Proceedings of the SIGIR 2007 Workshop on Focused Retrieval

Held in Amsterdam, The Netherlands,

27 July 2007.

Edited by
Andrew Trotman,
Shlomo Geva,
and
Jaap Kamps.

Proceedings of the
SIGIR 2007 Workshop on Focused Retrieval.
Held in Amsterdam, The Netherlands,
27 July 2007.

Published by:
Department of Computer Science,
University of Otago,
PO Box 56,
Dunedin,
New Zealand.

Editors:
Andrew Trotman
Shlomo Geva
and
Jaap Kamps

ISBN 978-0-473-12333-8

http://www.cs.otago.ac.nz/sigirfocus/

Preface

These proceedings contain the papers of the SIGIR 2007 Workshop on Focused Retrieval held in Amsterdam, The Netherlands on 27th July 2007. Nine papers were selected by the program committee from fifteen submissions (60% acceptance rate). Each paper was reviewed by at least two members of the program committee.

When reading this volume it is necessary to keep in mind that these papers represent the opinions of the authors (who are trying to stimulate debate). It is the combination of these papers and the debate that is will make the workshop a success.

We would like to thank the ACM and SIGIR for hosing the workshop. Thanks also go to the program committee, the paper authors, and the participants, for without these people there would be no workshop.

Andrew Trotman
Shlomo Geva
Jaap Kamps

Workshop Organization

Programme Chairs

Andrew Trotman
Shlomo Geva
Jaap Kamps

Programme Committee

James Allan
Charles Clarke
Shlomo Geva
David Hawking
Jaap Kamps
Mounia Lalmas
Christof Monz
Maarten de Rijke
Andrew Trotman
Ellen Voorhees

Table of Contents

Structural Relevance in XML Retrieval Evaluation

M. S. Ali
University of Toronto
sali@cs.toronto.edu

Mariano P. Consens
University of Toronto
consens@cs.toronto.edu

Mounia Lalmas
Queen Mary, University of London
mounia@dcs.qmul.ac.uk

ABSTRACT

Determining the effectiveness of XML retrieval systems is crucial for improving information retrieval from XML document collections. Traditional effectiveness measures do not address the problem of overlap in the recall-base. At the Initiative for the Evaluation of XML retrieval (INEX), extended cumulated gain (XCG) was developed to address overlap. It works by comparing the cumulated score of a retrieval result to an ideal result. The use of XCG is contingent on being able to define an ideal recall-base for every topic.

This paper introduces an alternative approach called structural relevance (SR) which addresses overlap by extending relevance to overlapping, non-disjoint elements. SR models the user process of browsing overlapped elements in a ranked list using XML summaries (bisimilarity-based graph representations of the structure of a collection of XML documents) to describe the user process in terms of the structure of the collection. We show how SR is incorporated into traditional relevance-based measures and illustrate the behavior of SR in comparison to XCG. Our results suggest that SR can evaluate XML retrieval systems as effectively as XCG without requiring an ideal recall-base.

1. INTRODUCTION

The Initiative for the Evaluation of XML retrieval (INEX) is a collaborative, international effort to develop effective XML retrieval systems. At INEX, a recognized challenge in evaluating XML retrieval systems has been the overlap problem [11]. Overlap occurs when a user finds a ranked element more than once in the process of evaluating a ranked list of elements. Numerous proposals have been made to address the problem [2, 20, 12]. Overlap occurs because a user can access retrieval elements directly from either the ranked list or from following the structural paths between elements while browsing. Overlapped elements result in poor user satisfaction of retrieval systems [21] because the user perceives that the search results contain repetitive answers [5]. Moreover, overlap invalidates the use of traditional relevance-based effectiveness measures because it repudiates the basic information retrieval assumption of independence of retrieval elements when determining their relevance.

The official metric for evaluation of system effectiveness for INEX 2002 to 2004 was precall [7]. Precall is the expected precision of a result at a given recall level where the system is weakly ordered [18] or, in other words, where a user assesses tie-ranked elements in a random order. Precall does not address overlap in the recall-base and this has motivated the development of other measures which do. In 2005, INEX adopted the extended cumulated gain (XCG) as its official measure [10]. XCG is based on cumulated gain (CG) [8] where each element in a ranked list contributes to the overall relevance, or what is referred to as the gain, of the list. The cumulated gain is calculated by summing the scores of elements from the head of the list to a fixed rank position. An ideal recall-base defines the maximum possible cumulated gain for all ranks. The ratio of cumulated gain to ideal cumulated gain shows how closely a given ranked list compares to the ideal list. XCG extends CG by incorporating heuristics to model the effects of overlap on relevance scores. It is important to recognize XCG's dependency on the methodology used to build an ideal recall base (as shown in [9]).

We approach the problem of overlap in the recall-base by revisiting the notion of relevance for non-disjoint (*i.e.*, overlapping) elements. In this regard, we differentiate between the relevance of a single retrieval element and the relevance of a retrieval element as a member of a set of elements. A human judge assesses the relevance of an element to a topic with the assumption that the element is independent of all other elements; whereas, the relevance of an element in a set of non-disjoint elements is the result of how a human uses the set to fulfill their information need. In this regard, isolation, which is the probability of first encountering an element from a ranked list composed of a given set of elements, is a measure of the expected relevance of the element as a member of the given set. The overall relevance of a ranked list of structurally non-disjoint elements must be assessed in terms of a user model that describes both how the list is used and how relevant the list is to the user for answering a given query. We call this the *structural relevance* of the list. If we consider the ranked list as an affordance for a user to traverse through retrieval results in an orderly manner, then, in XML retrieval, the structural relevance is the expected number of relevant elements found using a weakly ordered list.

SIGIR 2007 Workshop on Focused Retrieval
July 27, 2007, Amsterdam, The Netherlands

In this paper, we present the definition of structural relevance, show how it is incorporated into the evaluation of XML retrieval systems, and provide experimental results to compare SR to XCG. We restricted our experiments here to comparing measures for three runs for top-10 results across many INEX topics for three reasons. The first reason was that we noticed during the development of structural relevance that our results were sometimes different from XCG for certain topics, so we restricted the runs to systems that have performed well over the years at INEX. We restricted our presented results here to a small k (in our case, k=10) over a large number of topics so that we could clearly show through simple examples the differences between what the XCG and SR measures capture. Thirdly, we restricted the runs to the thorough task in INEX because it allows overlap. In future work, we intend to extend the application of SR to all retrieval tasks in INEX. The next two sections of the paper introduce SR and describe a (summary-based) approximation technique for computing SR. Section 4 surveys and compares existing measures and Section 5 presents the experimental results. Conclusions and future work are discussed in Section 6.

2. STRUCTURAL RELEVANCE

In this section we derive the measure of structural relevance based on isolation and overlap, then we show how structural relevance is used to modify measures such as precision and precall, and, finally, we derive a general expression for calculating isolation for weakly ordered ranked lists.

Structural relevance is a measure of the relevance of a group of overlapped elements. The problem of overlap occurs in a ranked list when a user finds some ranked element while browsing a different ranked element. This will occur because the element is reachable either directly from the ranked list or indirectly via structural paths from a different element in the list. We define structural relevance as the expected number of relevant elements that are found by the user while browsing a ranked list of elements for the first time. If there is no overlap in a group of elements, then structural relevance reduces to the number of relevant elements in the set. The following theorem shows how to calculate structural relevance.

THEOREM 2.1. *Measure of Structural Relevance*
The expected number of relevant elements up to some given element u in the ordered set of elements R where $p(e;R)$ denotes the probability of the user first encountering element e from the ranked list R is

$$E[n_R(u)] = \sum_{e \in R[u]} rel(e) \cdot p(e; R[u]) \quad (1)$$

PROOF. The number of relevant elements of a ranked list R can be written as $n_R = \sum_{e \in R} rel(e)$, where $rel(e)$ is the binary relevance of element e to some given topic such that, if e is relevant then $rel(e) = 1$, and $rel(e) = 0$ otherwise. Binary relevance is used here for simplicity and there is no reason that other ways of measuring relevance could not be used.

For recall-precision calculations, the number of relevant elements are calculated at given ranks. Consider element $u \in R$ where $R[u]$ is the ordered subset (the ranking) of elements from R that contains elements up to an element u from the weakly ordered list R. We can rewrite n_R as a function of the element u in list R as,

$$n_R(u) = \sum_{e \in R[u]} rel(e)$$

Assume that system evaluation is conducted in multiple, independent trials for each returned element. So, in top-k search there will be k trials to determine the relevance of all returned elements. Each trial is done in order to determine the relevance of the element e in the ranked list R. The trials are conducted in descending order by rank of elements. Given that k trials are conducted for k ranked elements, it is certain that all elements will have known relevance (*i.e.*, total probability for any element having known relevance after evaluation is 1). Consider the element e, its relevance is determined by either first encountering it from the ranked list, or by first encountering it from an higher or tie ranked element. For element e we get the probabilistic relationship of evaluation,

$$\begin{aligned} 1 = {} & P(\text{encounter } e \text{ first from ranked list}) + \\ & P(\text{encounter } e \text{ first from an overlapped element }) \end{aligned}$$

Let $p(e;R)$ denote the probability of encountering e first from the ranked list. Let $q(e;R)$ denote the probability of encountering e first from an overlapped element so, $1 = p(e;R) + q(e;R)$. We refer to $p(e;R)$ as the *isolation* of e in the ranked list R. We refer to $q(e;R)$ as the *overlap* of e in the ranked list R. To find the relevance up to some element $u \in R$, we take the expectation of relevance on the isolation of elements in $n_R(u)$, and thus we get our desired result,

$$E[n_R(u)] = \sum_{e \in R[u]} rel(e) \cdot p(e; R[u])$$

□

2.1 Precision and Precall with SR

The measure of structural relevance (SR) can be substituted into an evaluation measure for the number of relevant elements in a ranked list. SR makes the measure sensitive to structural overlap among relevant retrieval elements, assuming a user model of ranked traversal and weak ordering. For instance, in precision, $precision = n_R/k$, we substitute $E[n_R]$ from equation 1 for the number of relevant elements to get SR in precision as $SRP = E[n_R]/k$.

Consider now precall, which is the expected precision [18] of weakly ordered ranked elements, where tied elements are assessed in a random order. Based on the expected search length [4], precall estimates the length of the ranked list that would contain a desired number of relevant elements, s, in terms of the expected number of irrelevant elements in the list. Precall calculates precision using the ratio of relevant elements r to irrelevant elements irr in the list to find the expected number of irrelevant elements $\frac{s \cdot irr}{r}$ to achieve the information need of the desired number of relevant elements s, so $precall = n_R/(n_R + [s \cdot irr/r])$.

For SR in precall, we replace n_R with $E[n_R]$ for the number of relevant elements, but *not* the number of relevant elements r used in the *esl* calculation. Substituting for r would

invalidate the *esl* calculation because the search length is derived with the assumption of atomic elements. Substituting SR into precall, we get the SR in precall (SRPL) as

$$SRPL = \frac{E[n_R]}{E[n_R] + \frac{s \cdot irr}{r}} \quad (2)$$

2.2 Isolation of Elements

We now consider some ranked list R with a user model of browsing R as presented in precall [18]. In SR, the user model defines the set of traversal permutations of the ranked list. We use Raghavan's precall model here, but as will be seen, other user models could be similarly incorporated into structural relevance.

Let m_R denote the number of ranks in the ranked list R. Let R_i denote the set of elements in rank i. A user browses a ranked list by visiting elements in descending order from the highest to lowest ranks. For elements with tied scores, they are weakly ordered and the user visits these elements in random order until all elements in the rank have been visited. Let Ω denote the set of traversal permutations derived from the user model of browsing R. Let ℓ be the number of permutations of traversal of the ranked list R such that $\ell = |\Omega|$. The number of permutations in Ω can be calculated in terms of the permutations of orderings across all ranks as,

$$\ell = |\Omega| = \prod_{i=1}^{m_R} |R_i|! \quad (3)$$

The isolation of e is $p(e; R)$ for some given ranked list. By conditioning isolation on a permutation $R' \in \Omega$ of R, we get $p(e; R) = \sum_{R' \in \Omega} [p(R') \cdot p(e; R|R')]$.

For the user to choose a particular traversal path $R' \in \Omega$ where every element in R is visited only once, assume a uniform distribution, so $p(R') = 1/|\Omega| = 1/\ell$. Thus, our conditioned expression for $p(e; R)$ becomes

$$p(e; R) = \frac{1}{\ell} \cdot \sum_{R' \in \Omega} p(e; R|R') \quad (4)$$

Now, let us denote $p(e; R|R')$ as simply $p(e; R')$. The difference between R and R' is that R allows weak ordering, but the elements in R' are strictly ordered. The probability of reaching e from elements in R' can be considered a *Bernoulli* process. The process works as follows; every attempt to browse to e fails until e is reached in R', at which point e is reached with perfect certainty. Let $P(e; f)$ be the probability that e is encountered while browsing starting at element f. The trivial cases are $P(e; e) = 1$, and $P(e; f) = 0$ when e is not accessible from f. So, we calculate $p(e; R')$ for a given R' as follows, where $R'[e]$ is the set of elements in descending rank in R' up to e,

$$\begin{aligned} p(e; R') &= \left[\prod_{f \in (R'[e]-e)} 1 - P(e; f) \right] \cdot P(e; e) \\ &= \prod_{f \in (R'[e]-e)} 1 - P(e; f) \end{aligned} \quad (5)$$

EXAMPLE 2.2. *Isolation in a strictly ordered list.* What is the probability of isolating element e in a strictly ordered list R' of 4 elements where the probability of encountering element e from any other element is 0.8? *Ans.* Using equation 5 where $P(e; f) = 0.8$ we get $p(e; R') = (1 - P(e; f))^3 = 0.2^3 = 0.008$.

We now have a complete expression for the isolation of element e in ranked list R by substituting equation 5 into equation 4 and replacing the probability of not visiting element e from element f with $1 - P(e; f)$. So, we get

$$p(e; R) = \frac{1}{\ell} \cdot \sum_{R' \in \Omega} \left[\prod_{f \in (R'[e]-e)} 1 - P(e; f) \right] \quad (6)$$

where $R'[e] - e$ refers to the ranked list $R'[e]$ minus the element e, ℓ is the number of permutations of orderings (equation 3), and $1 - P(e; f)$ is the probability of not reaching e while browsing the element f.

EXAMPLE 2.3. *Isolation in a weakly ordered list.* What is the probability of isolating element e in a weakly ordered list $R = [a \mid b\ e]$ of 3 elements where the probability of encountering element e from a is 0.8 and from b is 0.4 ?

Ans. Referring to equation 6, there are 2 possible routes to e, either $a \rightarrow e$ or $b \rightarrow e$ with probabilities 0.8 and 0.4, respectively. So, $\Omega = \{[a, e, b], [a, b, e]\}$, $\ell = 2$, and we are given that $P(e; a) = 0.8$ and $P(e; b) = 0.4$. Applying equation 6 we get $p(e; R) = \frac{1}{2} \cdot [(1 - 0.8) + (1 - 0.8) \cdot (1 - 0.4)] = 0.16$.

3. APPROXIMATING ISOLATION

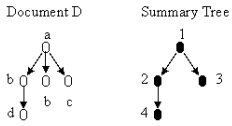
In this section, we introduce the integration of XML summaries with structural relevance to quantitatively model the process of browsing among elements. We show next how isolation in a ranked list can be extended to multiple exclusive sets of overlapped elements, and then using these results we derive an approximation for isolation for calculating SR.

3.1 Incoming XML Summary

Incoming XML summaries are graphs that describe the structure of incoming paths in an XML collection. Summary graphs are formed using XPath queries to generate bisimulations of the elements. The nodes of the summary graph are assigned *labels* that correspond to the tag paths from the root tag to each child tag in a corpus. The *extent* of a node is the set of *elements* in the corpus that match the node's *label*. The size of the *extent* is the number of times that the *label* matches a tag path of an *element* in the corpus. For convenience, each node is assigned a unique structural identifier (SID). There are many types of XML summaries and but we restrict ourselves here to using incoming path summaries. In future work, we intend to extend SR to use other summaries.

The formal definition of the XML summary (also known as XML synopsis, see [17, 3]) is shown below.

DEFINITION 3.1. *A graph synopsis for $G = (V_G, E_G)$ is a node-labeled, directed graph $S(G) = (V_S, E_S)$, where each node $v \in V_S$ corresponds to a set $extent(v) \subseteq V_G$ such that: (1) All elements in $extent(v)$ have the same label (denoted by $label(v)$), i.e., the label of the summary node); (2) $\cup_{v \in V_S} extent(v) = V_G$ and $extent(u) \cap extent(v) = \oslash$ for each $u, v \in V_S$; (3) $(u, v) \in E_S$ if and only if there exists $u' \in extent(u)$ and $v' \in extent(v)$ such that $(u', v') \in$*



Incoming Summary

SID	Label	Path	Extent S.
1	a	a	1
2	b	a,b	2
3	c	a,c	1
4	d	a,b,d	1

Figure 1: Example summary tree, document and incoming summary

E_G; and, (4) Each node $v \in V_S$ stores only an element $count(v) = |extent(v)|$.

EXAMPLE 3.2. Consider the summary tree S, the collection consisting of a single document and the incoming summary shown in Figure 1. The figure shows how each root-to-child tag path in the document defines a partition with an *extent* in the summary S. The *extent* of *summary nodes* result in frequency histograms that *describe* the occurence of tag paths in the collection.

If we assume that the summary graph edges are bi-directional and equally weighted in both directions, then we can consider the graph as describing a time-reversible discrete Markovian process [19]. So, given a well-formed XML document from a summarized collection, we can describe the Markovian process of browsing between summary partitions in the document based on the size of the extents of the summary. This allows us to estimate the relative time spent browsing in any partition, which is used later in Section 3.3 to calculate the isolation of elements in the summary partition.

The probability of a user being in some summary partition i while browsing the collection shall be denoted as π_i, and we calculate it by using the steady-state probabilities of the time-reversible discrete Markovian process,

$$\pi_i = \frac{\sum_j w_{ij}}{\sum_i \sum_j w_{ij}} \tag{7}$$

where $i, j \in S$ are partitions of the summary, and w_{ij} is the size of the extent of the child node among the partitions i and j. We interpret π_i as the fraction of time that a user who uses a description of the document structure (*i.e.* a summary) to browse will spend π_i of their time in partition i of the document.

EXAMPLE 3.3. Consider the summary shown in Figure 1. Table 1 shows the weighting matrix and the probabilities π_i of the time-reversible Markov chain for the summary in the figure.

	SID j					
SID i	1	2	3	4	TOTAL	π_i
1	0	2	1	1	4	44%
2	2	0	0	1	3	33%
3	1	0	0	0	1	11%
4	0	1	0	0	1	11%
TOTAL	3	3	1	2	9	

Table 1: Probabilities of browsing a given summary partition using equation 7 for Figure 1.

3.2 Multiple Sets of Overlap

In this section, we generalize SR to lists with multiple exclusive sets of overlapped elements in a ranked list. A set of overlapped elements refers to a weakly ordered subset of overlapped elements in a list whose order is based on the overlapped elements' ranks.

EXAMPLE 3.4. For the ranked list $\|e_1\|e_{11}\ e_2\ e_{21}\|e_{22}\|e_{12}\|$, where e_1 overlaps with e_{11} and e_{12}; and e_2 overlaps with e_{21} and e_{22}; there are two sets of overlapped elements: namely, $\|e_1\|e_{11}\|e_{12}\|$; and, $\|e_2\ e_{21}\|e_{22}\|$.

Consider the elements that are higher ranked in R to some element $e \in R$. Let m_R denote the rank of element e. We know that all of the higher ranked elements to rank m_R will be visited prior to e. Referring to equation 4 in section 2.2, each trial in the strictly ordered list R' to reach e from higher ranked elements must fail. So, the contribution to isolation from the higher ranks is a constant factor for all strictly ordered lists $R' \in \Omega$. Now, consider equation 5, the isolation in a strictly ordered list R',

$$\begin{aligned} p(e;R') &= \prod_{f \in R' \lceil e \rceil} 1 - P(e;f) \prod_{f \in (R' \lfloor e \rfloor - e)} 1 - P(e;f) \\ &= p_{hi}(e;R) \cdot \left[\prod_{f \in (R' \lfloor e \rfloor - e)} 1 - P(e;f) \right] \end{aligned}$$

where $R' \lfloor e \rfloor$ refers to elements equally ranked to e in R', $R' \lceil e \rceil$ refers to elements higher ranked to e in R', and we introduce the higher ranked isolation factor p_{hi}, where we replace R' with R because the higher-ranked elements to e in R' will be equal for all possible cases of R'. We calculate p_{hi} for e using failed trials from the higher ranks,

$$p_{hi}(e;R) = \prod_{f \in R \lceil e \rceil} 1 - P(e;f)$$

For any element f in R that is not overlapped with e, we know that $P(e;f) = 0$, so these elements can be removed from evaluation. So, in the evaluation of structural relevance, for any given element e, we only need to consider the elements overlapped with the element of interest. For multiple clusters of overlap, for each element of interest, we need only consider its higher or tie ranked elements in the list.

The set of higher-ranked elements to an element $e \in R$ is $R \lceil e \rceil = \{f \mid e, f \in R\ \forall f\ \exists\ G[f] > G[e]\}$ where $G[\cdot]$ is the score of an element. Let function $ov(X, x)$ denote the set of overlapped elements to some element x in some list X. So, we combine overlap with higher ranked elements to get the set of overlapped, higher ranked elements to element e,

$$R \lceil e \rceil_{ov} = ov(R \lceil e \rceil, e) \tag{8}$$

The set of tied elements for element e is $R \lfloor e \rfloor = \{f \mid e, f \in R\ \forall f\ \exists\ G[f] = G[e]\}$. So, the set of overlapped, tie ranked elements of element e, including element e, is

$$R \lfloor e \rfloor_{ov} = ov(R \lfloor e \rfloor, e) \bigcup e \tag{9}$$

For SR for some element e, we only consider the overlapped subset $R_{ov}(e)$ of the ranked list R which is found by taking the union of equations 8 and 9,

$$R_{ov}(e) = R\lceil e\rceil_{ov} \bigcup R\lfloor e\rfloor_{ov} \quad (10)$$

Thus, isolation of an element in a ranked list is strictly dependent on its higher ranked and tie ranked overlapped elements in the list.

3.3 Isolation Revisited

In section 2, we introduced structural relevance, showed how it is calculated using isolation, and then showed in equation 6 how isolation is dependent on the probability $P(e;f)$ of encountering a specific element while browsing a different given element. In section 3.1, we presented the XML summary as a Markovian process of browsing the summarized collection in terms of how the *browser* (*i.e.*, user who is browsing the collection) transitions between the summary partitions. Now, we revisit isolation and using results from section 3.2 we present a complete expression for calculating SR.

Assume that browsing from element f to e requires entering the summary partition of e (*i.e.*, being outside of the partition of e) and, simulateously, browsing along the structural paths in the document instance of e and f. So, $P(e;f)$ is the probability of being outside of element e's partition and the probability that structural paths are followed to reach e. Using equation 7, let us denote the probability of being in the partition of element e as $\pi_{(e)}$. So, being outside of the partition is $1-\pi_{(e)}$. Let X denote the probability of following a set of structural paths from element f to element e. The limiting probability of reaching e from f over an infinite number of trials is 1 if we assume that the number of structural paths are finite and that the browser has a positive probability of taking any structural path whenever possible. So, we get,

$$P(e;f) \quad = \quad (1-\pi_{(e)})\cdot X \approx 1-\pi_{(e)} \quad (11)$$

So, referring to equation 6, substituting equation 11 into the isolation of element e in ranked list R, we get

$$\begin{aligned} p(e;R) &= \frac{1}{\ell}\cdot\sum_{R'\in\Omega}\left[\prod_{f\in(R'[e]-e)} 1-P(e;f)\right] \\ &= \frac{1}{\ell}\cdot\sum_{R'\in\Omega}\left[\prod_{f\in(R'[e]-e)} \pi_{(e)}\right] \end{aligned} \quad (12)$$

Recall equation 3 where ℓ is defined across all the elements in the ranked list. Let $R_{(e)}$ be the elements in the rank of element e. To get all cases in ℓ where e is fixed, we would simply reduce the size of the rank $R_{(e)}$ by one, and we get $\ell(e) = (1/|R_{(e)}|)\cdot(\prod_{i=1..m_R}|R_i|!)$, where the number of ranks in R is m_R.

As we noted in equation 8, for some element e in R we need only consider the overlapped elements in $R'[e]$. So, for every ranked list $R'[e]$ there will be $\ell(e)$ number of possible traversals with e at a given rank. Among the tie-ranked elements, there will be $|R_{ov}\lfloor e\rfloor|$ relative positions in which e may occur. So, we substitute $|R_{ov}\lfloor e\rfloor|$ and $\ell(e)$ into isolation equation 12, and then rearrange, to get

$$\begin{aligned} p(e;R) &= \frac{\ell(e)}{\ell}\sum_{n=1}^{|R_{ov}\lfloor e\rfloor|}\pi_{(e)}^{n+m-1} \\ &= \frac{1}{|R_{(e)}|}\sum_{n=1}^{|R_{ov}\lfloor e\rfloor|}\pi_{(e)}^{n+m-1} \end{aligned} \quad (13)$$

where $R_{(e)}$ are the set of elements in the rank of element e, $R_{ov}\lfloor e\rfloor$ is the set of overlapped elements in the rank of element e, m is the number of higher ranked, overlapped elements to e, and $1-\pi_{(e)}$ is the approximated probability of browsing to element e from any overlapped element f.

Now, referring to the modified precall and precision metrics in section 2.1, consider structural relevance in equation 1 and substitute in the approximated isolation for $p(e;R[u])$ from equation 13 to get our final expression for SR

$$SR[u] \quad = \quad \sum_{e\in R[u]}\frac{rel(e)}{|R[u]_{(e)}|}\sum_{n=1}^{|R[u]_{ov}\lfloor e\rfloor|}\pi_{(e)}^{n+m-1} \quad (14)$$

4. XML RETRIEVAL METRICS

In this section we briefly present and discuss three XML retrieval measures. XCG is presented first (a detailed experimental comparison with SR is deferred to the following section), followed by PRUM and HiXEval.

4.1 Extended Cumulated Gain

Extended cumulated gain (XCG) is a cumulated gain (CG) [8] measure that addresses structural dependencies in the recall-base, such as near-misses and overlap, in content-oriented XML retrieval evaluation [10]. It is a flexible measure that incorporates multi-criteria assessments and modeling of user satisfaction. The relevance assessment of elements is used to determine the *ideal recall-base*, which contains elements with the highest scores without overlapping [9], and the *full recall-base*, which contains all relevant elements. Using two recall-bases, ideal and full, requires dependency normalisation heuristics to ensure that the total score for any element does not exceed the maximum score achievable when the ideal node itself is retrieved [10].

The scores of elements are determined using a relevance value function. The score is indicative of the utility of the element to a user. The relevance value function uses quantizations of relevance assessments to return a score in $[0,1]$. The function takes into account overlap, ranking of elements and provides a weighting factor α to represent the user's intolerance to overlapped elements.

The cumulated gain and ideal gain are calculated by summing the scores of ranked elements up to some prescribed position. Denote the score of the a-th element in a list of k_R length as $xG[a], a\in[1,k_R]$. Furthermore, denote the score of the a-th element in an ideal list of k_I length as $xI[a], a\in[1,k_I]$. So, up to a given position a,we express the cumulated score for a list (xCG) and the cumulated score for an ideal list (xCI) as follows,

$$xCG[a] = \sum_{i=1}^{a} xG[i],\ xCI[a] = \sum_{i=1}^{a} xI[i]$$

There are a number of different ways to compare the cumulated scores for a list and its ideal. In this work, we con-

sidered only the normalized extended CG ($nxCG$), which is simply the ratio of the cumulated scores for the list and its ideal, such that

$$nxCG[a] = \frac{xCG[a]}{xCI[a]} \quad (15)$$

4.2 Precision Recall with User Modeling

Precision recall with user modeling (PRUM) is one of the newest proposals for XML retrieval evaluation metrics. It measures the percentage of ideal elements in a collection that are seen by a user while browsing a ranked list [16, 15]. Like XCG, it relies on the definition of an ideal recall-base which consists of relevant elements that do not overlap. In addition, PRUM incorporates multi-criteria assessments and modeling of user satisfaction based on structural constraints. Unlike XCG, PRUM is a probabilistic measure and is defined as the expected number of ideal elements that are seen by the user, up to a given rank, while she browses the collection from the ranked list [1].

SR and PRUM use similar probabilistic event models and user models. But they differ in how relevance is considered. In SR, there is a distinction between relevance of an element and the relevance of a set of elements. SR does not explicitly include the use of multi-criteria assessments. SR employs summaries to model user satisfaction, whereas PRUM uses browsing habits derived from the assessment process. PRUM is at an early stage of development, and real world results to compare with SR are not available at the present time.

4.3 Highlighting XML Retrieval Evaluation

Highlighting XML retrieval evaluation (HiXEval) [14] is another recently proposed approach to measure the effectiveness of XML retrieval systems. HiXEval was motivated by the need to simplify XML evaluation and make it conform to well-established evaluation measures such as precision and recall. HiXEval was proposed as an extension of the traditional definitions of precision and recall to include the knowledge obtained from the highlighting assessment procedure adopted at INEX 2005. The biggest difference between HiXEval and other measures is its contention that the purpose of the XML retrieval task is to find elements that contain as much relevant information as possible, without also containing a significant amount of non-relevant information. With the way relevance has been assessed since 2005, this translates to the aim of returning elements that contain as much highlighted (relevant) content as possible, and as little non-highlighted (non-relevant) content as possible.

In calculating precision and recall, the explicit structure of the documents is ignored because these measures are based upon the amount of highlighted text in and across elements and documents. We leave a comparison of SR with HiXEval for future work.

5. SR AND XCG APPLIED TO INEX

The following investigations were conducted on a single run from 3 different systems for top-10 results for the thorough task in 114 INEX Wikipedia topics [13]. At this stage of our work, the focussed task was not considered because it does not allow overlapping elements, and thus SR modified measures (such as precision or precall) would have the same value as unmodified measures [1]. The results obtained are illustrative of the differences between SR and XCG. The incoming summary for calculating isolation was generated from the Wikipedia collection using the methodology presented in section 3.1. For XCG, we used the normalized extended cumulated gain ($nxCG$) with the configuration genLifted, overlap on, and $\alpha = 1$. The experimental measures were structural relevance with precall (SRPL) and precision (SRP) as described in section 2.1.

Table 3 shows the graphical results for topics 295, 307, and 335 on the three columns on the left (discussed later on). The right-most column shows the system rankings for each measure in terms of the number of topics that resulted in a given system rank order. The ranking was done for each measure and for each topic by ordering the systems in descending order based on the area of each system's performance curve. The histograms are labeled according to the ranking of systems from left (best) to right (worst): (M)AXPLANCK, (I)BMHAIFA, and (L)IP6. For instance, the first column of the top-most histogram shows that for SRP there were 60 topics where the systems were ranked MIL, or in other words, were ranked in descending order of performance: MAXPLANCK, IBMHAIFA, LIP6. The histograms for SRP and SRPL are sub-divided to show the number of topics for which the measure in question did not obtain the same ranking relative to rankings in XCG. Returning to the SRP histogram, for example, the first column in the histogram shows that SRP ranked the systems as MLI

Table 2: System outputs for topic 295

Notation for Systems

x/-	:	relevant/irrelevant element
‖	:	rank boundary
o[p/c/s]	:	overlap with ancestor/descendant/sibling

IBMHAIFA : ‖**xocp**‖**xoc**‖**xop**‖**xoc**‖**xop**‖**xops**‖**xoc**‖**-ops**‖**xop**‖**xoc**‖

172477.xml	**1136198.xml**	**14724.xml**
8: /article[1]/body[1]	0: /article[1]/body[1]	2: /article[1]/body[1]
9: /article[1]	1: /article[1]	3: /article[1]
76266.xml	5: /article[1]/body[1]/section[2]	
4: /article[1]/body[1]	7: /article[1]/body[1]/section[5]	
6: /article[1]		

$k = 10, r = 9, ranks = 10, relevant\ docs = 4$

LIP6 : ‖**-op**‖**-oc**‖**-oc**‖**-op**‖**xoc**‖**xop**‖**-oc**‖**-op**‖**-op**‖**-oc**|

3130820.xml	**196073.xml**	**1331267.xml**
7: /article[1]/body[1]	0: /article[1]/body[1]	6: /article[1]
9: /article[1]	1: /article[1]	8: /article[1]/body[1]
14724.xml	**1773624.xml**	
4: /article[1]	2: /article[1]	
5: /article[1]/body[1]	3: /article[1]/body[1]	

$k = 10, r = 2, ranks = 10, relevant\ docs = 1$

MAXPLANCK : ‖**-op -**‖**x**‖**-oc**‖**x**‖**-**‖**-**‖**x**‖**-**‖**-**‖

1773624.xml	**1331267.xml**	**172477.xml**
3: /article[1]	6: /article[1]	7: /article[1]
0: /article[1]/body[1]/section[1]	**2251312.xml**	**1711143.xml**
23273.xml	9: /article[1]	8: /article[1]
5: /article[1]/body[1]/section[9]		**14724.xml**
63285.xml	**419136.xml**	4: /article[1]
2:/article[1]/body[1]/section[4]	1: /article[1]/body[1]/section[6]	

$k = 10, r = 3, ranks = 9, relevant\ docs = 3$

[1]We will investigate in future work the use of SR as a meanss to measure relevance in the recall-base itself.

for 60 topics and 50 of those topics were also ranked as MLI by XCG.

Overall, SRP agreed with XCG for 78 out of 114 topics or 68%. SRPL agreed with XCG for 38 out of 114 topics or 33%. Since XCG, SRP and SRPL produce different results, we turn our attention to a small representative subset of topics to provide some insight into the differences.

5.1 Individual Topic Comparison

The detailed results for topic 295, including ranked list, overlap of elements, label paths, and documents, are shown in table 2. The notation used for representing overlapped elements is at the top of table 2. The first column of table 3 shows the evaluations for SRP, SRPL, and XCG for topic 295. Referring to table 2, a user who either randomly explores a list or systematically explores from the head of a list to the end would find the results of IBMHAIFA the best for topic 295. It contains more relevant documents and relevant elements occurring at earlier ranks than either LIP6 or MAXPLANCK. This observation is reflected in both SRP (table 3, column 1, row 1) and SRPL (table 3, column 1, row 2). XCG (table 3, column 1, row 3) differs in that it concludes that MAXPLANCK and IBMHAIFA are the best and nearly equal.

Topic 295 is a good example of how overlap affects the relevance of a list. This depends on the composition of the ranked list in terms of elements, ranking, and overlap. For instance, SRP for topic 295 shows a steep decline in the output of IBMHAIFA at 20% recall because of overlapped elements. We see SRP fluctuate across recall levels, upward with novel elements and downward with overlapped elements. At recall, it settles around 40% precision. But, we can see that the SRP is significantly above 40% in early ranks, and as the overlap becomes more pronounced at late recall levels the SR precision eventually achieves the precision based on 4 documents out of 10 elements. We see this behavior because documents are independent sets of elements and this is reflected in SR-based precision. SRPL (table 3, column 1, row 2), on the other hand, does not exhibit this behavior because precall is based on the expected search length which is strongly determined by the number of irrelevant elements. SR does not account for irrelevant elements, and we can see that overlap degrades SRPL for IBMHAIFA only slightly (*i.e.*, 9 out 10 elements in Table 2 are relevant with SRPL at about 80% for recall 1).

We recognize two general cases that we believe explain the differences between SRP and XCG. The first case involves over-penalization where results contain parent elements consistently ranked higher than children elements; XCG seems to over-penalize these configurations for overlap. The second case involves early recall, where results containing relevant elements at early ranks and results containing relevant elements at late ranks will perform overall equally in XCG. In contrast, results containing relevant elements at early recall score higher in SRP.

Case 1: Overlap Penalization. Topic 307 is a good example of overlap penalization. Over-penalization occurs because of the dependency normalisation heuristic that differentiates between the order of parent and child elements in a ranked list. The heuristic is that if a parent element is *seen*, then its child elements are considered *fully seen*, whereas if a child element is *seen* then its parent is only *partially seen* [10]. This heuristic results in over-penalization in all configurations of XCG except where $\alpha = 0$. Referring to Table 3, IBMHAIFA contains the best results because it returns both more relevant documents, and more relevant elements. But because parent elements are being ranked higher than child elements, XCG ranks IBMHAIFA last in performance. This example also demonstrates the inverse of this phenomena, where LIP6 is rewarded for overlap in its first two ranks.

Table 4: System outputs for topic 307

$IBMHAIFA$: ‖**-op**‖**-oc**‖**xop**‖**xop**‖**xoc**‖**xoc**‖**xop**‖**x**‖**xoc**‖**x**‖
$k = 10, r = 8, ranks = 10, docs = 4, relevant\ docs = 3$
$LIP6$: ‖**xoc**‖**xop**‖**-oc**‖**-op**‖**-oc**‖**-op**‖**xoc**‖**xop**‖**-oc**‖**-op**‖
$k = 10, r = 4, ranks = 10, docs = 5, relevant\ docs = 2$
$MAXPLANCK$: ‖**x -**‖**-oc**‖**x**‖**-**‖**-**‖**-**‖**x**‖**-op**‖**-**‖
$k = 10, r = 3, ranks = 10, docs = 9, relevant\ docs = 3$

Table 5: System outputs for topic 335

$IBMHAIFA$: ‖**xop**‖**xoc**‖**-ops**‖**xocps**‖**-ocps**‖**xops**‖**-ops**‖**-ops**‖**xops**‖**xocps**‖
$k = 10, r = 6, ranks = 10, docs = 1, relevant\ docs = 1$
$LIP6$: ‖**-ocs** ‖**-ops** ‖**-ocs** ‖**-ops** ‖**xocs** ‖**xops** ‖**xocs** ‖**xops** ‖**-ops** ‖**-ocs** ‖
$k = 10, r = 4, ranks = 10, docs = 5, relevant\ docs = 2$
$MAXPLANCK$: ‖**- -** ‖**-** ‖**- x** ‖**-** ‖**-** ‖**-** ‖**xocs** ‖**-ops** ‖
$k = 10, r = 2, ranks = 8, docs = 9, relevant\ docs = 2$

Case 2: Early Recall. Topic 335 is a good example of how XCG is not sensitive to early recall in a ranked list. XCG evaluates MAXPLANCK as the best search engine for topic 335 (see table 3, column 2, row 3). Referring to table 5, this makes some sense because MAXPLANCK returns the most number of relevant documents with the least overlap. But, MAXPLANCK is not the best list because the relevant elements in MAXPLANCK occur at the end of the list. In this regard, we would posit that IBMHAIFA has a better result. This can be see in topic 335 for IBMHAIFA that SRP is highest in early recall, although performance degrades significantly in later recall.

