The University of Amsterdam at TREC 2002

Christof Monz Jaap Kamps Maarten de Rijke
Language & Inference Technology group
ILLC, University of Amsterdam
Nieuwe Achtergracht 166, 1018 WV Amsterdam
The Netherlands
E-mail: {christof,kamps,mdr}@science.uva.nl
URL: www.science.uva.nl/∼{christof,kamps,mdr}

Abstract: We describe our participation in the TREC 2002 Novelty, Question answering, and Web tracks. We provide a detailed account of the ideas underlying our approaches to these tasks. All our runs used the FlexIR information retrieval system.

1 Introduction

At TREC 2002 we took part in the Novelty, Question Answering, and Web tracks. Our main aims for the Novelty and Web tracks was to set up baseline systems on which we plan to build in future editions of the tracks. Our main aim for the Question Answering track was to test a revised architecture of our knowledge-intensive question answering system Tequesta [16], and to experiment with a number of newly added features relating to the document retrieval steps carried out within Tequesta.

For all three tracks, our experiments exploited the FlexIR information retrieval system developed at the University of Amsterdam [15]. The main goal underlying FlexIR's design is to facilitate flexible experimentation with a wide variety of retrieval components and techniques. FlexIR is implemented in Perl, and built around the standard UNIX pipeline architecture; it supports many types of pre-processing, scoring, indexing, and term-weighting methods, of which we made good use this year. Depending on the task at hand, we used different weighting schemes; see the detailed descriptions of our efforts for each of the tracks below for the exact settings.

The rest of this paper is organized as follows. In three (largely self-contained) sections we describe our work for the Novelty, Question Answering, and Web tracks. We also provide a brief concluding section.

2 Novelty Track

In this section we describe our submissions for the TREC 2002 novelty track. The overall aim of the track is to investigate systems' abilities to locate relevant *and* new information within the ranked set of documents retrieved in a reply to a search engine query. Thus, systems should return information that is both new and relevant rather than whole documents containing duplicate and extraneous information [8]. The novelty task can naturally be divided into two parts. Indeed, the guidelines require that participants identify two lists of documents for a given topic [20]. The first contains the *relevant* sentences, and the second one (a subset of the first) contains only those sentences that add *new* information.

Our main interest in participating in the novelty track was in exploring the second part of the task: identifying *new* sentences. However, due to time constraints we had to limit ourselves to fairly straightforward approaches to both parts of the novelty task. We ended up setting a simple baseline, using established IR strategies for the relevance part, and weighted overlap for the novelty part; our aim is to build on this with more linguistically motivated techniques in the near future. The relevance part, which is the most important part of the track as it also has an obvious impact on the performance of the novelty part, requires far more work than we had anticipated.

The remainder of this section is organized as follows. After recalling some key facts about the experimental set-up, we describe our approaches to the relevance and novelty parts of the novelty task, and then list and briefly discuss our results.

2.1 Topics and Documents

For ease of reference, we briefly highlight some key facts about the documents and topics used in the novelty track; the overview paper provides further details [8]. Initially, there were 50 topics, taken from TRECs 6, 7, and 8 (topics 300–450); after the evaluation was completed, one topic was removed as it was not found to have relevant sentences. The documents are a subset of the relevant documents for the topics. Participants are provided with a ranked list of relevant documents, with between 10 and 25 relevant documents per topic.

2.2 Computing Relevance

We approached the task of identifying *relevant* sentences in the following manner. For a given topic, the sentences in the relevant documents for that topic were viewed as documents

themselves, thus creating a sentences-as-documents collection for each topic. We ran the topic (only using the title and description fields) against this sentences-as-documents collection using our retrieval engine FlexIR. We initially followed Salton and Buckley, who recommend the tfx.nfx weighting scheme for short queries and short documents [18], but some informal pre-submision experiments on comparable topics and documents suggested that tfv.nfx was somewhat more effective.

Three different runs were submitted: one where all documents and topics were porter stemmed [17] (run identifier `UAmsT11ntste`), and a second where they were lemmatized using Helmut Schmidt's TreeTagger [19] (run identifier `UAmsT11ntlem`); here, each word is assigned its syntactic root through lexical look-up; mainly number, case, and tense information is removed, leaving other morphological processes such as nominalization intact. And in the third run the results of the other two runs were simply merged (run identifier `UAmsT11ntcom`). Our motivation for the first two runs was to see to which extent morphological normalization has an impact on the relevance and novelty parts of the task. The third run was included to determine the impact on the novelty part of the task of high recall approaches to the relevance part.

2.3 Computing Novelty

Our approach to the novelty part of the task was based on a non-symmetric weighted overlap score, which we use to provide graded answers to the following question: is the information contained in a sentence *entailed* by a sentence (or set of sentences) seen before? We say that a sentence is *new* (within a context) if it is not entailed by the context.

Assuming the usual definition of *idf* term weights, we compute the *entailment score*, $entscore(s_i, s_j)$, of two (sets of) sentences s_i and s_j by comparing the sum of the weights of terms that appear in both s_i and s_j to the sum of the weights of all terms in the second sentence (or set of sentences) s_j:

$$entscore(s_i, s_j) = \frac{\sum_{t_k \in (s_i \cap s_j)} idf_k}{\sum_{t_k \in s_j} idf_k}. \quad (1)$$

In words: how many of the content-bearing terms in s_j occur in s_i? Clearly, $entscore(s_i, s_j)$ varies from 0 to 1.

