine three important parameters: segment length, query length and query type. We expect that a better understanding of their connections in human segmentations can be exploited when designing query segmentation algorithms. In the long run, this may even have consequences for the construction of tailored information retrieval models. Our study particularly sheds light onto the following questions (to cut a long story short, we give the answers right away):

1. Does the type affect how much of a query is segmented?
   *Yes, in noun phrase queries more keywords are segmented.*
2. Does length or type of a query affect if it is quoted at all?
   *Length does not, but noun phrase queries are quoted more.*
3. Does length or type of a query affect formation of consensus about its quotation among independent human annotators?
   *Type does not, but there is more agreement on short queries. Also, unanimity is an exception: many queries even do not have a segmentation supported by an absolute majority.*

Given these findings, we draw the following conclusions:

⇒ The answers to Questions 1 and 2 imply that segmentation accuracy against human segmentations requires to differentiate between noun phrase queries and others. On other queries, a more conservative segmentation (with less keywords in segments) should be applied.

[3] http://phrasys.net/uob/om/software

**Table 3: Distribution of segments in the Webis-QSeC-10, dependent on their length, and on query type. The absolute and relative amounts of segments and keywords per length class are given for each of the query types as well as all queries.**

| **Segment length** | **All queries** | | | | **Noun phrase queries** | | | | **Other queries** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Segments | Ratio | Words | Ratio | Segments | Ratio | Words | Ratio | Segments | Ratio | Words | Ratio |
| Study 1: *Webis-QSeC-10 training set (4850 queries, manually labeled noun phrase queries)* | | | | | | | | | | | | |
| 1 | 89 109 | 66.39% | 89 109 | 44.81% | 70 610 | 63.27% | 70 610 | 41.74% | 18 499 | 81.74% | 18 499 | 62.34% |
| 2 | 30 498 | 22.72% | 60 996 | 30.68% | 28 009 | 25.10% | 56 018 | 33.11% | 2 489 | 11.00% | 4 978 | 16.77% |
| 3 | 11 222 | 8.36% | 33 666 | 16.93% | 10 265 | 9.20% | 30 795 | 18.20% | 957 | 4.23% | 2 871 | 9.67% |
| 4 | 2 467 | 1.84% | 9 868 | 4.96% | 2 101 | 1.88% | 8 404 | 4.97% | 366 | 1.62% | 1 464 | 4.93% |
| 5–10 | 929 | 0.69% | 5 201 | 2.62% | 609 | 0.54% | 3 337 | 1.97% | 320 | 1.41% | 1 864 | 6.27% |
| Study 2: *Webis-QSeC-10 entirely (53 437 queries, automatically labeled strict noun phrase queries)* | | | | | | | | | | | | |
| 1 | 1 020 249 | 67.10% | 1 020 249 | 46.55% | 342 592 | 58.47% | 342 592 | 38.57% | 677 657 | 72.52% | 677 657 | 52.00% |
| 2 | 368 554 | 24.24% | 737 108 | 33.63% | 191 465 | 32.67% | 382 930 | 43.11% | 177 089 | 18.95% | 354 178 | 27.18% |
| 3 | 104 033 | 6.84% | 312 099 | 14.24% | 45 601 | 7.78% | 136 803 | 15.40% | 58 432 | 6.25% | 175 296 | 13.45% |
| 4 | 20 278 | 1.33% | 81 112 | 3.70% | 5 662 | 0.97% | 22 648 | 2.55% | 14 616 | 1.56% | 58 464 | 4.49% |
| 5–10 | 7 354 | 0.48% | 41 050 | 1.87% | 650 | 0.12% | 3 357 | 0.38% | 6 704 | 0.72% | 37 693 | 2.90% |

⇒ The answer to Question 3 in combination with the size of the Webis-QSeC-10 implies that traditional segmentation accuracy measures (of which some are based on annotator unanimity) are not reasonable for the Webis-QSeC-10.

In what follows, we substantiate these findings empirically.

### *Question 1: Segment Length vs. Query Type.*

Table 3 shows the segment length distribution per query type. Both the manually labeled noun phrase queries as well as the automatically labeled strict noun phrase queries are shown. As has been reported earlier [11], humans favor short segments of just two or three keywords over longer ones (see column group "All queries"). Also note that about 67% of all segments in the Webis-QSeC-10 contain just one keyword (i.e., they are unquoted) and only 0.5–0.7% (dependent on manual or automatic labeling) of the segments contain five or more keywords. These segments are often song titles, lyrics, and sometimes the aforementioned noise from false segmentations. Regarding the word level, about 45–46% of all keywords appear unquoted. About 30–34% of the keywords are part of 2-word segments and another 14–17% in 3-word segments. In this connection, note the strong correlation of the trends obtained from the manually labeled subset of the Webis-QSeC-10 and the automatically labeled entire corpus.

Regarding different query types (column groups "Noun Phrase Queries" and "Other Queries"), more keywords of SNP queries are segmented than in other queries: 38.5% of the keywords in SNP queries are unquoted, compared to 52% of the keywords in other queries. In return, long segments of five or more keywords appear relatively more often in other queries than in SNP queries. Again, similar trends are observed in the manually labeled queries.

### *Question 2: To Quote, or not to Quote.*

One reason for the different segment length distribution of SNP queries and other queries could be the ratio of unquoted queries. For about 71% of all the queries in the Webis-QSeC-10, at least one annotator chose not to quote. It might be hypothesized that many of these cases have occurred not by intent but due to AMT workers failing to submit quotations. Hence, we have investigated how often not to quote a query was the top voted option. This was the case in about 24% of all the queries. Interestingly, the ratio of queries for which most annotators chose not to quote lies in a 20–30% range for all query lengths. Hence, long queries are not more likely to be left unquoted than short queries. However, regarding query types, the picture looks different. Only about 16.8% of the SNP queries are left unquoted by a majority of annotators, whereas for other queries this is the case about 30.5% of times.