6. CONCLUSIONS AND FUTURE WORK

We have presented a general model of structural relevance and shown how it can be used to modify precall and precision for measuring effectiveness in XML retrieval. The SR approach uses XML summaries to represent how users perceive overlap in XML retrieval. The experimental results presented suggest that SR handles situations such as over-penalization of overlap due to heuristics and sensitivity to results with early recall more effectively than XCG. More significantly, we show that SR does not require an ideal recall-base or dependency normalization, as is the case for existing measures.

Future work includes obtaining results on the performance of SR for a larger number of systems, carrying out additional comparisons of SR (with XCG, PRUM, and HiXEval), undertaking reliability tests for the SR metric, and further developing the application of summary-based techniques to SR measures.

7. REFERENCES

[1] D. Carmel, Y. S. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer. Searching xml documents via xml fragments. In *SIGIR '03: Proc. of the 26th Annual Intl. ACM SIGIR Conf. on Res. and Dev. in IR*, pages 151–158, New York, NY, USA, 2003. ACM Press.

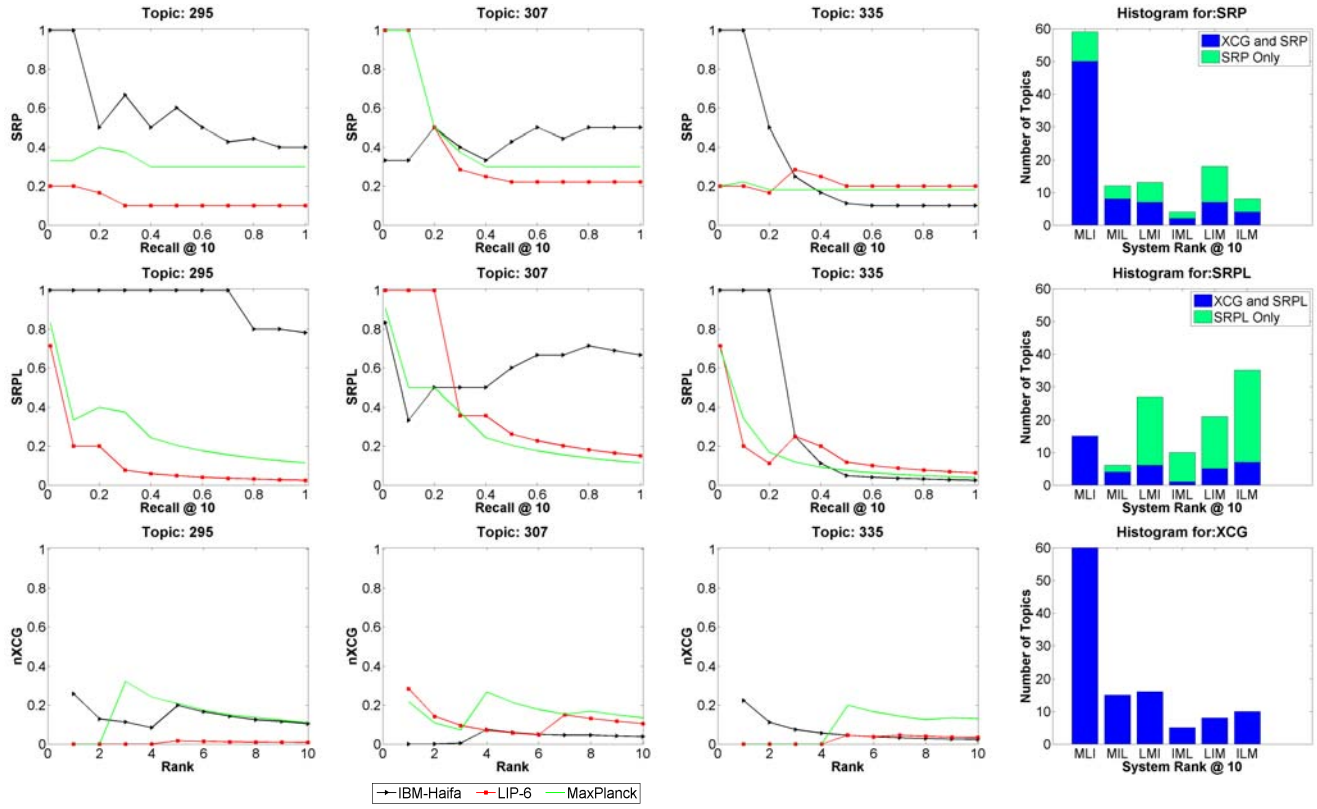


Table 3: System evaluations (SRP, SRPL, and XCG) and histograms across all topics

[2] C. Clarke. Controlling overlap in content-oriented xml retrieval. In *SIGIR '05: Proc. of the 28th Ann. Intl. ACM SIGIR Conf. on Res. and Dev. in IR*, pages 314–321, New York, NY, USA, 2005. ACM Press.

[3] Mariano P. Consens, Flavio Rizzolo, and Alejandro A. Vaisman. Exploring the (semi-)structure of XML web collections. Technical report, UofT - DCS, 2007. http://www.cs.toronto.edu/∼ consens/describex/.

[4] W. S. Cooper. Expected search length: A single measure of retrieval effectiveness based on weak ordering action of retrieval systems. *J of the Amer. Soc. for Info. Science.*, 19:30–41, 1968.

[5] A. de Vries, G. Kazai, and M. Lalmas. Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit. In *RIAO 2004*, Vaucluse, France, April 2004.

[6] N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors. *Adv. in XML IR, 3rd Intl. Workshop of INEX 2004*, volume 3493 of *LNCS*. Springer, 2005.

[7] N. Gövert and G. Kazai. Overview of the initiative for the evaluation of xml retrieval (inex) 2002. *Proceedings of the 1st Workshop of the INitiative for the Evaluation of XML Retrieval*, pages 1–17, 2003.

[8] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[9] G. Kazai. Choosing an ideal recall-base for the evaluation of the focussed task: Sensitivity analysis of the xcg evaluation measures. In *Compar. Eval. of XML IR Sys., 5th Intl. Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2006)*. Springer, 2007.

[10] G. Kazai and M. Lalmas. Extended cumulated gain measures for the evaluation of content-oriented xml retrieval. *ACM Trans. Inf. Syst.*, 24(4):503–542, 2006.

[11] G. Kazai, M. Lalmas, and A. de Vries. The overlap problem in content-oriented xml retrieval evaluation. *Proc. of the 27th Ann Intl ACM SIGIR Conf on Res and Dev in IR*, 2004.

[12] J. Kekäläinen, M. Junkkari, P. Arvola, and T. Aalto. TRIX 2004 - Struggling with the Overlap. In Fuhr et al. [6], pages 127–139.

[13] M. Lalmas and G. Kazai. Report on the ad-hoc track of the INEX 2005 workshop. *SIGIR Forum*, 40(1):49–57, 2006.

[14] J. Pehcevski and J. A. Thom. HiXEval: Highlighting XML retrieval evaluation. In *Adv. in XML Info. Retrieval: 4th Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005)*. Springer-Verlag, November 2005.

[15] B. Piwowarski and G. Dupret. Expected precision-recall with user modelling (eprum). In *SIGIR '06: Proc. 29th Ann. Intl ACM SIGIR Conf on R&D in Info Retr*, pages 260–267, NY, NY, USA, 2006. ACM Press.

[16] B. Piwowarski, P. Gallinari, and G. Dupret. Precision recall with user modeling (prum): Application to structured information retrieval. *ACM Trans. Inf. Syst.*, 25(1):1, 2007.

[17] N. Polyzotis and M. N. Garofalakis. Statistical synopses for graph-structured XML databases. In *SIGMOD Conf.*, 2002.

[18] V. Raghavan, P. Bollmann, and G. Jung. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.*, 7(3):205–229, 1989.

[19] S. M. Ross. *Introduction to Probability Models*. Academic Press, New York, 8th edition, 2003.

[20] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. Mixture Models, Overlap, and Structural Hints in XML Element Retrieval. In Fuhr et al. [6], pages 196–210.

[21] A. Tombros, S. Malik, and B. Larsen. Report on the INEX 2004 interactive track. *ACM SIGIR Forum*, 39(1):43–49, 2005.

Collaborative Knowledge Management: Evaluation of Automated Link Discovery in the Wikipedia

Wei Che Huang
Faculty of IT
Queensland University of Technology
Brisbane, Australia
w2.huang@student.qut.edu.au

Andrew Trotman
Department of Computer Science
University of Otago
Dunedin, New Zealand
andrew@cs.otago.ac.nz

Shlomo Geva
Faculty of IT
Queensland University of Technology
Brisbane, Australia
s.geva@qut.edu.au

ABSTRACT

Using the Wikipedia as a corpus, the Link-the-Wiki track, launched by INEX in 2007, aims at producing a standard procedure and metrics for the evaluation of (automated) link discovery at different element levels. In this paper, we describe the preliminary procedure for the assessment, including the topic selection, submission, pooling and evaluation. Related techniques are also presented such as the proposed DTD, submission format, XML element retrieval and the concept of Best Entry Points (BEPs). Due to the task required by LTW, it represents a considerable evaluation challenge. We propose a preliminary procedure of assessment for this stage of the LTW and also discuss the further issues for improvement. Finally, an efficiency measurement is introduced for investigation since the LTW task involves two studies: the selection of document elements that represent the topic of request and the nomination of associated links that can access different levels of the XML document.

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

General Terms: Measurement, Experimentation

Keywords: Wikipedia, Link-the-Wiki, INEX, Evaluation, DTD, Best Entry Point

1.INTRODUCTION

The Wikipedia is a well-known online collaborative knowledge sharing system, a free encyclopedia that can be extended by any wiki contributor and modified by other wiki users [24]. At the time of writing, there are more than 75,000 active contributors working on more than 5,300,000 articles in more than 100 languages [25]. The growth in English Wikipedia articles had been around 100% per year from 2003 through most of 2006. There has been a close to linear increase in the number of articles since roughly September 2006, previous to that the trend was exponential.

Built upon traditional Wiki architectures, the functionality of the Wikipedia search engine is limited to title and full-text search. In general, it performs searching at the article level. After keywords have been entered in the search box, the ***Go*** function takes the user to the particular article while the ***Search*** function returns a list of ranked articles (including an estimate of relevance given as a percent) [26]. In addition to search facilities on the Wikipedia web site, there are a number of other search engines that search the encyclopedia (such as Google, Qwika, Lycos and Yahoo!).

Little research has been done in the area of semi-structured retrieval that can be directly applied to enhance the search features within the Wikipedia (although XML information retrieval studies have gained much attention in the last few years [19]).

Wikipedia contributors, like those of other Wikis must specify a variety of links that are relevant to a new article. They manually find and create links to other internal Wikipedia documents or external web pages. None the less, it is easy to find many un-related links that have been created and inserted in the documents (technical terms and years in particular). As an example, the term, *atomic transition probabilities*, in the *Albert Einstein* page had been split into *atomic transition* and *probabilities*, and *atomic transition* had been linked to the page *Transition rule*. However, in the list of search results for the term *atomic transition*, *Crystal field excitation* has the highest relevance (19.8%) and *Transition rule* second with relevance 12.3%. Similar to *atomic transition*, the term, *quantum theory*, had been linked to the article, *Quantum mechanics*, which is not found in the first page of results for *quantum theory*, but the *Quantum theory* page is returned as the most relevant result (100%).

By these examples, it is inappropriate to utilize standard search facilities to automatically nominate related links for anchor texts. A pilot track, Link-the-Wiki (LTW), launched by the Initiative for the Evaluation of XML Retrieval (INEX) in 2007 aims to provide a reusable resource and standard methods for the evaluation of automated link discovery within the English Wikipedia collection [8]. Previous work on link discovery exists of course (see section 8 for a brief review), but typically the methods operate on linking at the document level. As far as we know there has been no work published on automated discovery of document hyperlinks in the Wikipedia studying the choice of anchor texts and the link destination to specific positions *within* existing Wikipedia pages.

In this paper an assessment procedure for evaluating automated link discovery is proposed for use at INEX 2007 and beyond. In general, the procedure can be divided into several steps. First a number of orphan documents nominated by participants will be used as example link-less documents. Participants will generate links for these documents and submit results. Then the results will be pooled together for evaluation. Pooling will be performed manually. Finally, performance will be measured using agreed upon metrics. In this paper the pooling process will be discussed

SIGIR 2007 Workshop on Focused Retrieval
July 27, 2007, Amsterdam, The Netherlands

and future possibilities will be discussed. The challenges and the evaluation tool will also be introduced.

The remainder of this paper is organized as follows. In the next section, we survey the Wikipedia and its use as a corpus for focused information retrieval. Then we introduce the Link the Wiki track (in Section 3). Previous work that is related to the task of LTW is briefly summarized in section 4. In Section 5, we explain the terminology and present the submission format. In the next section (6), assessment steps and the evaluation process are introduced and discussed. Section 7 covers measures of search engine efficiency that will be considered for LTW track. The LTW 2007 track and its future scope are described in Section 8. Finally, conclusions and future work are provided in Section 9.

2.WIKIPEDIA AS AN IR COLLECTION

The Wikipedia is a free online document repository written collaboratively by wiki contributors around the world. Composed of millions of articles in numerous languages it offers many attractive features as a corpus for information retrieval tasks. In the first place, this wiki-based corpus is freely available so there are no distribution restrictions. The INEX Wikipedia collection has already been converted from its original wiki-markup text into XML [6]. That collection is composed of a set of XML files where each filename is a unique number corresponding to the id of the Wikipedia article (e.g. 16238.xml). Each file corresponds to an online article in Wikipedia (see Figure 1). A semantic annotation of the Wikipedia was also undertaken by others (e.g. [17]). Search as well as retrieval could benefit from rich semantic information in the XML Wikipedia collection, where it exists.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <article>
    <name id="588">Africa</name>
    <conversionwarning>0</conversionwarning>
  - <body>
      <templatePortal NAME="Portal" />
```

Figure 1. article XML format with corresponding id

In addition, the semi-structured format provided by the XML-based collection offers a useful property for the evaluation of various semi-structured retrieval techniques. Specifically, the linkage within a document is an especially interesting aspect of the Wikipedia and offers opportunities for investigating article categorization as well as the user interaction (e.g. browsing and searching) with a hyperlinked corpus.

The Wikipedia collection might be used for a variety of purposes such as XML information retrieval, machine learning, clustering, structure mapping, and categorization. The Wikipedia has already been used as an IR corpus in several evaluation initiatives. At INEX 2006 it was used for the evaluation of *ad hoc* XML retrieval and for the XML Document Mining track. At CLEF 2006, it was used as a corpus for question answering [23]. As the collection has already been used at INEX it is the natural choice for the INEX [8] 2007 Link-the-Wiki track.

3.LINK THE WIKI

For Link the Wiki at INEX 2007 the XML Wikipedia collection already used at INEX will be used as the document collection. It is composed of about 660,000 documents in English and is around 5GB is size. Many articles in the Wikipedia collection are already extensively hyperlinked.

The aim of the Link-the-Wiki task, first described by Geva and Trotman [21], is to offer an evaluation forum for proposing, testing, and discussing algorithms for evaluating the state of the art in automated link discovery in XML documents. The test collection including documents, judgments, and metrics for evaluating different systems and comparing various approaches to automated discovery of hypertext links will be made available for other researchers.

Participants will be given a set of (about 50) orphan Wikipedia documents nominated by participants. The task is two fold: first to analyze each orphan and to recommend anchor text and destinations within the Wikipedia; second to recommend incoming links from other Wikipedia documents. For 2007 we expect 25 anchor texts to be recommended for each orphan document. There will, therefore, be 1,250 outgoing links and 1,250 incoming links created for the 50 orphans, per submission. In future years the number of links might no longer be limited to 25 and links outside the Wikipedia (for example to the web) may be included.

Results will be submitted to the organizers who will pool them in the usual way. The pooled results will be analyzed and evaluated either automatically or by participants, depending on the kind of link. Article-to-article links can be evaluated by comparison with the original Wikipedia pages – they already contain relevant links created by the page authors. Links directly to XML elements must be evaluated manually.

The detailed experiment steps of assessment and evaluation will be described below in Section 6 and 7.

4.RELATED WORK

Since the goal of this paper is to propose the evaluation forum of Link-the-Wiki track as well as to significantly extend the tasks of link discovery to XML element level, we briefly introduce several instances of previous research on link analysis and generation, as well as document relevance identification, especially in the case of the Wikipedia. The past research described here is mainly targeted at the document level and the related evaluation for these approaches is manually performed.

While the Wikipedia has only gained much popularity in recent years, link analysis on the web and hypertext documents has been a relatively mature research field. Various link based techniques based on the correlation between the link (density) and the entities are analyzed and developed to deal with diverse research problems [1]. Links have been used to provide additional information for improving the quality of search engine results. Moreover, link analysis can also be used for topically classifying communities on the Web. The idea is to identify the implicit communities by the analysis of Web graph structure [13]. Kumar et al. also apply the concept of co-citation in the web graph for the similarity measure. Beside co-citation, bibliographic coupling and SimRank can be used to determine the similarity of objects (e.g. web pages), which are based on the citation patterns of documents and the similarity of structural context respectively [9][11]. Moreover, the Companion algorithm derived from HITS (**H**yperlink-**I**nduced **T**opic **S**election) is proposed for finding related pages by exploiting links and their order on a page [4][12].

This conducts a strategy of using a page's URL, instead of query terms, to search a set of related Web pages.

An overview of Wikipedia research was presented by Voss, which consists of different aspects of wiki studies [27]. This includes the visualization of wiki editing, relations of readers and authors, citation of wiki articles, the (hyperlinked) structure of Wikipedia and the statistic of Wikipedia. Recently, more research with regard to Wikipedia has been undertaken in particular for identifying the relevance of wiki articles. Bellomi and Bonato utilize network analysis algorithms such as HITS and PageRank to find out the potential relevance of wiki pages (content relevant entries) in order to explore the high level (hyperlinked) structure of Wikipedia and gain some insights about its content regarding to cultural biases [2]. Ollivier and Senellart have conducted a set of experiments for examining the performance of approaches on finding related pages within Wikipedia collection [14]. There are totally 5 methods included in the evaluation, including Green-based methods, *Green* and *SymGreen*, and three classical approaches, *PageRankOfLinks*, *Cosine* with tf-idf weight and *Co-citations*. The concept of these methods is to find out the most related neighborhood of a given node. They can be derived to achieve the task of finding the related pages.

Another interesting topic of utilizing an automated approach in finding related pages is to explore potential links in a wiki page. Adafre and de Rijke propose a method of discovering missing links in Wikipedia pages via clustering of topically related pages by LTRank and identification of link candidates by matching the anchor texts [1]. Jenkins presents a Wikipedia link suggestion tool, *Can We Link It*, for searching missing links in a page [10]. This suggestion tool can automatically eliminate those link candidates through the learning of user rejection and grammatical structure. However, some of these suggested links are still without merit with respect to the topic.

Furthermore, Wikipedia's category structures also offer useful information for topic identification. Schönhofen utilizes only the titles and categories of Wikipedia articles to characterize documents [18]. However, this simple method has not fully exploited the potential of Wikipedia, such as the internal text of articles, the category hierarchy and the linking structure of Wikipedia. *Wikirelate* proposed by Strube and Ponzetto uses Wikipedia to compute semantic relatedness of words through existing measures: Path based, Information content based and Text overlap based measures [20]. These measures mainly rely on either the texts of the articles or the category hierarchy. According to the shortcomings of *Wikirelate*, Gabrilovich and Markovitch introduce a new approach called Explicit Semantic Analysis (ESA), which computes relatedness by comparing two weighted vectors of Wikipedia concepts that represent words appearing within the content [6].

It is difficult to compare and contrast various approaches without a standard benchmark. This is the intent of the LTW track, while tightening and extending linking requirements to include BEPs.

5.TERMINOLOGY

Since the Link-the-Wiki track at INEX 2007 involves a series of new schemes and procedures, in this section, we will describe these in some details.

5.1Anchor Text Specification

Text file inversion is probably the most widely used technique in text retrieval systems [7]. For each term in an XML document a list of occurrences is maintained. The representation of each occurrence of a term is composed of the article id and term position within the XML document. We use this general representation in the specification of anchor text in the Link-the-Wiki task. Each term, phrase (or word gram) in an XML document can be located by identifying three parts: the filename (or article id in our case), the absolute XPath to the element in which the term is found, and the term or phrase position within the element.

The filename is used to identify the document within the XML collection. In the XML Wikipedia collection, a document file is presented by a unique id. For instance:

C:/Wikipedia/xml/23816.xml

The filenames are unique hence "**23816.xml**" is sufficient to unambiguously identify the document.

An XML element within the document may be identified by the absolute XPath expression relative to the file's root element (see Table 1).

Table 1. The absolute XPath expression

Absolute XPath Context
/article[1]/body[1]/section[5]/section[2]/p[4]
/article[1]/body[1]/p[1]/emph2[1]
/article[1]/body[1]/section[4]/item[3]/collectionlink[3]

Finally, with the XML document object model (DOM) it is possible to specify a particular text node character position. In the following expression the last number is the term position that identifies the start position (in characters) of the term within the specific XPath context:

/article[1]/body[1]/section[2]/p[1]/text()[6].3

In the Link-the-Wiki task we are proposing to identify anchor text start and end character positions in this manner.

5.2Example Specification of Link Discovery

With the element specification format described above, the LTW task can accept submissions that work with anchor text and links to specific XML elements. We use the term *best entry point* (**BEP**) as already used in INEX to describe a destination element within a document from which to start reading. Anchor text must be identified precisely by using the DOM as it is a passage of text and not an XML element or a simple location within the text.

An example submission is depicted in Figure 2. As shown each topic (orphan page) is identified by a topic-id, file name, and title. While these attributes are the same for each topic, and are thus interchangeable, all three are included for the sake of convenience and clarity. For each orphan two sets of links are identified - *outgoing* and *incoming*.

Outgoing links are composed of a set of *link*s from the orphan page to existing Wikipedia pages. Each *link* consists of an

anchor and a target *file* and a *best entry point* within that file. Collectively these identify a unique XML element in a Wikipedia document.

Incoming links are composed of a set of *link*s from anchor texts within existing Wikipedia pages to a *best entry point* in the orphan page.

To work with document to document (e.g. "see-also") links all that is required is the specification of all XPath expressions as **/article[1]**. In this case the entire topic specification is degenerates to a set of links between documents without any explicit anchor or best entry points. This is a deliberate decision made to accommodate low-cost entry into the Link-the-Wiki track.

```
<topic id="38" file="13876.xml" name="Albert Einstein">
  <outgoing>
    <link>
      <anchor>
        <start> /article[1]/body[1]/p[3]/text()[2].10 </start>
        <end> /article[1]/body[1]/p[3]/text()[2].35 </end>
      </anchor>
      <linkto>
        <file> 123456.xml </file>
        <bep> /article[1]/sec[3]/p[8] <bep>
      </linkto>
    </link>
    …
  </outgoing>
  <incoming>
    <link>
      <anchor>
        <file> 654321.xml </file>
        <start> /article[1]/body[1]/p[3]/text()[2].10 </start>
        <end> /article[1]/body[1]/p[3]/text()[2].35 </end>
      </anchor>
      <linkto>
        <bep> /article[1]/sec[3]/p[8] <bep>
      </linkto>
    </link>
    …
  </incoming>
</topic>
```

Figure 2. Sample submission

5.3DTD

A Document Type Definition (DTD) will be defined for specifying the XML document structure. It will contain a list of legal elements and attributes from within the Wikipedia collection. This will allow participants to validate their runs before being submitted. Although, since document-to-document linking will be the default at INEX 2007, this is not immediately needed. A full version of LTW will be run in future years so the full DTD will be needed at that stage. The DTD for LTW 2007 is depicted in figure 3.

```
<!ELEMENT topic (outgoing, incoming)>
<!--
          topics DTD
                                    -->
<!ELEMENT outgoing (link+)>
<!ELEMENT incoming (link+)>

<!ATTLIST topic
           id CDATA #REQUIRED
           file CDATA #REQUIRED
           name CDATA #REQUIRED>
<!--
          The links
                                    -->
  <!ELEMENT link (anchor,linkto)>