A few remarks are in order. First, note that our entailment score is not just a notion of similarity: in general, $entscore(s_i, s_j) \neq entscore(s_j, s_i)$.

Second, to work with *entscore* and conclude that s_i entails s_j, it may not be sufficient to have a non-zero entailment score: we may need some positive 'entailment threshold.' In our experiments we used 0.6; this figure was obtained by testing our methods on the 4 samples provided by NIST as training material. The mechanism of entailment thresholds offers a large amount of flexibility for fine-tuning the entailment notion to one's purposes; see below for some discussion on this point.

To identify the list of new sentences as required by the guidelines, we simply went down our list of relevant sentences, taking the first one as our starting point, and including later ones only if they were not entailed by the ones already included. Our three runs used exactly the same ideas for their novelty parts, and differed only in the list of relevant sentences they took as input.

2.4 Results and Discussion

To assess the results of the relevance and novelty parts of the task, the product of precision and recall (P*R) is used as measure, with separate scores for the two parts of the task. The average of P*R is meaningful even when the judgment sets sizes vary widely, as is the case for the task at hand. One downside of P*R is that in practice the scores tend to be close to 0.

Table 1 shows the results for each of our three runs. Taking the stemmed run as our baseline, we see that both lemmatizing and combining produce significant improvements, for both the relevance and novelty parts.

Table 1: Summary of the results for the novelty track.

	Average P*R	
Run identifier	Relevance	Novelty
`UAmsT11ntste`	0.029	0.028
`UAmsT11ntlem`	0.033 (+13.8%)	0.031 (+10.7%)
`UAmsT11ntcom`	0.034 (+17.2%)	0.032 (+14.3%)

Let's take a closer look at the results. The improvements obtained by lemmatizing topics and documents instead of stemming them, are not uniform. For many individual topics stemming is at least as good as, or even better than lemmatizing, for both relevance and novelty; similar observations can be made about the combined run vs. the other runs. Table 2 provides a breakdown of the number of top scores per run; the first number is the total number of top scores for a given run, the second number is the number of unique top scores (that are not shared by other runs).

Table 2: Top scores per run.

	# Top P*R Scores (shared, unique)	
Run identifier	Relevance	Novelty
`UAmsT11ntste`	25, 3	23, 12
`UAmsT11ntlem`	37, 15	37, 26
`UAmsT11ntcom`	21, 9	23, 0

Figures 1 and 2 plot our P*R scores against the median by topic. They suggest a number of things. First, while we seem to do relatively poorly on the relevance part of the novelty task, our performance on the novelty seems somewhat better.

The definition of the novelty task suggests that a system's performance on the novelty part is, to a large degree, determined by its performance on the relevance part, and the considerable similarity between the plots in Figures 1 and 2 confirms this.

We carried out a number of post-submission experiments, using the golden standards provided by NIST. First of all, we

Figure 1: Comparison of relevance scores to median by topic.

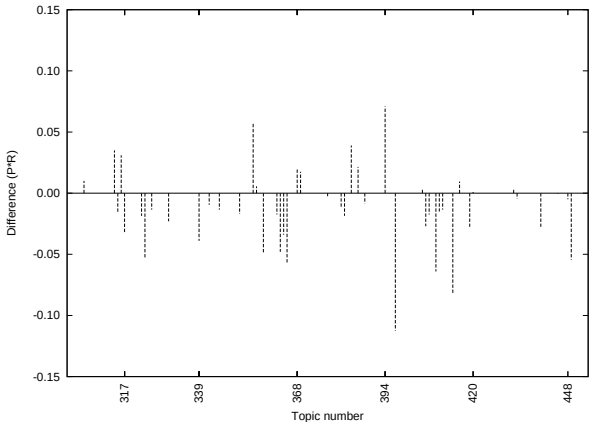
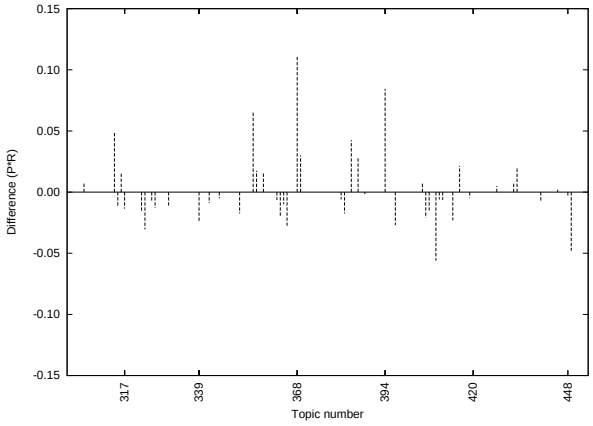


Figure 2: Comparison of novelty scores to median by topic.



ran some experiments to see whether we used an (almost) optimal value for the entailment threshold for our official submissions. Figure 3 shows the average precision, recall, and P*R scores for our combined run (`UAmsT11ntcom`) with increasing values of the threshold. The value of 0.6 that we used in the submitted run is close to the optimal one, although

Figure 3: Impact of the entailment threshold on novelty.