### *Question 3: Consensus Formation.*

To examine the consensus formation in the Webis-QSeC-10, we analyze the distribution of votes per query. Let the different segmentations $S_i$ for a query $q$ be ordered by decreasing number $v_i$ of votes they got (i.e., the number of times an $S_i$ was submitted). Hence, $S_1$ is the segmentation with most votes, $S_2$ the one with second-most votes, etc. For the 1072 queries in the corpus that have more than 10 votes, we normalize the votes to sum up to 10.

For 2774 queries (about 5%), all 10 annotators choose the same segmentation. Hence, unanimity is rather an exception. We thus also analyze how often at least an absolute majority of annotators agree on a query. We consider a segmentation $S_1$ to be chosen by an absolute majority iff $v_1 \geq 6$ or $(v_1, v_2) = (5, 1)$ (i.e., at least 6 annotators agree on one segmentation, or 5 agree and the other 5 each vote for pairwise different segmentations). In the Webis-QSeC-10, about 58.5% of the queries possess a segmentation that got absolute majority. This portion decreases with query length: for 3–4 words, about 64% of the queries have an absolute majority segmentation, while the ratio is about 40% for all other lengths.

Regarding query types, one might assume that SNP queries have a higher tendency of possessing an absolute majority segmentation but this is not the case: for SNP queries, about 57.4% have an absolute majority segmentation, while this is the case for 59.3% of the other queries. When combined with Question 2, however, the picture changes: for the 5607 SNP queries where an unquoted segmentation gets the most votes, only 44% achieve an absolute majority. In contrast, for the non-SNP queries, an absolute majority for not segmenting is the case for 62% of the 9778 cases. Hence, for other queries, the decision not to quote gets higher support on average.

Altogether, our analysis of how humans quote queries has two implications. First, it suggests that algorithms aiming at accuracy against human segmentations should take into account the query type. On SNP queries, a more aggressive strategy (more keywords in segments) might be reasonable than on other queries. Our new query segmentation approach operationalizes this idea (cf. Section 6). The second implication is to carefully reconsider the traditional accuracy measures. Unanimity among the annotators is the clear exception. Together with the size of the Webis-QSeC-10, this renders traditional measures less meaningful (cf. Section 4).

# 4. ACCURACY MEASURES REVISITED

Measuring the accuracy of a query segmentation algorithm seems to be a straightforward matter: set up a query corpus which contains a reference segmentation for each query, segment the queries using the algorithm in question, and then compare the algorithm's output for each query with its reference segmentation. A slight difficulty might be how to compare two segmentations in order to quantify how well one matches the other, but this has been addressed long ago. Yet, this is not the whole story.

The above setup assumes that, for every query in existence, there is exactly one true reference segmentation, which is obviously wrong. Bergsma and Wang [5] were the first to stumble upon this issue when they decided to have their corpus segmented not by one but three annotators. They observed that unanimity among their annotators was reached only on 40% of 500 queries, while on the remainder at least one annotator disagreed. Unfortunately, Bergsma and Wang then simply considered all obtained segmentations valid references as if they came from expert annotators. But it is an easy exercise to come up with a query that can be segmented properly only by a domain expert, whereas three people cannot be experts in all domains. This was previously observed by Hagen et al. [11], who found a number of errors in the Bergsma-Wang corpus, and who went on to enrich the Bergsma-Wang corpus by crowdsourcing ten segmentations per query from ten random people of their AMT worker pool for the Webis-QSeC-10. However, evaluation against the Webis-QSeC-10 so far again resorted to the basic idea of Bergsma and Wang, namely the top 3 segmentations which got the most votes were considered equally good points of reference. We argue that this is an oversimplification, and that scoring reference segmentations from a set of (weighted) alternatives is an integral part of properly measuring segmentation accuracy.

We first review the traditional segmentation accuracy measures, including accuracy quantification and simplistic reference selectors (cf. Sections 4.1 and 4.2). Second, we introduce new, more robust reference selectors that take into account annotator (dis)agreement (cf. Section 4.3). Third, we experimentally validate the new reference selectors compared to the traditional ones (cf. Section 4.4).

## 4.1 Quantifying Segmentation Accuracy

Given a query $q$, its reference segmentation $S'$, and an algorithm's segmentation $S$, there are three levels at which the accuracy of $S$ under $S'$ can be quantified:

*Query Level.* At query level, $S$ is correct if it contains exactly the same segments as $S'$. The *query accuracy* is 1 if $S = S'$, and 0 otherwise.

*Segment Level.* At segment level, $S$ is matched with $S'$ using precision and recall. The *segment precision* is the ratio of segments in $S$ that are also in $S'$, while *segment recall* is the ratio of segments in $S'$ that are also in $S$. Both measures can be combined via their harmonic mean as *segment $F$-Measure.*

*Break Level.* At break level, for all pairs of consecutive words in $q$, $S$ "decides" whether or not a segment break is included. The *break accuracy* hence is the ratio of correct decisions over all break positions in $q$ with respect to $S'$.

For example, consider the query $q =$ `new york times square` and $S' =$ `"new york" "times square"` as reference segmentation. Let an algorithm's segmentation be $S =$ `"new york" times square`. Obviously, $S \neq S'$ so that the query accuracy is 0. At segment level, $S$ contains one of the two reference segments, which gives a segment recall of 0.5. The other two one-word segments in $S$ are false, yielding a segment precision of 0.333, and a segment $F$-Measure of 0.4. The break accuracy is 0.666, since $S$ decides incorrectly for one of three break positions. The example shows a hierarchy of detail: an error at break level immediately implies a query accuracy of 0, while at segment level the effect is not that severe and even less so at break level.

On a corpus of segmented queries, the measures are averaged over all queries. But as mentioned above, the crucial point is that of selecting an appropriate reference segmentation $S'$.

## 4.2 Traditional Reference Selection Debunked

Given a query $q$, and a list of $m$ reference segmentations $(S'_1, \ldots, S'_m)$ from $m$ different annotators, the question is which one to select for comparison to an algorithm's segmentation $S$. In the literature, four reference selection strategies have been used:

*One Annotator.* This strategy selects the $S'_i$ from annotator $i$.

*Unanimity.* This strategy evaluates only on queries where $S'_1 = \ldots = S'_m$ (i.e., all annotators agree on the same segmentation). All other queries are dropped from the evaluation.