  <!ELEMENT anchor (file?, start, end)>
  <!ELEMENT linkto (file?, bep)>

  <!ELEMENT file  (#PCDATA)>
  <!ELEMENT start  (#PCDATA)>
  <!ELEMENT end  (#PCDATA)>
  <!ELEMENT bep  (#PCDATA)>
```

Figure 3 The LTW assessment DTD

5.4Specific XML Elements

The XML data model offers extensible element tags which can be arbitrarily nested in order to capture semantics [4]. Information such as titles, references, sections and sub-sections are explicitly captured using nested, application specific XML tags.

The use of XML elements as the retrieval unit is believed to provide a more accurate result than using whole documents. But, as yet using XML structure has not proven useful in XML *ad hoc* retrieval [22], except for some very specific queries such as multimedia queries that specifically target images. None the less, it is the XML-IR functionality that is required for Link-the-Wiki. Links from automatically identified anchor-text to best entry points in a document are needed.

The evaluation of the Link the Wiki task will require a different and possibly more complicated method of evaluation then XML-IR. Link evaluation is very different from conventional precision / recall so far used to evaluate XML-IR at INEX. Specifically a score is needed for the identification of anchor-texts as well as for the corresponding best entry point destinations. Although standard INEX metrics such as BEPD (see [8]) might be used for the latter, scoring the former remains unaddressed.

5.5Best Entry Points (BEPs)

At INEX a best entry point (BEP) is a specific document element from which the user can perform some optimal access to a series of relevant document elements [15]. The purpose of a BEP is to complement the users' searching activities and facilitate direct entry to relevant items within documents. The identification of BEP is already a sub-task in the *ad hoc* track at INEX and the

methods that are used there may be used in Link-the-Wiki essentially unmodified.

The BEP results in the LTW submission can be expressed in the following format.

<bep> /article[1]/sec[4]/p[3]</bep>

6.EVALUATION METHODOLOGY

The Link-the-Wiki track will be held once a year and will be generally based on the following steps:

1. Participants nominate 10 or more topics (Wikipedia pages) of reasonable length for which link discovery will be performed. These pages must (obviously) exist in the XML Wikipedia collection.
2. Topics are distributed to participants who run their link discovery search engines. A submission for each topic consists of a list of selected anchor texts and corresponding links to best entry points within the Wikipedia collection. The number of incoming and outgoing links will be restricted to some reasonable and manageable number for each topic, perhaps based on topic length.

   In the initial year (2007) participants will specify links at any level of granularity, but evaluation will only be performed between whole articles (all links will be treated as "see-also" links). The Wikipedia currently contains only this kind of link and not links to best entry points.
3. The *pooling* process is performed to merge results from different participants and that correspond to the same document. The specific details of this are tied to the design of an assessment tool and are outside the scope of this paper. The pooled results for see-also links can be automatically assessed by comparison to links in existing documents. In future years it will be necessary for assessors to manually assess links.
4. The link discovery search engines will be scored with respect to performance using standard metrics (that are yet to be defined). We expect and encourage experimentation with several metrics since the best way to score runs is not immediately obvious.
5. The results are returned to participants who in turn analyze, present, and discuss their approaches at the INEX workshop.

The detailed processes will be described in the following sections.

6.1Procedure

An initial set of LTW topics are nominated by participants and a final set of at least 50 topics will be selected. These topics (Wikipedia documents) will then be orphaned by eliminating the anchor texts and their associated destinations (the XML tags, *collectionlink*, will all be discarded). The *"what links here"* information will also be discarded from the topic documents. A topic submission should identify no more than 25 anchor texts from any part of the document and identify the most relevant destinations. Furthermore, no more than 25 incoming links can additionally be identified.

In 2007 submission of BEP links will not be required, but rather document to document links will suffice. However, anchor texts should be specified. Ideally, the link engines of participants should be able to automatically find the 25 most relevant anchor texts in response to the content of given topics and specify the associated link at the XML element level in the INEX documents. This means that clicking the anchor text does not lead to an article but to the particular document Best Entry Points.

At INEX 2007 evaluation will be performed between articles only so submissions may contain BEPs but they will be automatically reduced to whole articles in evvaluation. In future years, and with the use of an assessment tool, the evaluation of more precise link specifications will be supported.

For use with an assessment tool the pooling process will need to execute once the results are all submitted. Each nominated topic will then be associated with a set of links for assessment. The pooled results might be assessed in one of two ways: automated assessment might be performed by comparing results in the pool with those already in the orphan to get a *precision* and *recall* score. Manual assessment might be used to individually assess links. The exact details of the metric and the assessment tool are outside the scope of this paper and are yet to be defined in precise detail. This will be done through discussion between track participants.

6.2Challenges

The preliminary procedure of assessment has been stated and described above. However, the detailed methodology (e.g. approaches and metrics) are still not finalized. In fact, much like it was with the *ad hoc* track at INEX, one would expect that only after some considerable experimentation with evaluating LTW submissions could a methodologically sound evaluation approach be put in place.

In terms of different element levels, article level evaluation is not dissimilar to standard *ad hoc* retrieval and some form of F-Score might be utilized. Given an orphan document, taking into account the *precision* and *recall* of identified links (both incoming and outgoing), computing some form of mean may be sufficient. The hypothesis is that a relative comparison of runs will be sufficient to derive an appropriate ranking score [16].

With automated evaluation, there is no exhaustive assessment. Consequently, some returned links may be appropriate, but not already appear in the Wikipedia. The consequence is that evaluation results may appear pessimistic.

Manual assessment is expected to be more accurate, but is time consuming. With a suitable assessment tool we believe that this effort can be reduced to reasonable levels. The design of an efficient assessment tool is currently underway.

With exhaustive assessment pooling becomes important. Pooling with the LTW is more problematic than with a traditional Cranfield experiment since there is a real possibility that there will be very little overlap between submissions. In particular, the anchor texts from the runs may only partially overlap, or not overlap at all, and links may be pointing to different BEPs. This can lead to unreliable evaluation as observed when traditional *ad hoc* pools are too shallow.

At present we are exploring ways to collect the entire set of links from all submissions, eliminate duplicates where possible, and assess all remaining links. This will at least ensure that evaluation of the systems that contribute to the pool is meaningful. It is neither clear how re-usable such a set of assessments will be nor how exhaustive a set of manually assessed links can be. This can only be studied after the track has produced the first set of results.

Figure 4. Link the Wiki Submission Interface

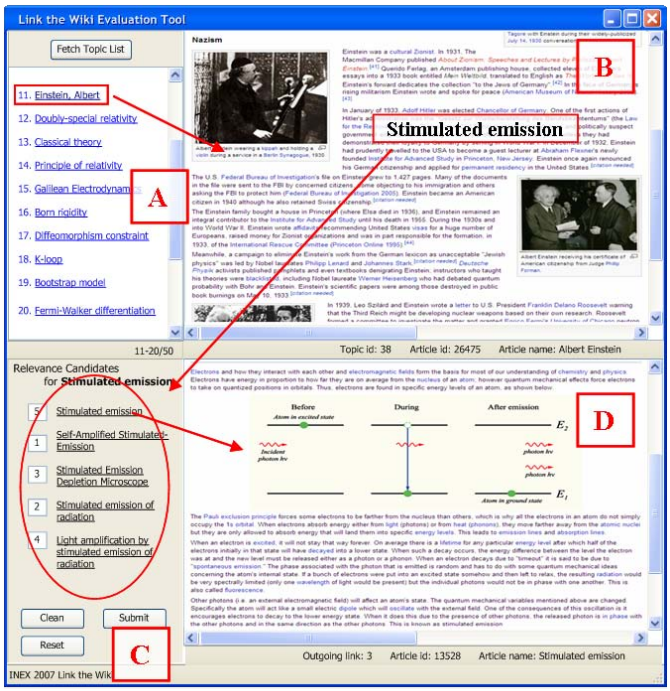


Figure 5. Link the Wiki evaluation tool

6.3Tools

An (online) assessment system will be provided for the LTW community for various evaluation scenarios. The preliminary prototype is illustrated in Figure 5.

In section A, a list of topics is displayed. The topic content is shown on the right hand side in section B. Once the user clicks on the anchor text in the topic content, a set of candidates associated with the anchor will be given in section C. A selected link in section C will show the corresponding linked-to text in section D. In this manner a user can see both the anchor text in context, and the linked-to text in context. A text box will be used to enter a relevance score for the selected link. The user can navigate through different links by clicking on link names.

This tool can also be used to view submissions as well as the pool. Section C displays the associated links with the rsv (score) from the participants' system while the content of a link is shown in section D. All anchor texts (or elements) that link to this content will be highlighted in the document in section B. This interface provides an easy way for participants to examine their result sets as well as to navigate through different anchor texts and linked contents.

7.EFFICIENCY

Missing from INEX has been any measure of search engine efficiency. Although the precision of the *ad hoc* runs is measured each year, how long it took the search engine to produce the results is completely unknown, participants don't normally publish this detail. The Link-the-Wiki track will be the first track at INEX in which efficiency will be considered.

Ideally each participant will run their solutions on the same computer configuration; however this is not feasible for several reasons: first, it is not practical to prescribe a given computer and operating system configuration and to expect participants to build it; second, prolonged use of such a machine will inevitably result in changes to the configuration (for example operating system patches might in some cases be installed but not in others); third, shipping a machine between participants is costly and time consuming and will result in changes to the configuration as an increasing number of search engines are installed; finally, bringing the search engines to the machine (for example at the workshop) is also not possible as search engines may not be portable across operating systems and doing so might start and operating system battle.

For these reasons participants will be asked to submit their runs and to state (as part of their run) the time it took for their system to produce the set of results. All this will be defined in the run submission DTD. Participants will also be asked to include configuration of the machine on which the run was generated.

It might appear at first inspection that the problem is that of building the optimal implementation of the optimal solution and running it on the fastest computer available. However, optimality is hard to define and there is a time/performance trade-off. For link discovery this is of particular interest.

An optimal set of links could be identified by a human with complete knowledge of the document collection – however it would be costly to gain such knowledge and to employ such an individual. An immediate set of results might be gained by building a finite state automaton from the titles of all documents

in the collection and a simple parser. A better (but more time consuming) result might be found with a part of speech (POS) tagger and some natural language processing. For a semi-commercial entity wanting to build an open source repository such as the Wikipedia, different subscription levels might be offered depending on the cost (in CPU cycles) of the quality of the linking.

For the purpose of comparison, using different algorithms on different machine configurations leads to some problems: like is not being compared with like and the group with the fastest machine should be able to produce the fastest runs. It is not yet clear how to address this problem and it is expected as a topic of debate. One solution is to also solicit from participants an estimate of the dollar cost of the machine (or machines) on which the result was generated. With a time and a cost, the unit of measure might be the precision-dollar. This, again, is likely to cause debate as the price of a machine increases exponentially with performance and the measure would favor network of workstation (NOW) or pile of PC (POPC) configurations - but perhaps justly.

A real-time question answering exercise was conducted at CLEF 2006. We believe that that exercise and the efficiency testing of Link-the-Wiki are the beginning of a new era in forum evaluation that started with the TREC Web Track. Once the limits of precision begin to be approached, small mutually-exclusive sets improvements begin to proliferate and are of less interest than substantial cost reductions in producing the results. This is already being seen in full-document retrieval where techniques such as impact-ordering and index pruning have been proposed. For *ad hoc* XML-IR no such techniques have yet been proposed or tested. We anticipate the *ad hoc* track at INEX adopting an efficiency task and consequently fast and effective XML-IR search engines. It should however be noted that the Link-the-Wiki task in itself will demand significantly more processing per topic than an *ad hoc* topic. Relatively few topics will mote likely suffice to severely tax slow underlying IR systems.

8.LINK-THE-WIKI 2007 AND BEYOND

In 2007 participants will submit 10 orphans each. A set of at least 50 will be distributed. For each one, each participant will identify 25 anchor-texts and for each anchor-text 5 best entry points. The time it took to generate the result and an approximate US dollar cost for the hardware will also be submitted. Runs will be reduced to a set of document-to-document links and performance measured with yet to be announced metrics.

In future years, metrics that score both anchor-text identification and the best entry point identification will be added. Links to destinations outside the Wikipedia are likely to be added too.

9.CONCLUSIONS AND FUTURE WORK

The Wikipedia is an attractive corpus for performing automated link discovery experiments since the collection is extensively hyperlinked. In this paper, we briefly described the objective and requirements of Link-the-Wiki track as well as the preliminary procedure of assessment and evaluation. The task of LTW has gone beyond traditional information retrieval that normally searched relevant whole article. The Link-the-Wiki task represents, in our view, an ideal use case for XML-IR. It aims at accessing different element levels within an XML document, which presents the most relevant components (sections, paragraphs, etc.) in relation to anchor text selected from the topic of request.

Briefly, the process of assessment can be depicted as follows. At least 50 orphan pages will be given to participants for LTW tasks. The automated discovery of document hyperlinks at the different XML element levels is performed by the participants' systems. The results are submitted to the organizers. The submissions are analyzed and the elements as well as the associated links on each topic are examined and selected as the candidates for the final evaluation. At the first stage, 25 anchor texts for each topic with the related 5 destinations will be chosen for manual evaluation. Since this pooling process and the final evaluation are manual and time-consuming, the automated approaches and the standard metrics will be investigated further first, especially for the element level evaluation (e.g. anchor text to BEP).

In addition to the evaluation, there are many options for improving the work introduced in this paper. Although the precision of the results for both the selection of elements that represent the topic and the retrieval of links associated with the elements is important, the efficiency measure is another consideration in the real world retrieval systems. Response time, the time a user must wait for a result, considers the CPU and I/O latency. An efficient LTW system will certainly be an asset to the Wikipedia and other collaborative knowledge management systems.

10.REFERENCES

[1] Adafre, S. F. and de Rijke, M. Discovering missing links in Wikipedia, *In Proceedings of the SIGIR 2005 Workshop on Link Discovery: Issues, Approaches and Applications*, Chicago, IL, USA, 21-24 August 2005.

[2] Bellomi, F. and Bonato, R. Network Analysis for Wikipedia, *In Proceedings of the 1st International Wikipedia Conference (Wikimania'05)*, Frankfurt am Main, Germany, 4-8 August 2005.

[3] Chernov, S., Iofciu, T., Nejdl, W. and Zhou, X. Extracting Semantic Relationships between Wikipedia Categories, *In First Workshop on Semantic Wikis: From Wiki to Semantic [SemWiki2006]*, Budva, Montenrego, 12 June, 2006.

[4] Dean, J. and Henzinger, M. R. Finding related pages in the World Wide Web. *Computer Networks*, 1999, 31(11–16):1467–1479.

[5] Denoyer, L. and Gallinari, P. The Wikipedia XML Corpus, ACM SIGIR Forum archive Volume 40, Issue 1 (June 2006), 64-69.

[6] Gabrilovich, E. and Markovitch, S. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis, *In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, Hyderabad, India, 6-12 January 2007.

[7] Geva, S. and Leo-Spork, Murray XPath Inverted File for Information Retrieval, *In proceedings of INEX 2003 Workshop*, Schloss Dagstuhl, 15-17 December 2003, 1-8.

[8] INitiative for the Evaluation of XML retrieval (INEX), 2007. http://inex.is.informatik.uni-duisburg.de/2007/

[9] Jeh, G. and Widom, J. SimRank: a measure of structural-context similarity, *In Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'02)*, Edmonton, Canada, 23-26 July 2002, 538-543.

[10] Jenkins, N. *Can We Link It*, 2007, http://en.wikipedia.org/wiki/User:Nickj/Can_We_ Link_It

[11] Kessler, M. M. Bibliographic coupling between scientific papers. *American Documentation*, 14(10-25), 1963.

[12] Kleinberg, J. Authoritative sources in a hyperlinked environment, *In Proceedings of the 9th Annual ACM–SIAM Symposium on Discrete Algorithms*, San Francisco, CA, USA, 25-27 January 1998, 668–677.

[13] Kumar, R., Raghavan, P., Rajagopalan, S. and Tomkins, A. Trawling the Web for emerging cyber-communities. *Computer Networks*, 31(11–16), 1999, 1481-1493.

[14] Ollivier Y. and Senellart P. Finding Related Pages Using Green Measures: An Illustration with Wikipedia, *In Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI'07)*, Vancouver, Canada, 22-26 July 2007.

[15] Reid, J., Lalmas, M., Finesilver, K. and Hertzum, M. Best Entry Points for Structured Document Retrieval – Part II: Types, Usage and Effectiveness, *Information Processing & Management*, 42(1):89-105, 2006.

[16] Salem, M., Woodley, A. and Geva, S. IR of XML documents? A Collective Ranking Strategy, In Proceeding s of Third International Workshop of the Initiative for the Evaluation of XML Retrieval, Dagstuhl Castle, Germany, 113-126.

[17] Schenkel, R., Suchanek, F. M. and Kasneci, G. YAWN: A Semantically Annotated Wikipedia XML Corpus, *In 12. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW 2007)*, Aachen, Germany, 2007, 277-291.

[18] Schönhofen P. Identifying document topics using the Wikipedia category network, *In Proceedings of the 2006 IEEE/EIC/ACM International Conference on Web Intelligence (WI'06)*, Hong Kong, 18-22 December 2006.

[19] Sigurbjornsson, B., Kamps, J. and de Rijke, M. Focused Access to Wikipedia, *In Proceedings of the sixth Dutch-Belgian Information Retrieval workshop (DIR 2006)*, TNO ICT, Delft, The Netherlands, 13-14 March, 2006.

[20] Strube, M. and Ponzetto, S. P. WikiRelate! Computing Semantic Relatedness Using Wikipedia, *In Proceedings of the 21th National Conference on Artificial Intelligence (AAAI'06)*, Boston, Massachusetts, USA, 16-20 July 2006.

[21] Trotman, A. and Geva, S. Passage Retrieval and other XML-Retrieval Tasks, *In Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, Seattle, Washington, USA, 10 August 2006, 48-50.

[22] Trotman, A. and Lalmas, M. Why Structural Hints in Queries do not Help XML-Retrieval, *In Proceedings of the 29th Annual International ACM SIGIR Conference*, Seattle, Washington, USA, 6-11 August 2006, 711-712.

[23] WiQA: Question answering using Wikipedia, 2006. http://ilps.science.uva.nl/WiQA/index.html

[24] Wikipedia, the free encyclopedia, 2007. http://wikipedia.org/

[25] Wikipedia: Size_of_Wikipedia, 2007. http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

[26] Wikipedia: Searching, 2007. http://en.wikipedia.org/wiki/Wikipedia:Searching

[27] Voss, J. Measuring Wikipedia, *In Proceedings of the 10th International Conference of the International Society for Scientometrics and Informetrics (ISSI 2005)*, Stockholm, Sweden, 24-28 July 2005.

From Passages into Elements in XML Retrieval

Kelly Y. Itakura
David R. Cheriton School of Computer Science,
University of Waterloo
200 Univ. Ave. W.
Waterloo, ON, Canada
yitakura@cs.uwaterloo.ca

Charles L. A. Clarke
David R. Cheriton School of Computer Science,
University of Waterloo
200 Univ. Ave. W.
Waterloo, ON, Canada
claclark@plg2.uwaterloo.ca

ABSTRACT

Trotman and Geva [8] suggest that XML retrieval must move from element-based to passage-based because human assessors see passages when judging relevance. Since the current XML retrieval evaluation involves returning XML elements, they suggest ways to convert passage retrieval results into XML elements. In this paper, we implemented one of their algorithms and argue that the algorithm returns a lot of excessive text. We also implemented an element-based XML retrieval algorithm and analyze why it works better, linking its behavior to the other algorithm of Trotman and Geva. We finally compare the results of these two implemented algorithms to a gold standard obtained by passage retrieval to compare the excess and the lack of text in the result sets. We conclude the paper by suggesting a better way to represent results of XML retrieval.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation, Theory

Keywords

XML retrieval, passage retrieval

1. INTRODUCTION

INEX [1] is an evaluation forum for XML retrieval. In the adhoc track, a *topic* contains a user's information need, including structural information as well as a set of query terms called a *title*. A title is what we normally type into a search engine; phrases are contained within double quotes, terms that must be in the returned elements are headed by the plus sign, terms that must not be in the returned elements are headed by the minus sign. The content-oriented (CO) task requires processing only titles. In the focused task, the result must be a set of *single* elements that are "the most exhaustive and specific" [6]. In this paper, we address adhoc, content-oriented, focused task.

Trotman and Geva [8] mentions that human assessors see passages when making a relevance judgement on results of XML retrieval. To transition into passage based XML retrieval, they propose how to convert elements into passages and passages into elements. In particular, they suggest a couple of ways to convert passages into XML elements for the focused retrieval task. This process involves finding an appropriate size of an XML element to return that is not redundant but contains necessary passages. The first algorithm of Trotman and Geva [8] to convert a passage into an XML element, takes the smallest XML element that contains the passages. The second algorithm takes the largest XML element that is contained in the passages. We call the first algorithm TG+ retrieval and the second algorithm TG− retrieval. Trotman and Geva cast doubts on specificity of the TG+ algorithm and exhaustivity of the TG− algorithm. That is, even though TG+ retrieval returns elements that contains relevant text, it may contain too much irrelevant text as well. On the other hand, TG− retrieval may return elements without much irrelevant text, but it may miss too much relevant text. In this paper, we implemented the TG+ algorithm to show that the returned results contain a lot of irrelevant text. We also implemented an algorithm analogous to TG− retrieval, and compared the exhaustivity and specificity of these two algorithms against a defined gold standard, a set of passages retrieved by a variant of BM25 [7].

SIGIR 2007 Workshop on Focused Retrieval
July 27,2007, Amsterdam, The Netherlands.

2. METHODOLOGY

As a test collection, we used INEX 2005 IEEE collection and the query topics provided for the INEX 2005 adhoc track. The corpus contains 16,819 files from various IEEE journals from 1995 to 2004. There are 39 topics, each containing a set of query terms. In INEX 2005, the results were assessed using the nxCG metric [5]. We only consider the generalized quantization because this is the only metric with published ranking in both INEX 2005 and 2006. In addition, we only ran the focused task because this task requires balancing exhaustivity and specificity, which is the topic of this paper.

In this section, we describe three different algorithms to obtain passages/XML elements. All these algorithms process a title into a set of disjunctive terms, separating phrases into terms, removing the plus sign, and ignoring terms preceded by the minus sign. We did not remove duplicate query terms within a topic and across topics. Finally, we used the Wumpus Information Retrieval System [2] to stem query terms and retrieve all positions of query term occurrences in the collection.

All three algorithms used a variant of Okapi BM25 [7]

to score passages or elements. Normally, Okapi BM25 is used for scoring documents. Its effectiveness against passage retrieval is not yet fully established. When used in passage retrieval, a score of a passage or an element P is defined as follows:

$$score(P) \equiv \sum_{t \in Q} W_t \frac{f_{P,t}(k_1+1)}{f_{P,t} + k_1(1 - b + b\frac{|P|}{avgdl})}, \qquad (1)$$

For a weight of a query term t, W_t, we used a *document-level* IDF value,

$$W_t = \log \frac{\text{total \# of documents}}{\text{\# of documents containing a term } t}.$$

When none of the documents contain a term t, we set the IDF of the term to zero. The average document length, *avgdl* is also computed at the document level. In our corpus, the average document length is 6147.97 terms. When computing the length of an XML element or a passage, we ignored XML tags. The number of time a term t appears in the passage P is denoted $f_{P,t}$. Parameters k_1 is positive, and $0 < b < 1$. This way, we can view scoring P as scoring a small document in the context of all documents in the collection.

2.1 Passage Retrieval

In passage retrieval, we disregarded all XML structures, and retrieved passages that start and end with query terms. First, we scored all such passages and ignored those that are less than 25 words long. Then we removed all the nested passages to return the top 1500 passages for each topic. The elimination of nesting involved adding to the ranking only when a passage does not contain the higher ranking passages within it and the passage is not contained in the higher ranking passages. We call the resulting set the *gold standard*, used as the basis for comparison because assessors look for elements that contain what they consider important passages.

2.2 Element Retrieval

In element retrieval, we computed the scores of all XML elements of interest taken from [3]. These are `abs app article bb bdy bm fig fm ip1 li p sec ss1 ss2 vt`. We ignored elements that are less than 25 words long, or have a zero score. We then eliminated the nesting of XML elements to get the top 1500 XML elements to return.

2.3 TG+ Retrieval

In this section, we describe how we implemented the TG+ algorithm, the first algorithm of Trotman and Geva [8] to convert passages into XML elements.

In the TG+ algorithm, after computing all passage scores, we assigned to an XML element the score of the highest scoring passages whose smallest ancestor is the XML element. We ignored XML elements that are not of interest, and those that are less than 25 words. Finally, we eliminated the nestings of the XML elements to return the top 1500 XML elements.

Figure 1 illustrates this approach. Suppose we have four passages with Okapi scores; passage 1 with a score of 5.2 to passage 4 with a score of 4.0. Passage 1 spans through paragraph 1, 2, and 3, which are under section 1 and an article. Similarly for other passages. After we computed Okapi scores for these passages, we assigned scores to corresponding XML elements. The score of 5.2 is assigned to `sec[1]`, the smallest element containing passage 1. Similarly, the score of 5.0 is assigned to `article`. Because passage 3 has a higher score than passage 4, we assign the score of passage 3, 4.7 to `p[7]`. Next, we get rid of nesting while taking the top scored elements. We first take the element with the highest score, `sec[1]` and assign a rank of 1. The element with the second highest score, `article` is eliminated because it causes a nesting of `sec[1]` within it. We can safely take the element with the next highest score, `p[7]` because it does not cause a nesting with `sec[1]`, and assign a rank of 2.

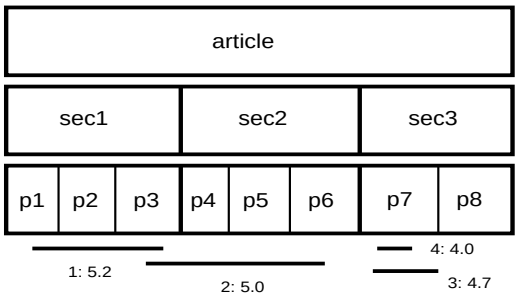


Figure 1: Assigning Scores and Nesting Elimination in TG+ Retrieval

Table 1: TG+ and Element Retrieval at $k_1 = 10$ and $b = 0.9$ in INEX 2005 CO-focused

	nxCG[10]	nxCG[25]	nxCG[50]	MAep	iMAep
TG+	0.1856	0.1774	0.1633	0.0612	0.0387
Element	0.2586	0.2323	0.217	0.0929	0.0715

3. EVALUATION AND ANALYSIS

3.1 Performance Against INEX 2005

We trained both element retrieval and TG+ retrieval over the INEX 2005 corpus and the query set. Figure 2 and Figure 3 show the scores for different values of k_1 with $b = 0.8$ using the nxCG metric, mean average precisions (MAep) and interpolated MAep (iMAep) in the CO.Focused task. We then chose $k_1 = 10$ for TG+ retrieval and $k_1 = 4$ for element retrieval for training b as seen in Figure 4 and Figure 5. The results of nxCG and MAep/iMAep metrics seem correlated as both have similar curves and give maximum values at the same parameters. The values of both k_1 and b need to be quite large for both algorithms to perform well. Clarke [3] also points out this phenomenon for his Okapi-based passage retrieval algorithm that is quite different from these two. Having a large k_1, therefore, seems to be necessary for using Okapi BM25 for passage retrieval.

These figures suggest that the simple element retrieval performs much better than TG+ retrieval. To see why, we compared the results of both algorithm at $k_1 = 10$ and $b = 0.9$, which are optimal parameters for TG+ retrieval. Table 1 shows that even if at an optimal setting, TG+ re-

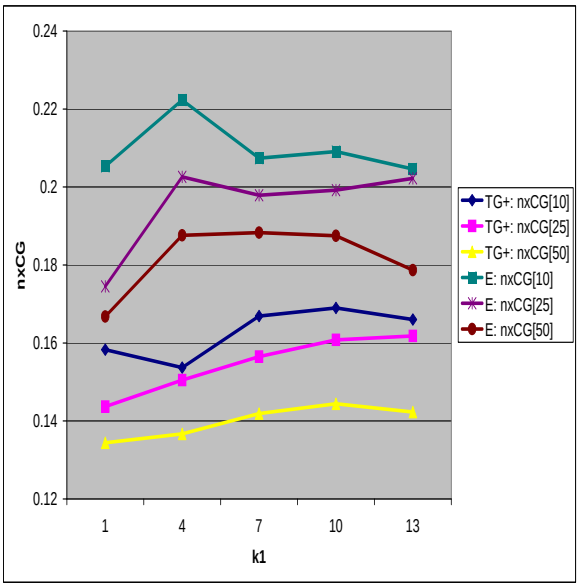


Figure 2: nxCG: Training on Different k_1 with $b = 0.8$ in INEX 2005 CO-Focused

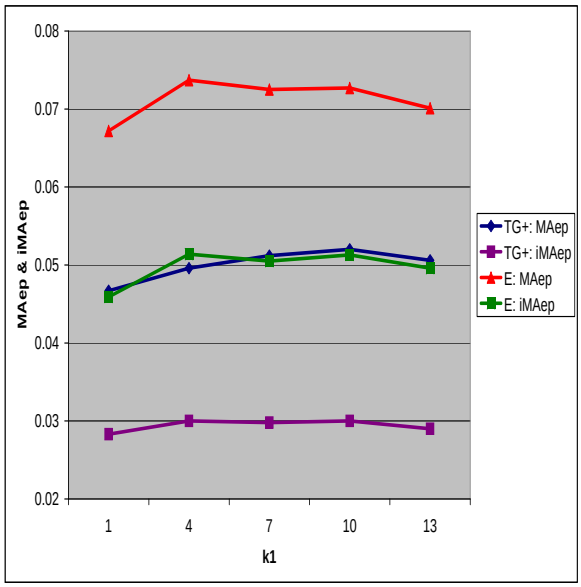


Figure 3: MAep/iMAep: Training on Different k_1 with $b = 0.8$ in INEX 2005 CO-Focused

trieval performs significantly worse than element retrieval.

We analyzed the submission files of TG+ and element retrieval and realized that in the TG+ algorithm, the majority of the elements returned had a large granularity; many of them were either `/article` or `/article/bdy`. In the element retrieval, however, the returned elements had much finer granularity such as paragraphs and sections.

For example, as wee see in Figure 6, the first element returned for the first topic in both retrieval methods is from the same file, `ex/2001/x1026` that spans through positions 27040000 and 27046835. In TG+ retrieval, the element returned was `/article/bdy`, corresponding to positions 27040246 through 27045776 with a score of 42.53. The passage corresponding to this element that gives the score of 42.53 spans through positions 27042389 and 27043619. In element retrieval, the element returned was `/article/bdy/sec[4]` corresponding to positions 27042506 through 27043559 with a score of 40.90. We see that `/article/bdy`, the smallest element that contains the highest scoring passage, is much longer than the passage. However, /article/bdy/sec[4], the element returned in element retrieval contains much of the passage without much excessive text.

The element, `/article/bdy/sec[4]`, returned by element retrieval was not returned by TG+ retrieval because the highest scoring passage within the element that spans through positions 27042550 and 27043534 only scored 41.02, lower than the highest passage contained in `/article/bdy`. On the other hand, element retrieval did not return `/article/bdy` because its score, 39.44 is lower than the score of `/article/bdy/sec[4]`, 40.90.

The above observation suggests that the TG+ algorithm is not a good approach for converting passages into an XML element because it returns a lot of excessive text. The TG− algorithm, taking the largest XML element contained in a passage, would likely perform well for the same reason element retrieval performs well; the returned elements would be unlikely to contain too much excessive text. However, both element retrieval and TG− retrieval may miss too much text to reduce the overall performance.

3.2 Performance Against the Gold Standard

In the previous section, we observed that TG+ retrieval tends to return excessive text. We also speculate that element retrieval may be missing too much text. To measure how much text is missing or in excess for both algorithms, we compared their results against our gold standard as follows.

First, because our goal is to convert passages into XML elements, and human assessors see passages when making relevance judgement, the gold standard must be passages. The best XML elements are those that cover the gold-standard passages sufficiently, but not much more. We created the gold standard using passage retrieval with the same parameters as TG+ retrieval, $k_1 = 10$ and $b = 0.9$.

Next, we compared a set of passages/XML elements returned by both TG+ and element retrieval against the gold standard at each rank up to 1500. The percent *lack* at rank r is defined as the percentage of the gold standard up to rank r that is not covered by the returned elements up to rank r. We can think of percent lack at rank r as how much text a user is missing (the user wants to read passages in

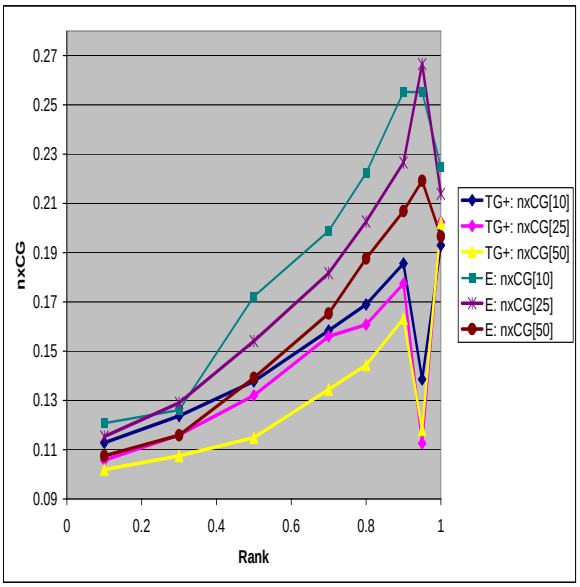


Figure 4: nxCG: Training on Different b in INEX 2005 CO-Focused

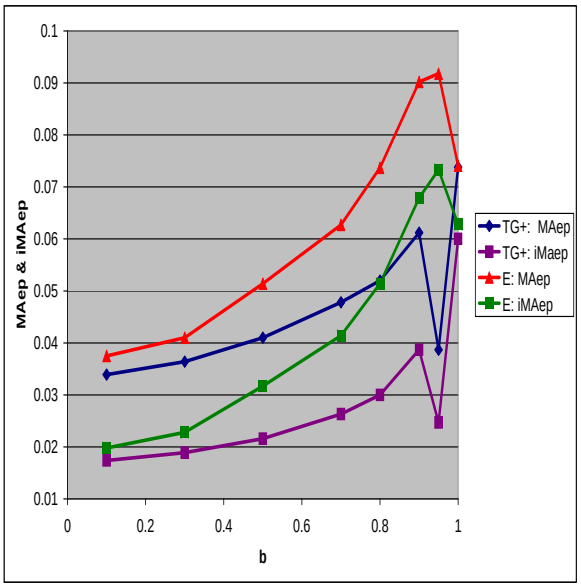


Figure 5: MAep/iMAep: Training on Different b in INEX 2005 CO-Focused

the gold standard) when the user reads from rank 1 to rank r. Similarly, the percent *excess* at rank r is defined as the percentage of the returned elements up to rank r that does not cover the gold standard up to rank r. We can think of percent excess as how much text a user reads that the user did not have to read (because the user only wants to read passages in the gold standard) when the user reads from rank 1 to rank r. We averaged both the percent lack and the excess over all topics.

Figure 7 show that overall, the excessive text returned by TG+ retrieval is larger than the excessive text returned by element retrieval. However, as the rank increases, the amount of excessive text for element retrieval increases to the point that towards the end of the ranking, the level of excess for both algorithms are about the same. Figure 8 shows that element retrieval misses much more text than TG+ retrieval does, and the general trend for both algorithms is to have less missing text as rank increases.

The reason that element retrieval performs better on the nxCG, MAep, and iMAep metric for the focused task is because having excessive text is punished more than missing text. In the INEX focused task, specificity is preferred to exhaustivity [6]. In the same manner, the TG− algorithm, that takes the largest element contained in the passage, will likely score high in the focused task. The TG+ algorithm would score high in other tasks that place preference on exhaustivity over specificity.

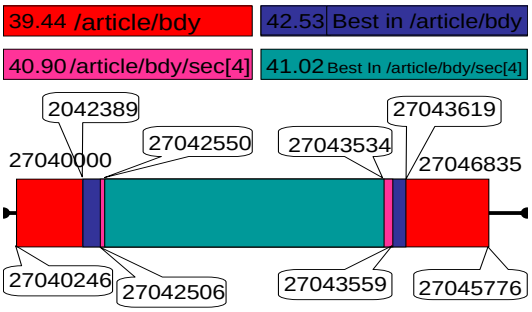


Figure 6: Elements and Passages Relating to Rank One Result

4. DISCUSSION

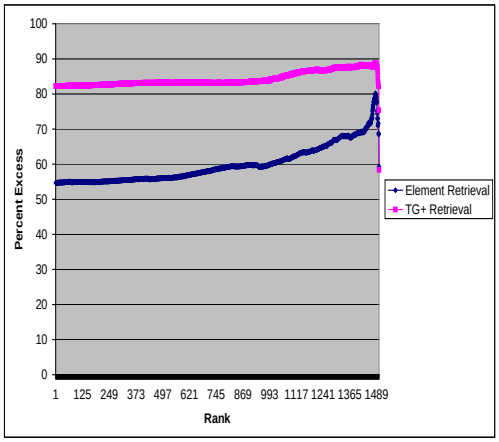


Figure 7: Percent Excess of the Gold Standard

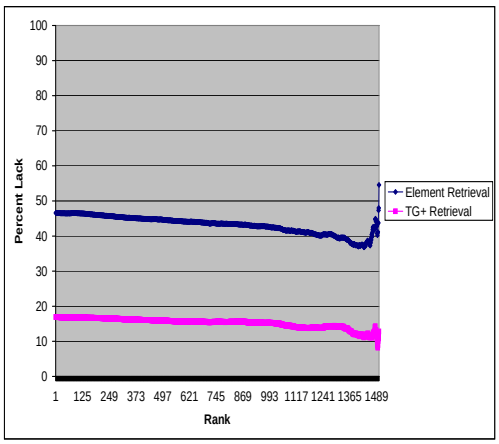


Figure 8: Percent Lack of the Gold Standard

Table 2: Average Length in Passage, Element, and TG+ Retrieval

	Final	Intermediate	Nested
Passage	1760.56	N/A	2424.85
Element	750.87	N/A	1354.69
TG+	3322.06	770.72	874.56

In this section, we discuss the behavior of parameters k_1 and b in passage retrieval. We used an average document length for $avgdl$ in computing an Okapi score for a passage in Equation 1. Because most passages are less than a size of a document, $|P|/avgdl$ is less than 1 most of the time. The result is that in the denominator, we multiply k_1 with something very small. Then if k_1 is also small, the denominator approaches to $f_{P,t}$, turning Equation 1 into

$$score(P) \equiv \sum_{t \in Q} W_t 1. \quad (2)$$

Therefore, one reason for a large k_1 value is to prevent Equation 1 from degenerating into Equation 2 when the length of P is smaller than the average document length. Similarly, a large b augments the effect of length normalization in $b|P|/avgdl$ in the denominator, and this sets off the large gap between the average document length and a passage length. Therefore, it appears that the choice of the average document length is balanced by the choices of k_1 and b.

To see how small the passage lengths are compared to the average document length, we computed various passage lengths. Table 2 shows the average length of the top 1500 final results (Final), the passages that produced the top 1500 final results (Intermediate) (only applicable to TG+), and the top 1500 results before the elimination of nesting (Nested), for passage, element, and TG+ retrieval. From the average document length of 6147.97, we compute the average passage length to be 1538.47. Then both the TG+ and the element algorithms retrieve fairly small elements/passages, 770.72 for TG+ retrieval, and 750.87 for element retrieval, compared to the average passage length of 1538.47. However, the average length of the elements returned by TG+ retrieval is about four times as large (3322.06) as the average length of the corresponding passages (770.72). This shows that the TG+ algorithm is inherently ineffective for the following reasons. The passages returned by TG+ algorithm is about the right size because this is about the same size as the elements returned by element retrieval that performs well. Therefore, it is the way we assign passage scores to elements, rather than the choice of $avgdl$, k_1, and b, that makes TG+ retrieval less effective than element retrieval.

Finally, it is interesting to note that the average length of the passages returned by the passage retrieval, 1760.56 is close to the average passage length of 1538.47. The difference between the average lengths of the passages returned by TG+ and passage retrieval may be accounted for by how nesting was eliminated. In passage retrieval, we eliminated nestings of the passages that eliminates more nesting than nesting elimination of elements in TG+ retrieval. Furthermore, the fact that the average passage length returned by passage retrieval is close (in fact more) than the average passage length implies that TG− retrieval would likely be able to find the largest element within the passages. However, we believe that the best elements to return in XML retrieval

is multiple consecutive elements that contain passages just enough.

5. RELATED WORK

Huang et al. [4] implemented their own version of the TG+ algorithm, where the method of identifying passages are different from ours. Instead of considering all possible passages as we did, they considered a fixed size passages obtained from sliding windows. Moreover, they first performed document retrieval, and then ran passage retrieval on the top scoring documents, whereas we directly retrieved high scoring passages. When scoring passages, they used a simple term frequency approach and two language modeling approaches, whereas we adopted a variant of Okapi BM25, which is used for document retrieval.

The effectiveness of the TG+ algorithms of Huang et al. and ours can be easily compared because both of us used the same test set, the INEX 2005 IEEE collection, ran the same CO.Focused task, and used the nxCG generalized quantization to score the results. Huang et al. compared their passage based XML retrieval results against the INEX 2005 CO-Focused task submissions of IBM Haifa that ranked 4th and of University of Amsterdam that ranked 28th in the Mean Average Precision (MAep) ranking. They concluded that their algorithm ranked between these two. Our results of the TG+ algorithm at $k = 10$ and $b = 0.9$ with MAep of 0.0612 also ranks between these two. On the other hand, our element retrieval algorithm with the same parameters that has a MAep of 0.0929 easily ranks the first preceding the first result of IBM Haifa ranked 1st, that has a mean average precision of 0.0917.

Huang et al. conclude the paper by mentioning that a passage retrieval algorithm can produce effective element retrieval results because it ranked between the 4th and the 28th out of 44 submissions. However, the fact that both versions of TG+ retrieval, despite with very different implementations of passage retrieval, *only* ranked between the 4th and the 28th implies that TG+ retrieval is not a good idea for turning passages into an element. The comparison of our TG+ retrieval against our gold standard along with the average length statistics in Section 4 also implicate that it does not perform well because of excessive text inherent to the very idea of TG+ retrieval.

6. CONCLUSIONS AND FUTURE WORK

We implemented three algorithms to test the effectiveness of the first algorithm of Trotman and Geva [8], which converts the results of passage retrieval into XML elements. This TG+ algorithm takes the smallest XML elements that contains the passages. It appears that the parameter k_1 in Okapi BM25 must be high for any passage retrieval to yield a good performance. We showed that the TG+ algorithm does not perform as well as the simple element retrieval, where we scored all XML elements, not passages. By comparing the result sets of the TG+ and the element retrieval algorithm, we realized that the TG+ algorithm tends to return a lot of excessive text. Even though element retrieval performs well, we speculated that it may miss too much text.

To see how much of text is missing or excessive, we created the gold standard, a set of results obtained from passage retrieval using the same Okapi BM25 parameters. We then computed an average percent excess and an average percent lack of text for returned results over all topics. Although element retrieval does miss out a lot of text, it scores well on the focused task of INEX because the task has a preference of specificity over exhaustivity. Because the TG− algorithm that returns the largest XML elements contained in the passages will not have much excessive text as in the case of element retrieval, the performance of the TG− algorithm may be similar to the one obtained by element retrieval. Ultimately though, the best elements to return must be a series of XML elements.

Future work involves studying the effectiveness of Okapi BM25 in passage retrieval setting, including analyzing why a large k_1 value works. We also would like to implement the TG− algorithm to compare it against the TG+ and the element algorithms. Finally, we would like to implement multiple passage retrieval and measure its performance.

7. REFERENCES

[1] INEX: INitiative for the Evaluation of XML Retrieval . Accessible at `http://inex.is.informatik.uni-duisburg.de`, 2007.

[2] S. Büttcher. the Wumpus Search Engine. Accessible at `http://www.wumpus-search.org`, 2007.

[3] C. L. A. Clarke. Controlling overlap in content-oriented XML retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 314–321, New York, NY, USA, 2005. ACM Press.

[4] W. Huang, A. Trotman, and R. O'Keefe. Elemement retrieval using a passage retrieval approach. In *the 11th Australian Document Computing Symposium*, 2006.

[5] G. Kazai and M. Lelmas. INEX 2005 evaluation metrics. *Springer-Verlag, Lecture Notes in Computer Science*, 3977:16–29, 2006.

[6] S. Malik, G. Kazai, M. Lelmas, and N. Fuhr. Overview of INEX 2005. *Springer-Verlag, Lecture Notes in Computer Science*, 3977:1–15, 2006.

[7] S. Robertson, S. Walker, and M. Beaulieu. Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive track. *7th Text REtrieval Conference*, 1998.

[8] A. Trotman and S. Geva. Passage retrieval and other XML-retrieval tasks. *SIGIR 2006 Workshop on XML Element Retrieval Methodology*, August 2006.

The Task First, Please

Valentin Jijkoun
ISLA, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
jijkoun@science.uva.nl

Maarten de Rijke
ISLA, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
mdr@science.uva.nl

ABSTRACT

We examine the current state of evaluation exercises for automatic Question Answering (QA) systems, specifically targeting the QA task (QA@CLEF) as it is being evaluating with the setting of the Cross-Language Evaluation Forum (CLEF). We describe several key issues for the evaluation of QA systems and show how they are problematic in the current setup of the tasks at QA@CLEF. We argue that many of the problems are caused by the lack of a clear understanding of the QA task that should include potential users, types of information needs, types of available information resources. Finally, we propose several scenarios for QA and focused retrieval tasks that address these problematic issues. Our main conclusion is simple but important: a clear task definition is paramount for a meaningful evaluation of automatic systems, as evidenced by the overview of the QA evaluation setups.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.4 [**Information Systems Applications**]: H.4.2 Types of Systems; H.4.m Miscellaneous

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

Focused retrieval, Question answering, Evaluation

1. INTRODUCTION

Question answering (QA) provides an important example of focused retrieval. In response to a user's question, QA systems are supposed to return an answer instead of a ranked list of documents from which the user has to extract the answer herself. Situated at the interface between computational linguistics and information retrieval, the task has attracted a great deal of attention over the past few years.

The launch of a dedicated question answering track at TREC 1999 has proved to be an especially important stimulus to research in the area. Most of the questions considered at the TREC QA track (and its descendants, the NTCIR and CLEF QA tracks) are fact-based, whose answers are typically named entities such as people, organizations, locations, dates (*"Who killed Lee Harvey Oswald? When was Mozart born?"*). Variations that have also been receiving attention include "list" questions such as *"Name all airports north of the polar circle*", definition questions such as "What is a sequencer?", as well as more complex questions that ask for "information nuggets" to be gathered from multiple documents.

Since focused retrieval and QA tasks are relatively new for the Information Retrieval community, there is an ongoing discussion about the nature of the tasks and approariate evaluation environments. Following up on analysis on the QA task in the literature [12], in this paper we identify a number of issues with the task scenarios being used today at one of the evaluation platforms for QA: QA@CLEF. We argue

(1) that a task model is important for informing the key ingredients of a retrieval task, including QA;

(2) that the QA task definitions used at CLEF leave a number of things to be desired, as a result of which key notions such as "answerhood" and "exactness" are seriously underspecified.

To remedy these shortcomings we argue that explicit *task definitions* should come first and that key ingredients (such as the definition of answerhood, the metrics to be used, etc) should be provided as part of the task definition. We propose a few QA task models that come with natural definitions of answerhood and metrics. One of these tasks (WiQA) was run as a pilot at CLEF 2006.

The aim of this note is ask questions and to stimulate discussion about current QA evaluation practices. In our analysis of current QA evaluation practices we use QA@CLEF as a main vehicle for discussion—this should not be interpreted as "CLEF-bashing." On the contrary, QA@CLEF has proved to be tremendously useful as a platform for fostering QA research in Europe, especially in languages other than English. Our comments and suggestions should be interpreted as suggestions for making the task even more valuable.

The remainder of this paper is organized as follows. In Section 2 we review the assumptions underlying the QA@CLEF track, relating it to the TREC QA on which it builds. Then, in Section 3 we indentify some of the key issues problematic for current QA evaluation exercises at CLEF. We proposed a few alternative scenarios in Section 4 and conclude in Section 5.

SIGIR 2007, Workshop on Focused Retrieval July 27, 2007, Amsterdam, The Netherlands

2. QUESTION ANSWERING AT CLEF

At TREC, the QA scenario that is being used as a model that informs decisions about what constitutes a correct answer and about what suitable metrics are, is that of an information analyst. Actually, very little of the analyst's rich context is included in the scenario used at TREC—no background knowledge, no factbooks or definition of an overarching task is included the task definition at the TREC QA track.

In 2003, a QA track was launched at CLEF, the Cross-Language Evaluation Forum [8]. Traditionally, retrieval tasks that are being evaluated at CLEF have involved multiple languages, either in the form of multiple monolingual tasks, bilingual tasks, or crosslingual tasks. The QA@CLEF track was no different: initially, it "copied" its task definition from the TREC QA track into three monolingual tasks (Dutch, Italian, Spanish), without, however, taking over TREC's assumption of the analyst's user scenario. In later years several languages were added, as were a cross-language variation of the task (with questions in one language and answer bearing documents in another). Originally, the conditions copied from the TREC QA track were more or less those from its 2001 edition: answers were 50 bytes long or exact, and participants could return up to three answers per question. The corpus used consisted of 1994/1995 newspapers. The questions were factoids only, and they were back-generated from the corpus.