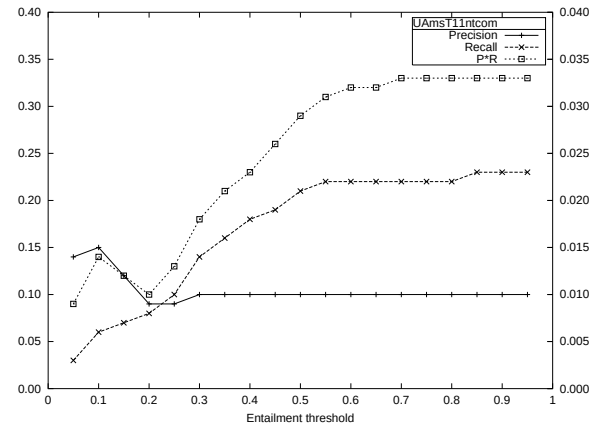
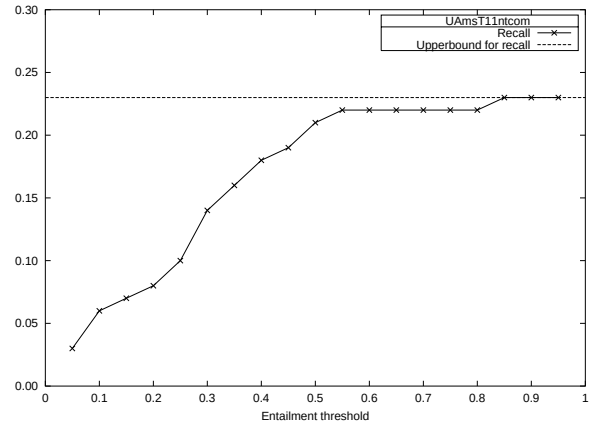


Figure 4: Upperbound on the novelty performance.



values of 0.7 or higher would have produced slightly higher scores (0.033, $+3.1\%$).

Furthermore, we determined an upperbound on the performance of the novelty part of our system, to get some understanding of its behavior in absolute terms. If we take the relevance results of our best run (`UAmsT11ntcom`) and intersect these with the novelty qrels provided by NIST, we get the best possible list of new sentences (given our relevance output). Since the precision for this optimal list is 1, it only makes sense to look at the recall for this list, which turns out to be 0.23, very close to the score actually obtained (0.22); see Figure 4.

In conclusion, while we are especially interested in the novelty part of the novelty track, it seems that the relevance part is the hardest and most important part of the task. We plan to address it more extensively than we have done so far by bringing in linguistic features; it is not obvious, however, how much this will differ from document summarization.

3 Question Answering Track

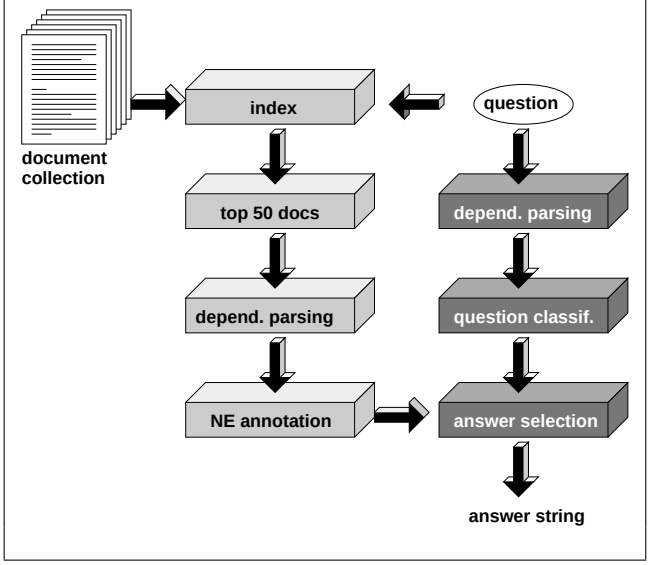
This section describes our submissions for the question answering track at TREC 2002. Our main focus was on evaluating a basic question answering system that exploits shallow NLP techniques in combination with standard retrieval techniques.

3.1 System Description

The system architecture of Tequesta (TExtual QUESTion Answering) is fairly standard; its overall architecture is displayed in Figure 5. Like most current QA systems, Tequesta is built on top of a retrieval system. The first step is to build an index for the document collection, in this case the AQUAINT collection. Then the question is translated into a retrieval query which is sent to the retrieval system. For retrieval we use the FlexIR system described in the introduction.

The retrieval system is used to identify a set of documents that are likely to contain the answer to a question posed to

Figure 5: Tequesta system architecture.



the system. The top documents returned by FlexIR are further processed as described in Section 3.1.2.

Just like the top documents, the question is also parsed. The parsed output is used to determine the focus of the question. Question analysis is explained in Section 3.1.3.

3.1.1 Document Retrieval

For pre-fetching relevant documents that are likely to contain the answer, Tequesta uses FlexIR, which was given a total of 1,033,461 documents to index. All our official runs for TREC 2002 used the Lnu.ltc weighting scheme [3] to compute the similarity between a question and a document. For the experiments on which we report in this article, we fixed *slope* at 0.2; the pivot was set to the average number of unique words occurring in the collection.

To increase precision, we decided to use a lemmatizer; the lemmatizer used is TreeTagger, the same as in our experiments for the novelty track.

In document retrieval it is common practice to return a ranked list of documents, each item being adorned with the similarity score. Additionally, FlexIR returns a minimally matching span (MSM) for each document. An MSM indicates the starting (s) and ending position (e) of a text excerpt, containing all matching terms, such that there are no positions s' or e', $s < s'$ and $e' < e$, and neither the span s', e nor the span s, e' also covers all matching terms; see also [4]. In a later stage of the question answering process, MSMs are used to restrict documents to passages which are likely to contain the answer.