*Best Fit.* This strategy selects the $S'_i$ which maximizes the break accuracy compared to $S$ (i.e., the $S'_i$ most similar to $S$).

*Top 3 Best Fit.* This strategy applies best fit on the top 3 reference segmentations that got the most votes from the annotators.

These strategies work well as long as the number of annotators $m$ is small, but as $m$ increases, they fall apart. With the Bergsma-Wang corpus (three annotators) or that of Bendersky et al. [3] (just one annotator), segmentation accuracy is typically reported for each annotator separately using the one annotator strategy. This is possible since the number of annotators is small and each annotator has segmented all queries. With the Webis-QSeC-10, however, there are 1795 annotators, none of whom segmented the entire corpus but only very small parts. This renders the one annotator strategy pointless. The same holds for the unanimity strategy, since all annotators of a query agree on only 5% of the Webis-QSeC-10 so that 95% of the corpus would be discarded.

The best fit strategy does not produce meaningful results either. Consider for instance the query `new york times` for which nine annotators choose $S'_1 =$ `"new york times"` but one chooses $S'_2 =$ `new york times` (i.e., no quotes at all). Let A and B denote segmentation algorithms to be compared and let A output $S_A = S'_1$ while B outputs $S_B = S'_2$. Obviously, A reflects the majority of annotators better than B. However, against the best fit reference, both systems achieve optimum accuracy at all three accuracy levels: best fit independently selects $S'_1$ for system A and $S'_2$ for system B. Observe that, in practice, this example is not an exception, but the rule: 71% of the queries in the Webis-QSeC-10 have at least one vote for the unquoted segmentation (cf. Section 3.3). Therefore, an approach that always chooses not to segment would achieve a query accuracy of 0.71 under best fit reference selection despite the fact that unquoted queries get the majority of votes only for about 24% of the queries.

As a remedy, Hagen et al. [11] use the top 3 best fit strategy. At first glance, this strategy resolves some problems with best fit. Given a query with a 5:2:2:1 vote distribution, the top-3 selection discards the one-vote segmentation. Note, however, that about 70% of the queries got at most 3 different segmentations (e.g., the 9:1 example from above) such that top 3 best fit often is equivalent to best fit. Also, vote distributions such as 4:2:2:2 pose a problem, since three segmentations are tied on second place so that none can be discarded. Altogether, the major problem with the two best fit strategies is that they do not take into account the number of votes a segmentation obtained as well as the vote distribution.

## 4.3 Rethinking Reference Selection

We now introduce new reference selection strategies that are robust against the aforementioned problems of the traditional strategies. Given a query $q$, and a list of $m$ reference segmentations $(S'_1, \ldots, S'_m)$ from $m$ different annotators, we propose the following strategies to select a reference segmentation:

*Weighted Best Fit.* This strategy selects the $S'_i$ like the best fit strategy (i.e., the $S'_i$ which maximizes break accuracy). But then, the obtained accuracy scores are weighted by the ratio of votes allotted to $S'_i$ compared to the maximum number of votes on any segmentation in $(S'_1, \ldots, S'_m)$.

*Weighted Best Fit Unless Absolute Majority.* This strategy selects the $S'_i$ chosen by an absolute majority of votes if there is one, otherwise it applies weighted best fit.

*Break Fusion.* This strategy works at break level. Instead of selecting a reference segmentation from $(S'_1, \ldots, S'_m)$, it fuses them into one. The fusion happens independently for each break position in $q$: if at least half of the annotators inserted a segment break between a pair of words, so does this strategy. If not, no break is inserted.

The idea of the weighted best fit strategy follows directly from the aforementioned problems of best fit. With this strategy, system B from the 9:1 `new york times` example, achieves only 1/9 of its perfect accuracy on segmentation $S'_2$.

Although this is much fairer than the traditional best fit, one might argue that system B disagrees with an absolute majority of annotators on each break position so that it "deserves" an accuracy of 0 and not $1/9$. This leads to the idea of the absolute majority strategy which selects the absolute majority segmentation as reference if there is one (true for 58.5% of the queries in the Webis-QSeC-10) and resorts to weighted best fit otherwise.

The break fusion strategy introduces a novel idea to reference selection, namely reference generation. The basic concept is that even for queries without absolute majority segmentation, many annotators at least agree on specific important segments that then should form the reference. Therefore, break fusion simply follows the majority of annotators at any one break position of a query. In case of a tie vote, a break is inserted. In the Webis-QSeC-10 there are 61 123 keyword pairs where an annotator majority chooses not to break (37.0% of all the 165 107 keyword pairs), 10 261 pairs (6.2%) with a tie, and 93 723 pairs (56.8%) with a majority for breaking.

For an illustration of the break fusion strategy, consider the query `how much costs new york times`: five annotators vote for `how much costs "new york times"`, four vote for `"how much costs" "new york times"`, and one votes for the unquoted `how much costs new york times`. At query level, there is no absolute majority. But at break level, the picture looks different (non-break votes | break votes):

| how | much | costs | new | york | times |
|---|---|---|---|---|---|
| 4 | 6 4 | 6 0 | 10 9 | 1 9 | 1 |

Following the majority at each break position, the generated reference segmentation is `how much costs "new york times"`.

For queries with an absolute majority segmentation, break fusion produces exactly that segmentation as the reference (i.e., it is equivalent to the absolute majority strategy on such queries). For other queries, break fusion identifies important segments and might even generate a segmentation none of the annotators actually suggested.

## 4.4 Reference Selection at Work

To demonstrate the impact of the new reference selectors on accuracy measurement, we apply them in a comparison of the segmentation algorithms from the literature on the so-called enriched Bergsma-Wang corpus (492 queries, 10 annotators each). This corpus is an error-corrected version of the Bergsma-Wang corpus, forming a subset of the Webis-QSeC-10 [11]. Table 4 shows the results. There is a column for the accepted pointwise mutual information (PMI) baseline and one for every algorithm that has been evaluated against the Bergsma-Wang corpus so far. Each row in the table corresponds to a combination of one of the new reference selectors with one of the accuracy measures.