In 2004, nine source languages and seven target languages were considered at QA@CLEF [9]. In addition to factoids, about 10% of the questions were definition questions, and another 10% did not have any answer in the corpora. To reduce the asssessment effort, the systems' output was reduced to a single, exact answer-string per question.

In 2005 the number of languages considered grew again (yielding 8 monolingual and 73 cross-lingual tasks) [13]. There was little or no innovation in the main task being assessed. So-called *temporally restricted* questions were added, which contain either an event that constrains the answer (e.g., *Who was Uganda's president during Rwanda's war?*), or a date (e.g., *Which Formula 1 team won the Hungarian Grand Prix in 2004?*), or a period of time (e.g., *Who was the president of the European Commission from 1985 to 1995?*). As in the previous year, a single exact answer per question was required.

In 2006, a number of changes were implemented for the main task at QA@CLEF [10]. For a start, list questions were included for the first time, and systems had to return short snippets containing answers to the test questions; the snippets were required to be short but sufficiently informative so as to allow the assessors to determine the correctness of the answer (without additional means). In addition, three pilots were run: the Answer Validation Exercise (AVE), the Real-Time QA Exercise (RTE), and Question Answering Using Wikipedia (WiQA). For AVE, systems were given triples of the form (Question, Answer, Supporting Text) and were asked to decide whether the Answer to the Question is correct and supported or not according to the given Supporting Text. At RTE, QA systems had to answer as many questions as possible in as little time as possible. WiQA was based on a scenario of a user exploring Wikipedia, and wanting to harvest additional bits of important information relevant to a Wikipedia article she is currently reading [3].

At the time of writing, the 2007 edition of the QA@CLEF task is in progresss. The main task has changed somewhat. It now uses a heterogeneous corpus, consisting of newspapers and Wikipedia—from non-overlapping periods of time: for some languages the newspaper collection dates back to 1994/1995, while the Wikipedia dump used is from late 2006. The Real-Time Evaluation pilot will run for a second time, and the WiQA pilot has merged with the WebCLEF web retrieval task at CLEF. A new pilot is being run which aims to assess the performance of QA systems when working on speech transcripts.

The number of participating teams at QA@CLEF has risen from 8 in 2003 to 37 in 2006—that's a tremendous success, but it also underlines the importance of a solid and well-defined task definition.

3. WHAT'S WRONG?

Increasingly, the QA@CLEF track has moved away from the TREC QA information analyst scenario. In principle, this is a laudable development, as we, as a community, will not be pushing the state of the art in QA in case we are merely repeating the same task at different venues and in different languages. However, this move has not been a move toward an alternative task scenario—no explicit task scenario has now been adopted at QA@CLEF, leaving many key dimensions of the task underspecified. Below, we review some of these dimensions and identify ones where, in our view, the current practice is not sufficiently explicitly specified as well as ones where clear choices have been made.

3.1 Exact Answers

Despite the drive of many researchers (and the TREC track) to focus on *exact* answers, users might not actually like or want simply exact answers: Lin et al. [7] show that users generally prefer answers *embedded in context*. The QA@CLEF task has already made an important step in this direction: in the 2006 setup systems were required to provide short document *passages* that justify the "answerhood" of the returned answers. The maximum length of supporting snippets, though, was set somewhat arbitrarily to 700 characters.

Another important issue is the "*exactness*" of answers. Assessors are typically asked to check whether answers are exact, both syntactically (i.e., they do not contain any "noise") and semantically. As was noted by participants in the TREC QA task,[1] this decision highly depends on assessors' background and expectations, and on the context in which a question arises. E.g., for the TREC question *Q160.7 "Where is the IMF headquartered?"*, the answer *"Washington"* was judged as exact, but for the question *Q152.1 "Where was Mozart born?"* the answer *"Salzburg"* was judged as inexact because assessors had the answer *Salzburg, Austria* in mind.

3.2 How Many Answers?

Whereas in the 2003 edition of the QA@CLEF tasks systems could return up to three answers for one questions, in the latest evaluation campaigns (both CLEF and TREC) a single answer is required. The decision to allow only answer might be a compromise between the amount of manual assessment of the submitted runs and the potential usability of QA systems. Since in the "information analyst" setting for document retrieval systems (at TREC and CLEF), as many as 1000 document are typically examined for relevancy, QA's focus on the top-1 answer in the simiar setting is hard to justify. At the same time, in the context of, e.g., Mobile QA [16] or real-time quizzes such a restriction would seem natural and even essential.

[1]Discussion on the TREC QA mailing list on October 6, 2006 started by Mark Greenwood.

The TREC complex interactive QA (ciQA) task [6] partially addresses this issue by allowing assessors to *interact* with a QA system for 15 minutes for one topic.

List questions, such as *"Name all airports in London"* present a particular challenge for the evaluation. E.g., QA@CLEF task switched from precision/recall-based evaluation for list questions to a "one complete answer" evaluation, and distinguishes closed list questions (e.g., *"What are the names of all of Bach's children?"*) and open list questions (e.g., *Name several most famous Bach's works.*, with the idea that they require different evaluation measures.

3.3 NIL Questions

What should a system do if it is not capable to locate an answer in a given document collection? Should a system back off and explicitly indicate with a *NIL* response, or try to use other available resources (e.g., Web, encyclopedias, newspapers) to find answers? Would a real-world user be interested in knowing that the system cannot find an answer, or would she prefer to at least receive "the best guess" so as to get started [12]?

3.4 Types of Questions

What types of questions should a QA system deal with? What questions should a QA evaluation exercise deal with? A small study of questions extracted from search engine logs [4] indicates that most users ask *procedural* questions (38%) such as *"How to cook a ham?"*, but factoids (e.g., *"What did caribs eat?"*) also constitute a substantial portion of questions (10%). Other common question types include *description* (13%, e.g., *"Who is victoria gott?"*) and *explanation* (10%, e.g., *"Why people do good deeds?"*).

Although QA evaluation exercises traditionally focus on factoid questions, with TREC's "OTHER" questions, CLEF's "definition" question and NTCIR's "why" questions [2] the attention is moving beyond factoids. Still, the reasons why specific types of questions are included in evaluations in specific proportions are not motivated by the requirements of potential users of QA systems.

3.5 Question Generation

Generating questions for a QA evaluation exercise is a laborious process. In its first year, the TREC QA track used questions back-generated from a corpus of newspaper/newswire documents, which made the questions somewhat unnatural and the task somewhat easier since the target document contained most of the question words [14]. In later years, questions at TREC's QA track were created by assessors, informed by query logs and based on their own interests. In contrast, for lack of a clear scenario, QA@CLEF has only dealt with question back-generated from the corpus of newspaper documents used. We believe that this is problematic (for the reasons described above)

3.6 Matching Needs and Sources

Librarians are good at selecting appropriate sources for addressing a specific user's information need. For questions like *"When was Mozart born?"* or *"What is a sequencer?"* they would probably consult an encyclopedia, while for a question like *"Which countries did Bush visit in 2005?"* newspapers seem a more appropriate source. A QA system intended for real world use should also match different available information resources to user's information needs. Why would we want to find Mozart's date of birth in a newspaper collection (at WebCLEF 2007) or Marlon Brando's age in a blog (as in the 2007 of the TREC QA track), if more natural and even more reliable sources are available?

The QA evaluation exercises are moving in the direction of diversifying data sources: Wikipedia is used at CLEF QA, a blog corpus is used at TREC (although the TREC questions are still mostly factoid), Google's view of the Web is used at WebCLEF 2007. Still, there is a long way to working with types of questions that match the types of collections used in the collection-based QA.

3.7 Multilinguality

At CLEF and NTCIR, multilinguality has been one of the key starting points. However, for the cross-lingual tasks (i.e., questions in language X are supposed to be answered using a document collection in language Y, a so called "X to Y" task), the evaluation questions are typically constructed by translating questions of a monolingual QA task into a different language (e.g., translating questions from Y into X, and thus creating an evaluation set for the "X to Y" QA task). This simplifies evaluation, but unfortunately creates many questions that are highly unnatural regarding the information sources. Why would a Dutch-speaking user be interested in answering the questions *"Who was Flaubert?"* from a collection of Spanish newspaper articles?

4. NEW SCENARIOS

Given the many dimensions outlined above, how should we go about evaluating QA? We see two possible options here:

1. Evaluating QA as a user-driven information access task: we first define who our users are. This will imply determining what kind of information needs they have, what resources they allow, and what constitutes proper result presentation(s), and evaluation measures.
2. Answering questions as a means for evaluating certain NLP tools or techniques: "I have a parser/tagger/analyzer/... and I want questions for which I can use the parser/tagger/analyzer/... to demonstrate its usability." Usually, this strategy leads to a clear but narrow definition of QA, not driven by information needs but by expected applicability of a specific tool or technique. E.g., the IR step can be dropped, questions can be pre-categorized, e.g., as "targeting synonymy and paraphrasing," "requiring basic world knowledge"—creators of different NLP tools may be interested in different categories of questions.

We believe that much confusion results from mixing options 1 and 2, and this is what has happened at QA@CLEF. The result is that many things are dealt with in a very ad hoc way: types of questions, evaluation measures, result presentation, choice of collection, etc.

If we are right, and the lack of an explicit task scenario at QA@CLEF is problematic, how should we move forward? Below we list a number of possibilities of task scenarios that we believe address the isssues identified in the previous section *and* that are worth pursuing.

Before we list our suggestions, we specify what we believe are natural criteria on scenarios to be considered for retrieval experiments at CLEF:

- The task should correspond as close as possible some real-world information need with a clear definition of a user;
- Multi- and cross-linguality should be natural (or even essential) for the task;

- The collection(s) used in the task should be the source of choice for the user's information need;
- Test questions should be generated by people having a genuine interest in the topic at hand;
- Collections, topics and assessors' judgements, resulting from the task should be re-usable in future; and finally,
- The task should be challenging for the state-of-the-art technology.

Against this background, then, we list a number of alternative task scenarios that we believe would make a meaningful QA evaluation effort.

4.1 Intelligence gathering

In analytical question answering [11], the users are information analysts and questions are not factoids for which answers come in a fairly limited number of "formats," but they are exploratory in nature, seeking to find out what is generally available on the topic of the question. E.g., *"What has been Russia's reaction to the U.S. bombing of Kosovo?* Here, appropriate responses can be taken to be frames, consisting of bags of atributes associated with a (news) event. Newspapers form a natural corpora to use in this scenario. The TREC QA task and especially the TREC ciQA task target at this type of scenarios: questions are assumed to follow one of the pre-defined templates (reflecting recurring interests of analysts) and assessors (users) may interact with the QA system within a specified time interval. The final decision about the correctness and the number of answers found with the help of the system is up to the assessor.

4.2 Event-targeted QA

In a different scenario, a user (e.g., a journalist or a history student) needs to collect background information around a specific event: e.g., persons involved, their occupations at the time of the event or later, ages, relations, places (populations, exact locations, distances), other details mentioned in connection with the event, possibly other related events, or even different perspectives on the event, etc.—whatever she might find important. The scenario is that the user starts with an article mentioning or describing event and has further questions about it, "stemming" from this initial information and her own knowledge.

In this setting the use of heterogeneous collections (newswire, blogs, encyclopedias, etc.), is much justified: more general questions ("*Where exactly in Iraq is Basra?*") are naturally answered using an encyclopedia, but for more specific questions (e.g., "*Which countries did Hussein visit in 1991?*") newspaper texts are a good (and maybe the most appropriate) source. Possible question types would include temporally and geographically restricted questions, as well as definition, relationship, list questions, and questions about subjective aspects and opinion questions (for which, e.g., blogs would be a natural source). Questions can be of any type in this scenario, and a ranked list of answers would seem most appropriate here, while a limited form of multilinguality seems natural, especially when the event at handtook place acros the border, or if the user is interested in the international perspective on the event being considered.

4.3 Trivia game show

Trivia are a source of entertainment for many, as is witnessed by game shows, trivia board games, as well as a large number online resources, where users both ask and answer such questions.[2] It is usually clear what the answer to a trivia question is, which makes the evaluation of trivia-based QA easier. The unique correct answer is known in advance, as defined by the game organizers. Answers to questions are always short strings (entities, actions, events). No specific information source is enforced, which means that a system may use any sources available (encyclopedias, the web, thematic corpora, etc.).

While a QA system that is intended for answering specifically trivia questions is not necessarily a useful real-world application (other than for entertainment purposes), if provides a clear definition of the task and straightforward evaluation measures, that take into account both answer correctness and the time spent by the system. Using such scenario would be a natural option for, e.g., the Real-Time QA Exercise (RTE) held at CLEF [1]. Multilinguality does not seem appropriate here, while the requirement that the answers are sufficiently exact seems reasonably natural.

4.4 WiQA 2006 and WebCLEF 2007

The CLEF 2006 WiQA task [3, 5] and CLEF 2007 WebCLEF task [15] take, as the scenario, a user collecting important information on a specific topic. E.g., the user might be writing an essay or updating a encyclopedia article on the topic, and is gathering "important" information nuggets that are worth including in her report. An automatic system is supposed to help the user to locate new important bits of information in Wikipedia (for the WiQA task) or on the Web (for the WebCLEF task). While not instantiating a traditional QA scenario—it really only asks a single question about the topic at hand: what should I know about it?—, these two tasks provide two different frameworks for evaluating focused retrieval systems, in which, moreover, multilinguality comes natural, as important information may be expressed in a language other than the language of the topic statement. Finally, the task suggets natural document sources—Wikipedia and/or the web.

5. CONCLUSIONS

We examined the current state of evaluation exercises for automatic Question Answering systems, specifically targeting the QA@CLEF task. We have described several key issues for the evaluation of QA systems and showed that they are problematic in the current setup of the tasks.

We argue that many of these problems are caused by the lack of a clear understanding of the QA task that should include potential users, types of information needs, types of available information resources. The lack of clarity on these dimensions makes it difficult to justify the setup and evaluation decisions for a QA task. Finally, we proposed several scenarios for QA and focused retrieval tasks that address the problematic issues.

Our main conclusion is simple but important: a clear task definition is paramount for meaningful evaluation of automatic systems, as evidenced by our overview of the QA evaluation setups.

6. ACKNOWLEDGEMENTS

We are grateful to Diana Santos and other members of the organizing group responsible for QA@CLEF. This research was supported by the Netherlands Organisation for Scientific Research (NWO) under project number 220-80-001.

[2] E.g., `http://www.funtrivia.com`

7. REFERENCES

[1] C. L. A. Clarke, G. V. Cormack, and T. R. Lynam. Exploiting redundancy in question answering. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 358–365, New York, NY, USA, 2001. ACM Press.

[2] J. Fukumoto, T. Kato, F. Masui, and T. Mori. An overview of the 4th question answering challenge (QAC-4). In *Proceedings of the NTCIR Workshop 6*, 2006.

[3] V. Jijkoun and M. de Rijke. Overview of the WiQA task at CLEF 2006. In *Proceedings CLEF 2006*, to appear.

[4] V. Jijkoun and M. de Rijke. Retrieving answers from frequently asked questions pages on the web. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 76–83, New York, NY, USA, 2005. ACM Press.

[5] V. Jijkoun and M. de Rijke. WiQA: Evaluating Multi-lingual Focused Access to Wikipedia. In T. Sakai, M. Sanderson, and D. Evans, editors, *Proceedings EVIA 2007*, pages 54–61, 2007.

[6] D. Kelly and J. Lin. Overview of the TREC 2006 ciQA Task. *SIGIR Forum*, 41(1):107–116, June 2007.

[7] J. Lin, D. Quan, V. Sinha, K. Bakshi, D. Huynh, B. Katz, and D. Karger. What makes a good answer? The role of context in question answering. In *Proceedings of INTERACT 2003*, 2003.

[8] B. Magnini, S. Romagnoli, A. Vallin, J. Herrera, A. Penas, V. Peinado, F. Verdejo, and M. de Rijke. The multiple language question answering track at CLEF 2003. In C. Peters, J. Gonzalo, M. Braschler, and M. Kluck, editors, *Comparative Evaluation of Multilingual Information Access Systems, CLEF 2003*, volume 3237 of *Lecture Notes in Computer Science*, pages 471–486. Springer, 2004.

[9] B. Magnini, A. Vallin, C. Ayache, G. Erbach, A. Penas, M. de Rijke, P. Rocha, K. Simov, and R. Sutcliffe. Overview of the CLEF 2004 multilingual question answering track. In C. Peters, P. Clough, G. Jones, J. Gonzalo, M. Kluck, and B. Magnini, editors, *Multilingual Information Access for Text, Speech and Images: Results of the Fifth CLEF Evaluation Campaign*, volume 3491 of *Lecture Notes in Computer Science*, pages 371–391, 2005.

[10] B. Magnini, D. Giampiccolo, P. Forner, C. Ayache, P. Osenova, A. Penas, V. Jijkoun, B. Sacaleanu, P. Rocha, and R. Sutcliffe. Overview of the CLEF 2006 Multilingual Question Answering Track. In *Proceedings CLEF 2006*, to appear.

[11] S. Small, T. Strzalkowski, T. Liu, S. Ryan, R. Salkin, N. Shimizu, P. Kantor, D. Kelly, R. Rittman, and N. Wacholder. HITIQA: towards analytical question answering. In *Proc. of COLING 2004*, 2004.

[12] K. Sparck Jones. Is question answering a rational task? In *Proceedings 2nd CoLogNET-ELSNET Symposium on Questions and Answers: Theoretical and Applied Perspectives*, 20003.

[13] A. Vallin, B. Magnini, D. Giampiccolo, L. Aunimo, C. Ayache, P. Osenova, A. Penas, M. de Rijke, B. Sacaleanu, D. Santos, and R. Sutcliffe. Overview of the CLEF 2005 Multilingual Question Answering Track. In C. Peters, F. Gey, J. Gonzalo, H. Müller, G. Jones, M. Kluck, B. Magnini, and M. D. Rijke, editors, *Accessing Multilingual Information Repositories*, volume 4022 of *Lecture Notes in Computer Science*, pages 307–331. Springer, September 2006.

[14] E. Voorhees. Overview of the TREC-9 Question Answering Track. In *The Ninth Text REtrieval Conference (TREC 9)*, 2001.

[15] WebCLEF. Definition of the clef 2007 webclef task, 2007. `http://ilps.science.uva.nl/WebCLEF/WebCLEF2007`.

[16] E. Whittaker, J. Mrozinski, and S. Furui. Factoid question answering with web, mobile and speech interfaces. In *HLT-NAACL*, pages 288–291, 2006.

On the Relation between Relevant Passages and XML Document Structure

Jaap Kamps[1,2] Marijn Koolen[1]

[1] Archives and Information Studies, University of Amsterdam
[2] ISLA, Informatics Institute, University of Amsterdam

ABSTRACT

Whereas traditional document retrieval methods always return whole atomic documents as results, focused retrieval methods aim to provide more direct access to the relevant information by zooming in on those parts of the document that contain the relevant text. The main aim of this paper is to investigate how relevant text inside a document relates to the document structure. We analyze the INEX 2006 assessments, where topic assessors were asked to mark in yellow all and only relevant text, in relation to the underlying document structure of English Wikipedia pages transformed into XML.

Our main findings are: First, although relevant passages are typically small—with a median length of a few sentences and a mean length of a paragraph—they have varying lengths and may cover any fraction of an article. Second, the document structure corresponds reasonably well to the relevant passages. Although the shortest element containing the relevant passages is twice as long on average, half of the passages are closely fitting an XML element (the passage covers 95-100% of the element). Third, in particular the start of a relevant passage tends to coincide with the start of an XML element.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries

General Terms

Measurement, Experimentation

Keywords

Evaluation, Relevance, Passage Retrieval, XML Retrieval

1. INTRODUCTION

In focused retrieval, the task is to go beyond the document level and zoom in on only those parts of the document that contain relevant text. Focused retrieval dates back, at least, to the early days of passage retrieval [6]. As Salton et al. [6, p.49] put it:

> Large collections of full-text documents are now commonly used in automated information retrieval. When the stored document texts are long, the retrieval of complete documents may not be in the users' best interest. In such circumstances, efficient and effective retrieval results may be obtained by using passage retrieval strategies designed to retrieve text excerpts of varying size in response to statements of user interest.

Early passage retrieval approaches have been using either the document structure (sentences, paragraphs, sections, etc.), or arbitrary text windows of fixed length [1]. In particular, the use of document structure derived from SGML mark-up was pioneered in [9]. The early experimental results primarily confirmed the effectiveness of passage-level evidence for boosting document retrieval. Over the years, research in this area has forked off several approaches like passage retrieval, question answering and XML element retrieval. In question answering, returning short and to-the-point results is a firm requirement [8]. In XML element retrieval, the goal is to retrieve those XML elements that are relevant (i.e., discuss the topic of request exhaustively) but contain no non-relevant information (i.e. they are specific for the topic of request) [2].

To evaluate focused retrieval methods, we also require relevance assessments below the document level. A simple binary decision whether the document is relevant no longer suffices. Assessors have to indicate which parts of the document are relevant, or in the case of question answering whether the given answer is correct, and evaluation measures have to reflect how well a *retrieved* document part fits a *relevant* document part. During the INEX 2006 campaign [5] such sub-document assessments have been collected. The document collection consists of the English Wikipedia pages transformed into XML [4]. Topic assessors are asked to mark in yellow all and only relevant text in a pooled set of documents. The judges only view the rendered text, unaware of the precise underlying XML structure. As a result, the highlighted passages are elicited unobstructed by the XML document structure.

The main aim of this paper is to investigate how relevant text inside a document relates to the document structure. Recall from the above, passages have traditionally been defined using either the document structure (like the XML structure at INEX), or based on various windows of text (like the assessors' highlights). This prompts a number of questions:

- What is the length of relevant passages? What fraction

SIGIR 2007 Workshop on Focused Retrieval
July 27, 2007, Amsterdam, The Netherlands

Table 1: Length of relevant passages in the INEX 2006 adhoc assessments.

	Min	Max	Median	Mean	Stdev
passage length	1	78,943	297	1,090	3,263
article length	96	234,461	4,528	9,485	12,962
article highlights	7	78,943	510	1,753	4,242
article ratio	0.0001	1.0000	0.1339	0.3160	0.3574

of the article is considered relevant?

- How well do the highlighted passages correspond to XML elements of the document structure?
- Since highlighted passages may span a range of elements, how do the passage boundaries correspond to XML element boundaries?

The adhoc task at INEX is to retrieve XML elements containing relevant text at the right level of granularity. The adequacy of the document structure to determine the unit of retrieval has been challenged in [7]. To study the value of the XML document structure to define retrieval results, INEX is allowing also arbitrary passage results in 2007. The analysis of this paper differs from the INEX retrieval tasks: rather than evaluating retrieval results in terms of their relevant or highlighted text, we investigate the highlighted passages as a whole directly.

2. ANALYSIS

We analyze the INEX 2006 adhoc retrieval assessments (v5-filtered) containing judgments for 114 topics (numbered 289-298, 300-366, 368-369, 371-376, 378-388, 390-392, 394-395, 399-407, 409-411, and 413). The assessors have assessed relevance by highlighting relevant text at the granularity of sentences. The assessment interface automatically merges consecutive highlighted passages. A passage's start and end point is identified by either XML element boundaries or character-offsets on the respective text nodes. First, we will look at the length of passages, both in absolute and relative terms. Second, we will investigate how highlighted passages relate to XML elements. Third, we will zoom in on the passages start and end points, and relate them to XML element boundaries.

2.1 Relevant Passage Length

We start by looking at the length of highlighted passages, both absolute and relative length, and want to find out characteristics of the relevant information inside articles. Table 1 shows the length of highlighted passages for the INEX 2006 adhoc topics. Over 114 topics, there are 9,086 passages in 5,648 articles (we restrict our analysis to these articles). Passages contain 1,090 characters on average (median 297), while relevant articles contain almost 10,000 characters on average (median 4,528). Since articles can have multiple relevant passage, the average length of relevant text per article is 1,753 characters, showing that these relevant articles have 1.6 relevant passages on average. Looking at the relative length of the highlights, we see that on average 31.60% of the relevant articles' text is highlighted (median 13.39%). The highlighted passages have a median length of a couple of sentences, and an average length of a paragraph.

We now look at the impact of the topic at hand on the length of the highlighted passage. Figure 1 shows the distribution of passage length over topics. Although most of the passages are very short, some topics contain quite a few passages that are over 10,000 characters in length. There is certainly no "fixed" passage length per topic. Moreover, there is variation in length of highlighted passages over topics, although also plotting the relevant article's length over topics (not shown) results in similar pattern.

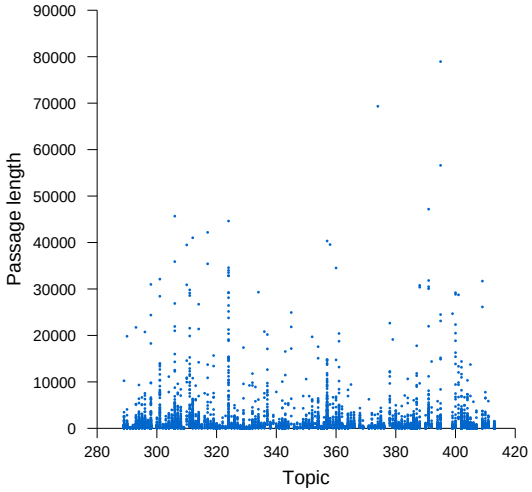


Figure 1: Length of highlighted passages over topics.

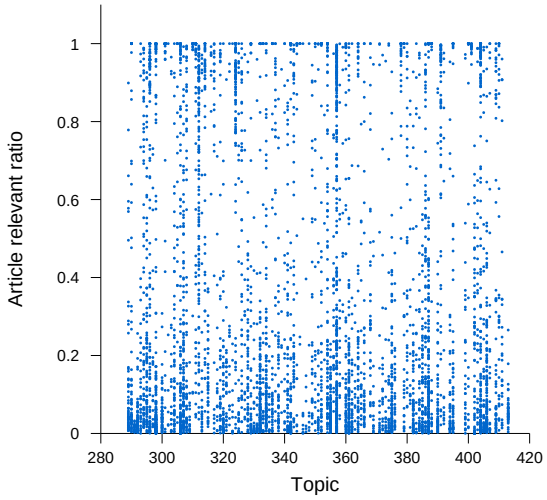


Figure 2: Fraction of the article that is highlighted over topics.

Since articles have substantial variation in length, we look at the relative length of the highlighted text. Figure 2 shows the fraction of articles that is highlighted over topics. What is most striking is the spread over the whole range. For many of the articles across most topics, only a small fraction (less than 20%) of the text is highlighted. Also, for many topics, there are a few articles that are wholly relevant. The density of the plot seems somewhat greater on the extremes.

Does the fraction of highlighted text depend on the length of the article? Figure 3 shows the fraction of articles that is highlighted over the length of the articles. Many of the Wikipedia articles are rather short, including many of the relevant articles. Most of the relevant articles are much shorter than 50,000 characters, and for most of the articles the relevance ratio is below 0.2, corresponding to Fig-

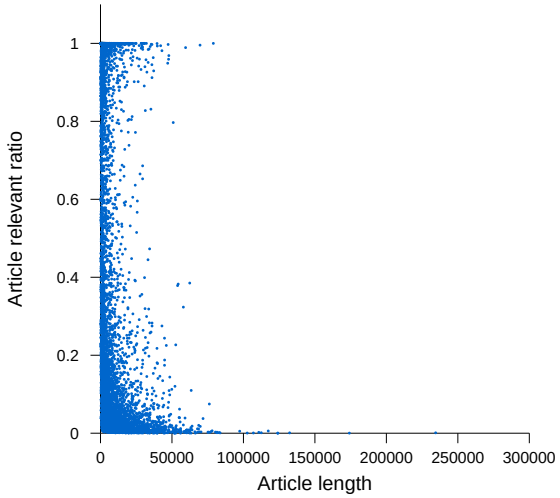


Figure 3: Article length versus highlighted fraction.

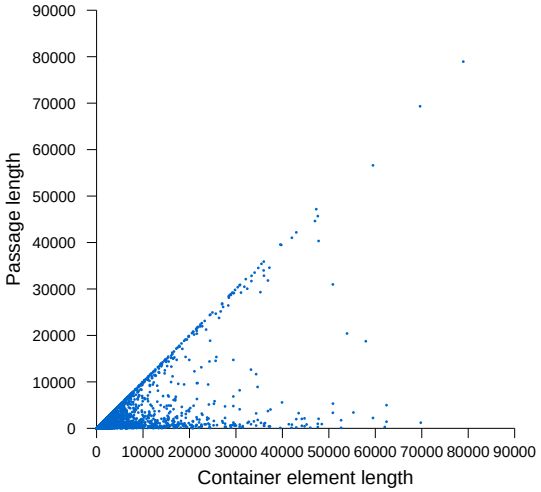


Figure 4: Passage length versus component element length.

Table 2: Length of passages and container elements.

	Min	Max	Median	Mean	Stdev
passage length	1	78,943	297	1,090	3,263
container length	1	78,943	620	2,348	5,525
container ratio	0.0009	1.0000	0.9730	0.7028	0.3637

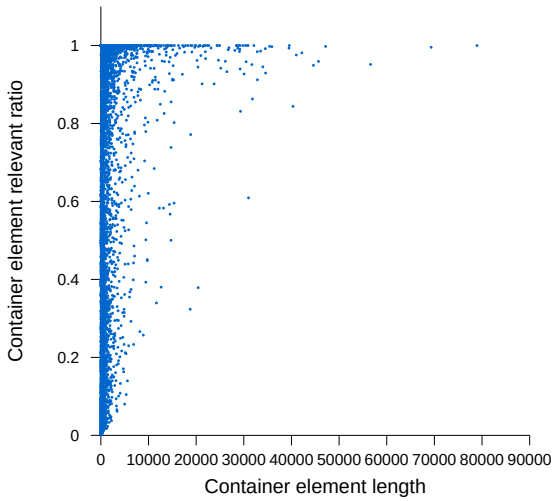


Figure 5: Fraction of container element that is highlighted.

ure 2. Above a relevance ratio of 0.2, the articles are spread more or less evenly over the relevance ratio scale, indicating that the relevant portion of an article varies greatly. This is rather surprising, as we would expect that longer articles have a smaller percentage of relevant text. Recall from the introduction that sub-document retrieval is motivated by the assumption that long documents only contain a relatively small fraction of relevant text.

Summarizing, our analysis showed that i) relevant passages are relatively short with a median length of a couple of sentences, and an average length of a paragraph; ii) there is no "fixed" length of relevant passages; iii) the highlighted text may cover any fraction of the article; and iv) the fraction of the article that is highlighted does not depend on the length of the article.

2.2 Relating Passages to Elements

We now relate the relevant passages to the document structure, and want to find out how well the highlighted passages correspond to XML elements of the document structure. From the article level, we now zoom in on the XML elements that contain relevant text. We use the notion of *container elements* to identify those elements that contain the whole relevant passage. More specifically, we will focus on the *shortest container elements*, i.e. the shortest element to contain the *whole* passage.

How long are the XML elements containing the passages? Table 2 gives some statistics on the length of passages and their container elements. We include the passage lengths again for comparison. The container elements have a mean length of 2,348 characters, and a median length of 620 characters. That is, the average container element is twice the length of the average passage. The minimum and maximum lengths are equal, meaning that both the shortest passage and the longest passage exactly fit their container element, i.e. the container contains only relevant text. This suggests that if we approximate the relevant passage by an XML element from the document structure, we retrieve in total twice the length of relevant text. The ratio of the container elements that is covered by the relevant passage, also shown in Table 2, is on average 70% but the median ratio is 97%. This suggests a reasonable fit between passages and their container elements.

In the previous section we saw that relevant passages vary widely in length. How does the length of the passages relate to the length of the container element? Figure 4 plots the passage length against the container element length. The diagonal axis shows the passages that exactly fit their container elements, and especially for longer passages the container element fits like a glove. The part below this diagonal axis is empty, as passages can never be longer than their container elements. The bulk of the passages is shorter than 10,000 characters, and here their containers are often substantially longer than the relevant passages. Looking at the same data from another angle, Figure 5 plots the ratio of container elements that is highlighted. This shows the same pattern: the longer containers tend to have higher relevance

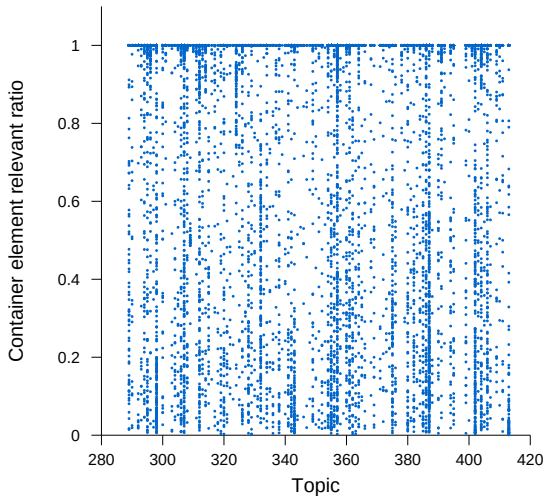


Figure 6: Fraction of container element that is highlighted over topics.

Table 3: Distribution of container elements over relevance ratio.

Ratio	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Frequency	419	755	656	467	432	375	315	247	288	424	4,705

Table 4: Container tag frequency and mean relevance ratio.

Tag	Frequency	Mean length	Mean ratio
⟨p⟩	2,761	558.7	0.7045
⟨body⟩	1,693	6,184.8	0.4213
⟨section⟩	1,424	2,453.6	0.6746
⟨item⟩	944	138.2	0.9248
⟨article⟩	724	7,009.6	0.8526
⟨normallist⟩	304	1,004.8	0.4667
⟨name⟩	270	21.4	1.0000
⟨collectionlink⟩	209	19.4	1.0000
⟨row⟩	180	62.0	0.7122
⟨caption⟩	174	93.7	0.9849

ratios. This is in itself no big surprise, since a long relevant passage spanning a range of elements is required for these long container elements.

Some of the topics provide hints of the type of XML element that is likely to be relevant. Does the topic at hand impact the relative fit of the container element? Figure 6 shows the relevance ratio of the container elements split over topics. For many topics, the number of container elements with smaller ratios is small, but there is great variation in relevance ratios over containers. The dark line at the top indicates that quite a number of relevant passage boundaries coincide with the container element boundaries. From the plots it is still not clear whether the number of containers with a relevance ratio of 1 is higher than the number of containers at lower relevance ratios. Table 3 shows the distribution of container elements over the different relevance ratios. In total, 4,705 relevant passages closely fit their container element, that is, half of the relevant passages (51.8%) cover 95–100% of the text of their container elements.

Finally, we investigate the correspondence between specific container element types and highlighted passages. Table 4 shows the tag names of the container elements, their frequencies, mean length, and the mean of their relevance ratios. The ⟨p⟩ element is the most frequent container of relevant passages and on average, 70% of these containers is relevant text. The ⟨body⟩ element is also very frequent but has a much lower relevance ratio (42%). The ⟨article⟩ element, somewhat surprisingly, has a much higher relevance ratio (85%), while it is only slightly longer than the ⟨body⟩ element. The ⟨article⟩ contains the ⟨body⟩ element and the elements ⟨name⟩ (the name of the Wikipedia article) and ⟨conversionwarning⟩. A plausible explanation is that if a large part of the article is relevant, the ⟨name⟩ of the page will be included in the passage highlighted by the assessor, resulting in ⟨article⟩ being the container element. If the ⟨name⟩ element is not highlighted, but different sections somewhere down the article are highlighted, the container element will be the ⟨body⟩. Other document structures that correspond well to highlighted passages are ⟨section⟩, ⟨item⟩, ⟨name⟩ and ⟨collectionlink⟩ elements.

Summarizing, our analysis above revealed mixed results for the correspondence between relevant passages and container elements (i.e., the shortest XML element containing the whole passage). On the one hand, the average container element is twice as long as the average passage. On the other hand, half of the passages have a closely fitting container element (the passage covers 95-100% of the element).

Table 5: Offsets of relevant passages.

	Min	Max	Median	Mean	Stdev
start element	0	10,723	0	62.74	317.68
end element	0	61,743	2	365.80	2,423.29
start container	0	47,510	1	252.90	1,344.91
end container	0	68,566	24	1,023.48	3,928.68

2.3 Passage and Element Boundaries

We now zoom even further in, and look at the relation between passage boundaries and element boundaries. We define two more notions, *start element* and *end element* as:

- *start element*: the XML element that directly contains the first highlighted character of the passage.
- *end element*: the XML element that directly contains the last highlighted character of the passage.

If the highlighted passage crosses no element boundaries (e.g., a passage from a single paragraph), the start and element elements coincide and are also the container element.

We look at where the highlighted passages start and end (character offset) in the document structure and within their container elements. Table 5 shows the offsets of highlighted passages for the INEX 2006 adhoc topics. First, we look at the closest XML element boundaries and see that the median offset in the start element is 0. Thus, at least half of the highlighted passages start at an XML element boundary. The much higher mean offset shows that the distribution is skewed. Nonetheless, the bulk of the passages start very close to the start element boundary. Second, the offset to the end of the end element is 2, showing that most the passages end at the boundary of the end element. The average is much higher, showing again a skewed distribution. Third, we look at the shortest XML element containing the whole passage and see that the median offset in the container element is 1, indicating that many of the container elements

are also start elements. Fourth, the median offset to the end of the container elements is 24, showing that most of the passages end some distance before the end the container element.

Summarizing, the correspondence between the relevant passages and document structure is particularly strong at the passages' start points: relevant passages start at an element boundary.

3. CONCLUSIONS

In focused retrieval the aim is to retrieve only those parts of a document that contain relevant text and no non-relevant text. In XML retrieval the XML structure of documents is exploited to locate relevant elements and use their boundaries as passage boundaries. In this paper we have investigated how well these XML element boundaries correspond to the boundaries of relevant passages in the INEX 2006 adhoc assessments.

Our first question was:

- What is the length of relevant passages? What fraction of the article is considered relevant?

The data show that most relevant passages are rather short, less than 1,000 characters, but there is a great variety over topics, and there seems to be no 'fixed' passage length and there is no relation between passage length and article length, and therefore no clear answer on what fraction of an article is considered relevant.

The second question was:

- How well do the highlighted passages correspond to the XML elements of the document structure?

The average length of the shortest element containing the highlighted passage is twice as long as the average passage length, but half of these container elements are a close fit to the passage (95-100% of their content being relevant text). Document structures that correspond naturally to highlighted passages are paragraphs, sections, list-items, titles and the whole article itself. However, even though these structures correspond reasonably well to highlighted passages, there is large variation over passages, articles and topics.

Our last questions was:

- Since highlighted passages may span a range of elements, how do the passage boundaries correspond to XML element boundaries?

The start of the passage often corresponds with the first character of the "start" element and the container element. The end of the passage corresponds well to the last character of the "end" element, and is at some distance from the end of the container element.

There are, as always, various limitations to the analysis provided. First, there is an obvious impact of the particular document structure of the collection. Wikipedia is an encyclopedia, with a highly organized structure, and created by a multitude of writers and editors. The generated XML encoding is based on the simple Wiki-syntax, and of course depends the particular writing style—how well is the particular article textually structured? and how well does this correspond to the sectioning structure? Second there is an obvious impact of relevance assessor and the assessment interface. Does a judge highlight the best text in the article's context, or judge relevance on equal grounds throughout the whole collection?

What do we learn from the analysis in terms of the retrieval approaches? First, the short length of the typical relevant passage seems to suggest retrieving fixed window passages, but the variation in length of passages and coverage of the article seems to suggest a flexible unit of retrieval such as XML elements. Second, the fact that half of the passages fit closely with an XML element seems to support retrieving XML elements, but the fact that the corresponding elements are twice the length of the relevant passage seems to support passages results. Third, the start of a relevant passage tends to coincide with the start of an XML element, so if we assume results are displayed in the context of the article, retrieval of XML elements seems a good approach. Although also fixed window passage retrieval proved an effective approach to find hot-spots inside articles [3]. In short, there is mixed support for both retrieving elements of the document structure and for retrieving arbitrary passages. We look forward to the retrieval experiments at INEX 2007 to help determine what approaches turn out to be more effective in practice.

Acknowledgments

Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO, grants # 612.066.513, 639.072.601, and 640.001.501), and by the E.U.'s 6th FP for RTD (project MultiMATCH contract IST-033104). Marijn Koolen was supported by NWO (# 640.001.501).

REFERENCES

[1] J. P. Callan. Passage-level evidence in document retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 302–310. Springer-Verlag, New York NY, 1994.

[2] C. Clarke, J. Kamps, and M. Lalmas. INEX 2006 retrieval task and result submission specification. In N. Fuhr, M. Lalmas, and A. Trotman, editors, *INEX 2006 Workshop Pre-Proceedings*, pages 381–388, 2006.

[3] C. L. A. Clarke and E. L. Terra. Passage retrieval vs. document retrieval for factoid question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 427–428. ACM Press, New York NY, 2003.

[4] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 40(1):64–69, June 2006.

[5] INEX. INitiative for the Evaluation of XML Retrieval, 2006. `http://inex.is.informatik.uni-duisburg.de/2006/`.

[6] G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58. ACM Press, New York NY, 1993.

[7] A. Trotman and S. Geva. Passage retrieval and other XML-retrieval tasks. In A. Trotman and S. Geva, editors, *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, pages 43–50, 2006.

[8] E. M. Voorhees. Overview of the TREC 2001 question answering track. In *The Tenth Text REtrieval Conference (TREC 2001)*, pages 42–51. National Institute for Standards and Technology. NIST Special Publication 500-250, 2002.

[9] R. Wilkinson. Effective retrieval of structured documents. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 311–317. Springer-Verlag, New York NY, 1994.

Evaluating Focused Retrieval Tasks

Jovan Pehcevski
AxIS project team
INRIA-Rocquencourt, Le Chesnay, France
jovan.pehcevski@inria.fr

James A. Thom*
School of Computer Science and IT
RMIT University, Melbourne, Australia
james.thom@rmit.edu.au

ABSTRACT

Focused retrieval, identified by question answering, passage retrieval, and XML element retrieval, is becoming increasingly important within the broad task of information retrieval. In this paper, we present a taxonomy of text retrieval tasks based on the structure of the answers required by a task. Of particular importance are the *in context* tasks of focused retrieval, where not only relevant documents should be retrieved but also relevant information within each document should be correctly identified. Answers containing relevant information could be, for example, best entry points, or non-overlapping passages or elements. Our main research question is: How should the effectiveness of focused retrieval be evaluated? We propose an evaluation framework where different aspects of the *in context* focused retrieval tasks can be consistently evaluated and compared, and use fidelity tests on simulated runs to show what is measured. Results from our fidelity experiments demonstrate the usefulness of the proposed evaluation framework, and show its ability to measure different aspects and model different evaluation assumptions of focused retrieval.

Categories and Subject Descriptors: H.3 [Information Storage and Retrieval]: H.3.3 Information Search and Retrieval; H.3.7 Digital Libraries

General Terms: Measurement, Performance, Experimentation

Keywords: Evaluation, In Context, Test collection, XML Retrieval

1. INTRODUCTION

Traditional information retrieval (IR) typically returns whole documents as answers, and leaves it up to users to locate the relevant information within each retrieved document. Focused retrieval [22], including question answering [23], passage retrieval [1, 2, 6, 24], and XML element retrieval [16], investigates ways to provide users with direct access to relevant information in retrieved documents. Evaluating focused retrieval is a challenging task since different retrieval techniques typically produce answers of various sizes and granularity, which calls for a common evaluation framework where different aspects of focused retrieval can be consistently measured and compared.

The INitiative for the Evaluation of XML retrieval (INEX) has studied different aspects of focused retrieval since 2002, by considering XML element retrieval techniques that can effectively retrieve information from structured document collections [16]. Since 2005, a highlighting assessment procedure is used at INEX to gather relevance assessments for the INEX retrieval topics [15]. In this procedure, assessors from the participating groups are asked to highlight sentences representing the relevant information in a pooled set of documents. An assessment program then computes the relevance of the judged elements (including whole documents) as the ratio of highlighted to fully contained text, where the element relevance values are drawn from a continuous scale in the range 0 to 1.

INEX 2006 introduced two new retrieval tasks, *relevant in context* and *best in context*, that combine document retrieval with XML element retrieval [4]. The *relevant in context* task is document retrieval with a twist, where not only the relevant documents should be retrieved, but also a set of non-overlapping XML elements representing the relevant information within each document should be correctly identified. The *best in context* task is similar, except that here systems are asked to return only one element per document, which corresponds to the best entry point for starting to read the relevant information in the document.

These two *in context* tasks correspond to end-user tasks where focused retrieval answers are grouped per document, in their original document order, providing access through further navigational means. This assumes that users consider documents as the most natural units of retrieval, and prefer an overview of relevance in their context. Moreover, the *in context* tasks loosely correspond to the assessment procedure used at INEX 2006, with the difference that the INEX assessors highlighted sentences whereas the systems only returned XML elements.

Interactive experiments at INEX [21], along with user studies carried out within and outside INEX [3, 9, 13], have also confirmed the usefulness of grouping the retrieved elements by their contained documents. The need for element grouping is mainly motivated by the fact that users not only want to locate more focussed information within a document, but they also want to "see what the document is" [3]. These findings justify the inclusion of the *in context* retrieval tasks at INEX, and highlight their importance in focused retrieval. In Section 2, we present a taxonomy for text retrieval tasks based on the structure of the answers required by a task, and discuss how it covers the *in context* tasks of focused retrieval.