3.1.2 Document Analysis

Document analysis focuses on the top 50 documents that were returned by FlexIR. For each of them, we used the MSM to extract a text passage which was then analyzed further. The passage begins with the sentence containing the beginning position of the MSM and ends with the sentence containing the ending position of the MSM. This way we make sure that the passage contains full sentences which can be parsed. Here, we used Dekang Lin's dependency parser MINIPAR [14]. Identifying sentence boundaries was accomplished by TreeTagger.

Depending on the question type — see below for more details — a named entity recognizer was applied to identify phrases that are of the same semantic type as the expected answer. This process is guided by the question classification component. For instance, if a question is looking for a numerical expression (such as age, speed, length, etc.) only expressions of that type are annotated.

3.1.3 Question Analysis

Just like the top 50 documents, the questions themselves were also part-of-speech tagged, morphologically normalized, and parsed. Since there is a significant difference between word order in questions and in declarative sentences, we needed to adjust the tagger for questions. To this end, TreeTagger was trained on a set of 500 questions with part-of-speech tags annotated. We used 300 questions taken from the Penn Treebank II data set together with the 200 TREC-8 questions, which we annotated semi-automatically.

We used 33 categories to classify the focus or target of a question, some of which are listed in Figure 6.

To identify the target of a question, pattern matching is applied to assign one of the 33 categories to the question. In total, a set of 102 patterns is used to accomplish this. Some of the patterns used are shown in Table 3.

If more than one pattern matches the question, it was assigned multiple targets. The patterns are ordered so that more specific patterns match first. Also, the answer selection component described in the next subsection obeys the order in which questions were categorized to find answers for more specific targets first.

Questions of type `what-np` form a special category. Here we use a dependency parser to identify the appropriate target, symbolized by `np` in the type. Usually, `what-np` questions are of the form *What NP VP?* or *What NP PP VP?*. After parsing the question, we use the head of the NP as target, which has *what*, or *which* as a determiner. For instance, question 1413 from the TREC 2002 question set, shown in (2), is assigned `what:river` as question target.

(2) What river is called "China's Sorrow"?

If none of the matching strategies described so far is able to assign a target to a question, the question is categorized as `unknown`. As a consequence, none of the answer selection strategies which are particularly suited for the respective question targets can be applied, and a general fall back strategy is used.

Table 3: Types for question classification.

Question target	Example patterns
name	/(W|w)hat(wa| i|\')s the name/
pers-def	/[Ww]ho(wa| i|\')s [A-Z][a-z]+/
thing-def	/[Ww]hat(wa| i|\')s an? /,/ (was|is|are|were) a kind of what/
pers-ident	/[Ww]ho(wa| i|\')s the/
thing-ident	/[Ww](hat|hich)(wa| i|\')s the /
number	/[Hh]ow (much|many) /
expand-abbr	/stand(s)? for(what)?\s*?/,/is (an|the) acronym/
find-abbr	/[Ww]hat(i|\')s (the|an) (acronym|abbreviation) for
agent	/[Ww]ho /,/ by whom[\.\?]/
object	/[Ww]hat (did|do|does) /
known-for	/[Ww]hy .+ famous/ /[Ww]hat made .+ famous/
aka	/[Ww]hat(i|\')s (another|different) name /
name-instance	/Name (a|one|some|an) /
location	/[Ww]here(\'s)? /,/ is near what /
date	/([Aa]bout)?(W|w)hen /,/([Aa]bout)?(W|w)(hat|hich) year /
reason	/[Ww]hy /
what-np	-
unknown	-

3.1.4 Answer Selection

Given the parsed and annotated top documents returned by FlexIR and given the parsed and classified questions, the actual process of identifying the answer starts.

Questions of type `agent` ask for an animate entity, such as a person or organization, being the logical agent of an event described in the question. If the dependency structure from the question matches a dependency structure from a document and there is an animate NP in subject position, or, in case of passive voice, within a PP headed by the preposition *by*, we take this to be the logical agent. Of course, such an NP is disregarded if it already occurs in the question itself. Questions of type `object` are dealt with analogously.

Questions of type `what-np` are particularly interesting because they are very frequent (at least in the TREC 2002 data, where 14.8% of the questions are of this type) and explicitly require some lexical knowledge base. Questions of type `what-np` ask for something that is an instance of the `np` and that fits the further description expressed in the remainder of the question. For example, question 1525, given in (3), asks for something which is a university.

(3) What university did Thomas Jefferson found?

In (3) *university* is the focus of the question and the further constraint *did Thomas Jefferson found?* is the topic of the question. In order to establish the relationship between an entity found in a matching dependency structure and the predicate *university* it is necessary to access a lexical knowledge base. Tequesta exploits WordNet for this purpose. In particular, WordNet's hyponym relations are used.

Answer candidates for all remaining question types where identified by named entity extraction where the named entity has to be of the same type as the expected answer.

Each answer candidate received a matching score depending on its position in the document. Candidates occurring within the MSM passage received a higher score than candidates occurring outside it. If the same candidate was extracted several times, possibly from different documents, their individual scores were summed up. The answer candidates were sorted by score and the answer candidate with the highest score was returned as answer. Answer candidates with identical scores were sorted randomly.