Note that the accuracies of almost all methods appear to be convincing with the traditional top 3 best fit reference selector. However, the accuracies of all algorithms drop as a result of applying the new reference selectors. Also, many of the relative differences between segmentation algorithms increase. More importantly, more of these differences become statistically significant, which demonstrates the positive impact of the new reference selectors: the algorithms can be better differentiated even on this small corpus. Consider, for instance, the top five algorithms. With the top 3 best fit selector, their differences in query accuracy are not statistically significant. But with the weighted best fit selector, at least the difference in query accuracy between the two algorithms [11] and [20] becomes significant. With the break fusion selector, also the difference in query accuracy between [11] and [5] becomes significant. Also, the ranking of the algorithms (as visualized by cell shading) changes severely compared to the traditional top 3 best fit selector.

Altogether, the new reference selectors provide a more robust and fair means to evaluate query segmentation accuracy. We employ them in the evaluations of the following sections.

# 5. NEW BASELINES: WIKIPEDIA + SNPS

Our second contribution to evaluation methodology consists of two new baseline algorithms. They are inspired by the facts that many segmentation algorithms exploit titles of Wikipedia articles [10, 11, 18] and that Zhang et al. have shown the potential of Wikipedia titles for improving retrieval [21]. However, using only Wikipedia titles as segments has not been evaluated before.

### *Wikipedia Titles (WT).*

Given a query $q$, the Wikipedia title baseline is supposed to use only Wikipedia titles as segments. As simple as this may be, a decision rule is required for the special case of overlapping Wikipedia titles (happens for 3645 queries in the Webis-QSeC-10) in order to decide which of the overlapping titles are used as segments.

*Regions* in a query $q$ are formed by the transitive hulls of overlapping Wikipedia titles. Each such region is handled separately. If it contains just one title, it is simply segmented as one segment. If it contains more than one title, the WT baseline uses the following decision rule proposed in [11]. First, a weight is assigned to each title $w$, which is the maximum web occurrence frequency of any of its sub-2-grams multiplied by $|w|$ (the number of words in $w$). These weights can be statically precomputed (e.g., using the frequencies in the Google $n$-gram corpus [6]) and stored in a hash table. In a second step, each valid segmentation $S$ of the region is scored by the summed weights of the segments in $S$. Finally, the segmentation with the highest score is chosen.

For example, consider the query $q$ = `where in new york is new york yankees stadium`, which contains the potential Wikipedia title segments `new york`, `new york yankees`, and `yankees stadium` (one-word titles like `york` are not considered for segmentation). Two disjunct regions can be identified: the first region is the first occurrence of `new york`, consisting of just one title. The second region is `new york yankees stadium` consisting of three titles. Although the appearance of `new york` in the second region does not overlap with `yankees stadium`, they

**Table 4: Accuracy of state-of-the-art segmentation algorithms on the enriched Bergsma-Wang corpus, dependent on the reference selector applied. The darker a cell, the better the value compared to the rest of the row. Algorithms are grouped so that inter-group differences are mostly statistically significant and intra-group differences not (paired t-test, $p = 0.05$). Note however, that, under the three new reference selectors, some differences of [11] and WT+SNP to other members of their group become significant.**

| **Accuracy measure** | | **Algorithm** (significance groups) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Selector | Acc. level | [15] | PMI | [14] | [8] | WT | [10] | [20] | [5] | [11] | WT+SNP |
| top 3 best fit | query | 0.522 | 0.732 | 0.722 | 0.711 | 0.793 | 0.764 | 0.783 | 0.805 | 0.807 | 0.819 |
| | seg prec | 0.629 | 0.779 | 0.783 | 0.781 | 0.844 | 0.827 | 0.844 | 0.857 | 0.861 | 0.87 |
| | seg rec | 0.678 | 0.78 | 0.796 | 0.782 | 0.863 | 0.827 | 0.843 | 0.868 | 0.856 | 0.867 |
| | seg F | 0.652 | 0.779 | 0.79 | 0.781 | 0.853 | 0.827 | 0.844 | 0.862 | 0.859 | 0.869 |
| | break | 0.765 | 0.876 | 0.872 | 0.86 | 0.902 | 0.896 | 0.909 | 0.918 | 0.914 | 0.92 |
| weighted best fit | query | 0.346 | 0.575 | 0.576 | 0.6 | 0.617 | 0.671 | 0.662 | 0.682 | 0.713 | 0.718 |
| | seg prec | 0.45 | 0.616 | 0.632 | 0.663 | 0.666 | 0.727 | 0.717 | 0.73 | 0.76 | 0.763 |
| | seg rec | 0.499 | 0.619 | 0.645 | 0.664 | 0.686 | 0.729 | 0.717 | 0.74 | 0.756 | 0.761 |
| | seg F | 0.473 | 0.617 | 0.638 | 0.664 | 0.676 | 0.728 | 0.717 | 0.735 | 0.758 | 0.762 |
| | break | 0.583 | 0.706 | 0.713 | 0.736 | 0.723 | 0.788 | 0.775 | 0.786 | 0.809 | 0.808 |
| weighted best fit unless absolute majority | query | 0.286 | 0.513 | 0.526 | 0.563 | 0.548 | 0.637 | 0.62 | 0.64 | 0.677 | 0.678 |
| | seg prec | 0.452 | 0.594 | 0.635 | 0.664 | 0.663 | 0.728 | 0.718 | 0.732 | 0.759 | 0.761 |
| | seg rec | 0.529 | 0.601 | 0.658 | 0.669 | 0.705 | 0.731 | 0.719 | 0.756 | 0.753 | 0.759 |
| | seg F | 0.487 | 0.598 | 0.647 | 0.667 | 0.683 | 0.729 | 0.719 | 0.744 | 0.756 | 0.76 |
| | break | 0.645 | 0.763 | 0.772 | 0.774 | 0.789 | 0.825 | 0.824 | 0.833 | 0.842 | 0.846 |
| break fusion | query | 0.183 | 0.435 | 0.441 | 0.47 | 0.431 | 0.547 | 0.555 | 0.53 | 0.587 | 0.585 |
| | seg prec | 0.377 | 0.548 | 0.58 | 0.601 | 0.588 | 0.663 | 0.668 | 0.653 | 0.701 | 0.701 |
| | seg rec | 0.466 | 0.562 | 0.607 | 0.611 | 0.643 | 0.671 | 0.673 | 0.688 | 0.697 | 0.702 |
| | seg F | 0.417 | 0.555 | 0.593 | 0.606 | 0.614 | 0.667 | 0.67 | 0.67 | 0.699 | 0.701 |
| | break | 0.613 | 0.772 | 0.77 | 0.773 | 0.769 | 0.818 | 0.826 | 0.82 | 0.837 | 0.837 |