How to evaluate the *in context* tasks of focused retrieval? There are two main requirements [10]: i) the score should reflect the ranked list of documents inherent in the result list, and ii) the score should also reflect how well the retrieved information per document corresponds to the relevant information. In Section 3, we propose an evaluation framework where different aspects of the *in context* focused retrieval tasks can be consistently evaluated and compared. To measure the extent to which text retrieval systems return relevant information, we design evaluation measures that consider the amount of highlighted text in relevant documents [17, 18]. Our proposal is motivated by the need to use measures that are simple

*This work was undertaken while James Thom was visiting the AxIS team at INRIA in 2007.

SIGIR 2007 Workshop on Focused Retrieval
July 27, 2007, Amsterdam, The Netherlands

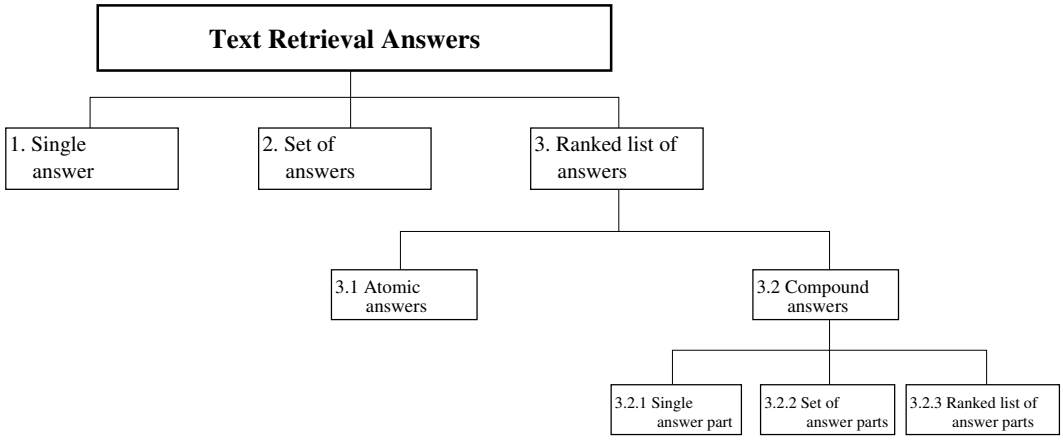


Figure 1: Taxonomy of text retrieval answers

and easy to interpret [7] and that are natural extensions of the well-established measures used in traditional information retrieval [20].

Since a variety of evaluation measures can be used to evaluate retrieval effectiveness, it is essential to carry out tests to determine whether they measure what are they intended to measure, and whether the reported evaluation scores can be trusted. Accordingly, two important tests are used to qualify the *evaluation* of evaluation measures: *fidelity* and *reliability* [23]. Simulated runs constructed in a controlled way are typically used to determine the *fidelity* of an evaluation measure [5, 11, 19]. In XML retrieval, these runs contain various granularity of elements in their answer lists (such as ideal elements, full document elements, or leaf elements). A measure successfully passes the fidelity test if the obtained evaluation scores demonstrate that the best retrieval performance is indeed achieved when using the right (and desired) answer granularity, while preserving a reasonable relative ordering of the other simulated runs. The results from our fidelity tests shown in Section 4 demonstrate the usefulness of the proposed evaluation framework, and its ability to measure different aspects and model different evaluation assumptions of focused retrieval.

We conclude this paper with our discussions in Section 5, where we use our findings to reflect on the comparison between passage and element retrieval, the usefulness of focused and traditional document retrieval in identifying relevant information, and the importance of choosing appropriate evaluation assumptions.

2. A TAXONOMY OF RETRIEVAL TASKS

In this section, we present a taxonomy of text retrieval tasks based on the structure of the answers required by a task. We only consider tasks where non-overlapping answers are allowed. We also discuss some assumptions about what users want; these assumptions, together with the answer structure, define a retrieval task and influence how it should be evaluated.

Answers

In text retrieval answers can include either or both documents (or equivalently document identifiers) and excerpts of documents. The excerpts could be passages (identified by start and end positions) or in the case of XML retrieval, elements (identified by XPath expressions). Furthermore, depending on the retrieval task, answers may be a single result, an unordered set of results, or a ranked list of results. This leads us to a partial taxonomy of tasks based on answers as shown in Figure 1. For each type of answer in the taxonomy (such as an atomic answer or a compound answer), we describe one or more text retrieval tasks that can be used to generate that particular answer. The taxonomy parts are explained as follows.

1. Single answer

   For tasks where the user is only interested in one document (or excerpt of a document) as an answer, such as in Google's "I'm Feeling Lucky™".

2. Set of answers

   For Boolean retrieval tasks where the user is interested in finding all matching documents (or excerpts).

3. Ranked list of answers

   3.1 Atomic answers

   For tasks where the answers are a ranked list of documents, such as a list of web pages found by a search engine, or a ranked list of elements as retrieved for the INEX *thorough* or *focused* tasks [4], or a ranked list of passages for the TREC *question answering* task [23].

   3.2 Compound answers

   For *in context* tasks where the result of a query is a ranked list of answers (usually documents) and clustered for each answer in the list, further information (answers parts) needs to retrieved from the document. These could be:

   3.2.1 Single answer part, such as the best entry point returned in the INEX *best in context* task [4] or text snippets returned as document summaries by search engines.

   3.2.2 Set of answer parts, such as the elements returned in the INEX *relevant in context* task [4] (in 2007 INEX will allow passages as well as elements).

   3.2.3 Ranked list of answer parts. It is conceivable that the answer parts could be returned as a sub-list of ranked elements, which could be represented by using a document heat-map.

This paper is concerned with evaluation of the last group of tasks, which are considered in more detail in the taxonomy. These are the *in context* tasks that are based on compound answers. Specifically, we consider the *relevant in context* task where the result of a query is a ranked list of answers documents, and, for each document in the answer list, a set of passages or elements is returned.

Assumptions

In defining a text retrieval task it is also necessary to define the assumptions about what the user is wanting to see. We make the following basic assumption about all text retrieval tasks:

Users want to see as much relevant information as possible with as little irrelevant information as possible. Such an assumption is the basis of methods for evaluating the effectiveness of information retrieval systems based on recall and precision.

This basic assumption is not sufficient to determine how best to evaluate most text retrieval tasks. For this, we need to make further assumptions about what users actually prefer, which for example we may choose to test via user experiments. These assumptions may depend on the type of retrieval task, as illustrated by the following examples.

1. *Users do not want to see the same (or similar) answers more than once.* This motivates the work behind evaluating aspectual retrieval [14], and influences the way commercial search engines present answers.

2. *Users want the shortest and the most complete answer.* This might be motivated by a question answering task where an answer needs to be seen in isolation, and it need not be required to provide any context.

3. *Users consider longer more detailed answers to be more useful than shorter answers.* This models users that prefer documents containing longer passages with more relevant information over documents containing shorter passages.

4. *Users consider all answers to be equally useful.* This models users that place equal value on each relevant document, and here documents with longer relevant passages are considered as equally useful as those with shorter passages.

The last two assumptions are likely to depend on the task. We explore these assumptions in more detail for the *relevant in context* task later in this paper.

3. EVALUATION FRAMEWORK

In this section, we describe an evaluation framework for the *in context* tasks of focused retrieval. The framework focuses on compound answers given in the taxonomy shown in Figure 1. The evaluation of the *in context* tasks calculates scores for ranked lists of documents, where per document we obtain a score reflecting how well the retrieved information corresponds to the relevant information in the document.

Score per document

Three different scores per document can be calculated, depending on whether a single answer part, a set of answer parts, or a ranked list of answer parts are retrieved from the document. We focus on the case where a set of non-overlapping answer parts is retrieved.

For a retrieved document, the text identified by the selected set of retrieved parts is compared to the text highlighted by the assessor [17, 18]. More formally, let d be the retrieved document, and let p be a part (element or passage) that belongs to $\mathcal{P}_d$, the set of retrieved parts from document d. Let $rsize(p)$ be the amount of highlighted relevant text contained by p (if there is no highlighted text, $rsize(p) = 0$). Let $size(p)$ be the total amount of text contained by p, and let $Trel(d)$ be the total amount of highlighted relevant text for the document d.

We calculate the following:

- Precision, as the fraction of retrieved text (in characters) that is highlighted:

$$P(d) = \frac{\sum_{p \in \mathcal{P}_d} rsize(p)}{\sum_{p \in \mathcal{P}_d} size(p)} \quad (1)$$

  The $P(d)$ measure ensures that, to achieve a high precision value for the document d, the set of retrieved parts for that document needs to contain as little non-relevant information as possible.

- Recall, as the fraction of highlighted text (in characters) that is retrieved:

$$R(d) = \frac{\sum_{p \in \mathcal{P}_d} rsize(p)}{Trel(d)} \quad (2)$$

  The $R(d)$ measure ensures that, to achieve a high recall value for the document d, the set of retrieved parts for that document needs to contain as much relevant information as possible.

- F-Score, as the combination of precision and recall using their harmonic mean, resulting in a score in [0,1] per document:

$$F(d) = \frac{2 \cdot P(d) \cdot R(d)}{P(d) + R(d)} \quad (3)$$

For retrieved non-relevant documents, all the above scores evaluate to zero: $P(d) = R(d) = F(d) = 0$.

We use the F-score as an appropriate document score for the case where a set of answer parts is retrieved: $S(d) = F(d)$. The resulting $S(d)$ score varies between 0 (document without relevance, or none of the relevance is retrieved) and 1 (all relevant text is retrieved without retrieving any non-relevant text).

Scores for ranked list of documents

We have a ranked list of documents $\mathcal{D}$, and for each document we have a document score $S(d_r) \in [0, 1]$, where d_r is the document retrieved at rank r ($1 \leq r \leq |\mathcal{D}|$). Hence, we need generalized evaluation measures, and we utilise the most straightforward generalization of precision and recall [12]. More formally, let us assume that for a retrieval topic there are in total $Nrel$ documents with relevance, and let us also assume that the function $rel(d_r) = 1$ if document d_r contains relevant information, and $rel(d_r) = 0$ otherwise. Let $rsize(d_r)$ be the amount of highlighted relevant text contained by d_r (if there is no highlighted text, $rsize(d_r) = 0$), and let $Trel$ be the total amount of highlighted relevant text for the retrieval topic (calculated across the $Nrel$ relevant documents).

Over the ranked list of documents, we calculate the following:

- generalized Precision ($gP[r]$), as the sum of document scores up to a document-rank r, divided by the rank r:

$$gP[r] = \frac{\sum_{j=1}^{r} S(d_j)}{r} \quad (4)$$

- generalized Recall ($gR[r]$), as the number of documents with relevance retrieved up to a document-rank r, divided by the total number of documents with relevance:

$$gR[r] = \frac{\sum_{j=1}^{r} rel(d_j)}{Nrel} \quad (5)$$

The generalized Recall definition, as given in Equation 5, follows the assumption that each document with relevance is treated as equally relevant, and thus as equally useful to retrieve (Assumption 4). However, since the documents in the answer list are ranked in a descending order of their estimated likelihood of relevance, an alternative (and equally plausible) assumption would be that documents with more highlighted relevant text should be considered to be more relevant (and therefore more useful to retrieve) than documents with less highlighted text (Assumption 3). To model this evaluation assumption, we use the alternative generalized Recall definition shown in Equation 6:

$$gR'[r] = \frac{\sum_{j=1}^{r} rsize(d_j)}{Trel} \quad (6)$$

These generalized measures are compatible with the standard precision/recall measures used in traditional information retrieval. Specifically, the Average generalized Precision for a retrieval topic can be calculated by averaging the generalized Precisions at natural recall points where generalized Recall increases (the generalized Precision of non-retrieved relevant documents is 0).

A consequence of introducing two generalized Recall definitions ($gR[r]$ and $gR'[r]$) is that two Average generalized Precision definitions need to be respectively used in calculating the overall performance score: AgP, which uses $gR[r]$ and is shown in Equation 7; and AgP', which uses $gR'[r]$ and is shown in Equation 8.

$$AgP = \sum_{j=1}^{|\mathcal{D}|} \frac{1}{Nrel} \cdot rel(d_j) \cdot gP[j] \quad (7)$$

$$AgP' = \sum_{j=1}^{|\mathcal{D}|} \frac{rsize(d_j)}{Trel} \cdot rel(d_j) \cdot gP[j] \quad (8)$$

When looking at a set of topics, the Mean Average generalized Precision ($MAgP$ or $MAgP'$) is simply the mean of the Average generalized Precision scores per topic.

Traditional IR measures

The traditional IR measures treat each retrieved document as either relevant or not, and therefore assign a binary score per document: $S(d_r) = rel(d_r)$. Over the ranked list of documents, we use the following traditional IR measures:

- Precision ($P[r]$), as the fraction of retrieved relevant documents up to a document-rank r:

$$P[r] = \frac{\sum_{j=1}^{r} rel(d_j)}{r} \quad (9)$$

- Recall ($R[r]$), as the fraction of relevant documents retrieved up to a document-rank r (which is the same as the generalized Recall definition given in Equation 5), and

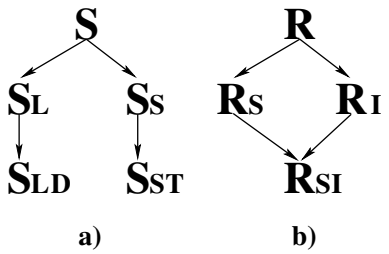


Figure 2: Expected orderings for runs of a dimension of the S–R space: (a) different sets of parts (b) different document rankings

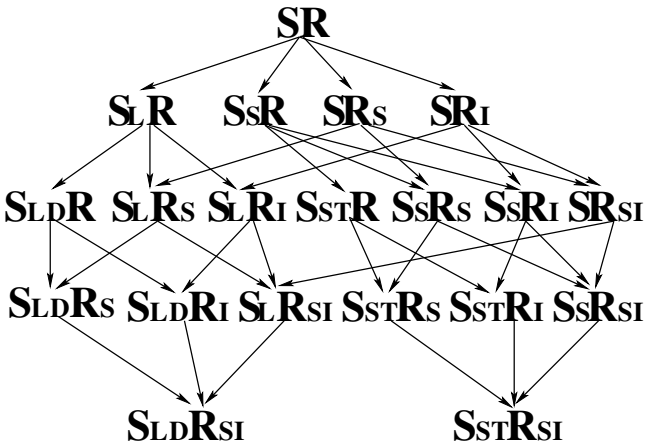


Figure 3: Expected orderings for runs of the S–R space

- Average Precision (AP), as the average of the precisions calculated at natural recall points:

$$AP = \sum_{j=1}^{|\mathcal{D}|} \frac{1}{Nrel} \cdot rel(d_j) \cdot P[j] \quad (10)$$

For a set of topics, the Mean Average Precision (MAP) is simply the mean of the Average Precision scores per topic.

4. FIDELITY TESTS

Fidelity tests should be designed to assess whether evaluation measures indeed measure what they are supposed to measure. In testing the fidelity of evaluation measures for *in context* retrieval, where there are sets of passages/elements returned for each document in the ranked list, there are two dimensions that we need to consider within the overall space of possible runs:

- runs with different amounts of relevant and non-relevant information in the set of passages/elements returned for each document, and
- runs with different rankings of the documents.

For a given evaluation measure these two dimensions may interact in unexpected ways.

Simulated runs and expected orderings

We designed the following suite of simulated runs that took the two dimensions into account.

Run	$gP[r]$			$gR[r]$			$MAgP$	$gR'[r]$			$MAgP'$	MAP
	1	2	10	1	2	10		1	2	10		
$\mathbf{SR}$	1.0000	1.0000	0.9763	0.0419	0.0838	0.3722	1.0000	0.2403	0.3661	0.7194	1.0000	1.0000
$\mathbf{SR_S}$	1.0000	1.0000	0.9763	0.0419	0.0838	0.3722	1.0000	0.2403	0.3661	0.7194	1.0000	1.0000
$\mathbf{SR_I}$	0.0000	0.5000	0.8833	0.0000	0.0419	0.3420	0.8954	0.0000	0.2403	0.6952	0.7647	0.8954
$\mathbf{SR_{SI}}$	0.0000	0.5000	0.8833	0.0000	0.0419	0.3420	0.8954	0.0000	0.1258	0.6952	0.7838	0.8954
$\mathbf{S_LR}$	0.8584	0.8506	0.7830	0.0419	0.0838	0.3722	0.7976	0.2403	0.3661	0.7194	0.8314	1.0000
$\mathbf{S_LR_S}$	0.8427	0.8506	0.7830	0.0419	0.0838	0.3722	0.7969	0.1258	0.3661	0.7194	0.8262	1.0000
$\mathbf{S_LR_I}$	0.0000	0.4292	0.7136	0.0000	0.0419	0.3420	0.7113	0.0000	0.2403	0.6952	0.6289	0.8954
$\mathbf{S_LR_{SI}}$	0.0000	0.4213	0.7136	0.0000	0.0419	0.3420	0.7110	0.0000	0.1258	0.6952	0.6428	0.8954
$\mathbf{S_{LD}R}$	0.7935	0.7280	0.5664	0.0419	0.0838	0.3722	0.5352	0.2403	0.3661	0.7194	0.6719	1.0000
$\mathbf{S_{LD}R_S}$	0.6624	0.7280	0.5664	0.0419	0.0838	0.3722	0.5278	0.1258	0.3661	0.7194	0.6422	1.0000
$\mathbf{S_{LD}R_I}$	0.0000	0.3968	0.5241	0.0000	0.0419	0.3420	0.4700	0.0000	0.2403	0.6952	0.4931	0.8954
$\mathbf{S_{LD}R_{SI}}$	0.0000	0.3312	0.5241	0.0000	0.0419	0.3420	0.4664	0.0000	0.1258	0.6952	0.4926	0.8954
$\mathbf{S_SR}$	0.9578	0.9489	0.8693	0.0419	0.0838	0.3722	0.8687	0.2403	0.3661	0.7194	0.9194	1.0000
$\mathbf{S_SR_S}$	0.9400	0.9489	0.8693	0.0419	0.0838	0.3722	0.8680	0.1258	0.3661	0.7194	0.9140	1.0000
$\mathbf{S_SR_I}$	0.0000	0.4789	0.7905	0.0000	0.0419	0.3420	0.7742	0.0000	0.2403	0.6952	0.6966	0.8954
$\mathbf{S_SR_{SI}}$	0.0000	0.4700	0.7905	0.0000	0.0419	0.3420	0.7739	0.0000	0.1258	0.6952	0.7117	0.8954
$\mathbf{S_{ST}R}$	0.4942	0.4715	0.4518	0.0419	0.0838	0.3722	0.4589	0.2403	0.3661	0.7194	0.4722	1.0000
$\mathbf{S_{ST}R_S}$	0.4488	0.4715	0.4518	0.0419	0.0838	0.3722	0.4578	0.1258	0.3661	0.7194	0.4660	1.0000
$\mathbf{S_{ST}R_I}$	0.0000	0.2469	0.4118	0.0000	0.0419	0.3420	0.4093	0.0000	0.2403	0.6952	0.3578	0.8954
$\mathbf{S_{ST}R_{SI}}$	0.0000	0.2243	0.4118	0.0000	0.0419	0.3420	0.4088	0.0000	0.1258	0.6952	0.3642	0.8954

Table 1: Performance scores for simulated runs of the S–R space, obtained with different measures using the 114 INEX 2006 topics. The runs are grouped in five clusters, depending on the answer parts retrieved ($\mathbf{S}$, $\mathbf{S_L}$, $\mathbf{S_{LD}}$, $\mathbf{S_S}$, $\mathbf{S_{ST}}$).

The first dimension for the simulated runs covers the set of elements/passages returned for each document. We considered five different sets:

$\mathbf{S}$ the set of non-overlapping passages that are highlighted as relevant by the assessor;

$\mathbf{S_L}$ for each passage in $\mathbf{S}$ return the smallest element containing the passage, that is an element which is larger than (or equal in size to) the passage;

$\mathbf{S_{LD}}$ return the whole document;

$\mathbf{S_S}$ for each passage in $\mathbf{S}$ return the largest non-overlapping elements fully contained within the passage, that is one or more elements which are smaller than (or one element equal in size to) the passage; and

$\mathbf{S_{ST}}$ for each passage in $\mathbf{S}$ return the smallest elements fully contained within the passage that do not contain any sub-elements.

The expected ordering of these runs is shown in Figure 2(a).

The second dimension for the simulated runs covers different document rankings. We considered four different rankings:

$\mathbf{R}$ in order of decreasing relevant information from the document containing the most relevant information (that is the most text highlighted as relevant by an assessor) to the document containing the least;

$\mathbf{R_S}$ same as ranking $\mathbf{R}$ but with the first two documents swapped;

$\mathbf{R_I}$ same as ranking $\mathbf{R}$ but with a document containing no relevant information inserted at the start of the list; and

$\mathbf{R_{SI}}$ same as ranking $\mathbf{R_S}$ but after swapping the first two documents, a document containing no relevant information is inserted at the start of the list.

The expected ordering of these runs is shown in Figure 2(b). This ordering is based on the evaluation measure addressing the assumption that users want longer more detailed answers in preference to shorter answers.

As we are interested in how these two dimensions interact, we combine the runs in an $\mathbf{S}$–$\mathbf{R}$ space as shown in Figure 3, which gives the expected ordering of the various combinations of the two dimensions.

For example, the run $\mathbf{SR}$ corresponds to returning as answers the documents in the order from the document with the most text highlighted as relevant to the document with the least text highlighted as relevant ($\mathbf{R}$), and for each document only returning as answer parts those passages corresponding to all the highlighted text ($\mathbf{S}$). This run $\mathbf{SR}$ should be perfect retrieval (under most assumptions), and no other run should perform better than $\mathbf{SR}$ for any topic (even though for some assumptions they may perform as well as this run).

As other examples, the runs $\mathbf{S_{ST}R_{SI}}$ and $\mathbf{S_{LD}R_{SI}}$ correspond to returning the following as answers: a document containing no relevant text, followed by the document containing second highest amount of relevant, followed by remaining documents in order of most to least highlighted text ($\mathbf{R_{SI}}$). In the $\mathbf{S_{ST}R_{SI}}$ run each document in the list contains as parts of an answer only the (too small) elements within the highlighted passages, that is elements with no other elements nested within them ($\mathbf{S_{ST}}$). In the $\mathbf{S_{LD}R_{SI}}$ run the whole document is returned as the only answer part ($\mathbf{S_{LD}}$). As illustrated in Figure 3, of all the runs we consider, we would expect one or both of these two runs to be the worst performing.

Experimental results

We now present experimental results for the simulated runs of the $\mathbf{S}$–$\mathbf{R}$ space. We use version 5.0 of the INEX 2006 relevance assessments, which contains a set of judgements for 114 topics from INEX 2006.

Table 1 shows performance scores obtained with different evaluation measures on the 114 INEX 2006 topics. We base our analysis

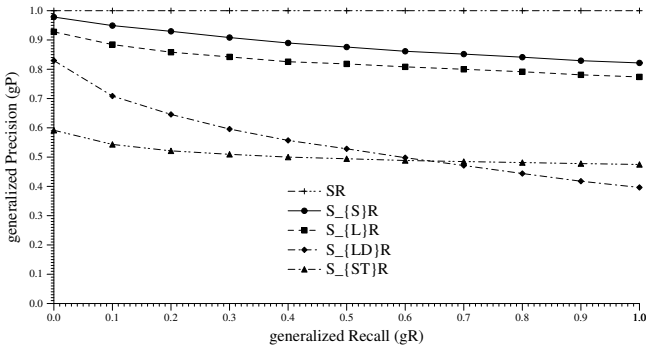


Figure 4: Evaluation of the overall performance of five simulated runs of the S-R space, using a fixed document ranking (R). The graph shows values for interpolated generalized precision (gP) at 11pt generalized recall (gR).

on the results obtained with the three overall performance measures (*MAgP*, *MAgP'*, and *MAP*), although results obtained with the three rank cutoff measures (*gP[r]*, *gR[r]*, and *gR'[r]*) are also reported. The runs are grouped in five clusters, depending on the answer parts retrieved ($\mathbf{S}$, $\mathbf{S_L}$, $\mathbf{S_{LD}}$, $\mathbf{S_S}$, or $\mathbf{S_{ST}}$).

Several observations can be drawn from these results.

First, when analysing the performance differences of runs with a fixed document ranking, we aim at separately investigating the first dimension of the **S–R** space (different sets of parts). The expected orderings for this dimension are correctly captured by both *MAgP* and *MAgP'*, but not by *MAP*. This is perhaps not surprising, since we are losing information in the abstraction toward the document level needed for MAP. Figure 4 shows an 11 point interpolated recall/precision graph plots for five simulated runs containing different sets of parts. Our initial expectations are confirmed: the passage run $\mathbf{S}$ results in perfect retrieval, and no other element run performs better than this run; returning $\mathbf{S_L}$ elements that fully contain the highlighted passages results in better performance than returning whole documents ($\mathbf{S_{LD}}$); and returning larger fully highlighted elements ($\mathbf{S_S}$) results in better performance than returning smaller fully highlighted elements ($\mathbf{S_{ST}}$). Although we did not initially speculate about the expected ordering between $\mathbf{S_S}$ and $\mathbf{S_L}$, both Figure 4 and the scores in Table 1 show that, for the INEX 2006 topic set, returning larger fully highlighted elements ($\mathbf{S_S}$) seems to be a better retrieval strategy than returning elements that fully contain the highlighted passages ($\mathbf{S_L}$).

Second, when analysing the performance differences of runs in each cluster, we aim at separately investigating the second dimension of the **S–R** space (different document rankings). As expected, we observe that the first run of each cluster, which ranks documents in a descending order of their contained relevant information ($\mathbf{R}$), either outperforms or performs as well as the other runs in the same cluster, irrespective of the overall performance measure used. The case of inserting a non-relevant document at the top of the ranking ($\mathbf{R}$ versus $\mathbf{R_I}$ and $\mathbf{R_S}$ versus $\mathbf{R_{SI}}$) is also correctly captured by the three measures; however, the swap of the first two document ranks ($\mathbf{R}$ versus $\mathbf{R_S}$) is correctly captured only by *MAgP* and *MAgP'*, but not by *MAP*. We also observe a (somewhat unexpected) behaviour for the *MAgP'* measure when comparing $\mathbf{R_I}$ with the $\mathbf{R_{SI}}$ document ranking. Our initial expectation was that the $\mathbf{R_I}$ ranking would perform at least as good as its swapped counterpart $\mathbf{R_{SI}}$, which is indeed correctly captured by *MAgP* and *MAP*. However, for all but the third $\mathbf{S_{LD}}$ cluster *MAgP'* captures the exact opposite performance behaviour. These results therefore suggest that *MAgP'* is not as reliable as *MAgP*, which seems to correctly capture the expected run orderings for the second (as well as the first) dimension of the **S–R** space.

Last, in order to reflect the interaction between the two dimensions in the **S–R** space, we perform a per-topic analysis to investigate whether the expected run orderings (shown in Figure 3) are correctly captured by the two overall performance measures, *AgP* and *AgP'*. Table 2 shows the results of this analysis. For an expected run ordering (a row in the table), we report the following values: mean absolute difference between the run performances (*Diff*, in percentage); the number of topics (of the total 114 INEX 2006 topics) where the first run performs better ($>$), is equal to ($==$), or performs worse ($<$) than the second run; and the actual t-test p values used to check if the mean absolute performance differences are statistically significant. The general trend among these results is clear: *AgP* is capable of correctly capturing the expected run orderings of the simulated runs in the **S–R** space, where for each comparison among the run pairs (the rows in the table), the first run performs better or as good as the second run. We also observe four notable disagreements between *AgP* and *AgP'* when comparing run pairs that insert non-relevant document at the top of their rankings (the rows containing negative *AgP' Diff* numbers for mean absolute performance differences). As discussed previously, *AgP'* fails to correctly capture the expected run orderings after a non-relevant document is inserted at the top of the ranking.[1] However, we also observe that there are cases where the mean absolute performance differences obtained by *AgP'* are much larger than those obtained by *AgP*, which is especially true when comparing $\mathbf{R}$–>$\mathbf{R_I}$ and $\mathbf{R_S}$–>$\mathbf{R_{SI}}$ run orderings. This suggests that, even though the fidelity tests demonstrate that it is not as capable as *AgP* at capturing the expected behaviour, there may be cases where the *AgP'* measure is likely to be more sensitive than *AgP* at distinguishing between different retrieval approaches.

5. DISCUSSION AND CONCLUSIONS

In this section, we use our findings from the previous section to motivate a discussion about the following research topics: the comparison between passage and element retrieval; the usefulness of focused and traditional document retrieval in identifying relevant information; and the importance of modelling appropriate evaluation assumptions for a retrieval task.

Passage versus element retrieval

The results of our fidelity tests in Section 4 demonstrate that perfect retrieval for the *relevant in context* task can only be achieved when retrieving all the highlighted passages within a document, in their exact size. The absolute difference in *MAgP* scores between the passage and our best simulated element run was 13%, which shows that no element run can achieve perfect retrieval (although the score achieved by the perfect element run could be higher than the one achieved by our best element run). One explanation for this could be that there is an inherent bias of the highlighting assessment procedure towards passage retrieval, since assessors are allowed to highlight sentences which could span across or even be contained within element boundaries.

How can passage and element retrieval be sensibly compared? If there is an inherent bias towards passages, then this should be taken into account when comparing these two types of retrieval.

[1] Although *AgP'* may correctly capture the expected run orderings when a non-relevant document is inserted after the first highly ranked document.

Run ordering	AgP					AgP'				
	Diff (%)	$>$	$==$	$<$	p	*Diff* (%)	$>$	$==$	$<$	p
$\mathbf{SR}$–>$\mathbf{S_L R}$	+20	112	2	0	2.2e-16	+17	112	2	0	2.2e-16
$\mathbf{SR}$–>$\mathbf{S_S R}$	+13	112	2	0	2.2e-16	+8	112	2	0	2.2e-16
$\mathbf{SR}$–>$\mathbf{SR_S}$	0	0	114	0	—	0	0	114	0	—
$\mathbf{SR}$–>$\mathbf{SR_I}$	+10	114	0	0	2.2e-16	+24	114	0	0	2.2e-16
$\mathbf{S_L R}$–>$\mathbf{S_{LD} R}$	+26	113	1	0	2.2e-16	+16	113	1	0	2.2e-16
$\mathbf{S_L R}$–>$\mathbf{S_L R_S}$	+0.07	52	13	49	0.6023	+0.5	52	13	49	0.2962
$\mathbf{S_L R}$–>$\mathbf{S_L R_I}$	+9	114	0	0	2.2e-16	+20	114	0	0	2.2e-16
$\mathbf{S_S R}$–>$\mathbf{S_{ST} R}$	+41	114	0	0	2.2e-16	+45	114	0	0	2.2e-16
$\mathbf{S_S R}$–>$\mathbf{S_S R_S}$	+0.07	43	29	42	0.4146	+0.5	43	29	42	0.0963
$\mathbf{S_S R}$–>$\mathbf{S_S R_I}$	+9	114	0	0	2.2e-16	+22	114	0	0	2.2e-16
$\mathbf{SR_S}$–>$\mathbf{S_L R_S}$	+20	112	2	0	2.2e-16	+17	112	2	0	2.2e-16
$\mathbf{SR_S}$–>$\mathbf{S_S R_S}$	+13	112	2	0	2.2e-16	+9	112	2	0	2.2e-16
$\mathbf{SR_S}$–>$\mathbf{SR_{SI}}$	+10	114	0	0	2.2e-16	+22	114	0	0	2.2e-16
$\mathbf{SR_I}$–>$\mathbf{S_L R_I}$	+18	112	2	0	2.2e-16	+14	112	2	0	2.2e-16
$\mathbf{SR_I}$–>$\mathbf{S_S R_I}$	+12	112	2	0	2.2e-16	+7	112	2	0	2.2e-16
$\mathbf{SR_I}$–>$\mathbf{SR_{SI}}$	0	0	114	0	—	−2	0	0	114	5.9e-13
$\mathbf{S_{LD} R}$–>$\mathbf{S_{LD} R_S}$	+0.7	67	8	39	0.0004	+3	67	8	39	5.9e-05
$\mathbf{S_{LD} R}$–>$\mathbf{S_{LD} R_I}$	+7	114	0	0	2.2e-16	+18	114	0	0	2.2e-16
$\mathbf{S_L R_S}$–>$\mathbf{S_{LD} R_S}$	+27	113	1	0	2.2e-16	+18	113	1	0	2.2e-16
$\mathbf{S_L R_S}$–>$\mathbf{S_L R_{SI}}$	+9	114	0	0	2.2e-16	+18	114	0	0	2.2e-16
$\mathbf{S_L R_I}$–>$\mathbf{S_{LD} R_I}$	+24	113	1	0	2.2e-16	+14	113	1	0	2.2e-16
$\mathbf{S_L R_I}$–>$\mathbf{S_L R_{SI}}$	+0.03	52	13	49	0.6023	−1	25	0	89	2.4e-06
$\mathbf{S_{ST} R}$–>$\mathbf{S_{ST} R_S}$	+0.1	60	0	54	0.4904	+1	60	0	54	0.2141
$\mathbf{S_{ST} R}$–>$\mathbf{S_{ST} R_I}$	+5	114	0	0	2.2e-16	+11	114	0	0	2.2e-16
$\mathbf{S_S R_S}$–>$\mathbf{S_{ST} R_S}$	+41	114	0	0	2.2e-16	+45	114	0	0	2.2e-16
$\mathbf{S_S R_S}$–>$\mathbf{S_S R_{SI}}$	+9	114	0	0	2.2e-16	+20	114	0	0	2.2e-16
$\mathbf{S_S R_I}$–>$\mathbf{S_{ST} R_I}$	+36	114	0	0	2.2e-16	+34	114	0	0	2.2e-16
$\mathbf{S_S R_I}$–>$\mathbf{S_S R_{SI}}$	+0.03	43	29	42	0.4146	−1	12	0	102	1.9e-09
$\mathbf{SR_{SI}}$–>$\mathbf{S_L R_{SI}}$	+18	112	2	0	2.2e-16	+14	112	2	0	2.2e-16
$\mathbf{SR_{SI}}$–>$\mathbf{S_S R_{SI}}$	+12	112	2	0	2.2e-16	+7	112	2	0	2.2e-16
$\mathbf{S_{LD} R_S}$–>$\mathbf{S_{LD} R_{SI}}$	+6	114	0	0	2.2e-16	+15	114	0	0	2.2e-16
$\mathbf{S_{LD} R_I}$–>$\mathbf{S_{LD} R_{SI}}$	+0.4	67	8	39	0.0004	+0.05	46	0	68	0.8790
$\mathbf{S_L R_{SI}}$–>$\mathbf{S_{LD} R_{SI}}$	+24	113	1	0	2.2e-16	+15	113	1	0	2.2e-16
$\mathbf{S_{ST} R_S}$–>$\mathbf{S_{ST} R_{SI}}$	+5	114	0	0	2.2e-16	+10	114	0	0	2.2e-16
$\mathbf{S_{ST} R_I}$–>$\mathbf{S_{ST} R_{SI}}$	+0.05	60	0	54	0.4896	−1	48	0	66	0.0189
$\mathbf{S_S R_{SI}}$–>$\mathbf{S_{ST} R_{SI}}$	+36	114	0	0	2.2e-16	+35	114	0	0	2.2e-16

Table 2: Comparison of AgP and AgP' scores of expected run orderings in the S–R space, using the 114 INEX 2006 topics. For each expected run ordering, a row shows the mean absolute performance difference (*Diff*), the number of topics where the first run performs better ($>$), is equal to ($==$), or performs worse ($<$) than the second run, and the t-test p value.

Accordingly, two different sub-tasks could be identified that allow a sensible comparison between passage and element retrieval:

- A *passage retrieval sub-task*, where the retrieval answers are passages and it makes sense to compare whether element retrieval techniques (based on the underlying XML structure) help in identifying more relevant passages; and
- An *element retrieval sub-task*, where the retrieval answers are XML elements and it makes sense to compare whether passage retrieval techniques help in identifying more relevant elements [8].

The evaluation measures proposed in this paper could be consistently used for evaluation of both sub-tasks.

Focused versus traditional document retrieval

The results of our fidelity tests in Section 4 demonstrate that the traditional IR measures, such as *MAP*, cannot fully capture the level of detail required by focused retrieval. More precisely, although the *MAP* score correctly reflects the different ordering of documents in the result list, it still does not reflect how well the retrieved information per document corresponds to the relevant information. On the other hand, we demonstrated that our proposed mean average generalized precision measure (*MAgP*) is able to fully capture both evaluation aspects, which makes it more useful than *MAP* in measuring the retrieval performance.

In a separate study, Kamps et al. [10] have used the top 20 run submissions in the INEX 2006 *relevant in context* task to compare the correlation of relative system rankings based on *MAgP* with that of *MAP*, and the extent to which the two measures are capable at distinguishing between different retrieval approaches. The rank correlation (Kendall's tau) between *MAP* and *MAgP* was found to be 0.6740 over the top 20 official submissions, while when comparing the numbers of significant differences, *MAgP* was able to distinguish more performance differences than *MAP* (112 versus 95 of the total 190 pairwise comparisons).

Modelling evaluation assumptions

In Section 2 we have listed several assumptions which are typically used in evaluating different text retrieval tasks. Assumption 3 (users consider longer more detailed answers to be more useful than shorter answers) and Assumption 4 (users consider all retrieved answers to be equally useful) are of particular importance for *in context* retrieval tasks, as it is not entirely clear which of the two assumption should be preferred for evaluation of the *in context* tasks. We have modelled these two assumptions with the two generalized recall definitions and their corresponding average generalized precision definitions, shown in Equations 5 to 8 in Section 3. However, our fidelity tests in Section 4 have demonstrated that the *AgP'* measure, based on Assumption 3, is not entirely measuring what it is supposed to measure, and that the *AgP* measure, based on Assumption 4, correctly captures the expected run orderings.

An argument for Assumption 3 is that it also motivates the preference given to more exhaustive answers in some evaluations, and one could argue whether the *AgP'* definition, shown in Equation 8, is really correctly modelling this assumption. However, fixing this definition requires further investigation, which might be solved in one of these two ways: first, a definition for interpolated average generalized precision could be used instead of the current non-interpolated definition; and second, the current non-interpolated *AgP'* definition could be re-defined as follows:

$$AgP' = gR'[|\mathcal{D}|] \cdot \frac{\sum_{j=1}^{|\mathcal{D}|} rel(d_j) \cdot gP[j]}{\sum_{j=1}^{|\mathcal{D}|} rel(d_j)} \qquad (11)$$

A more fundamental challenge, however, relates to the user preference of the two evaluation assumptions. Would users regard a focused and more concise answer as more useful than a lengthy exposition? Or would they indeed perceive the answer that contains more relevant (and possibly repeating) information as more useful? Currently, we do not have exact answers to these questions. We believe that it may be possible to determine the answers to these and similar questions either via user experiments or by questioning assessors about how they valued the answers for their topics.

Acknowledgements We thank the anonymous reviewers for providing useful comments on a draft of this paper.

REFERENCES

[1] J. Allan. HARD track overview in TREC 2003 high accuracy retrieval from documents. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, pages 24–37, 2004.

[2] J. Allan. HARD track overview in TREC 2004 high accuracy retrieval from documents. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*, 2004.

[3] S. Betsi, M. Lalmas, A. Tombros, and T. Tsikrika. User expectations from XML element retrieval. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 611–612, Seattle, USA, 2006.

[4] C. Clarke, J. Kamps, and M. Lalmas. INEX 2006 retrieval task and result submission specification. In *INEX 2006 Workshop Pre-Proceedings*, pages 381–388, 2006.

[5] N. Gövert, N. Fuhr, M. Lalmas, and G. Kazai. Evaluating the effectiveness of content-oriented XML retrieval methods. *Information Retrieval*, 9(6):699–722, 2006.

[6] W. Hersh, A. Cohen, P. Roberts, and H. Rekapalli. TREC 2006 genomics track overview. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.

[7] D. Hiemstra and V. Mihajlovic. The simplest evaluation measures for XML information retrieval that could possibly work. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 6–13, Glasgow, UK, 2005.

[8] W. Huang, A. Trotman, and R. A. O'Keefe. Element retrieval using a passage retrieval approach. In *Proceedings of the 11th Australian Document Computing Symposium (ADCS 2006)*, pages 80–83, Brisbane, Australia, 2006.

[9] J. Kamps and B. Sigurbjörnsson. What do users think of an XML element retrieval system? In *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2005*, volume 3977 of *Lecture Notes in Computer Science*, pages 411–421, 2006.

[10] J. Kamps, M. Lalmas, and J. Pehcevski. Evaluating Relevant in Context: Document retrieval with a twist. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, 2007 (to appear).

[11] G. Kazai, M. Lalmas, and A. de Vries. Reliability tests for the XCG and inex-2002 metrics. In *Advances in XML Information Retrieval: Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004*, volume 3493 of *Lecture Notes in Computer Science*, pages 60–72, 2005.

[12] J. Kekäläinen and K. Järvelin. Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology*, 53(13):1120–1129, 2002.

[13] H. Kim and H. Son. Users interaction with the hierarchically structured presentation in XML document retrieval. In *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2005*, volume 3977 of *Lecture Notes in Computer Science*, pages 422–431, 2006.

[14] E. Lagergren and P. Over. Comparing interactive information retrieval systems across sites: the TREC-6 interactive track matrix experiment. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 164–172, Melbourne, Australia, 1998.

[15] M. Lalmas and B. Piwowarski. INEX 2006 relevance assessment guide. In *INEX 2006 Workshop Pre-Proceedings*, pages 389–395, 2006.

[16] S. Malik, G. Kazai, M. Lalmas, and N. Fuhr. Overview of INEX 2005. In *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2005*, volume 3977 of *Lecture Notes in Computer Science*, pages 1–15, 2006.

[17] J. Pehcevski. *Evaluation of Effective XML Information Retrieval*. PhD thesis, RMIT University, Melbourne, Australia, 2006. `http://www.cs.rmit.edu.au/~jovanp/phd.pdf`.

[18] J. Pehcevski and J. A. Thom. HiXEval: Highlighting XML retrieval evaluation. In *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2005*, volume 3977 of *Lecture Notes in Computer Science*, pages 43–57, 2006.

[19] B. Piwowarski and G. Dupret. Evaluation in (XML) information retrieval: Expected precision-recall with user modelling (EPRUM). In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 260–267, Seattle, USA, 2006.

[20] S. Robertson. Evaluation in information retrieval. In *European Summer School on Information Retrieval (ESSIR)*, volume 1980 of *Lecture Notes in Computer Science*, pages 81–92, 2001.

[21] A. Tombros, B. Larsen, and S. Malik. Report on the INEX 2004 interactive track. *SIGIR Forum*, 39:43–49, 2005.

[22] A. Trotman and S. Geva. Passage retrieval and other XML-retrieval tasks. In *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, pages 43–50, Seattle, USA, 2006.

[23] E. Voorhees. Overview of the TREC 2003 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, pages 54–68, 2004.

[24] C. Wade and J. Allan. Passage retrieval and evaluation. Technical report, CIIR, University of Massachusetts, Amherst, 2005.

Chunking-based Question Type Identification for Multi-Sentence Queries

Mineki Takechi
Fujitsu Limited
17-25 Shinkamata 1-chome, Ota-ku
Tokyo, Japan
takechi.mineki@jp.fujitsu.com

Takenobu Tokunaga
Tokyo Institute of Technology
2-12-2 Ookayama, Meguro-ku
Tokyo, Japan
take@cl.cs.titech.ac.jp

Yuji Matsumoto
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma
Nara, Japan
matsu@is.naist.jp

ABSTRACT

This paper describes a technique of question type identification for multi-sentence queries in open domain question-answering. Based on observations of queries in real question-answering services on the Web, we propose a method to decompose a multi-sentence query into question items and to identify their question types.

The proposed method is an efficient sentence-chunking based technique by using a machine learning method, namely Conditional Random Fields. Our method can handle a multi-sentence query comprising multiple question items, as well as traditional single sentence queries in the same framework.

Based on the evaluation results, we discuss possible enhancement to improve the accuracy and robustness.

Categories and Subject Descriptors

H.3.4 [**Information Systems**]: Information Storage and Retrieval, Systems and Software

General Terms

Design, experimentation, management, performance

Keywords

question type identification, multi-sentence queries, web documents, question-answering system

1. INTRODUCTION

Question type identification is an essential component of various information access metods such as question-answering systems, information retrieval, dialogue systems, and other applications. It is the initial stage of the internal processing flow of the application, thus its accuracy exerts a major effect on the accuracy of the entire application. This paper proposes a question type identification method for multi-sentence queries in question-answering(QA) systems.

SIGIR 2007 Workshop on Focused Retrieval
July 27, 2007, Amsterdam, The Netherlands

In recent years, we have focused on the extraction of procedural expressions from web pages to provide answers to the How-to questions in open domain question-answering[18]. In the early stages of the study, we concentrated on extracting answer candidates, based on the assumption that the correct question type was given. In the latest study, we aimed to automatically identify classes of How-to type questions in web texts, such as blogs or e-mails, and then started research targeting the texts in question-answering services on the Web.

Most previous studies of open domain question-answering have dealt with single sentence queries. However, in the actual fields requiring question type identification, such as call centers of enterprises and Internet information services, they must frequently handle multi-sentence queries. Moreover, a single query often includes multiple questions.

A multi-sentence query often contains contents that are not directly used for question type identification, such as greetings or apologies. For extracting only sentences which need question type identification, irrelevant sentences must be removed so that the question type can be correctly identified.

Although some previous research works have studied the question type identification of multi-sentence queries, many of them rely on pattern matching. Open domain QA must handle a variety of questions, meaning approaches requiring manually created patterns are costly. Therefore, the automatic acquisition of such patterns is required, even on a partial basis.

This paper presents an approach to question type identification as a chunking problem of sentences, which combines N-grams of words and other features used for question sentence type identification via a machine learning technique called Conditional Random Field (CRFs).

We performed evaluations and experiments, and investigated the effectiveness of the proposed approach. We also report herein the accuracy of the question segment extraction required for question type identification and the accuracy of question type identification separately. Finally, we discuss individual effective features based on the results of analyses.

2. QUESTION SEGMENTATION AND TYPE IDENTIFICATION

s1 Given when I sleep enough every night I am very tired all day.

s2 Any doctors tell me what these symptoms resemble depression. But what is the definition of depression?

s3 In my office I have no time to relax. Because of my post-

s4 Any office is concerned about my recent condition and I recommend that I see the doctor.

s5 Do so other directors like me manage their work stress?

s6 Please let me know if you have good advice.

Figure 1: Example of a Multi-Sentence Query.

Figure 1 shows an example of a multi-sentence query in web question-answering services. In this example, the sentences are numbered sequentially. The single query includes two questions; one described by sentence s2 and another by sentences s5 and s6 respectively. In this paper, a set of sentences describing a single question, such as s5 and s6, is called a *question segment*. Therefore, the query shown in Figure 1 includes two question segments. A variety of question segments are conceivable: however, in this paper, a question segment is assumed to be the shortest series of sentences describing a question. Question type identification herein means extracting question segments and identifying their question types.

Comparing single sentence queries in previous work, it is not clear what characteristics are effective in extracting question segments from a multi-sentence query and identifying their question type. The characteristics for question type identification in previous research must be reviewed in an evaluation of question segments including multiple sentences. With this in mind, we therefore annotated actual multi-sentence queries and analyzed the characteristics that were necessary for question segment extraction and question type identification.

3. QUESTION TYPE ANNOTATION

As an operator of question-answering services that provides answers for questions from unrestricted users in the Internet, we chose "Oshiete! goo."[1] We studied 2,234 queries obtained from articles in 21 categories of "Oshiete! goo" such as town/local information, healthcare, and so forth. The average number of sentences per query is 5.7 and its deviation is 3.9. The average length and deviation of a sentence are 73.9 bytes and 51.8 respectively.

Question types were manually tagged based on the ten kinds of question types, namely Yes-No, Name, Description, Evaluation, How-to, Reason, Location, Time, Consultation and Other. Their definitions are detailed in other publications[17]. The annotators tagged passages considered necessary to identify one question and its question type. Consequently, one question was expressed by a set of several text passages. The boundary of tagged passages were allowed to be in any place and not necessarily at the start or end of a sentence. Moreover, only one question type was allowed to be assigned to a passage, meaning no overlapped passages tagged in different question types could be contained in a single sentence. The annotators annotated question types without seeing its answer or question title.

[1]http://oshiete.goo.ne.jp/

Table 1: Classified Given Question Types.

Question-Types	Number of Passages
Yes-No(Y)	1709 / .43
Description(D)	636 / .59
Name(N)	454 / .71
How-to(W)	325 / .79
Reason(R)	304 / .87
Location(L)	197 / .92
Evaluation(E)	141 / .95
Consultation(C)	106 / .98
Time(T)	63 / 1.00
Others(OT)	10 / 1.00
Total	3945

The corpus was divided into two, and two annotators A and B classified the respective articles. Furthermore, 234 queries collected in 2001 were tagged by another annotator C from annotators A and B. The question type annotation results of annotator C were then compared with those of annotators A and B to calculate the inter-annotator agreement.

The results of this question type annotation are shown in Table 1. The right column in the table indicates the frequency of tagged passages for each question type where they are arranged in the descending order of frequency from the top. The adjacent values of each frequency, meanwhile, indicate their cumulative ratio of frequencies to the total frequency of all passages.

In total, there are 1252 articles, each containing multiple question items and 3945 question segments related to their question items were confirmed. The number of question items per article was 1.77. There were 98 questions where the passage corresponding to one question item was contained in multiple sentences and 188 sentences each containing multiple question items, accounting for about 5% of all sentences containing question items.

The agreement for question type annotation was calculated on a sentence-by-sentence basis. The question type was annotated for passages, consequently, the question type for a sentence is not confirmed in this state. The question type of a passage is assigned to a sentence containing the passage, while a sentence containing multiple question items was handled as having multiple question types. In this case, the agreement for question type annotation was assumed to agree when all the question types of the sentence matched. The F-measure as used in the evaluation of MUC[2] was used for the inter-annotator agreement for question type annotation.

After calculating the inter-annotator agreement for question

[2]http://www–nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/muc7_score_intro.pdf

types, variations of inter-annotator agreement were found to occur depending on the question types, with the Yes–No and Location types achieving the highest agreement at 0.7. For sentences containing multiple question items, all the tagged question types need to match, meaning the agreement tends to be low. When the agreement was calculated excluding sentences containing multiple question items, the F-measure was 0.8 in the Yes–No type, the Location type, and the How–to type with the highest agreement, while the agreements of other question types stayed low.

4. CHUNKING-BASED IDENTIFICATION

Our goal is to extract question segments in a query and identify their question types. When a question segment is defined as a sequence of sentences, our task can be perceived as assigning a label to each sentence, which is indicated either inside or outside of the question segments, namely the so-called labeling or chunking problem.