Since the score of the highest ranked answer candidate can be the sum of several occurrences, possibly from different documents, we take the document which has the largest share in the score as the supporting document, which is returned together with the answer-string.

3.1.5 Confidence

One of this year's changes in the TREC question answering track was to adorn an answer with a confidence score, indicating the system's trust in the returned answer. We used a rather simple approach to computing confidence. All answer candidates for a question q were ranked with respect to their answer score, yielding a sorted list of answer candidates $a_1,\ldots,a_n$, where $score(a_i) \geq score(a_{i+1})$, for $1 \leq i \leq n$. If two answer candidates have the same score, they are sorted at random. Then, the confidence that the highest ranked answer candidate is indeed the correct answer is computed as follows:

$$confidence(a_1) = \begin{cases} a_1 - a_2 & \text{if } a_1 > a_2 \\ \frac{1}{m} & \text{if } a_1 = \cdots = a_m > a_{m+1} \end{cases}$$

Figure 6: Question targets, plus examples from the TREC-11 question set.

agent name or description of an animate entity
(Q-1424): *Who won the Oscar for best actor in 1970?*

aka alternative name for some entity
(Q-1448): *What is the fear of lightning called?*

capital capital of a state or country
(Q-1520): *What is the capital of Kentucky?*

date date of an event
(Q-1406): *When did the story of Romeo and Juliet take place?*

date-birth date of birth of some person
(Q-1880): *When was King Louis XIV born?*

date-death date of death of some person
(Q-1601): *When did Einstein die?*

expand-abbr the full meaning of an abbreviation
(Q-1531): *What does NASDAQ stand for?*

location location of some entity
(Q-1818): *Where did Golda Meir grow up?*

name the name of a person or an entity in general.
(Q-1436): *What was the name of Stonewall Jackson's horse?*

number-dist spatial distance between two entities
(Q-1876): *How far from the earth is the sun?*

number-height height of some entity
(Q-1802): *How tall is Tom Cruise?*

number-length length of some entity
(Q-1857): *What is the length of Churchill Downs racetrack?*

number-money monetary value of some entity or event
(Q-1645): *How much is the international space stations expected to cost?*

object `object` questions are near-reverses of the `agent` questions. Here, the object of an action described in the question is sought.
(Q-1590): *What do grasshoppers eat?*

pers-ident a person fitting some description expressed in the question
(Q-1769): *Who is the owner of the St. Petersburg Times?*

thing-ident thing identical to the description expressed in the question
(Q-1547): *What is the atomic number of uranium?*

what-np an instance of the np fitting the description
(Q-1484): *What college did Allen Iverson attend?*

3.2 Results

The 2002 edition of the main QA task differs from previous years in several aspects. First of all, the document collection has changed from Disks 1–5 of the TIPSTER/TREC collection to the AQUAINT collection covering a more recent period, namely 1998–2000. A total of 500 questions is provided that seek short, fact-based answers. Some questions are not known to have an answer in the document collection. A further restriction, with respect to previous TRECs, is that each participating system is allowed to return only one response per question. A response is either a [answer-string, docid] pair or the string "NIL," The answer-string has to be an exact answer and the docid must be the id of a document in the collection that supports the answer.

An [answer-string, docid] pair is judged *correct* or *right* (R) if the answer-string consists of exactly a correct answer and that answer is supported by the document returned. If the answer-string is responsive and contains a correct answer, but the document does not support that answer, the pair will be judged "unsupported" (U). If the answer-string contains a correct answer and the document supports that answer, but the string contains more than just the answer (or is missing bits of the answer), it is judged as *inexact* (X). Otherwise, the pair is judged *incorrect* or *wrong* (W).

Finally, the scoring method for a run has changed in order to incorporate the confidence with which a question is answered by a system. Within the submission file the questions should be ordered from most confident response to least confident response. The final *confidence-weighted score* (CWS) is computed as follows:

$$\text{CWS} = \frac{\sum_{i=1}^{500} \frac{1}{i} \sum_{j=1}^{i} [\![\, judgment(j) = \text{R} \,]\!]}{500}$$

where $judgment(j)$ is the judgment of the NIST assessors for question j, and $[\![\, expression \,]\!]$ is 1 if *expression* is true, and 0 otherwise.

3.2.1 Submitted Runs

We submitted three runs for the main task (`UAmsT11qaM1`, `M2`, and `M3`).

The runs differed along 2 dimensions: the number of documents used as input for the answer selection process: either 50 documents (`UAmsT11qaM1`) or 100 documents (`UAmsT11qaM2` and `UAmsT11qaM3`), and whether questions were sorted with respect to confidence or not: runs `UAmsT11qaM1` and `UAmsT11qaM2` were sorted with respect to confidence and run `UAmsT11qaM3` was simply sorted by question id.

3.2.2 Results and Discussion

Table 4 summarizes the confidence-weighted scores (CWS) for each of our three submitted runs (`UAmsT11qaM1`, `M2`, and `M3`) over the 500 questions.

To investigate the impact of the different judgments for partial correctness of an answer-string, we compared the strict confidence-weighted scores, as defined above, to confidence

Table 4: Summary of the CWS for the main task.