are transitively linked via the title `new york yankees`. The title `new york yankees` from the second region gets thrice the score of `new york` so that the segmentation `"new york yankees" stadium` is preferred over `"new york" "yankees stadium"`. Altogether, the WT baseline outputs `where in "new york" is "new york yankees" stadium`.

### *Wikipedia Titles and Strict Noun Phrases (WT+SNP).*

The WT baseline segments only entire Wikipedia titles, neglecting other potential segments. Its design is based on the three facts that baseline algorithms should be simple, Wikipedia titles are well-known concepts, and that not every frequent phrase on the web is a good segment. However, segmenting only Wikipedia titles might be considered a bit too conservative. Following Zhang et al. [21] who also showed the positive effect of noun phrases on retrieval performance, we propose a generalization of the WT baseline which segments strict noun phrases in addition to Wikipedia titles. Regions now consist of overlapping Wikipedia titles and SNPs and are treated similar to the WT baseline. Thereby, the normalized weights of SNPs that are not Wikipedia titles are obtained via multiplying their web frequencies by their length. Observe that both baselines can be viewed (and implemented) as special cases of the segmentation approach described in [11], whereas the difference is a limited "dictionary" of allowed segments (i.e., only Wikipedia titles and SNPs instead of all web phrases).

### *Evaluation.*

On the enriched Bergsma-Wang corpus, the WT baseline significantly improves on the PMI baseline under most reference selectors (cf. Table 4). While it ranges among the most accurate algorithms regarding the traditional top 3 best fit selector, its accuracy drops regarding the more advanced selectors. Under break fusion, the WT baseline falls in the same accuracy group as the PMI baseline. Keeping in mind the corpus consists of noun phrase queries only, not surprisingly, the WT+SNP baseline almost outperforms [11].

## 6. HYBRID QUERY SEGMENTATION

The decision whether or not to introduce segments into a query is a risky one: a bad segmentation leads to bad search results or none at all, whereas a good one improves them. Since keeping users safe from algorithm error is a core principle at most search engines, and since even a small error probability yields millions of failed searches given billions of searches per day, a risk-averse strategy is the way to go. In doubt, it is always safer to do without any query segmentation, and to rely on existing, well-tested technology. But that is not to say that all query segmentation is futile, on the contrary: whenever one can be sure beyond doubt that introducing segments will not harm overall retrieval performance, it should definitely be done in order to be of better assistance to the user. This observation obviously suggests to use a hybrid strategy that treats different types of queries in different ways. But none of the approaches proposed in the literature implements this idea, yet. So far, queries are always treated the same regardless of their type.

The most important question for a hybrid strategy is how to distinguish low-risk queries from high-risk ones? In this connection, some solutions come to mind, such as employing other query predictors. However, we do not wish to supplement one cutting-edge technology with another as the errors of one increase those of the other. Instead, we revisit our study of how humans quote (see Section 3.3). One of the main findings is that humans quote differently on strict noun phrase queries compared to others. Hence, it is reasonable to distinguish these two query types and to approach them differently. To operationalize this idea, we reuse existing algorithms as sub-routines in our hybrid strategy. First, we employ the strict noun phrase query detector introduced in Section 3.2 to tell SNP queries and other queries apart. Second, dependent on the detector's decision, queries are segmented with the segmentation algorithm best suited for the respective query type. The question which algorithm to choose for which query type under which circumstances is subject to our evaluation in the next section.

**Table 5: The Wikipedia title baseline (WT) compared to the previously most accurate algorithm [11] on the Webis-QSeC-10 training set (4850 queries). Accuracy is given on all queries, and dependent on the query types strict noun phrase (SNP) queries and other queries. All differences between the two algorithms are significant, except for seg rec and seg F on SNP queries (paired t-test, $p = 0.01$).**

| **Accuracy measure** | | **All queries** | | **SNP queries** | | **Other queries** | |
|---|---|---|---|---|---|---|---|
| Selector | Acc. level | [11] | WT | [11] | WT | [11] | WT |
| break fusion | query | 0.407 | **0.495** | **0.570** | 0.495 | 0.260 | **0.495** |
| | seg prec | 0.591 | **0.658** | **0.663** | 0.622 | 0.526 | **0.690** |
| | seg rec | 0.555 | **0.697** | 0.650 | **0.665** | 0.470 | **0.725** |
| | seg F | 0.573 | **0.677** | **0.656** | 0.643 | 0.496 | **0.707** |
| | break | 0.715 | **0.772** | **0.773** | 0.750 | 0.664 | **0.791** |

# 7. EVALUATION

In our evaluation, we compare instances of hybrid query segmentation to traditional approaches with respect to three performance measures. (1) We measure segmentation accuracy against the manually quoted queries of the Webis-QSeC-10. (2) We measure retrieval performance in a TREC setting using the two retrieval models employed by the commercial search engine Bing and the Indri ClueWeb09 search engine hosted at Carnegie Mellon University.[4] (3) We measure runtime performance and memory footprint.

We have systematically combined traditional segmentation algorithms (including the option "none" of not segmenting) to form instances of hybrid segmentation. As expected, there is no one-fits-all combination which maximizes performance with respect to all of the above measures. To cut a long story short, three instances of hybrid segmentation perform best in terms of segmentation accuracy and retrieval performance using Bing and Indri, respectively. The following confusion matrix shows the traditional segmentation algorithms employed by each of them, dependent on query type:

| **Query type** | **Hybrid segmentation instance** | | |
|---|---|---|---|
| | HYB-A (accuracy) | HYB-B (Bing) | HYB-I (Indri) |
| SNP | [11] (= WT+SNP) | None | None |
| other | WT | WT | [11] |

In what follows, we detail the experiments that lead to this result.