Chunking is a process of identifying chunks that indicate some sort of visual or semantic unit. In natural language processing, chunking is used to find various kinds of units, such as noun phrase, paragraph, named entities and lexical and grammatical units. In our case, the target unit is question segments.

Although there are various ways to represent chunks, we adopted a method assigning a status to each sentence, which permits the use of the same framework as one for the conventional problem of tagging morphemes and noun phrases. For this task, previous methods such as Inside/Outside [13, 14] and Start/End [21] were proposed. Kudo et. al.[8] summarized them into five expressions of IOB1, IOB2, IOE1, IOE2, and IOBES(Start/End). Firstly, the following ten kinds of conditions are defined;

I1 The sentence is part of the chunk.

I2 The sentence is a middle sentence other than that at the start or end of the chunk, consisting of three sentences or more.

B1 The sentence is at the start of the chunk immediately following a chunk.

B2 The sentence is at the start of chunk.

B3 The sentence is the one at the start of the chunk consisting of two sentences or more.

E1 The sentence is at the end of the chunk immediately preceding a chunk.

E2 The sentence is at the end of chunk.

E3 The sentence is at the end of the chunk consisting of two sentences or more.

S The sentence composes one chunk by itself.

O The sentence is not included in any chunk.

At this time, IOB1, IOB2, IOE1, IOE2, and IOBES are models that perform tagging to meet the following rules based on the combination of conditions above;

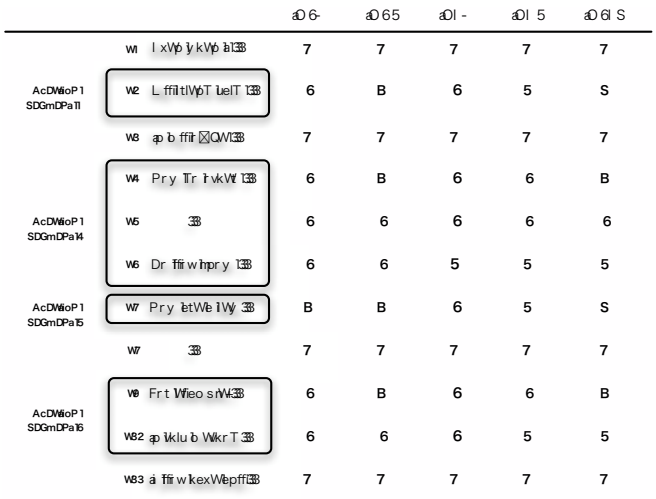


Figure 2: Example Assignment of Chunk Labels.

IOB1 I1, O, B1

IOB2 I1, O, B2

IOE1 I1, O, E1

IOE2 I2, O, E2

IOBES I2, O, B3, E3, S

Examples tagged by IOB1, IOB2, IOE1, IOE2, and IOBES are shown in Figure 2.

In order to indicate the question type of a chunk, a tag indicating the question type is linked to a tag indicating a portion in the chunk such as B, E, I, and S with a hyphen "-". For example, the B-W of IOB2 in Figure 3 indicates the start sentence of question segment 4 annotated How-to question type by the "-W" tag. Identically, "B-D" means the sentence is the first sentence in a question segment stated Description question type by "-D" tag.

4.1 Overview of the proposed technique

The processing flow in the proposed technique of question type identification follows the steps in the list below;

Step 1 Divide a question article into sentences, each of which is terminated with a period ".".

Step 2 Carry out chunking with respect to each article.

Step 3 Extract question segments labeled with their question types.

The chunker divides a sequence of sentences into question segments and other chunks and a chunk tag is assigned to each sentence. The chunk tags used are of five types, namely IOB1, IOB2, IOE1, IOE2, IOBES, and the IO-tag that does not distinguish B/E/S tags from the I-tag. Sentences not involved in the identification of question types are assigned

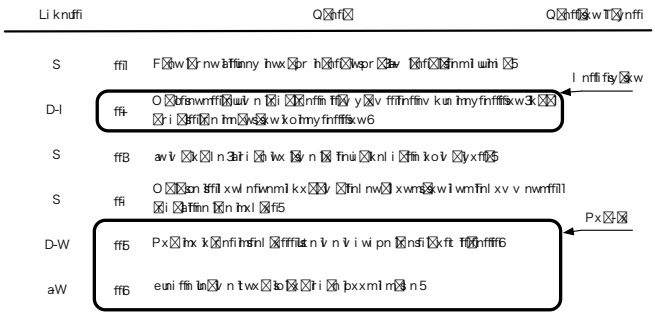


Figure 3: Extracting Question Segments and Identifying Question Types.

the O-tag, while those sentences constituting a question segment are assigned a tag consisting of the combination of one of the letters I, B, E, and S and one of the letters stating the question type such as W and D. For example, I-W tag and B-D tag represent the portion in the chunk and the question type. Figure 3 shows an example of composition of chunks using the IOB-tags. A chunker learns a chunking model from the pairs of sentences and their chunk tags as shown in Figure 3. To extract question segments from a query, a chunk tag is assigned to each sentence. Subsequently, sentences labeled with the same type, such as "-D" and "-W", are chunked by post-processing. Consequently, a question segment is extracted as a chunk and the question type is assigned to the question segment based on the chunk tag.

4.2 Conditional random fields(CRFs)

The Conditional Random Fields(CRFs) is a sequence modeling framework that has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence. CRFs perform better than Hidden Markov Models(HMMs) and Maximum Entropy Models when the true data distribution has higher-order dependencies than the model, as often appears in practical cases and have thus been recently used in bioinformatics and natural language processing. The advantages of CRFs on which we focused attention are as follows; (1) There is no need to assume the independency of random variables as with those in the Markov model, (2) Since a model is described with conditional random variables, the model parameters can be estimated without calculating the distribution of random variables in the condition. One report points out that CRFs provide performances similar to that of the HMMs with the number of training cases less than that needed for the HMMs in the order of sample of 1 to one-several-tenths [5].

For a set of feature function F, let the number of locations where a feature $f \in F$ holds for a combination (x, y) of random variables x and y be $\phi_f(x, y)$, and let a vector whose elements are $\phi_f(x, y)$ be $\Phi(x, y)$. The variable x is a input symbol for the conditions of a model and the varibale y is a label that the model outputs. Let the significance of feature f be represented by θ_f and a vector including θ_f as its elements be Θ. Subsequently, the degree of confidence of giving a label can be expressed by equation (1).

$$\langle \Theta, \Phi(x, y) \rangle = \sum_{f \in F} \theta_f \phi_f(x, y) \quad (1)$$

Using this, let equation (2) defines a conditional probability $Pr(y|x)$. This is an expression directly to represent the probability model of a CRF.

$$\Pr(y|x) = \frac{\exp\langle \Theta, \Phi(x, y) \rangle}{\sum_{y \in Y} \exp\langle \Theta, \Phi(x, y) \rangle} \quad (2)$$

where Y is a set of labels.

The detailed model of CRF can be found in the previous studies[10, 5].

4.3 Experimental settings

To evaluate the effectiveness of the proposed technique, we conducted an experiment to extract question segments and identify question types in actual question articles. Excluding articles satisfing one of conditions below a), b), and c) apply, we chose 954 queries from 2234 queries in the corpus described in Section 3 as the dataset for our experiments.

a) The queries include the Yes–No type or the Other type.

b) The queries include sentences that have different question types in one sentence.

c) The queries do not include a question described in multiple non-adjacent sentences.

The Yes–No type could be interpreted as other question types. For example, "Do you know how to install this software?" can be answered by Yes or No, however this question asks you a method, which make it a How-to type question, requiring different handling to other question types. Hence we decided not to include the Yes–No type in our present study. Since questions including multiple questions in a sentence require pre-processing not directly involved in sentence chunking, those are not covered in the present study, either. Under the definition of the question segment in Section 2, there is no guarantee that a question segment can consist of only adjacent sentences. In fact, in the results of the question type annotation we conducted, there are multiple non-adjacent sentences grouped into the same question segments. Because of the lack of such cases, the experiments in this paper eliminate queries, including question segments consisting of non-adjacent sentences. Sentences were segmented by periods alone, with one question type assigned to a single sentence. As in the question type annotation in the previous section, a question type of a sentence was defined to be the question type of passages in the sentence. For the question types in this experiment we used those proposed during the past QA Workshop [15] and those with unique tags defined based on the results of the previous study by Tamura et.al.[19].

The chunking features are composed of uni-gram and bi-gram of parts of speech. After feature selection using the

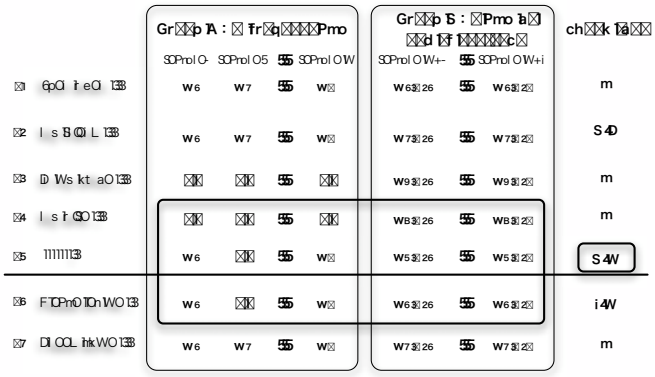


Figure 4: Example of the Data Format in the Learning and Testing of Chunking When the Window Size Equals to Three Sentences.

frequency of features in the learning corpus, a thousand frequent parts of speech are stored. Additionally, we performed an experiment exploiting only several words at the beginning and end of sentences. The reason is that symbols, function words such as question marks, and auxiliaries at the ends of sentences, are expected to be effective for the extraction of question segments. Identically, interrogatives at the beginning of sentences are thought to work well for question type identification.

For chunk tag sets, we exploited five types mentioned in the previous sections, namely IOB1, IOB2, IOE1, IOE2, IOBES, and the IO types that do not distinguish two adjacent question segments. As a CRF implementation, we used CRF++[3] developed by Kudo and the learning parameters were set in default values.

The features used in this experiment were only combinations of part–of–speech(POS). Uni-gram and bi-gram of POS, and n words from the beginning or the end of a sentence were exploited and the number n was varied from 1 to 5. In the case of only the uni-gram, two tests were conducted both in the feature set only including content words only and in the feature set including all words respectively. Figure 4 represents the format of the feature set of learning and test data for CRF++, which is a matrix of sentence features. Each column is assigned to one feature and each cell in this matrix indicates a feature value corresponding to the sentence. In this experiment, the values of the features are binary.

In Figure 4 w_1, w_2, ..., and w_m indicate the top m words in frequent words ranking in the dataset, and $w_{1,m+1}$, $w_{2,m+2}$, $w_{7,m+n}$ the n words at the end of each sentence. The 'nil' indicates that those features are not included in the sentence.

As shown in Figure 4, the feature columns can be divided into several groups of columns, some of which were exploited in combination. A sequence of sentences are used as the context of a targeted sentence in the process of chunking. We define a "window" as a sequence of contextual sentences exploited in chunking. The window size varied in the following sizes; only target sentences for chunking, three sentences, including one forward and one backward sentence, and five sentences, including two forward and two backward sentences of the target. Table 2 summarizes these experimental conditions.

[3]http://chasen.org/~taku/software/CRF++/

Table 2: Summary of Experimental Settings.

Features	Set1 : uni-gram of all/content words
	Set2 : uni-gram + bi-gram of all words
	Set3 : n POSs at the end of sentence(n=1-5)
	Set4 : n POSs at sentence head and end(n=1-5)
Tags	IO/IOB1/IOB2/IOE1/IOE2/IOBES
Window	one, three and five sentences

Table 3: Accuracy of Chunking.

	Uni-All	Uni-Con	Uni+Bi	#Seg's
Accuracy	.29	.18	.29	–
Segmentation	.56	.32	.57	1088
Consultation	.12	.07	.15	66
Description	.3	.11	.34	246
Evaluation	.27	.13	.27	80
Location	.34	.15	.33	108
Name	.34	.20	.30	258
Reason	.33	.06	.35	146
Time	N/A	N/A	N/A	13
How-to	.5	.26	.47	171

4.4 Experimental results

Table 3 indicates the evaluations of chunking when varying experimental settings. In their settings, thousand of words which appear most frequently in the experimental corpus are used. Table 3 represents the F-measure value for each of the question types, and the accuracy is computed by regarding a case as the correct estimation when their segments and question types for all questions in a query are correctly assigned. These F-measure values are independently computed in segment extraction and question type identification. During the computation of F-measure values of segmentation, meanwhile, only the segmentation result is checked.

The accuracy generally shows low performance, meaning this task cannot be performed accurately with simple word features. The accuracy of chunking was performed by using all kinds of parts of speech rather than the use of content words alone.

No question segment shows high accuracy regardless of feature selection, but the best performance was obtained by using all parts of speech in How-to type. Compared with the results using uni-gram alone and using both uni-gram and

Table 4: Results of Chunking When Varying Window Size.

	Window size		
	1	3	5
Accuracy	.29	.28	.28
Segmentation	.57	.57	.60
Consultation	.15	N/A	.03
Description	.34	.33	.32
Evaluation	.27	.17	.20
Location	.33	.22	.19
Name	.3	.28	.28
Reason	.35	.3	.28
Time	N/A	N/A	N/A
How-to	.47	.41	.41

Table 5: Accuracy of Labeling Sentences with Different Chunk Tag Sets.

	IO	IOB1	IOB2	IOE1	IOE2	IOBES
I	.76	.74	.14	.73	.11	N/A
O	.94	.94	.94	.94	.94	.94
B	-	.16	.74	-	-	.11
E	-	-	-	.13	.73	.15
S	-	-	-	-	-	.72

bi-gram, their segmentation with bi-grams showed slightly better performance than with uni-grams alone but their type identification not always. For instance, when adding bi-gram to uni-gram in features, the accuracy of type identification was increased in the Description type, contrarily declined in the How-to type.

Table 4 shows the results of question extraction and type identification when varying in the window size, with the values in the cells of this table computed as the same manner as in Table 3. As shown in Table 4, we obtain no salient difference in the accuracies of chunking. On the other hand, there are some differences in question type identification, along with the changing widow size.

Table 5 presents the performance of question extraction by using different chunk tag sets. The values in this table indicate F-measures of I/O/B/E/S tags when exloiting each chunk tag sets. The IO tag set, which cannot recognize adjacent question segments, achieves high F-measure values in the type identification of I tag. In the IOB1 tag set, a B-tag, which indicates the boundary of adjacent question segments shows a lower performance. In the case of the IOB2 tag set, I-tag, which indicates the inside or end of a question segment, also shows lower performance. This kind of tendency is also observed in the experimental results of E-tag in IOE1 and IOE2. When using the IOBES tag set meanwhile, the S-tag of a question segment with no adjacent question segment shows a high F-measure but the performance of I/B/E tags remains lower.

Table 6 shows the confusion matrix of B-* tags. Each column indicates a type of estimated tag. To clarify the changing between the correct and estimated tags, we choose only experimental results for queries that comprise a question segment consisting of a sentence and recounted the frequencies of estimated tags. The B-T tag is eliminated in Table 6, because B-T merely appeared in selected queries for recounting.

Table 6: Distribution of Estimated B-tags for true B-tags.

	Estimated tags						
	B-C	B-D	B-E	B-L	B-N	B-R	B-W
B-C	0	3/.20	0	2/.13	7/.47	1/.07	2/.13
B-D	1/.02	37/.61	0	0	14/.23	7/.11	2/.03
B-E	1/.07	3/.20	7/.46	0	3/.20	0	1/.07
B-L	1/.04	0	0	6/.21	20/.71	1/.04	0
B-N	0	12/.18	1/.01	5/.07	45/.67	4/.06	1/.01
B-R	0	9/.19	2/.04	2/.04	10/.21	25/.52	0
B-W	0	5/.07	1/.02	2/.03	13/.19	1/.02	45/.67

For most of the question types, the majority involved cases where the correct tags were estimated, although that is not the case with B-C and B-L tags. In particular, B-C completely failed in the estimation. This reveals a tendency whereby question types such as B-C,B-D,B-L,B-R and B-W are wrongly classified to B-N type when identification of the same fails.

Conversely, focusing on How-to type question marked by the B-W tag, few with tags other than B-W are miscategorized to B-W. To improve the accuracy of the extraction of How-to type questions, error categorizations of B-W to B-N must be avoided. To do so, more detailed error analysis of these cases is required.

5. DISCUSSION

When failing in question segment extraction, errors often occur in the boundaries of adjacent question segments and in the inside of segments comprising two or more sentences. At the boundaries of adjacent segments, by using IOB2, IOE2 and IOBES tag sets, performance enhancement was achieved. When using the IOB2, IOE2 and IOBES, however the performance of labeling the sentence in the inside of a chunk contrarily was declined. Because the number of such chunks is few in our corpus, positive examples for the CRFs considered to be insufficient.

The experimental results show the opposite natures between in question segmentation and in question type identification when using the same features. In general, it should be difficult to reveal such two different problems in the same computational model and the proposed method has not considered this aspect of the problem. Since the concurrent processing of question segmentation and question type identification is effective reducing computational cost, we chose this approach at the beginning of this study. However, we might need to change the strategy so that we could reduce the computational cost along with exploiting different models for question segmentation and question type identification

in the next step.

Another important observation in the experimental result is that many errors of question segmentation and type identification occurred in sentences including many ellipses. That process to identify ellipsis is, known as anaphora resolution [24, 4, 1, 2], is generally difficult, meaning insufficient accuracy has been achieved for use in practical tasks to date. As an alternative to avoid anaphora resolution, the addition of sentences probably including elided elements into a chunk could be considered. From this perspective, I will enhance question segmentation and question type identification as in the following paragraphs.

In question segment extraction, the portion and structure of a question segment in a query have not been identified yet, thus the bag-of-words approach using words in the query is plausible. However if a question segment includes many ellipses, the bag-of-words approach is insufficient to extract the features of question segments. To solve this problem, it is worthwhile to perform ellipsis analysis on the entire before the question segment extraction.

In the experimental results of question type identification, the performance using only features of a chunked segment, presents better than that using the features of contextual sentences before and after the chunked sentence together, meaning it is difficult to improve the accuracy of question type identification by simply adding contexts of chunked sentence. On the other hand, because ellipses in chunked sentences are problematic in question type identification as well as question segment extraction, this problem should be solved.

6. RELATED WORK

Identification of the question types of question sentences has often been made by pattern matching using lexico-semantic patterns that consider grammar and word meaning classes. A similar strategy has been applied to many other question-answering systems since the success of this method in question analysis in early studies of open domain question-answering [12, 15, 23, 6].

For studies using machine learning, techniques based on learning algorithms such as a decision tree [26], a maximum entropy model [3], SNoW [11], and Support Vector Machines [16, 25] have been proposed. In Support Vector Machines (SVMs) [22], Suzuki proposed a question type identification technique using the N-gram of words and their meaning classes as features. The reports of Suzuki indicate that SVMs can bring about the best result of question type identification of conventional learning algorithms such as the decision tree and maximum entropy model.

Previous studies for multi-sentence queries include the classification of sentences in question-answering logs that accumulated at the call center of a business. For instance, there is automatic answering at the help desk of an academic organization [9, 7] and question type identification for QA articles at question-answering sites on the Internet [19, 20].

Tamura et.al extracted questions from multi-sentence queries in articles at question-answering sites on the Internet and tried to identify the question types of these questions [19]. Tamura et.al., expanding on their initial method, proposed a technique applicable to cases including multiple question sentences in a single article [20]. Their technique, however, depends on manual work for type identification, though question sentences called *core sentences* are automatically extracted, making it unclear how accurately it can identify question types in a question article including multiple questions.

Tamura et.al's technique and ours differ in the following points. Whereas Tamura et.al target questions consisting of a single sentence when extracting question segments, our method extracts questions from a multi-sentence query. In our data, question type annotation is performed with any strings whereas their technique tags only sentences. Since our technique is designed to permit the question type annotation of multiple passages for the same question, it can optionally mark any relations between such passages if necessary for more detailed analysis.

7. CONCLUSIONS

We dealt with the question segmentation and type identification for multi-sentence queries simultaneously and also proposed a learning-based technique of question type identification and showed the evaluation of those methods. The experimental results clarified the different tendencies of performance between different question types using the same features of texts, which suggests two directions in the next step of study: two pass processing, such as the method proposed by Tamura et.al. and acquiring other discriminative features that are effective in question segment extraction and the type identification. In particular, as regards question type identification, anaphora resolution is demanded to acquire the key features to discriminate the question types.

8. ACKNOWLEDGMENTS

I would like to thank Dr. Taku Kudo and Dr. Tetsuro Takahashi for providing us their useful free software of machine learning and corpus annotation tools.

9. REFERENCES

[1] R. Iida, K. Inui, and Y. Matsumoto. Anaphora resolution by antecedent identification followed by anaphoricity determination. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(4):417–434, 2005.

[2] R. Iida, K. Inui, Y. Matsumoto, and S. Sekine. Noun phrase coreference resolution in Japanese based on most likely candidate antecedents. *IPSJ Journal*, 46(3):831–844, 2005. in Japanese.

[3] A. Ittycheriah, M. Franz, Wei-Jing, and A. Ratnaparkhi. Question answering using maximum entropy components. In *Proceedings of NAACL-2001*, pages 33–39, 2001.

[4] M. Kameyama. *Centering theory in discourse*, chapter Intrasentential Centering: A Case Study, pages 89–112. Oxford, Clarendon Press, 1998.

[5] H. Kashima, Y. Tsuboi, and T. Kudo. Development of discriminative models in natural language processing –from HMM to CRF–. In *Proceedings of tutorial in*

the 12th Annual Meeting of the Association for Natural Language Processing, 2006. in Japanese.

[6] T. Kato, J. Fukumoto, and F. Masui. An overview of NTCIR-5 QAC3. In *Proceedings of NTCIR-5 Workshop Meeting*, Tokyo, Japan, December 2005.

[7] Y. Kiyota, S. Kurohashi, and F. Kido. Dialog Navigator : A question answering system based on large text knowledge base. *Journal of Natural Language Processing*, 10(4):145–175, July 2003. in Japanese.

[8] T. Kudo and Y. Matsumot. Chunking with support vector machines. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association of Computational Linguistics*, pages 192–199, 2001.

[9] S. Kurohashi and W. Higasa. Dialogue helpsystem based on flexible matching of user query with natural language knowledge base. In *Proceedings of 1st ACL SIGdial Workshop on Discourse and Dialogue*, pages 141–149, 2000.

[10] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, 2001.

[11] X. Li and D. Roth. Learning question classifiers. In *COLING2002*, pages 556–562, August 2002.

[12] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Goodrum, R. Girju, and V. Rus. LASSO: A tool for surfing the answer net. In *Proceedings of TREC-8*, pages 175–184, 1999.

[13] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the 3rd Workshop on Very Large Corpora*, pages 88–94, 1995.

[14] E. T. K. Sang and J. Veenstra. Representing text chunks. In *Proceedings of EACL 1999*, 1999.

[15] Y. Sasaki, H. Isozaski, H. Taira, T. Hirao, H. Kazawa, J. Suzuki, and E. Maeda. SAIQA : A Japanese QA system based on a large - scale corpus. In *IPSJ SIG Notes FI-64*, pages 77–82, 2001. in Japanese.

[16] J. Suzuki. *Kernels for Structured Data in Natural Language Processing*. PhD thesis, Nara Institute of Science and Technology, 2005.

[17] M. Takechi. *Identification of Multi-Sentence Question Type and Extraction of Descriptive Answer in Open Domain Question-Answering*. PhD thesis, Nara Institute of Science and Technology, 2007.

[18] M. Takechi, T. Tokunaga, Y. Matsumoto, and H. Tanaka. Feature selection in categorizing procedural expressions. In *Proceedings of the 6th International Workshop on Information Retrieval with Asian Languages: IRAL2003*, pages 49–56, July 2003. in Japanese.

[19] A. Tamura, H. Takamura, and M. Okumura. Classification of multiple-sentence questions. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 426–437, October 2005.

[20] A. Tamura, H. Takamura, and M. Okumura. Extraction of question items and identification of their dependency relations. In *Proceedings of the 12th Annual Meeting of the Association for Natural Language Processing*, 2006. in Japanese.

[21] K. Uchimoto, Q. Ma, M. Murata, H. Ozaku, and H. Isahara. Named entity extraction based on a maximum entropy model and transformation rules. In *Proceedings of the ACL 2000*, 2000.

[22] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[23] E. M. Voorhees. Overview of TREC 2003 Question Answering Track. In *Proceedings of the twelfth Text REtreival Conference(TREC-12)*, 2003.

[24] M. Walker, M. Iida, and S. Cote. Japanese discourse and the process of centering. *Computational Linguistics*, 20(2):193–233, 1994.

[25] D. Zhang and W. S. Lee. Question classification using Support Vector Machines. In *Proceedings of SIGIR-2003*, pages 26–32, 2003.

[26] I. Zukerman and E. Horvitz. Using machine learning techniques to internet wh-questions. In *Proceedings of ACL-2001*, pages 547–554, 2001.

Can we at least agree on something?

Andrew Trotman
University of Otago
Dunedin
New Zealand
andrew@cs.otago.ac.nz

Nils Pharo
Oslo University College
Oslo
Norway
Nils.Pharo@jbi.hio.no

Dylan Jenkinson
University of Otago
Dunedin
New Zealand
djenkins@cs.otago.ac.nz

ABSTRACT

During a session of the INEX 2006 workshop in Schloss Dagstuhl the first at-INEX experiment was run. Participants were asked to assess topics in order to increase the number of multiple assessed topics available for analysis (and in order to increase the number of assessors per topic). This contribution presents the experimental set-up, the experiment, and an analysis of the results.

When examining the agreement level across all assessors it is shown that each assessor both brings new material, and disagrees with the there-to consensus. Extrapolation suggests that with 8 assessors, there will be no content that they all agree is relevant, but they continue to agree on which documents are reliant until 19 assessors are present. This suggests that relevance is in the mind of the assessor and not a ground truth.

Additionally examined are several problems encountered in conducting the experiment. These are explained in detail and recommendations for change in the INEX methodology are made.

Categories and Subject Descriptors

H.3 [Information Storage and Retrieval]: Information Search and Retrieval – Retrieval models, Search process.

General Terms

Human Factors, Theory.

Keywords

Element retrieval, XML-retrieval, agreement levels.

1. INTRODUCTION

Each year INEX participants travel to Schloss Dagstuhl near Frankfurt in Germany for the annual workshop. For Europe the venue is isolated. There is no airport or railway station. Participants are essentially locked-down, nowhere to go and nothing to discuss except XML and information retrieval. Nothing to do other than present talks, listen to talks, and to participate in lively discussions.

During the 2006 INEX round the decision was made to take advantage of the lock-down in order to conduct an experiment. The INEX workshop participants are a substantial human resource, knowledgeable in the domain of information retrieval, and with the time and motivation to participate in an experiment (while at the workshop).

SIGIR 2007 Workshop on Focused Retrieval
July 27, 2007, Amsterdam, The Netherlands

The nature of such an experiment is dictated by the physical environment, the time available, and the participants. The experiment must require many participants, must be conducted in parallel on each participant, and must require no more than one workshop session. It must also not become overbearing or a disincentive from attending the workshop.

The experiment conducted at INEX has become known as the at-INEX experiment. It was run for the first time in 2006 and is expected to continue as a feature of INEX at future workshops. This contribution outlines the first at-INEX experiment, the motivation behind the experiment, the experiment, and the results.

2. CHOICE OF EXPERIMENT

Two domains were considered for the at-INEX experiment: an interactive experiment, and an assessment experiment.

Unlike a Cranfield methodology laboratory experiment [17], an interactive experiment requires a substantial number of participants (and topics) for statistical significance. The INEX interactive experiment in 2006, for example, had over 80 participants each performing 4 queries selected from a total of 12 [6]. In that experiment each participant was given a total of 15 minutes to fulfill the information need. When the time taken to answer questionnaires before, during, and after the experiment is added to the time it took participants to familiarize themselves with the experimental conditions, and to the four lots of fifteen minutes, a total running time of between 1.5 and 2 hours was needed for each participant.

The at-INEX environment matches the needs of an interactive experiment perfectly. There are many available participants and the time frame is relatively short. Certainly if each participant performed only 2 searches and the questionnaires were kept short then such an experiment could be conducted in just one workshop session.

Assessment experiments (that is, judging topics) require only one participant per topic and can be done by hundreds of people working on different topics in parallel. This is the traditional model used at INEX [7] (and TREC [18]). Assessing a single topic at INEX 2005 took about 11 hours, and at INEX 2006 it took about 7 hours [12] – vastly more time than available at the workshop in Schloss Dagstuhl. On initial inspection an assessment experiment is a bad match to the experimental environment, however this is the nature of the experiment that was conducted.

The time to assess topics for INEX has been of concern and under investigation for many years [12]). INEX assessors are the participants themselves and are not paid for the task. Their reason for participating is their research, not their desire to perform assessment. Assessing is considered by some as a necessary evil

done only to get performance measures for their search engines. Some (including one of the authors) have employed students to assess because the task is dreary and laborious. Much to the surprise of the authors, some INEX workshop participants who participated in the at-INEX experiment had never judged a single document so had managed to duck the task year after year – clearly they did not consider assessing something to look forward to.

An ongoing task at INEX is the reduction of the assessment load while at the same time maintaining assessment quality. Considerable advances have been made. From 2002 to 2006 the changes included changing from a two-dimensional graded relevance scale to a one-dimensional continuous scale [12]. Changes from explicit assignment of assessments to each element to the yellow-highlighting method suggested by Clarke [4]. But there remains room for further load reduction. Specifically the at-INEX experiment in 2006 aimed to answer several questions:

1. Do the assessments for a single topic need to be conducted by a single assessor?

The assessment of a single topic might, for example, be split amongst two, three, or more judges, each assessing part of the topic. Advantage might be taken of graduate students studying IR to assess a topic during class as a hands-on method of learning about the process. This could only be done if relevance was the same in the mind of each of there assessors. Conveniently what constitutes a relevant document for a given topic is spelled out in the INEX topic narrative – but it remains open to interpretation.

2. Can the INEX document pool be reduced in size?

INEX uses a round-robin pooling method (called top-n). In this method the top element from each run are collected and the documents from which they come are added to a pool, then the pool is de-duplicated. The process continues for the second element from each run, and so on until eventually the pool contains n (at INEX 2006 n=500) unique documents (see Piwowarski and Lalmas [11] for details).

Investigations into the most appropriate size of n have focused on identifying the remaining number of unidentified relevant documents. Experiments might be conducted to investigate the effect (on relative search engine performance) of reducing the size of the pool. A shallower pool, although leading to a less complete set of relevance assessments, would take less time to assess.

3. How effective are assessments collected with a very short time frame?

The time available to assess at the workshop was limited to one workshop session. If it were possible to reliably assess topics in such a short time frame then many more topics could be assessed in the same time frame. Equally, if the number of topics remained fixed it might be possible to complete the entire assessment task in just a few hours.

3. METHODS OF COLLECTION

In total 41 INEX 2006 workshop attendees participated in the experiment. 15 topics were chosen for re-assessment on the basis that those topics had already been double-judged and further additional assessing of those topics could be used to gain a better understanding of how the concept of relevance crosses a population. There was no order to the manor in which topics were given to assessors. After the assessment process participants answered a short questionnaire containing questions about how they assessed. Table 1 shows the number of assessors that answered questionnaires for each topic.

Table 1: Distribution of topics to assessors

Topic	Assessors	Topic	Assessors
304	3	364	3
310	4	385	3
314	2	403	3
319	2	404	2
321	3	405	3
327	4	406	3
329	3	407	1
355	2	***Total***	***41***

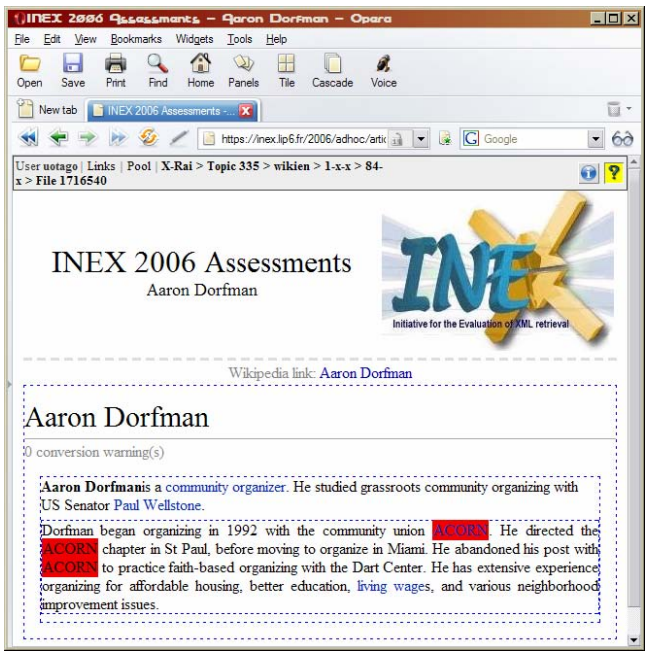


Figure 1: The X-Rai assessment software GUI. In this example the assessor has chosen to highlight keywords.

Assessment was performed using the X-Rai [11] assessment tool built by Benjamin Piwowarski specifically for INEX. Assessors were given a topic to assess, then chose a document from the document-pool to assess, then identified any relevant passages within that document by highlighting them in yellow. Finally they moved on to the next pool document by clicking at the bottom of the window. X-Rai is shown in Figure 1 with the topic keywords highlighted.

The document collection used was the INEX Wikipedia document collection whose details are published elsewhere [5]. Documents were presented to the user in alphabetical order, and not pool order. Presenting out of pool order has the advantage of not

biasing the assessor early (or late) in the experiment – they don't know if the document they are assessing is likely to be relevant or not.

In the at-INEX experiment the time available to assess a topic was limited to one workshop session (1 hours and 20 minutes), but the average time taken to assess a 500 document pool at INEX 2006 was about 7 hours – clearly it was not possible to fully reassess each topic in a single session. To resolve this problem the pool for the topics in Table 1 were reduced to about 100 documents each (that is, the top-n process was stopped after a complete round and 100 or more documents were in the pool).

As a means of ensuring the validity of the INEX pooling software, new and alternate pooling software was developed for the experiment.

4. RESULTS OF COLLECTION

4.1 The Pooling Process

On average these reduced pools contained 135 documents per topic. Comparison with the official pools showed agreement levels between 92% and 100% with a mean of 98%. That is, for example, for topic 405, 124 of the 135 documents (92%) in the reduced pool were in the official pool. For others all documents were, but on average 98% were.

Investigation into why the reduced pools were not a full subset of the official pools revealed a workflow anomaly that it is hoped will be resolved for future INEX rounds.

The workflow model at INEX proceeds thus: Participants submit topics, the organizers select the final topics from those[1], the final topics are released to participants who submit runs for those topics, the pools are generated, the topics judged, then the performance of each run is determined. Participants can submit both official runs and unofficial runs with only the former being included in the pooling and scoring process.

If a participant submits a run that contains errors (such as an invalid document-ID, an element that does not exist, or a malformed XPath) then the entire run is excluded from the pooling process. However, as different software is used to assess performance as that used to generate the pools, such runs are still scored even though they were not included in the pooling process.

The effect of this appears at first to be negligible because one expects malformed runs to be produced by buggy search engines which will perform badly. However this need not be the case. Should a run contain a simple error, but otherwise be well formed, the top documents in the run will not be assessed unless other runs also identify the same documents in their top ranks. If those documents were to be relevant they would continue to be considered non-relevant because they were not assessed. Such a run performs badly not because it fails to identify relevant documents (or elements), but because the results it does identify are never scored.

Exactly this situation occurred during INEX 2006. A run from University of Granada uniquely identified results in the top few rankings for at least one topic and was excluded from the pooling process – but was then later ranked relative to the other runs. It performed badly and the submitter questioned its official score.

The new software used to generate the reduced pools only parsed parts of the run-file necessary to build the pool. If errors did not occur in those parts of the file then the file was used for pooling[2]. The official pools were built from software that parsed the entire file and rejected all files that contained any errors.

Several changes are recommended to the workflow:

It is not possible to determine from a run whether or not it is official. This could be amended by adding one attribute to the inex-submittion tag. Rejected runs could also be marked in a similar way.

Runs can perform badly because their top ranking results are not in the pool. This can be rectified by ensuring that any run rejected from any part of the workflow is rejected from the entire workflow. One possibility is to fully check every run on submission.

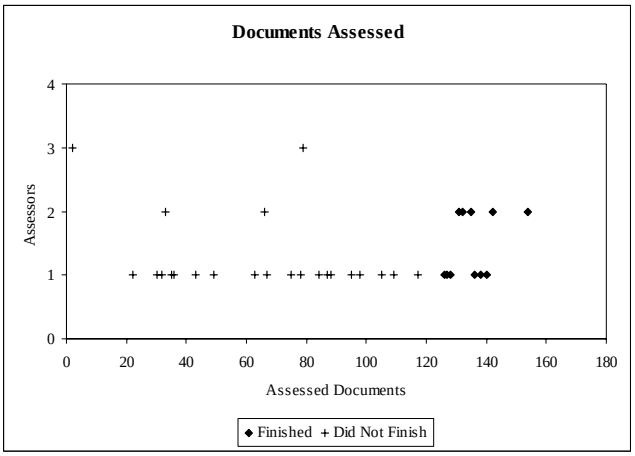


Figure 2: Number of documents assessed in the allotted time

4.2 Workload

Of the participants, 16 completed the assessment task in the allotted time (1 hour and 20 minutes) and 29 did not. Some (4) did not answer the questionnaire. For topics 319 and 355 no assessor completed the task in the allotted time. For topics 314, 327, and 329 two assessors completed the task, for the remainder only one assessor completed the task.

On average 87 documents were assessed in the time period with a minimum of 2 and a maximum of 154 documents assessed. Figure 2 shows the distribution of the number of documents assessed in the time period.

If the relative rank order of the official runs is maintained using the assessments completed in just this short time then the pooling could be stopped at n=100 documents and assessing completed in just an hour and twenty minutes per topic.

A set of assessments for the at-INEX experiment was constructed for the 15 topics by taking the assessment pool with the lowest

[1] All syntactically incorrect, partially completed, and duplicate topics are dropped. All non IR topics are dropped (such as "all papers written by Smith"). There is no formal method other than opinion of the several reviewers.

[2] The error was subtle, some paths in some documents were missing instances. That is, in the file /article[1]/body[1]/emph3 is seen whereas /article[1]/body[1]/emph3[1] was needed.

pool-id for those topics that were complete in the allotted time, and for those that no assessor completed, the pool with the most assessed documents.

A reduced set of official assessments was taken by excluding all topics except the 15 multiple-assessed topics.

The performance of the All-In-Context[3] runs submitted to INEX in 2006 was scored using the INEX assessment tool. The metric MAgP was used. Two scores were generated, one against the reduced assessment set of only 15 topics, the other against the full assessment set, but for only the 15 topics.

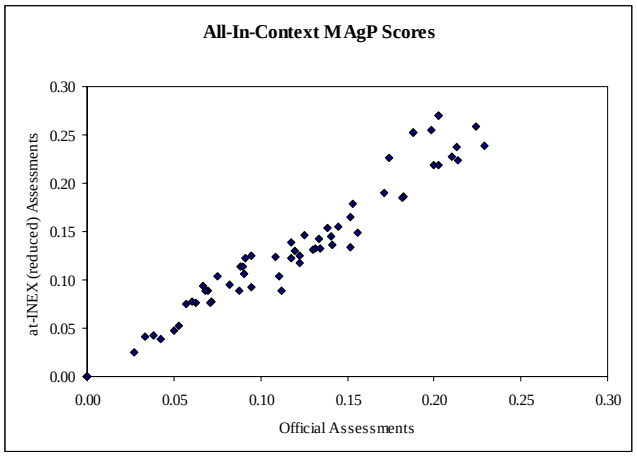


Figure 3: Run performance against the two assessment sets

Figure 3 shows the performance of each run against the two sets of assessments. The average amount of time needed to complete the assessment of a single topic (across all topics, not just the 15) for the official assessments was 6 hours and 51 minutes. The maximum time allotted to the at-INEX assessment task was 1 hour and 20 minutes. The Spearman's rank correlation coefficient for the relative performance of the search engines is 0.97. There is a strong positive correlation of one to the other.

The subset of runs that ranked in the top 10 against either set of assessments contains 13 runs. The Spearman's rank correlation for just those runs is -0.03, that is, there is a very weak negative correlation for the top performers. In an hour and twenty minutes of assessing the top runs can be separated from the bottom runs, but the relative performance of the top runs cannot be determined.

It can be seen from Figure 3 that the performance of a run against the at-INEX assessments is often better than the performance against the official assessments. One possible reason is that the average amount of relevant material per document in the at-INEX assessments is larger than that of the official assessments (1833 vs. 1059) so any fixed length passage from a run is more likely to intersect a relevant passage in the at-INEX set. An alternative and more likely reason is that the number of relevant documents in the at-INEX set is smaller than that in the official set (22 vs. 60) so one point of generalized recall (1/gR) is larger in the at-INEX assessments than in the official assessments.

[3] Trotman *et al*. [16] examined XML-IR use cases and consider this to be the most viable task examined.

5. RESULTS OF QUESTIONNAIRE

5.1 Factors Influencing Assessments

In the questionnaire the assessors were asked to state the factors that helped them to decide what would make a passage relevant. The factors were categorized and in all 14 different categories were found. Some were very idiosyncratic, e.g. "geographical facet" or "discourse", and others were very common. Three factors appeared to be much more influential than the others; titles, content and keywords.

The study showed (see Table 2), not surprisingly, that content was the most important factor. Many (12 assessors) also used keywords collected from the task description, either by system highlighting or browser enabled searching in the article. The third most important factor was the titles of documents, sections and sub-sections which were used by 11 assessors. The most common other factors were context, links, introductory text and bibliographic references, each being indicated by three assessors as being important.

Table 2: Factors affecting passage choice at INEX

Factors	Titles	Content	Keywords	Other
Assessors	11	30	12	21

From studies of information searching behavior (e.g. Barry [1]) it is known that there are many diverse factors influencing readers' relevance judgments of information sources. In fact, Barry's study showed that "every respondent mentioned factors beyond the topical appropriateness of documents during their evaluation" [1]. In the study herein a similar variety of criteria is not seen, perhaps due to the heterogeneity of the information sources, all being Wikipedia articles. Another, and more interesting, theory is that the assessors (all being information scientists) may have a very rationalistic set of factors for determining the relevance. This is in-line with Pharo and Järvelin's [10] findings relating to the relationship between information scientists and end-users perspectives on web information searching, pointing out the mismatch of viewing the searcher as a very rational individual when prescribing information searching procedures when in fact the searcher to a very large degree is looking for satisfaction (see e.g. Prabha *et al*., [13]) during information retrieval.

5.2 Dynamics of Relevance Assessments

This study also examined how the learning effect affected the relevance assessments. It is known from studies of information searching that often searchers will have an unclear formulation of the information need, which may become clearer throughout the search process as they start interacting with potential information sources. Would a similar development be spotted among assessors?

In response to the question of whether they had changed their mind during the assessment process 17 persons said they had changed their mind whereas 22 persons said they had not (two assessors did not answer this question). The main reason given by those who had changed their mind was, in fact, related to the learning process, they had to get acquainted with the topic, the document type or the assessment software. Thus it is seen that a learning effect is also involved in cases where the search task is formulated and where the goal of the information searcher clearly is directed at full recall, which is the case in this type of

experiment. This also suggests that the assessments would benefit from assessor training before assessment. They might even benefit from reassessment of documents judged at an early stage of the assessment process.

5.3 Size of Relevant Passages

Half of the searchers said they preferred to use a standard size for marking relevant passages; the other half disagreed and claimed that the size differed. This might be related to characteristics of the tasks, but a closer inspection of the individual searchers does not reveal any systematic connection between tasks and passage preference. Of the assessors favoring specific sizes the large majority had a preference for small or smallish (e.g. one paragraph) passages.

Table 3: Preference of element size

Elements	**< 1**	**1**	**2**	**3+**
Assessors	25	24	10	15

The assessors were also asked to mark the correspondence of their selected passages to article elements (see Table 3). The results strengthen the notion that searchers prefer smaller elements. More than half of the assessors choose to select passages equivalent to one element (24 assessors) or less of length (25 assessors). More than one out of three assessors, however, chooses to use passages covering more than two elements. Some assessors pointed out that the tasks they performed very much suggested a specific size of passage to be marked.

Assessors were also asked for their preference with respect to Best Entry Points (BEP), i.e. the element they recommended as the best place in the document from which to start reading. Only eight assessors stated a definitive need for more than one good entry point per article. 22 assessors rejected such an option whereas a few stated they sometimes would have liked to add more than one BEP.

6. AGREEMENT LEVELS

6.1 Agreement Levels

Search engine performance is often measured against a gold standard set of assessments produced by a single individual. Wherever possible at INEX the topics are assessed by the original topic author, thus the assessments can be considered the "right answer" in the mind of the person with the information need.

However, Spink *et al*. [14] show that (on the web at least) many queries will be seen repeatedly, and are issued by many different individuals. It is not clear if each individual has the same definition of relevance, and if they do not then how this affects the relative performance of search engines.

Trotman [15] and Pehcevshi and Thom [9] examined agreement levels for whole documents and for elements at various rounds of INEX. They initially showed very low agreement levels. Their work resulted in changes to the assessment methods. These changes in turn resulted in improvements in both the assessor load and agreement levels.

In the at-INEX experiment multiple judges were available to assess each topic. Computing the agreement levels with multiple assessors provides insights into how different users view the relevance of the same documents with respect to the same query. That is, it is possible to identify those parts of the document everyone agrees are relevant and those that only some agree are relevant.

At INEX 2006 assessors identified passages of relevant text using a yellow-highlighting method. From these passages the relevant elements were automatically deduced. To compute the agreement levels for this data it is therefore necessary to examine the passages in the assessment files and not the elements. There are complications.

6.2 Reading INEX Assessment Files

X-Rai [11] produces XML files containing three kinds of assessments: relevant passages, relevant elements, and best entry points. The passages are those parts of a document highlighted by an assessor. The elements are those document elements crossing a relevant passage. The BEPs are separately identified by the assessor. These are grouped by document (file) and stored in separate files for each topic.

Passages are identified in an assessment file in the following way:

```
<passage start="/article[1]/body[1]/section[
14]/normallist[1]/item[25]/text()[1].0" end=
"/article[1]/body[1]/section[14]/normallist[
1]/item[25]/text()[1].19" size="20"/>
```

This passage starts before the 1st character of the first text node of the 25th item of the 1st normallist of the 14th section of the 1st body of the 1st article in the file. The passage is 20 characters[4] (not bytes as the files are encoded in UTF-8) in length and finished after the 19th character of the same text node in the document tree.

Elements are identified in the following way:

```
<element path="/article[1]/body[1]/section[1
4]/normallist[1]/item[25]" exhaustivity="2"
size="34" rsize="20"/>
```

The path is specified in XPath [3]. For 2006 assessments the exhaustivity is redundant, the size is the size of the element and the rsize if the quantity of relevant text. 20 of the 34 characters in the element were highlighted as relevant making the specificity 20/34=0.59.

Best entry points are identified in the following way:

```
<best-entry-point path="/article[1]/body[1]/
section[14]/normallist[1]/item[25]/text()[1]
.0"/>
```

In this instance, the best entry point is before the first character of the same text node identified in the element description above.

6.2.1 Discrepancies

X-Rai requires all text that can be highlighted by an assessor to be in separate leaf nodes of the document tree. Unfortunately, the document collection is not structured in this way so a series of

[4] More accurately it is 19 in length, but the assessments state 20.

simple transformation are applied to the documents before assessment starts. For example[5]

```
<a>some.text<b>.and.some.other.with.spaces.after.</b>...</c>
```

becomes

```
<a><xrai:s>some.text</xrai:s><b>.and.some.other.with.spaces.after....</b></c>
```

Because the assessors are assessing against a transformed document collection and not the original, the assessments do not always match the original document structure. For example, the topic 310 assessments for document 2545650 contain the passage:

```
<passage start="/article[1]/name[1]" end="/article[1]/body[1]/section[4]/section[3]/normallist[1]/item[2]/text()[1].33" size="14581" />
```

The end point is 33 characters into the first text node of the given path. That contains the text (delineating quotes added for clarity):