	UAmsT10qa...	M1	M2	M3
CWS(R)		0.145	0.101	0.146
CWS(R,U)		0.219	0.213	0.197
CWS(R,X)		0.151	0.135	0.174
CWS(R,U,X)		0.225	0.248	0.226

scores where also inexact (X) or unsupported (U) answers count as correct. E.g.,

$$\mathrm{CWS(R,X)} = \frac{\sum_{i=1}^{500} \frac{1}{i} \sum_{j=1}^{i} [\![\, judgment(j) \in \{\mathrm{R,X}\} \,]\!]}{500}$$

As can be expected, confidence-weighted scores increase as judgments become less strict. In particular, allowing for unsupported answers has a strong impact on the scoring. Comparing run UAmsT11qaM1 (using the top 50 documents) with UAmsT11qaM2 (using the top 100 documents), indicates that using a smaller set of documents for answer selection is to be preferred; although this conclusion is not supported by CWS(R,U,X).

Runs UAmsT11qaM2 and UAmsT11qaM3 both use the top 100 documents, but we did not sort the responses in UAmsT11qaM3 with respect to confidence. This was meant to evaluate our confidence score computation algorithm. The results in Table 4 are very inconclusive, as UAmsT11qaM2 scores better for CWS(R,U) and CWS(R,U,X) but worse for CWS(R) and CWS(R,X).

In addition, we also calculated the precision of each run, neglecting confidence weights. E.g.,

$$\mathrm{Prec(R)} = \frac{\sum_{i=1}^{500} [\![\, judgment(j) = \mathrm{R} \,]\!]}{500}$$

The average precision scores are displayed in Table 5.

Table 5: Summary of the avg. precision for the main task.

	UAmsT10qa...	M1	M2	M3
Prec(R)		0.128	0.112	0.112
Prec(R,U)		0.170	0.176	0.176
Prec(R,X)		0.134	0.132	0.132
Prec(R,U,X)		0.176	0.196	0.196

As with the confidence-weighted scores, precision also increases as judging becomes less strict. Again, counting unsupported answers as correct has the strongest impact on precision. Note, that UAmsT11qaM2 and UAmsT11qaM3 have the same scores for all judgments since they differ only with respect to confidence sorting. The higher precision scores of UAmsT11qaM2 and UAmsT11qaM3 compared to UAmsT11qaM1, when allowing for unsupported answers, are probably due to the lower number of NIL answers: UAmsT11qaM1 contains 234 questions having NIL as an answer, whereas UAmsT11qaM2 and UAmsT11qaM3 contain only 88 questions having NIL as an answer.

Table 6 offers a closer look at our primary run for the main task, UAmsT11qaM1, and provides a breakdown in terms of the individual question types. Column 1 lists the question classes as discussed in Section 3.1.3 which have at least one question in the TREC 2002 question set; column 2 lists the percentage of questions belonging to a particular class. In column 3 the individual precision scores are displayed. Column 4 lists the confidence-weighted scores for each class of questions. The last column records the relative difference between the mean CWS for the class and the overall CWS for the run (shown at the bottom of column 4). All confidence-weighted scores are based on strict evaluation, i.e., CWS(R).

Table 6: Analysis of the scores for **UAmsT11qaM1**.

Question class	% quest.	Prec.	CWS	CWS diff.
agent	5.8%	0.172	0.150	+3.4%
aka	3.0%	0.133	0.131	−9.6%
capital	0.6%	0	0.136	−6.2%
date	16.2%	0.160	0.160	+10.3%
date-birth	2.0%	0.300	0.164	+13.1%
date-death	1.2%	0.833	0.165	+13.7%
expand-abbr	1.8%	0	0.132	−8.9%
location	14.4%	0.097	0.148	+2.0%
name	4.8%	0.041	0.133	−8.2%
number-dist	1.4%	0	0.163	+12.4%
number-height	2.0%	0.200	0.131	−9.6%
number-length	0.4%	0	0.145	±0%
number-many	1.0%	0.200	0.158	+8.9%
number-people	0.8%	0	0.135	−6.8%
number-money	0.8%	0.250	0.175	+20.6%
number-much	1.8%	0.111	0.132	−8.9%
number-speed	0.6%	0.666	0.155	+6.9%
number-age	1.2%	0.166	0.155	+6.9%
object	1.4%	0.428	0.132	−8.9%
pers-def	0.8%	0.250	0.132	−8.9%
pers-ident	4.4%	0.090	0.141	−2.7%
thing-def	0.2%	0	0.136	−6.2%
thing-ident	16.2%	0.061	0.133	−8.2%
what-np	14.8%	0.121	0.143	−1.3%
unknown	2.4%	0	0.133	−8.2%
Total		0.128	0.145	

4 Web Track

TREC 2002's Web track features two tasks, named page finding and topic distillation, using a recent crawl of the .gov domain (January 2002). For the named-page finding task, we experimented with plain text runs, anchor-text runs, and their combinations. For topic distillation task, we additionally experimented with ways to exploit the link and URL structure in the collection.

The remainder of this section is organized as follows. After discussing some key facts about the collection and our experimental set-up, we describe our runs for the named pages finding task, and for the topic distillation task, and then discuss our findings on the link structure of the collection.

4.1 The .GOV Collection

The size of the .GOV collection, 1.25 million documents and in total 18 gigabytes, posed a challenge for our FlexIR system. Although CSIRO did a commendable job in preparing this collection, we occasionally stumbled upon binary content, and extremely long strings of characters. We had to implement various modifications to overcome the linux filesize limits. The resulting text-based index is 6 Gb (3.25 Gb for the index and 2.5 Gb for the inverted index).