## 7.1 Segmentation Accuracy

Until now, measuring segmentation accuracy against manually quoted queries has been the predominant evaluation method employed in the literature. Today, the largest and most representative corpus of manually quoted queries is the Webis-QSeC-10. Robust performance measures for this corpus which compare automatically segmented queries to sets of manual quotations have been described at length in Section 4.

### *Accuracy-oriented Hybrid Segmentation (HYB-A).*

In a pilot study, we have employed the Webis-QSeC-10 training set and our new accuracy measures to determine which combination of traditional segmentation algorithms maximizes segmentation accuracy. It turns out that this is the case for [11] on SNP queries and the WT baseline for all other queries. Note that [11] is equivalent to WT+SNP on SNP queries. The respective accuracies on the training set are shown in Table 5. Interestingly, and in contrast to previous evaluations against the enriched Bergsma-Wang corpus, the WT baseline clearly outperforms [11] (columns "All queries"). However, regarding SNP queries only, it is the other way around.

**Table 6: Accuracy on the Webis-QSeC-10 test set. The darker a cell, the better the value compared to the rest of the row. Most pairwise differences (especially on query level) are statistically significant (paired t-test, $p = 0.01$).**

| **Accuracy measure** | | **Algorithm** | | | | | |
|---|---|---|---|---|---|---|---|
| Selector | Acc. level | [10] | [11] | PMI | WT+SNP | WT | HYB-A |
| weighted best fit unless absolute majority | query | 0.448 | 0.481 | 0.520 | 0.600 | 0.638 | 0.644 |
| | seg prec | 0.618 | 0.642 | 0.638 | 0.722 | 0.744 | 0.747 |
| | seg rec | 0.582 | 0.603 | 0.627 | 0.702 | 0.759 | 0.745 |
| | seg F | 0.599 | 0.622 | 0.632 | 0.712 | 0.751 | 0.746 |
| | break | 0.708 | 0.722 | 0.736 | 0.775 | 0.800 | 0.797 |
| break fusion | query | 0.383 | 0.414 | 0.425 | 0.517 | 0.501 | 0.534 |
| | seg prec | 0.582 | 0.608 | 0.588 | 0.682 | 0.672 | 0.692 |
| | seg rec | 0.546 | 0.566 | 0.583 | 0.662 | 0.707 | 0.699 |
| | seg F | 0.564 | 0.586 | 0.585 | 0.672 | 0.689 | 0.696 |
| | break | 0.704 | 0.718 | 0.731 | 0.770 | 0.777 | 0.784 |

Table 6 shows an evaluation of the resulting HYB-A instance on the Webis-QSeC-10 test set (48 587 queries) compared to the baseline segmentation algorithms and the two best performing traditional segmentation algorithms from the literature. We employ our two reference selectors weighted best fit unless absolute majority and break fusion. As expected from the above pilot study, HYB-A outperforms its sub-routines and is the most accurate approach overall, especially on query level. Note that, different to previous evaluations, the PMI baseline performs better than the state-of-the-art approach [11]. This is probably due to the new reference selectors that allow for a more fine-grained performance assessment.

### *Discussion.*

An explanation for the specific combination of traditional segmentation algorithms in HYB-A can be found in our analysis of human quoting behavior. There, it is shown that accuracy-oriented algorithms should segment SNP queries more aggressively (more keywords in segments) than other queries, which in turn should be segmented conservatively (less keywords in segments). This is exactly the strategy of HYB-A. On SNP queries, the algorithm [11] aggressively segments all phrases that appear at least 40 times on the web, whereas the WT baseline on the other queries conservatively segments only Wikipedia titles.

## 7.2 Retrieval Performance

Measuring segmentation accuracy alone does not tell much about whether a segmentation algorithm actually improves retrieval performance. The only way to check this hypothesis is to employ a given segmentation algorithm in a retrieval pipeline and measure its impact on search results using a set of queries for which relevant documents are known in advance. This has been largely neglected in the literature, and therefore we conduct a TREC style experiment on the ClueWeb09 collection. From the TREC topics in the Web tracks 2009–2011 and the Million Query track 2009 with at least one document being judged as relevant, we use the 355 titles of length at least 3 keywords as our query set (61 from the Web tracks, 294 from the Million query track). We employ two search engines: the online Indri ClueWeb09 search and the Bing API. In order to simulate searching the ClueWeb09 with Bing, the top 100 results are filtered for documents contained in that corpus. Furthermore, we introduce two more baselines of interest: the optimum segmentation oracle (OPT), which always returns a segmentation that max-

[4] http://boston.lti.cs.cmu.edu/Services/batchquery

**Table 7: Retrieval performance as nDCG@10 for 355 TREC queries, 212 of which are SNPs. Algorithms are grouped so that inter-group differences are mostly statistically significant and intra-group differences are not (paired t-test, $p = 0.05$).**

*Bing search engine*

| Query type | **Algorithm** (significance groups) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | PMI | [11] | WT | HYB-A | None | WT+SNP | [10] | HYB-B | OPT |
| all | 0.138 | 0.143 | 0.143 | 0.144 | 0.144 | 0.144 | 0.144 | 0.148 | 0.170 |
| SNP | 0.127 | 0.131 | 0.130 | 0.131 | 0.138 | 0.131 | 0.133 | 0.138 | 0.156 |
| other | 0.154 | 0.159 | 0.162 | 0.161 | 0.154 | 0.162 | 0.160 | 0.162 | 0.189 |

*Indri ClueWeb09 search engine*

| Query type | **Algorithm** (significance groups) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | None | PMI | WT | WT+SNP | HYB-A | [10] | [11] | HYB-I | OPT |
| all | 0.170 | 0.175 | 0.188 | 0.192 | 0.193 | 0.226 | 0.228 | 0.228 | 0.308 |
| SNP | 0.232 | 0.210 | 0.224 | 0.232 | 0.232 | 0.232 | 0.232 | 0.232 | 0.311 |
| other | 0.078 | 0.122 | 0.136 | 0.133 | 0.136 | 0.219 | 0.222 | 0.222 | 0.304 |

imizes retrieval performance on a given search engine, and the no-segmentation (None), which does not segment any query. For the optimum segmentation oracle, all valid segmentations of all queries were submitted to both search engines, recording the search results.