```
", designed by Richard Loomis"
```

which isn't 33 characters in length (its 28 characters in length). In this case the extra white-space occurring after the element has been included in the element for X-Rai which identified the end of the highlighting as occurring 33 characters into the transformed element.

Another way in which the transformation can cause discrepancies is with passage lengths, the length of a passage can be larger then the amount of text between the start and end points in the original XML files.

Runs submitted to INEX are generated against the original untransformed document collection. Given the assessments can indicate more content per element than exists; it may not be possible to submit a perfect run.

6.3 Agreement Level Algorithm

Assessment discrepancies make it difficult to compute agreement levels for multiple assessors that will agree with future results published by other researchers for the same assessments – unless the algorithms are stated up front. The approach taken for the work described in this contribution is:

For each topic

  For each relevant document

    Load and parse the original XML document

    Replace each character in all text nodes with '0'

  For each assessor

    For each passage

      Locate the start point, start

      Locate the end point, end

      Increment each character between start and end

All end points are truncated to at most the length of the element in which they terminate. In the example above, it would be truncated at 28 characters.

6.4 Assessment Subset

Not all assessors completed the assessment task. The assessments from those assessors who completed less that 50% of the task were discarded from the analysis.

As the results from the INEX 2007 double-assessment experiment were also available they were included in the analysis, as were the official INEX assessments.

The assessors of these two sets did not all assess the same documents for two reasons: first, different pools were used; second, some assessors did not complete the task. The documents used in the analysis were those that all assessors assessed. Table 4 shows the number of assessors per topic and the number of documents they all assessed on common for that topic. In total 60 assessors assessed 1,471 documents across 15 topics (an average of 98 documents and 4 assessors per topic).

Table 4: Pool sizes and number of assessors used for analysis

Topic	Documents	Assessors
304	135	3
310	91	4
314	130	4
319	78	4
321	132	3
327	78	5
329	86	5
355	83	3
364	56	5
385	87	4
403	113	4
404	104	4
405	99	4
406	67	5
407	132	3
Total	**1,471**	**60**

6.5 Results

Figure 4 shows the mean number of documents considered relevant as the number of assessors is increased. As a different number of assessors assessed each topic several lines are shown, each being the mean of only those with at least *m* given assessors where *m* is the number of points on the line (that is, for the line with 3 points all topics were used to generate the means).

[5] This example is lifted directly from private communication with B. Piwowarski.

The figure shows that as the number of assessors is increased from 1 to 5 the assessors continue to find further relevant documents (the union increases). It also shows that the number documents they all agree are relevant decreases (the intersection decreases).

Taking the case where the number of assessors was 4, and fitting a logarithmic trend line to the intersection-curve resulted in an R-squared of 0.991. This line was extrapolated and it crosses the x-axis at 19 assessors. That is, if the trend continued then with 19 assessors there would be no single document that all assessors agree relevant to any information need. Fitting a logarithmic line to the union, R-squared value is 0.999 and at 19 assessors 33 documents would be identified.

Figure 5 shows the mean number of characters of text that are considered relevant as the number of assessors is increases. A similar pattern to that of documents is seen. Fitting logarithmic lines to the intersection and union (of 4 assessors) resulted in an R-squared of 0.958 and 0.997 respectively. Extrapolating to 8 assessors and there are no characters in common, but a total of 64,167 characters of relevant content.

Even though the 8 assessors would not agree on relevant content within a document they will all agree that some documents are relevant. Care should be taken with this conclusion because of the inherent inaccuracy of extrapolating such a small number of points over such a long distance.

Pehcevski [8] reports that in the INEX 2005 interactive experiment participants agreed at the extreme ends of the old INEX multi-grade relevance scale (i.e. highly-relevant and not-relevant) but not in the middle. A similar result can be seen in the Cystic Fibrosis collection [2]. Figure 4, Figure 5, and the extrapolations strongly suggest that relevance is in the mind of the assessor and not a universal truth. These two results are not contradictory – assessors can, in general, agree where within a document the relevant content is found even if some don't.

If, indeed, each assessor continues to identify new relevant documents, and there is no one document that every assessor agrees is relevant then it is not clear that Cranfield experiments are meaningful for XML-IR and Passage Retrieval. Further investigation is required.

For the 4 topics that have 5 assessors, Figure 6 shows the proportion of the union that was identified by at least 1, 2, 3, 4, and 5 assessors (mean over all possible assessor groupings). In each case a decrease is seen suggesting that each time a new assessor is added, they will disagree on an otherwise commonly held belief.

Figure 7 shows the same but for documents. Particular note should be taken of topic 327 in which four assessors agree on a relevant document, but not where within that document the relevant material can be found. This is exactly as predicted by Figure 4 and Figure 5.

7. CONCLUSION

This investigation examined the first at-INEX experiment and reports on the results. Several methodological problems were encountered and suggestions made to tighten the practice.

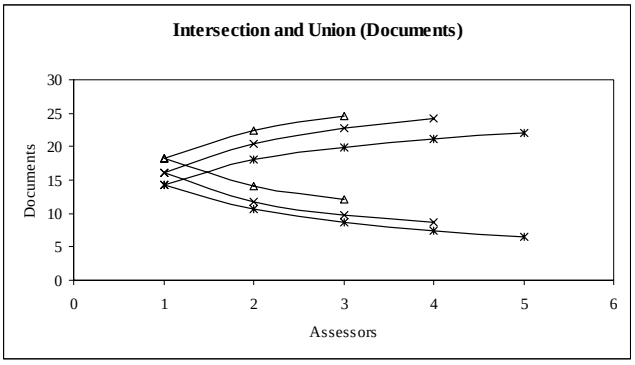


Figure 4: Cross-assessor intersection and union of documents

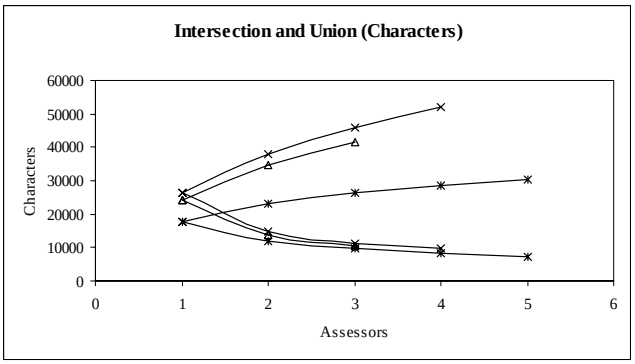


Figure 5: Cross-assessor intersection and union of characters

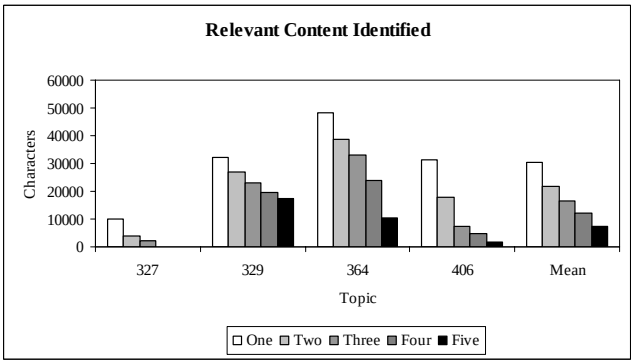


Figure 6: Decreasing agreement of relevant text as the number of assessors increases

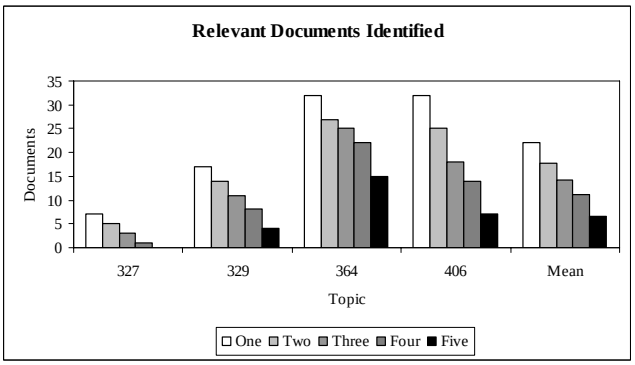


Figure 7: Decreasing agreement of relevant documents as the number of assessors increases

A program was written to generate pools for assessment, that program identified a different set of documents that those in the official pool. This was because the runs included in the pool were different for each program. In turn this is because it is not possible to know from a given run if, or not, that run is official or not. It is also not possible to know if it was rejected for some reason.

Changes to the submission and acceptance process are recommended: Official runs should be marked as such within the run. To avoid confusion runs should be verified at the submission point and no run should be accepted if it is possible for it to later be rejected.

The number of documents assessed in the allotted time period varied greatly – that is, we can't agree how many documents can be assessed in an hour and twenty minutes. The mean was 87 documents.

When examining the assessments, the passages in the assessment files did not match the elements in the documents. This was because changes had been made to the original documents in order to use the X-Rai assessment tool. It is not clear how this effects the assessment overall (further investigation is required). This problem might be rectified in two possible ways. First, the translation to move all content into leaf node might be changed to avoid moving the relative location of text (even though it is just white-space). Second, the leaf-node requirement might be removed from the assessment tool.

Measuring the performance of each search engine against the two sets of assessments showed a strong positive correlation for the runs, but a weak correlation for the top performing runs. In answer to questions 2 and 3 in Section 2, the assessments collected over one hour and twenty minutes (per topic) are effective at separating good from bad runs, but in order to separate good from very good runs the pool cannot be reduced in size (to about 100 documents).

To answer to question 1 in Section 2, the agreement level of assessors was measured as the number of assessors was increased. Only about 8 assessors are needed before they stop agreeing which parts of a document are relevant, but 19 assessors are needed before they disagree on which documents are relevant. Relevance is in the mind of the assessors and assessors do not agree with each other.

When deciding on relevance, assessors do not agree on which factors are important, some think the content, while others think titles and keywords. The size of a relevant passage also varies across assessors with some identifying whole elements as relevant and others non-elemental passages.

It is pertinent to ask if we can at least agree on something. In answer: yes. We agree which runs performed well even though we don't agree on how we decided this.

8. ACKNOWLEDGEMENTS

Funded in part by a University of Otago Research Grant.

9. REFERENCES

[1] Barry, C. L. (1994). User-defined relevance criteria: An exploratory study. *Journal of the American Society for Information Science*, 45(3), 149-159.

[2] Berkeley. (2005). Cystic fibrosis reference collection. Available: http://www.sims.berkeley.edu/~hearst/irbook/cfc.html [2005, 25 February].

[3] Clark, J., & DeRose, S. (1999). XML path language (XPath) 1.0, W3C recommendation. The World Wide Web Consortium. Available: http://www.w3.org/TR/xpath.

[4] Clarke, C. (2005). Range results in XML retrieval. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology, Second Edition*, (pp. 4-5).

[5] Denoyer, L., & Gallinari, P. (2006). The wikipedia XML corpus. In *Proceedings of the INEX 2006 Workshop*.

[6] Malik, S., Tombros, A., & Larsen, B. (2006). The interactive track at INEX 2006. In *Proceedings of the INEX 2006 Workshop*.

[7] Malik, S., Trotman, A., Lalmas, M., & Fuhr, N. (2006). Overview of INEX 2006. In *Proceedings of the INEX 2006 Workshop*.

[8] Pehcevski, J. (2006). Relevance in XML retrieval: The user perspective. In *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, (pp. 35-42).

[9] Pehcevski, J., & Thom, J. A. (2005). HiXEval: Highlighting XML retrieval evaluation. In *Proceedings of the INEX 2005 Workshop*.

[10] Pharo, N., & Järvelin, K. (2006). "irrational" searchers and IR-rational researchers. *Journal of the American Society for Information Science and Technology*, 57(2), 222-232.

[11] Piwowarski, B., & Lalmas, M. (2004). Providing consistent and exhaustive relevance assessments for XML retrieval evaluation. In *Proceedings of the 13th ACM conference on Information and knowledge management*, (pp. 361-370).

[12] Piwowarski, B., Trotman, A., & Lalmas, M. (2007 (un-published)). Sound and complete relevance assessments for XML retrieval.

[13] Prabha, C., Connaway, L. S., Olszewski, L., & Jenkins, L. R. (2007). What is enough? Satisficing information needs. *Journal of Documentation*, 63(1), 74-89.

[14] Spink, A., Wolfram, D., Jansen, B. J., & Saracevic, T. (2001). Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 53(2), 226-234.

[15] Trotman, A. (2005). Wanted: Element retrieval users. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology, Second Edition*, (pp. 63-69).

[16] Trotman, A., N.Pharo, & Lehtonen, M. (2006). XML-IR users and use cases. In *Proceedings of the INEX 2006 Workshop*.

[17] Voorhees, E. M. (2001). The philosophy of information retrieval evaluation. In *Proceedings of the The Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, (pp. 355-370).

[18] Voorhees, E. M. (2003). Overview of TREC 2003. In *Proceedings of the 12th Text REtrieval Conference (TREC-12)*.

To Click or not to Click? The Role of Contextualized and User-Centric Web Snippets

Nikos Zotos
Computer Engineering Dept
Patras University, Greece
zotosn@ceid.upatras.gr

Paraskevi Tzekou
Computer Engineering Dept
Patras University, Greece
tzekou@ceid.upatras.gr

George Tsatsaronis
Athens University of Economics and Business, Greece
gbt@aueb.gr

Lefteris Kozanidis
Computer Engineering Dept
Patras University, Greece
kozanid@ceid.upatras.gr

Sofia Stamou
Computer Engineering Dept
Patras University, Greece
stamou@ceid.upatras.gr

Iraklis Varlamis
Athens University of Economics and Business, Greece
varlamis@aueb.gr

ABSTRACT
When searching the web, it is often possible that there are too many results available for ambiguous queries. Text snippets, extracted from the retrieved pages, are an indicator of the pages' usefulness to the query intention and can be used to focus the scope of search results. In this paper, we propose a novel method for automatically extracting web page snippets that are highly relevant to the query intention and expressive of the pages' entire content. We show that the usage of semantics, as a basis for focused retrieval, produces high quality text snippet suggestions. The snippets delivered by our method are significantly better in terms of retrieval performance compared to those derived using the pages' statistical content. Furthermore, our study suggests that semantically-driven snippet generation can also be used to augment traditional passage retrieval algorithms based on word overlap or statistical weights, since they typically differ in coverage and produce different results. User clicks on the query relevant snippets can be used to refine the query results and promote the most comprehensive among the relevant documents.

Categories and Subject Descriptors
H.3.3 [**Information Search and retrieval**]: Selection Process, Information Filtering; H.3.1 [**Content Analysis and Indexing**]: Linguistic Processing; H.3.4 [**Systems and Software**]: Performance Evaluation (efficiency and effectiveness).

General Terms
Algorithms, Performance, Experimentation.

Keywords
Web passage retrieval, semantic similarity, coherence.

1. INTRODUCTION
The advent of the web has brought people closer to information than ever before. Web search engines are the most popular tool for finding useful information about a subject of interest.

SIGIR 2007 Workshop on Focused Retrieval
July 27, 2007, Amsterdam, The Netherlands

What makes search engines popular is the straightforward and natural way via which people interact with them. In particular, people submit their requests as natural language queries and they receive in response a list of URLs that point to pages which relate to the information sought. Retrieved results are ordered in a way that reflects the pages' importance or relevance to a given query. Despite, the engines' usability and friendliness, people are oftentimes lost in the information provided to them, simply because the results that they receive in response to some query comprise of long URL lists. To fill this void, search engines accompany retrieved URLs with snippets of text, which are extracted either from the description meta-tag, or from specific tags inside the text (i.e. title or headings).

A snippet is a set of (usually) contiguous text, typically in the size of a paragraph, which offers a glimpse to the retrieved page's content. Snippets are extracted from a page in order to help people decide whether the page suits their information interest or not. Depending on their decisions, users might access the pages' contents simply by clicking on their URLs (retrieved by the engine) or ignore them and proceed with the next bunch of results.

Most up-to-date web snippet generation approaches extract text passages[1] with keyword similarity to the query, using statistical methods. For instance, Google's snippet extraction algorithm [1] uses a sliding window of 15 terms (or 100 characters) over the retrieved document to generate text fragments in which it looks for query keywords. The two passages that show up first in the text are merged to produce the final snippet. However, statistically generated snippets are rough indicators of the query terms co-occurring context but, they lack coherence and do not communicate anything about the semantics of the text from which these are extracted. Therefore, they are not of much help to the user, who must decide whether to click on a URL or not.

Evidently, if we could equip search engines with a powerful mechanism that generates self-descriptive and document expressive text snippets, we could save a lot of time for online information seekers. That is, if we provide users with that piece of text from a page that is the most relevant to their search intention and

[1] We use the terms snippet and passage interchangeably to denote the selection of small size text from the full content of a document.

which is also the most representative extract from the page, we may assist them decide whether to click on the page or not.

In this paper, we propose a snippet selection technique, which relies on the implicit query semantics rather than the query terms and on the snippets semantic information rather that on the statistical distribution of terms within the text. Our technique focuses on selecting ***coherent***, ***query-relevant*** and ***expressive*** text fragments, which are delivered to the user and which enable the latter perform focused web searches. At a high level our method proceeds as follows:

- It takes as input a query and uses a number of semantic resources (thesauri, ontologies, etc.) in order to assist the user in determining the query intention. This practically translates into offering the user the means to annotate search terms with the appropriate sense (always specified in the query context).
- Given the disambiguated query intention and a set of results that correlate to the underlying intention, it identifies within the text of a page, the fragment that is the most relevant to the semantics of the query.
- Query-relevant text snippets are then evaluated in terms of their lexical elements' coherence, their importance to the semantics of the entire page and their closeness to the query intention.
- Snippets that exhibit the strongest correlation to both the query and the page semantics are presented to the user.

After applying our snippet selection approach to a number of searches, we conclude that retrieved snippets determined by the semantic correlation between snippets and queries yield improved accuracy compared to the snippets that are determined by using only the statistical distribution of query keywords in the pages' snippets. In brief, the contributions of this article are as follows:

- A measure of the snippet's closeness to the query intention (usefulness). In our work, a useful snippet is the text fragment in a retrieved page that exhibits the greatest terminological overlap to the query keywords and which is also semantically closest to the query intention.
- A measure of the importance and representativeness of a snippet against the entire document from which it derived. Our measure adheres to the semantic cohesion principle and aims at identifying the query focus in the search results.
- A combination of the above measures, in order to assist the user in performing comprehensive and focused web searchers in two ways: with and without clicking on the retrieved results. Without clicking on the results, the user can view the particular text fragment in the page that best matches her search intention. By clicking on a snippet, the user's focus is directly driven to the exact text fragment that contains relevant information to her search intention. In particular, query-relevant text fragments appear highlighted so that the user gets instantly the information that she wants without the need to go through the contents of a possibly long document.

The paper is organized as follows. We begin our discussion with a detailed description of our semantically-driven approach in snippets' selection. Then in Section 3, we experimentally study the effectiveness that our snippet selection approach has in focused retrieval and we discus obtained results. In Section 4 we review related work and we conclude the paper in Section 5.

2. MINING QUERY-RELEVANT AND TEXT-EXPRESSIVE SNIPPETS

It is common knowledge that web users decide on which results to click based on very little information. Typically, in the web search paradigm, information seekers rely on the retrieved page's title, URL and text snippet that contains their search keywords to infer whether the page is of interest to their search pursuit or not.

Although, the titles given to web pages are greatly representative of their content, the text snippets of the search results might oftentimes be misleading and communicate incomplete information about the pages' semantic content. This is essentially because titles are created manually, whereas web snippets are automatically generated by the search engine modules on the sole ground that they contain the query keywords.

Evidently, decisions based on little information are susceptible to be bad decisions. A bad decision is encountered when the user clicks on a link misguided by a title or a text snippet, which is of little relevance to the linked page's contents. In an analogous manner, a bad decision might be when the user decides not to click on the link to a *good* page simply because the text snippet of the page is poor or seems unrelated to her query intention.

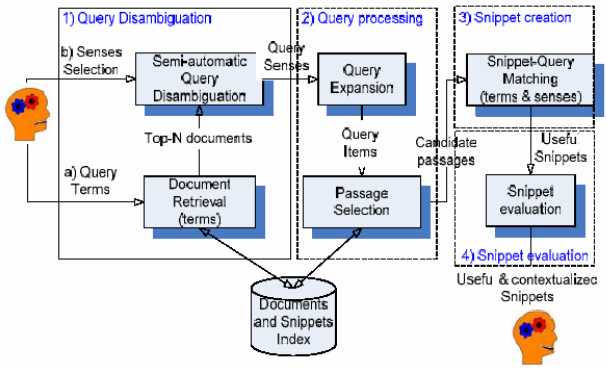


Figure 1. Snippet selection process

In this section, we present our approach towards the automatic extraction of query-relevant and document-expressive snippets, in the hope of assisting web information seekers make informative decisions about whether to click on a retrieved result or not. The basic steps of our approach, as depicted in, are:

1) Disambiguation of the query intention, with automatic, semi-automatic or completely manual methods.
2) Semantic similarity matching between query and text passages using both terms and implied concepts (candidate passages).
3) Creation of query-similar snippets from the document (useful snippets).
4) Evaluation of the selected snippet's expressiveness to the document contents.
5) Presentation of the query-relevant and text-expressive snippets to the user.

We begin our discussion, with a brief description of our approach towards the identification of the query intention (step 1). Then, in Section 2.2 we describe our semantically-driven approach for extracting candidate text nuggets from a query matching page (step 2) and selecting those that are semantically closest to the query intention (step 3). In Section 2.3, we introduce a novel

method for evaluating how expressive or else representative is a query-relevant text fragment to the entire content of the page from which it derived (step 4). Finally, in Section 2.4, we discuss how we can put together the derived information about the text nugget that is the most useful to the query intention and also expressive of the document's content in order to assist web users perform focused web searches.

2.1 Identifying the Query Intention

A number of studies have shown that a vast majority of queries to search engines are short and under-specified [13]. Moreover, short keyword queries are inherently ambiguous in the sense that the same query might intend the retrieval of distinct information sources. Although, the problem of query sense detection is not new, nevertheless the challenge of deciphering the intention of a query still remains.

In our work, we attempt the semi-automatic identification of the query intentions based on the semantic analysis of the query matching pages [14]. In particular, we rely on the top N (N=20) pages retrieved for a query, we parse them to remove html markup, we tokenize, POS-tag them, remove their stop-words and we utilize them as a small Web corpus of query co-occurrence data against which query sense resolution is attempted. The first step towards query sense disambiguation concerns the mapping of all content terms[2] inside every page to WordNet [16] nodes. The corresponding query senses that relate (in WordNet) to any of the senses of the page's content terms are candidate senses for describing the query intention.

For instance, assume that query q has 4 senses in WordNet, say s_1, s_2, s_3 and s_4, and that a query matching page P has a number of terms t_1, t_2, …, t_n with senses $t_1\{s_1, s_2)$, $t_2\{s_1,s_2,s_3\}$, …. $t_n\{s_1,s_2\}$. To identify which of the 4 query senses is attributed to q in the contents of P, we examine which senses of q relate in WordNet to any of the senses of $t_1,t_2,\ldots t_n$. We then take the query senses that relate to any of the page terms' senses and present them to the user in order to select which of the displayed senses is the most suitable for describing her information need.

In particular, assuming that query sense s_1 relates to some sense of t_1, query sense s_3 relates to some sense of t_1 and query senses s_2 and s_4 do not relate to any of the senses of the terms in P, our approach picks the query senses s_1 and s_3 and displays them to the user as candidate senses for describing the query intention in the context of P. The user implicitly indicates the intention of the query, by picking among the candidate concepts, those that she deems the most suitable to express her query semantics. By relying on the user for the final selection of a suitable query sense, we ensure that the query intention is accurately disambiguated and therefore it can successfully participate in the snippet selection process.

Before we proceed with the description of how the identified query sense participates in the snippet selection process, we should stress that our method on snippet selection is not bound to a particular query sense resolution method. Consequently, it can be fruitfully combined with any query disambiguation technique that one would like to use. For an overview of the different similarity metrics employed for word sense resolution see [37].

[2] Content terms are nouns, proper nouns, verbs, adjectives and adverbs.

2.2 Semantic Selection of the Best Snippet for a Query

Having identified the query semantics or else the query intention, we now turn our interest to the description of how this knowledge can be exploited in the selection of those document fragments that are semantically closest to the query intention.

The process begins with the expansion of the disambiguated set of query terms with their synonyms in WordNet. This ensures that text fragments containing terms that are semantically equivalent but superficially distinct to the query terms, are not neglected in the snippet selection process. The snippet selection that follows finds all the appearances of the original query terms and their synonyms (query items in) within the retrieved page. Upon identification of query matching items in the page's text, we define a window size of 20 words (see [17]) around the identified query items and we extract all the passages that contain any of the items of the expanded query set. All the extracted passages are candidate snippets with respect to the considered query.

To identify within a query relevant page those text snippets that better match the query intention, our method combines (i) the ***terminological overlap*** (expressed by the relevance measure) and (ii) the ***semantic correlation*** (expressed by the quality measure) between the query and snippet sets of concepts.

The ***terminological overlap*** between the query and a snippet is, in rough terms, the intersection of the two item sets; given that all snippets have similar size and that the items in both sets have been mapped to WordNet nodes. The normalized terminological overlap between a query and a passage, which is determined by the fraction of the passage's terms that have a semantic relation[3] in WordNet to the query sense, indicates the ***relevance*** that a passage has to the query intention and it is formally given by:

$$\text{Relevance}(q, p) = \frac{\sum_{j=1}^{k} qr \bullet \text{Tf} / \text{IDF}(t_j, p)}{qs \bullet \sum_{i=1}^{n} \text{Tf} / \text{IDF}(t_i, p)}$$

Where k is the number of terms in passage p that relate to at least one term in the query, n is the total number of terms in the passage, qr is the number of query terms to which the passage term t_j relates (query relevant terms) and qs is the number of terms in the query (query size). Finally, $\text{Tf/IDF}(t_x,p)$ denotes the importance of term t_x in passage p as this is determined by their cosine similarity in the vector space model. Passages containing terms that relate to the sense of the query keywords are deemed to be query relevant. However, this is not sufficient for judging the quality or the usefulness that the candidate passages have to the query intention.

To ensure that only good quality passages will participate in the snippet to be extracted from a query matching page, we semantically correlate the expanded query and the query-relevant passage. The query-passage term similarity metric is based on the Wu and Palmer similarity metric [15], which combines the depth

[3] Out of all the WordNet relation types, in our work we employ: direct hypernymy, (co-)hyponymy, meronymy and holonymy, as indicative of the query-passage terminological relevance.

of paired concepts in WordNet and the depth of their least common subsumer (LCS), in order to measure how much information the two concepts share in common. According to Wu and Palmer the similarity between a query term q_i and a passage term S_k is given by:

$$\text{Similarity}\,(q_i, S_k) = \frac{2 * \text{depth}\,(\text{LCS}\,(i,k))}{\text{depth}\,(i) + \text{depth}\,(k)}$$

The average similarity between the query and the passage items indicates the ***semantic correlation*** between the two. The query passage semantic correlation values, weighted by the score of their relation type (r) that connects them in WordNet, quantifies the quality of the selected passage. Formally, the ***quality*** of a passage S containing n terms to some query q containing m terms is:

$$\text{Quality}(S, q) = \frac{1}{n \times m} \sum_{j=1}^{m} \{ \sum_{k=1}^{n} [\text{Similarity}\left(q_j, S_k\right) \bullet \text{RelationWeight}(r)] \}$$

where, RelationWeights(r) have been experimentally fixed to 1 for synonymy, 0.5 for hypernymy and hyponymy and 0.4 for meronymy and holonymy, based on the relation weight values introduced in [18]. The final step towards the identification of the best text nuggets within a query matching page, is to compute the degree to which a candidate passage makes a useful snippet to the user issuing a query and receiving a list of answers in the form of page URLs and accompanying text fragments. In measuring the usefulness that a candidate snippet has to some query, we rely on the combination of the snippet's relevance and quality to the query intention. Formally, the usefulness of a snippet S to a query q is:

$$\text{Usefulness}\,(S, q) = \text{Relevance}\,(q, S) \bullet \text{Quality}\,(S, q)$$

Following the steps described above, in the simplest case, we select from a query matching page the text passage that exhibits the greatest usefulness value to the query intention, as the best snippet to accompany the page retrieved for that query. In a more sophisticated approach, we could select more than one useful passages and merge them in a coherent and expressive snippet.

2.3 Towards Coherent and Expressive Text Snippets

Having presented our approach towards selecting query–relevant text snippets, we now proceed with the qualitative evaluation of our selection. The aim of our evaluation is to ensure that the snippets presented to the user are both coherent and text-expressive. By coherent, we mean that the selected snippet should be well-written and meaningful to the human reader, whereas by text-expressive we mean that the selected snippet should represent the semantics of the entire document in which it appears.

Snippet coherence is important in helping the user infer the potential usefulness of a search result before she actually clicks on that. Snippet expressiveness is important after the user clicks on a snippet, since it guarantees that the snippet is in accordance to the target page. Given that our passage selection method operates upon the semantic matching between the query intention and the snippet terms, the evaluation of a snippet's coherence focuses on semantic rather than syntactic aspects. That is, in our evaluation we measure the degree to which terms within the snippet semantically relate to each other. To evaluate semantic coherence of a selected snippet, we map all its content terms to WordNet nodes. Thereafter, we apply the Wu and Palmer similarity metric (cf. Section 2.2) in order to compute the degree to which snippet terms correlate to each other. Based on the average paired similarity values between snippet terms, we derive the degree of the in-snippet semantic coherence as:

$$\text{Coherence}(S_1) = \frac{1}{n} \sum_{i,\,j=1}^{n} \arg\max_{w_j} \text{similarity}(w_i, w_j)$$

where Coherence denotes the ***in-snippet semantic correlation*** of terms n in snippet S_1. Since the appropriate senses for words w_i and w_j are not known, our measure selects the senses which maximize Similarity (argmax similarity(w_i, w_j)).

Measuring semantic coherence amounts to quantifying the degree of semantic relatedness between terms within a passage. This way, high in-snippet average similarity values yield semantically coherent passages. Semantic coherence is a valuable indicator towards evaluating the degree to which a selected passage is understandable by the human reader. However, even if a passage is semantically coherent, there is no guarantee that the information it brings is expressive of the entire document content.

Snippet ***expressiveness is*** the degree to which a selected passage is expressive of the entire document's semantics. For modeling the text-expressiveness of a selected passage we want to compute the terminological overlap and the semantic correlation between the selected passage and the rest of its source text. Our computational model is analogous to the query-snippet usefulness metric with the only difference that in our evaluation we compare passages rather that words.

More specifically, we take all the content terms inside a document (passage content terms included), we map them to their corresponding WordNet nodes and we define the *Expressiveness* of a snippet (S_1) in the context of document D as follows:

$$\text{Expressiveness}\left(S_1, \left(D - S_1\right)\right) = \text{Usefulness}\left(S_1, \left(D - S_1\right)\right)$$

where Usefulness (S_1,(D-S_1)) denotes the product of (i) the terminological overlap (i.e. Relevance) between the terms in the selected snippet and the terms in the remaining source document (i.e. D-S1) and (ii) the average semantic correlation between the passage and the remaining text items, weighted by their Relation (r) type.

Based on the above formula, we evaluate the level of expressiveness that a selected passage provides to the semantics of the entire text in which it appears. The expressiveness of a snippet increases with the number of semantically related terms between the snippet and the rest of the text in its source document. The combined application of the snippet coherence and expressiveness metrics gives an indication on the contribution of a snippet in conveying the message of a document retrieved in response to a user query.

2.4 Snippet-Driven Focused Retrieval

So far, we have described our technique for selecting, from a query matching document, the fragments that are semantically close to the query intention. Moreover, we have introduced qualitative measures for assessing how comprehensive is the selected

snippet to the human reader and how indicative it is of the entire document semantics.

We now turn our attention on how we can put together the criteria of usefulness; semantic coherence and text expressiveness, in order to assist users perform focused web searches. The foremost decision is to ***balance the influence of each criterion*** in our final decision on the best snippet. In other words, we need to decide whether a query-useful snippet should be valued higher than a semantically coherent or text expressive snippet and vice versa.

Apparently, the weight that could or should be given to each of the individual scores cannot be easily determined and even if this is experimentally fixed to some threshold, it still bears subjectivity as it depends on several factors such as the characteristics of the dataset, the user needs, the nature of the query and many other. Whatever the reasons and whichever the objectives for weighting the individual scores within a single formula, in the course of this study we let the final decision on the user, who can apply her own evaluation criteria for selecting which snippet will be displayed for a retrieved document. An approach is to present multiple snippets from each document in the query results (i.e. the best snippet when accounting only one criterion each time) and consequently exploit user feedback to conclude on how users perceive the contribution that different values have on snippet-driven retrieval performance. Based on the users' implicit feedback on what makes a good snippet, we could determine the most appropriate weighting scheme for each user [23].

Another critical issue, concerns the ***visualization of the selected snippets*** to the end user. We claim that it would be useful to highlight the query terms and their synonyms inside the selected snippets, so that the users can readily detect their search targets. Moreover, it would be convenient that text passages are clickable and upon their selection they direct the user to the query relevant snippet rather than the beginning of the document. This way, we can take off the user the burden of reading through the entire document until she detects the information that is most relevant to her query intention. The snippet selection process can be enriched by merging together snippets from multiple documents and by presenting the merged snippet to the user as an extended answer to her information interests.

In overall, deciding on what makes a good snippet for a particular user information need is a challenge that leaves ample space for discussion, experimentation and evaluation. Next, we present an experimental study that we conducted in order to validate the contribution of our snippet selection method in focused retrieval performance and we discuss experimental results.

3. EXPERIMENTS

To validate the usefulness of our snippet selection algorithm in focused retrieval, we conducted two distinct yet complementary experiments. In one experiment we evaluate the performance of our snippet selection algorithm in delivering query useful snippets, and in the second experiment we evaluate how users perceive the query usefulness, the coherence and the text expressiveness of the passages retrieved by our approach.

3.1 Experimental Setup

In our study, we compared our semantically driven passage retrieval algorithm against a baseline passage retrieval algorithm. Building upon the machinery of the previous sections, we automatically disambiguated a set of snippets and measured the improvement of incorporating the Usefulness and the Coherence semantic pieces of information into the text retrieval task against a standard baseline.

More specifically, following a similar experimental framework with the one described in [26] we compared the term TF/IDF vector space retrieval model against a retrieval technique utilizing manually disambiguated queries along with the automatically disambiguated snippets set. In our experiment we exploited existing knowledge on the snippets' relevance to their corresponding queries and we evaluated the Usefulness and the Expressiveness of the passages selected by our algorithm.

To quantitatively evaluate the performance of our passage retrieval algorithm, we have employed the NPL data collection [36] as a testbed. NPL contains 93 experimental queries and a total of 11,429 short documents. Out of all the NPL documents and queries we selected a total of 30 queries and their respective 10,737 relevant documents that we processed as described in Section 2.1 and we indexed them in a local SQL 2005 server. Although, NPL provides a well-structured collection of queries and relevant documents and as such it may not be representative enough of the web data, nevertheless it provides a gold standard collection for running preliminary experiments and evaluate the feasibility of our method. Another reason for employing the above dataset is that NPL documents are quite short (i.e. they contain on average 23 terms) and as such they approximate snippets' size. Moreover, the NPL queries vary in size (i.e. between 2 and 9 terms) and constitute partially formed questions rather than mere keywords. As such they are convenient for a passage retrieval experiment.

In the course of our study, we have semi-automatically annotated each of the 30 experimental queries with an appropriate WordNet sense that represents the query intentions. Moreover, we have annotated every word inside all NPL documents with an appropriate WordNet sense through the exploitation of the Wu and Palmer similarity metric. Based on the selected collection of queries, followed by the given gold standard relevant documents, we evaluated the effectiveness of our snippet selection approach in delivering query useful and text expressive snippets.

3.2 Query Useful Passages

This experiment aims at comparing our **semantically-driven passage retrieval algorithm**, which computes a query useful passage based on the semantic correlation between the query and the passage terms, against the baseline generated by the term TF/IDF vector space representation of the snippets and the use of cosine for query to snippet similarity. For our comparison, we measure the efficiency of the two algorithms in delivering query useful snippets, which practically translates into comparing the Relevance and Quality values of the snippets retrieved by each of the algorithms for the respective queries. To enable our comparison, we formulate the NPL collection as follows: We merge all NPL documents together into a huge virtual document. This document can answer all queries in our dataset. Every individual NPL document forms a candidate passage into the virtual document, which can answer each of the experimental queries. Given that we know in advance which passage of the huge document (i.e. the entire NPL collection) answers each query, our evaluation proceeds as follows. We employ the baseline algorithm and our semantically-driven algorithm, which combines the snippets'

relevance and quality values (i.e. usefulness) and we give scores for each passage. Furthermore, we combine the baseline query to snippet similarity with the computed semantic similarity when the retrieved document reports a Coherence value larger than the average snippet coherence in the collection. We compare the 3 metrics by drawing the interpolated standard 11 precision-recall point curves.

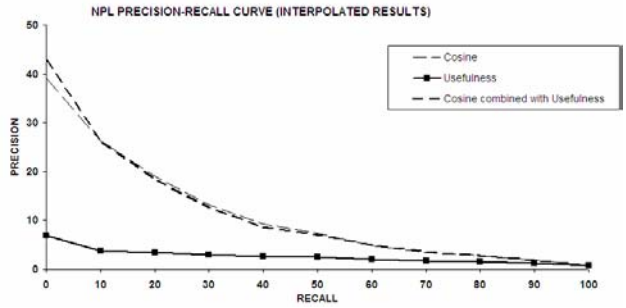


Figure 2. Performance of query-useful passage retrieval.

Obtained results indicate that our proposed semantic similarity measures and more specifically the incorporation of usefulness into the query to snippet similarity measure when the snippet coherence is high can aid the text retrieval task. We show that when the usefulness measure for semantically coherent snippets is applied, an improvement of almost up to 3.5% (see Figure 3) can be achieved even by the top three standard precision recall points.

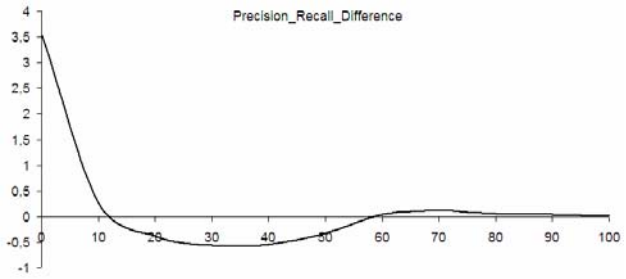


Figure 3. Performance improvement using semantics.

Although the retrieval improvement is quite low and therefore might be perceived as statistically insignificant, nevertheless we believe that a 3.5% improvement over a well structured and manually annotated data collection will significantly increase when a semi-structured un-annotated dataset is considered. As such we claim that the improvement our method can achieve in the context of the web retrieval will be much higher that the one obtained in a small and well-balanced document collection.

3.3 Impact of Passage Selection Criteria

Having accumulated perceptible evidence on the effectiveness that our semantically-driven passage selection approach has on retrieval performance, we carried out a blind user study in order to evaluate the impact that the different snippet selection criteria have on human judgments. For our study, we employed the 30 NPL queries and their relevant documents to which we applied our snippet selection algorithm three times.

In the first run, we parameterized our algorithm so that it selects from a document the text nugget that is the most useful to the query intention. That is, we applied our Usefulness metric (cf. Section 2.2) to each of the query relevant documents in order to extract from every document the text nugget that is semantically closest to the query semantics. In the second run, we parameterized our algorithm so that it selects from a query relevant document the text fragment that is the most coherent. That is, we applied our semantic Coherence metric (cf. Section 2.3) to each of the query relevant documents in order to extract from every document the piece of text that exhibits the maximum in-snippet semantic correlation. Finally, in the third run, we parameterized our algorithm so that it selects from a query relevant document, the passage that is the most expressive of the documents' semantics. That is, we applied our Expressiveness metric (cf. Section 2.3) to each of the query relevant documents in order to extract from every document the piece of text that most accurately captures the entire document's content.

As a baseline snippet selection technique, we relied on the Alicante passage retrieval algorithm [26] in order to determine a query relevant snippet from each of the query matching documents.

Based on the snippets derived by the baseline, the usefulness-driven, the coherence-driven and the expressiveness-driven selection approaches, we conducted a blind user test, in which we recruited 15 postgraduate students from our university. For our study, we provided our subjects with the list of the 30 sense annotated queries and the snippets selected by each of the algorithms for each of the query relevant documents. The snippets extracted from every query-relevant document were displayed to our users in a random order so as not to convey any information about the criteria under which these were selected. Moreover, in case the same snippet was selected by more that one algorithms, it was presented only once to the user.

We then asked our participants to read all the snippets delivered for each of the queries and indicate for which of the snippets they would like to read the entire source document. In other words, our participants were not informed about the different snippet selection criteria and they were not aware of the fact that all the snippets displayed for a query would direct them to the same document. In contrast, the instructions that were given to them required that: "*Select which of the displayed text fragments do you think will direct you to a document that can successfully answer the search intention of the query*?" Note that the query intention was explicitly communicated to our subjects through the WordNet sense that has been selected for representing the query semantics.

Our participants interacted with a local interface via which we displayed them the annotated experimental queries (one at a time) and the different snippets selected for every query relevant document. For every query, the users viewed at least one snippet (in case all algorithms selected the same passage) and at most four snippets (in case a different passage was selected by each of the algorithms) in a random order. Our subjects indicated their selections by clicking on the snippet that they deemed it would drive them to the most query-relevant document. A user's click on a snippet translates to a vote given by the user for that snippet's success in focusing retrieval results to the query intention.

We recorded the user's clickthrough on the displayed snippets in order to infer the criterion that influences the most human judgments about what makes a snippet successful. In case the user clicked on a snippet that was selected by more than one algorithm, the user's vote was equally attributed to all selection techniques that delivered the particular snippet. Based on the human preferences, we can evaluate to a certain extend how people cast

their click decisions to the snippets that they are returned for their search queries. Furthermore, human judgments (reported on Table 1) give us some early intuition about the weights that should be appended to our snippet selection metrics of query usefulness and text expressiveness. The following table reports the number of times each user selected a passage delivered by the baseline, the query usefulness, the semantic coherence and the text expressiveness criteria over all 30 queries examined.

Table 1. Snippet selection criteria preferred across users.

USER	Baseline	Query Usefulness	Semantic Coherence	Text-Expressiveness
1	5	12	9	8
2	9	17	8	5
3	6	7	7	10
4	8	17	9	10
5	8	13	7	5
6	11	15	5	6
7	3	15	7	6
8	14	14	4	3
9	9	9	10	7
10	11	15	5	6
11	4	11	9	6
12	7	15	11	8
13	5	10	11	9
14	9	18	6	8
15	6	12	7	5

The results of our human survey suggest that semantically derived snippets are valued higher than statistically obtained ones. This is in line with our intuition that passage selection based on the semantic correlation between the passage and query terms yields improved retrieval focus. With respect to the passage semantics, our results demonstrate that the snippet selection criterion that is valued higher by our participants is that of usefulness. This practically implies that what users would like to see in the text fragments accompanying retrieval results are passages that exhibit high semantic and terminological correlation to their query intention. This observation supports the need for more sophisticated approaches towards snippets' selection and indicates that passage retrieval algorithms could be fruitfully explored in this respect. However, our findings relying on a few users and a small number of queries merit further investigation before these are employed towards tuning the weights that should be given to the different metrics employed for snippets' selection.

4. RELATED WORK

The role of text snippets or passages in the context of web information retrieval has been studied before. A number of researchers have proposed the exploitation of passages to answer natural language queries [4], [5], [6] and generic queries [3]. Authors in [4] search for single snippet answers to definition questions through the exploitation of lexical rather than semantic patterns. In [5] and [6] the authors exploit WordNet to annotate and consequently answer definition questions. Most of the reported approaches on snippets' exploitation for question-answering rely on some similarity measure in order to derive from a query relevant document, the text fragment that is closest to the query. The relevance/ similarity between the query and the snippet is measured using linguistic [10] (distance in an ontological thesaurus) or statistic [9] (word frequency, proximity or co-occurrence) techniques or a combination of them.

Passage retrieval is a common component to many question answering systems. Currently, there exist several passage retrieval algorithms, such as MITRE [24], bm25 [25], MultiText [34], IBM [28], SiteQ [29], ISI [30]. Recently, [31] quantitatively evaluated the performance that the above passage retrieval algorithms have on question answering. Moreover, passage retrieval approaches have been proposed in the context of web-based question answering [32], [33]. Most of the systems explored in web-based passage retrieval typically perform complex parsing and entity extraction for documents that best match the given queries, which limits the number of web pages that can analyze in detail. Other systems require term weighting for selecting or making the best-matching passages [27] and this requires auxiliary data structures.

Many research works perform post processing of snippets extracted from query results. They either cluster snippets into hierarchies [3], use them to construct ontologies [7], or further expand the snippet collection with relevant information nuggets from a reference corpus [8]. Evaluation of retrieved snippets is performed once again using statistic [35]or linguistic methods [11] and long QA series [12]. Text coherence is a topic that has received much attention in the linguistic literature and a variety of both qualitative and quantitative models have been proposed [19] [20] [21] [22]. Most of existing models incorporate either syntactic or semantics aspects of text coherence.

In our work on passage retrieval, we rely purely on semantic rather that syntactic aspects of both the queries and the documents and we propose a novel evaluation framework which ensures that the passage delivered in response to some query and not merely query relevant but they are also semantically coherent and expressive of the entire document's contents.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a novel framework for the automatic selection out of a query matching document the text snippet that is the most useful to the query intention. Our approach capitalizes on the notion of semantic correlation between the query keywords and the selected snippet's content as well as on the semantic correlation between the in-snippet terms. We argue that our approach is particularly suited for identifying within the contents of a possibly long document the focus of the query and we introduced a qualitative evaluation scheme for capturing the accuracy in which the selected passage participates in focused web searches. We applied our snippet selection technique to a number of searches that we have performed using synthetic data generated by simulation. Our experiments revealed that our snippet selection method determined by the semantic correlation between the query and the selected text fragment yields increased retrieval performance compared to statistical-based passage retrieval methods.

The snippet selection approach introduced in this paper relies on semantic rather than statistical properties of web documents and it is relatively inexpensive assuming access to a rich semantic resource (such as WordNet). This makes the proposed approach particularly attractive and innovative for the automatic selection and evaluation of focused web snippets. An important future direction lies in the enrichment of our snippet selection model with advanced linguistic knowledge such as co-reference resolution, genre detection or topic distillation. Moreover, it would be interested to experiment with alternative formulas for measuring the correlation between the query keywords and the passage terms, such as the one proposed in [2]. Another possible direction would be to employ a query relevant snippet as a backbone resource for a query refinement technique. Yet a more stimulating challenge

concerns the incorporation of user profiles in the snippet selection process in an attempt to deliver personalized text passages. Last but not least, our snippet selection approach could be fruitfully explored in the context of web question-answering and element retrieval systems in the hope of helping the user find the exact information sought in an instant yet effective manner.

ACKNOWLEDGEMENTS

Authors George Tsatsaronis and Lefetris Kozanidis were funded by the 03ED_850 and 03ED_413 research projects respectively, implemented within the "Reinforcement Programme of Human Research Manpower" (PENED) and co-financed by National and Community Funds (25% from the Greek Ministry of Development-General Secretariat of Research and Technology and 75% from E.U.-European Social Fund). Author Iraklis Varlamis is partially funded by the EPEAEK PYTHAGORAS II research project, financed by the Greek Ministry of Education and Religion.

REFERENCES

[1] Google patent 2003. Detecting query-specific duplicate documents. US patent No. 6615209.

[2] Halkidi M., Nguyen,B, Varlamis, I., Vazirgiannis M. 2003. THESUS: Organizing Web document collections based on link semantics. VLDB J. 12(4): 320-332.

[3] Ferragina, P., Gulli, A. 2005. A personalized search engine based on web-snippet hierarchical clustering. In Special Interest Tracks & Posters, 14th Intl. WWW Conference.

[4] Androutsopoulos I., Galanis D. 2005. A practically unsupervised learning method to identify single-snippet answers to definition questions on the web. In the HLT/EMNLP Conference, pp. 323-330.

[5] Prager J., Chu-Carroll J., Czuba K. 2001. Use of WordNet hypernyms for answering what-is questions. In Proceedings of the TREC-2002 Conference, NIST.

[6] Prager J.M., Radev D.R., Czuba K. 2001. Answering what-is questions by virtual annotation. In Proceedings of Human Language Technologies Conference, pp. 26-30.

[7] van Hage W.R, de Rijke M., Marx M. 2004. Information retrieval support for ontology construction and use. In Intl. Semantic Web Conference, pp. 518-533.

[8] Mishne G., de Rijke M., Jijkoun V. 2005. Using a reference corpus as a user model for focused information retrieval, In Journal of Digital Information Management, 3(1):47-52.

[9] Lin D., Pantel P. 2001. Discovery of inference rules for question answering. Natural Language Engineering 7(4):343-360.

[10] De Boni M., Manandhar S. 2003. The use of sentence similarity as a semantic relevance metric for question answering. In New Directions in Question Answering, pp. 138-144.

[11] Voorhees E. 2003. Evaluating answers to definition questions. In Proceedings of the HLT-NAACL Conference.

[12] Voorhees E. 2005. Using question series to evaluate question answering system effectiveness. HLT/EMNLP Conf.

[13] Jansen B.J., Spink A. Saracevic T. 2000. Real life, real users, and real needs: A study and analysis of user queries on the Web. Information Processing & Management, 36(2):207-227.

[14] Kozanidis L., Tzekou P., Zotos N. Stamou S., Christodoulakis D. Ontology-based adaptive query refinement. In Proceedings of the 3rd WebIST Conference, 2007.

[15] Wu X., Palmer M. 1994. Web semantics and lexical selection. In the 32nd ACL Meeting.

[16] WordNet. available at: http://www.cogsci.princeton.edu/~wn

[17] Zamir O., Etzioni O. 1998. Web document clustering: a feasibility demonstration. In the SIGIR Conference.

[18] Song Y.I., Han K.S., Rim H.C. 2004. A term weighting method based on lexical chain for automatic summarization. In the 5th CICLing Conference, pp. 636-639.

[19] Grosz B., Joshi A., Weinstein S. 1995. Centering: A framework for modeling the local coherence of discourse. In Computational Linguistics, 21(2): 203-225.

[20] Higgins D., Burstein J. Marcu D., Gentile C. 2004. Evaluating multiple aspects of coherence in student essays. In Proceedings of the NAACL Conference, pp. 185-192.

[21] Foltz P., Kintsch W., Landauer K. 1998. Textual coherence using latent semantic analysis. In Discourse Processes, 25(2&3): 285-307.

[22] Lapata M., Barzilay R. 2005. Automatic evaluation of text coherence: models and representations. In the Intl. Joint Conferences on Artificial Intelligence.

[23] Kritikopoulos A., Sideri M., Varlamis I. 2007. Success Index: Measuring the efficiency of search engines using implicit user feedback. In the 11th Pan-Hellenic Conference on Informatics, Special Session on Web Search and Mining.

[24] Light M., Mann G.S., Riloff E., Breck E. 2001. Analyses for elucidating current question answering technology. Journal of Natural Language Engineering, Special Issue in Question Answering.

[25] Robertson S.E., Walker S., Hancock-Beaulieu M., Gatford M., Payne A. 1995. Okapi at TREC-4. 4th TREC Conf.

[26] Vicedo J.L., Ferrandez A. 2001. University of Alicante at TREC-10. In the 10th Text Retrieval Conference.

[27] Clarke C., Cormack G., Lynam T. 2001. Web reinforced question answering. In the 10th TREC Conference.

[28] Ittycheriah A., Franz M., Roukos S. 2001. IBM's statistical question answering system-TREC10. 10th TREC Conf.

[29] Lee G.G., Seo J., Lee S., Jung H., Cho B.H., Lee C., Kwak B.K., Cha J., Kim D., An J., Kim H., Kim K. 2001. SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow NLP. In the 10th TREC Conference.

[30] Hovy E., Hermjakob U., Lin C.Y. 2001. The user of external knowledge on factoid QA. In the 10th TREC Conference.

[31] Tellex S., Katz B., Lin J., Fernandes A., Marton G. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In the 26th SIGIR Conference.

[32] Kwok C., Etzioni O., Weld D. 2001. Scaling question answering to the web. In the 10th WWW Conference, pp. 150-161.

[33] Buchholz S. 2002. Using grammatical relations, answer frequencies and the world wide web for TREC question answering. In the 10th TREC Conference.

[34] Clarke C., Cormack G., Kisman D., Lynam T. 2000. Question answering by passage selection (Multitext experiments for TREC-9). In the 9th TREC Conference.

[35] Charles L.A., Clarke E.L. 2003. Terra: Passage retrieval vs. document retrieval for factoid question answering. In SIGIR Conference, pp. 427-428.

[36] Vaswani P., Cameron, J., The NPL Experiments in Statistical Word Associations and their Use in Document Indexing and Retrieval, No. 42. National Physical Laboratory, 1970.

[37] Pedersen T., Banerjee S., Patwardhan S. Maximizing semantic relatedness to perform word sense disambiguation. Available at: http://www.d.umn.edu/~tpederse/Pubs/max-sem-relate.pdf.

Author Index