We built two separate indexes for the .GOV collection: a text-only index, and an anchor-text index. For the free-text index, we indexed all of the documents' textual contents, decoding special html-characters into plain ASCII, and replacing diacritics with the unmarked characters. We used the Porter stemmer [17], and a stoplist of 391 words. Our text index contains 1,247,753 documents. We also built a separate anchor-text only index, assigning the anchor-texts to the linked documents. Again, we used the Porter stemmer. Our anchor-text index contains 667,737 documents, which is 53.51% of the text-based index. For the retrieval runs, we experimented with two weighting schemes, the familiar Lnu.ltc scheme and a scheme, baptized Lnm.ltc, based on minimal matching span (MSM) weighting (see section 3.1.2 for details). We did not use blind feedback in any of our runs.

4.2 Named Page Finding Task

For the named page finding task, there are 150 short queries containing the name of a page. The average query length is 3.81 words or 3.55 words after removing stopwords. There is considerable ambiguity when retrieving a unique page characterized by such a short query. As it turned out, there is a unique relevant page for 132 of the topics, for 16 topics there are two relevant pages, and there are three relevant pages for the remaining 2 topics.

The precursor of this task was TREC 2001's home page finding task [9]. For entry page finding, non-content features such as URLs and links provided valuable information [11]. We did not see a straightforward way to use non-content features for this year's task. An alternative is to use the anchor-texts in the collection [5]. For the named page finding task, we experimented with plain text runs, anchor-text runs, and their combinations.

Table 7: Overview of the named page finding runs.

Run	Type	Weighting
1. UAmsT02WnTl	Text-only	Lnu.ltc
2. UAmsT02WnTm	Text-only	Lnm.ltc
3. UAmsT02WnA	Anchors-only	Lnu.ltc
4. UAmsT02WnTlA	Combined 1/3	
5. UAmsT02WnTmA	Combined 2/3	

The submitted runs are shown in Table 7. The text and anchor-only runs were combined in the following manner. We only considered the first ten results of both runs; following Lee [13], the scores are normalized using $RSV_i' = \frac{RSV_i - min_i}{max_i - min_i}$. We assigned new weights to the documents using the summation function used by Fox and Shaw [7]: $RSV_{new} = RSV_1 + RSV_2$.

Table 8: Anchor-text only runs.

Run	MRR	Top 10	Unknown
UAmsT02WnTl	0.4254	82 (54.7%)	46 (30.7%)
UAmsT02WnTm	0.2601	58 (38.7%)	83 (55.3%)
UAmsT02WnA	0.3279	69 (46.0%)	70 (46.7%)
UAmsT02WnTlA	**0.4317**	99 (66.0%)	35 (23.3%)
UAmsT02WnTmA	0.3672	81 (54.0%)	59 (39.3%)

The results for our official run are shown in Table 8; the column labeled 'MRR' lists the mean reciprocal rank of the first correct answer (the official measure); the column labeled 'Top 10' lists the number of topics with a correct named pages in the top 10; and the column labeled 'Unknown' lists the number of topics for which no named page was found in the top 50.

The results show that the text runs using Lnu.ltc weighting scheme were more effective than those using the Lnm.ltc scheme. The combined text and anchor-text run performed the best with an MRR of 0.4317. The anchor-text only run, which indexes only half of the documents, scores 77.08% of the text only run. The combination of both runs improves the MRR by 1.48% over the text only run; the number of topics in the top 10 is improved by 20.73% over the text only run.

4.3 Topic Distillation Task

For topic distillation, only key resources in the collection will be regarded as relevant. A page can be a key resource solely by its set of links, e.g., a home page of a relevant site. The challenge is to find ways to exploit the additional structure in the documents. There are 50 topics, having on average 3.24 words (2.92 after removing stop words). Although key resources are supposedly much rarer than relevant documents, there turn out to be on average 32.12 key resources per topic.[1]

Similar to the named page finding task, we created runs using the text-only and anchors-only collections (see Table 9 for an overview of the official runs). We experimented with

Table 9: Overview of the topic distillation runs.

Run	Type	Weighting
1. UAmsT02WtT	Text	Lnm.ltc
2. UAmsT02WtTri	Realized indegree 1	
3. UAmsT02WtA	Anchors	Lnu.ltc
4. UAmsT02WtAri	Realized indegree 3	
5. UAmsT02WtAcs	Base URL clusters 3	

the following approach for exploiting the URL information (indicated as 'base URL clusters' in Table 9). Since there will rarely be more than one key resource per site, we cluster pages by their base URL, and return the page with the lowest URL depth. Specifically, we assign the top 100 documents

[1] This is over 49 topics, ignoring Topic 582 for which there were no key resourses in the collection. There are 11 topics with less than 10 key resources.

to the first 10 different base URLs. Next, we return the page with the lowest URL depth or slash-count per cluster.

We also experimented with the use of the link structure of the documents (indicated as 'realized indegree' in Table 9). There exist approaches that look at the global link structure, i.e., page-rank [2], and those that look at the local link structure surrounding an initially retrieved set of documents, i.e., Hyperlink Induced Topic Search (HITS) [10]. We follow Kleinberg [10] in considering the local set of pages containing the initially retrieved documents, plus all documents linked from, or linking to documents in this set. For the anchor text runs we used the top 100 results, and for the text runs, the local set is determined by the top 200 documents. We implemented an approach that combines both global and local link structure by comparing how much of the links of a page are present in the local set of initially retrieved documents. Specifically, we calculate the local indegree (the number of a page's incoming links that are in the local set) divided by the page's indegree (the total number of links to a page). This number, which gives an indication of the topicality, is multiplied by the local indegree. The (local) indegree by itself gives an indication of the relative importance of the page [1]. The resulting new ranking is solely based on the structural link information.