Table 7 shows the obtained retrieval performances measured as nDCG@10. Contained are the baselines PMI, WT, WT+SNP, None, and OPT, as well as Hagen et al.'s [10, 11], the accuracy-oriented HYB-A, and another two instances of hybrid segmentation: the Bing-oriented HYB-B, and the Indri-oriented HYB-I, which maximize performance on Bing and Indri, respectively.[5]

### *Bing-oriented Hybrid Segmentation (HYB-B).*

Regarding Bing, the unexpected outcome is that none of the algorithms perform significantly better than not segmenting at all. Since Bing must be treated as a black box in our setup, we can only speculate that its query processing pipeline somehow levels the performances of different segmentations. But there still is room for improvement: OPT significantly outperforms all other approaches. The instance of hybrid segmentation closest to the optimum oracle is HYB-B. This instance does not segment SNP queries (matching OPT performance 73 times), and employs the WT baseline on other queries (matching OPT 35 times). However, HYB-B is not significantly better than the second-best approach.

### *Indri-oriented Hybrid Segmentation (HYB-I).*

Regarding Indri, three groups of segmentation approaches can be distinguished: the group whose differences from unquoted queries are mostly not statistically significant, the group of Hagen et al.'s algorithms [10, 11] plus yet another instance of hybrid segmentation HYB-I, and the optimum oracle. Here, HYB-I again does not segment SNP queries (matching OPT performance 69 times), and employs [11] on other queries (matching OPT 43 times). However, HYB-I is not significantly better than [10, 11].

### *Discussion.*

Altogether, these results suggest that different search engines (i.e., retrieval models) each require specifically tailored (hybrid) query segmentation algorithms. Otherwise, query segmentation may not improve significantly over not segmenting at all. That said, it is interesting to observe the differences of the algorithms that maximize performance on Bing and Indri, and to compare them to accuracy-oriented segmentation:

**Table 8: Throughput and memory requirement.**

| **Perf. measure** | **Algorithm** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | HYB-A | HYB-I | HYB-B | [10] | [11] | WT+SNP | WT | PMI |
| queries/sec | 3083 | 3152 | 3625 | 3649 | 3658 | 4083 | 4379 | 27 388 |
| memory | 2.6GB | 12.5GB | 59MB | 12.5GB | 12.5GB | 2.6GB | 59MB | 1.1GB |

- Maximizing segmentation accuracy not necessarily maximizes retrieval performance as well.
- High-accuracy algorithms, such as HYB-A and WT+SNP, are not significantly better in terms of retrieval performance than not segmenting at all.
- Low-accuracy algorithms, such as [10, 11], may perform significantly better in terms of retrieval performance than high-accuracy ones.
- SNP queries can often be left unsegmented in terms of retrieval performance.
- HYB-A differs very much from HYB-B and HYB-I in terms of segmentation aggressiveness, regarding both query types.
- The segmentation accuracies of HYB-B and HYB-I are below 0.3 under the break fusion selector and thus significantly worse than all other approaches shown in Table 6.

Presently, we can offer at best partial explanations for these observations. A thorough study of the apparently contradictory behavior of query segmentation algorithms against the two evaluation paradigms segmentation accuracy and retrieval performance is an interesting question for future work. Moreover, the above findings must be taken with a grain of salt: our TREC experiments are small-scale compared to the number of queries that went into measuring segmentation accuracy. While analyzing retrieval performance is a must when developing segmentation algorithms, the available ground truth data should be scaled up significantly in order to draw more reliable conclusions. Our currently used query set contains only very few queries with 5 or more keywords, and for some of them hardly any segmentation achieves an nDCG $> 0$.

In any case, our experiments have shown that the decision of when to segment at all is an important one. We hypothesize that further types of queries may be distinguished, so that hybrid query segmentation may be tailored even better to specific retrieval situations. Since the optimum segmentation oracle convincingly demonstrates the potential of query segmentation, comparing its segmentations to those of the existing algorithms might yield new algorithmic ideas. For instance, we found segmentations from the optimum oracle to often contradict human intuition.

## 7.3 Runtime and Memory

Besides accuracy and retrieval performance, also runtime and memory consumption are crucial criteria to judge the applicability of a segmentation algorithm in a real-world setting. Runtime is typically measured as throughput of queries per second while memory consumption concerns the data needed for operation. We compare the PMI baseline, the new WT and WT+SNP baselines, Hagen et al.'s algorithms [10, 11], and the above hybrid segmentation approaches. Table 8 shows the results measured on a standard quad-core PC. Regarding memory consumption, our implementations employ the external hash table described by Brants et al. [7] to index (parts of) the Google $n$-gram corpus. The entire corpus fits into about 12.5 GB of main memory. However, as described

[5] Bing's lower retrieval performance compared to Indri is probably due to the "age" of the ClueWeb09. Many relevant web pages readily retrieved by Indri may have changed or disappeared.

earlier, many of our new approaches (as well as the PMI baseline) do not require the entire corpus. The PMI baseline requires only 1- and 2-grams along with their web frequencies, the WT baseline requires only normalized weights for Wikipedia titles, and the WT+SNP baseline requires only normalized weights for Wikipedia titles and $n$-grams which are strict noun phrases. The corresponding hash tables can be easily pre-computed offline. The instances of hybrid segmentation require the hash tables of their subroutines.

Regarding throughput, the PMI baseline is by far the fastest approach. The WT and WT+SNP baselines are faster than [10, 11] since they sum up fewer weights of potential segments. The hybrid approaches are slowest due to the POS tagging step. HYB-B and HYB-I are faster than HYB-A since they do not segment SNP queries; HYB-B is almost as fast as [10, 11].

It is rumored that the monthly throughput of major search engines is about 100 billion queries which translates to about 40 000 queries per second.[6] Typically, about half of the queries contain 3 or more keywords so that about 20 000 queries per second are amenable to segmentation. All the evaluated approaches can easily handle such a load when run on a small cluster of standard PCs.

## 8. CONCLUSION AND OUTLOOK

In the literature, the predominant approach to evaluate query segmentation algorithms is to check their output against manually quoted queries. In this paper, we conducted the first large-scale study of how humans quote queries. Especially for segmentation evaluation purposes, our study provides new insights. For instance, humans quote strict noun phrase queries (SNP queries, for short, composed of nouns, numbers, articles, and adjectives only) more aggressively than other queries. Most keywords of SNP queries end up in quoted segments, whereas the opposite is true for other queries. This finding forms the heart of our new hybrid query segmentation framework which treats SNP queries different than other queries. By reusing existing algorithms, we have tailored our hybrid segmentation framework to mimic human query quoting better than every state-of-the-art algorithm.

The paper also introduces new performance measures that measure an algorithm's segmentation accuracy against manually quoted queries. They are especially suited for large-scale corpora involving many annotators per query. The existing measures come with conceptual shortcomings that severely limit their applicability in such situations. In comparison, our measures do a much better job at differentiating segmentation algorithms. Their measurements are more fine-grained such that differences between algorithms are more often statistically significant.

The third main contribution of this paper is an experimental evaluation of query segmentation algorithms within a TREC setting. An important and somewhat unexpected outcome is that maximizing segmentation accuracy not necessarily maximizes retrieval performance as well. Our aforementioned instance of hybrid query segmentation, which outperforms all other algorithms in terms of accuracy, is itself outperformed by some algorithms in terms of retrieval performance. Nevertheless, we show the flexibility of our hybrid query segmentation framework and tailor it to two retrieval models, again outperforming all other algorithms. Here, we found that not segmenting SNP queries at all is the current best approach, which opposes our earlier finding that humans quote SNP queries more aggressively. Yet, there is still room for improvement, since our approaches are still significantly worse than the optimal segmentations obtained from a segmentation oracle.

Finally, we pinpoint many directions for future work: evaluating query segmentation in terms of retrieval performance lacks large-scale resources and should be done using a broader range of retrieval models. We hypothesize that query segmentation is especially beneficial on long non-SNP queries, which are underrepresented in the TREC corpora. Other open questions include why SNP queries apparently are better off without any segmentation, and how the contradictory performance measurements in terms of segmentation accuracy and retrieval performance can be explained. One starting point to answer these questions is an analysis of the segmentation oracle on different retrieval models in order to better understand what differentiates a "perfect" retrieval-oriented segmentation from those of the algorithms developed so far.

## 9. REFERENCES

[1] C. Barr, R. Jones, and M. Regelson. The linguistic structure of English web-search queries. In *EMNLP 2008*, pp. 1021–1030.

[2] M. Bendersky, W. B. Croft, and D. Smith. Two-stage query segmentation for information retrieval. In *SIGIR 2009*, pp. 810–811.

[3] M. Bendersky, W. B. Croft, and D. Smith. Structural annotation of search queries using pseudo-relevance feedback. In *CIKM 2010*, pp. 1537–1540.

[4] M. Bendersky, W. B. Croft, and D. A. Smith. Joint annotation of search queries. In *ACL-HLT 2011*, pp. 102–111.

[5] S. Bergsma and Q. Wang. Learning noun phrase query segmentation. In *EMNLP-CoNLL 2007*, pp. 819–826.

[6] T. Brants and A. Franz. Web 1T 5-gram Version 1. Linguistic Data Consortium LDC2006T13, Philadelphia, 2006.

[7] T. Brants, A. Popat, P. Xu, F. Och, and J. Dean. Large language models in machine translation. In *EMNLP-CoNLL 2007*, pp. 858–867.

[8] D. Brenes, D. Gayo-Avello, and R. Garcia. On the fly query entity decomposition using snippets. In *CERI 2010*, poster.

[9] A. Broschart, K. Berberich, and R. Schenkel. Evaluating the potential of explicit phrases for retrieval quality. In *ECIR 2010*, pp. 623–626.

[10] M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. The power of naïve query segmentation. In *SIGIR 2010*, pp. 797–798.

[11] M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. Query segmentation revisited. In *WWW 2011*, pp. 97–106.

[12] J. Huang, J. Gao, J. Miao, X. Li, K. Wang, and F. Behr. Exploring web scale language models for search query processing. In *WWW 2010*, pp. 451–460.

[13] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *WWW 2006*, pp. 387–396.

[14] Y. Li, B.-J. P. Hsu, C. Zhai, and K. Wang. Unsupervised query segmentation using clickthrough for information retrieval. In *SIGIR 2011*, pp. 285–294.

[15] N. Mishra, R. Roy, N. Ganguly, S. Laxman, and M. Choudhury. Unsupervised query segmentation using only query logs. In *WWW 2011 (Posters)*, pp. 91–92.

[16] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *Infoscale 2006*, paper 1.

[17] K. Risvik, T. Mikolajewski, and P. Boros. Query segmentation for web search. In *WWW 2003 (Posters)*.

[18] B. Tan and F. Peng. Unsupervised query segmentation using generative language models and Wikipedia. In *WWW 2008*, pp. 347–356.

[19] R. W. White and D. Morris. Investigating the querying and browsing behavior of advanced search engine users. In *SIGIR 2007*, pp. 255–262.

[20] C. Zhang, N. Sun, X. Hu, T. Huang, and T. Chua. Query segmentation based on eigenspace similarity. In *ACL-IJCNLP 2009*, pp. 185–188.

[21] W. Zhang, S. Liu, C. T. Yu, C. Sun, F. Liu, and W. Meng. Recognition and classification of noun phrases in queries for effective retrieval. In *CIKM 2007*, pp. 711–720.

[6] http://searchengineland.com/google-search-press-129925 (last accessed August 12, 2012)