Table 10: Official topic distillation run results.

Run	Prec. at 10, 20, and 30		
UAmsT02WtT	**0.1755**	0.1245	0.1020
UAmsT02WtTri	0.0673	0.0582	0.0463
UAmsT02WtA	0.1000	0.0714	0.0558
UAmsT02WtAri	0.0633	0.0469	0.0381
UAmsT02WtAcs	0.0653	0.0786	0.0660

The results of our official runs are shown in Table 10. The official measure is precision at 10, at which the text-only run scores best with 0.1755. The anchor-text only run, covering only half the documents, scores 56.98% of the text only run. A text only run using Lnu.ltc weighting, not submitted, scored better than the official run, with a precision at 10 of 0.2102. The run using the base URL clusters fails to improve the anchor-text base run, although it improves precision at 20 and 30. The runs based on link information all perform worse than the underlying base runs.

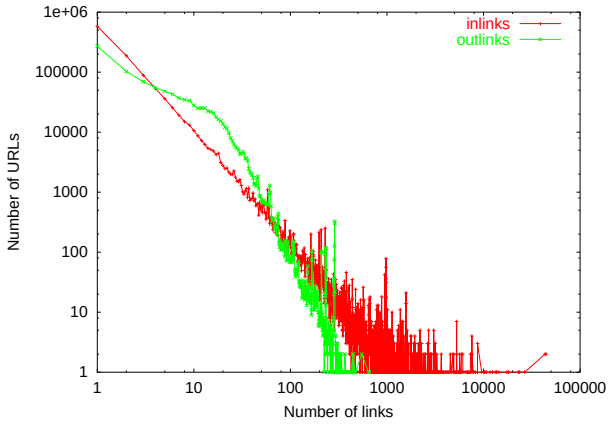
4.4 Link Structure

The link structure in `.GOV` should be a fairly representative sample of the current Internet.[2] Figure 7 shows the link distribution in the `.GOV` collection on a logarithmic scale. Both the distribution of outlinks, and the distribution of inlinks show a powerlaw behavior as observed by [6]. The five pages with the highest number of outlinks are:

- `visibleearth.nasa.gov/browse.html` (653);

[2]We used the provided `links_id` and `id2url` files. These contain a few bugs, e.g., `G15-52-0622377` is listed as `www.lib.noaa.govnewj.htm` instead of `www.lib.noaa.gov/edocs/..newj.htm`.

Figure 7: Link distribution in **.GOV**.



- `www.bls.gov/oes/2000/oes_alph.htm` (647);
- `www.bls.gov/oes/2000/oes_stru.htm` (646);
- `hn.usatlas.bnl.gov/cgi-bin/cvsweb.cgi/offline/graphics/Jive/` (548);
- `www.whitehouse.gov/news/nominations/index-date.html` (471);

The five pages with the highest number of inlinks are:

- `www.usgs.gov/` (44,499);
- `www.usda.gov/` (43,324);
- `www.nasa.gov/` (26,693);
- `www.usda.gov/news/privacy.htm` (23,418);
- `www.usgs.gov/accessibility.html` (23,234);

It is of crucial importance for link-based approaches to be able to distinguish between intrinsic links (links within a site, mainly for navigational purposes) and transverse links (links between sites). The `.GOV` contains in total 11,164,829 links between pages in the collection. We first identified the site of a page as it base URL, with the removal of any prefix starting with `www`. This results in a set of 2,413,054 transverse links (or 22%). This reduced set still contained many within-site links, so we further reduced the set by removing links between base URLs when either is a substring of the other. For example, a link between `www.nih.gov` and `www.nlm.nih.gov` regarded as instrinsic, while a link between `www.nlm.nih.gov` and `www.nichd.nih.gov` is regarded as transverse. The resulting set of transverse links contains 1,699,834 links (or 15% of all links).

Arguably, pages that do not receive links from other sites will rarely be key resourses. This motivated experiments with anchor-text only runs on three different indexes:

First Anchors Index Only extracting complete link descriptions in the collection. This includes all transverse links, and only a small proportion of intrinsic links (which are usually included as relative locations). All unique anchor-texts are assigned to the document to which the

link points. Considering the dramatic difference in the number of inlinks discussed above, we decided to remove repeated occurrences of the same anchor-text. This resulted in a set of 313,562 anchor-texts covering 186,328 documents, only 15% of the collection.

Second Anchors Index Here we try to recover as many links as possible, by unfolding relative links based on the URL path of the page in which the link occurs, and simplying the resulting URL paths. This includes both intrinsic and transverse links. We again remove repeated occurrences of the same anchor-texts. The result is a set of 1,110,566 anchor-texts covering 667,737 documents, which is 54% of the collection.

Third Anchors Index We use the same procedure as for the second anchors index, but now retain all links as they appear in the collection. Thus, if the same anchor-text occurs thousands of times, we include it thousands of times (similar to [5]). The resulting index is based on 2,766,946 anchor-texts covering 667,737 documents, which is 54% of the collection.

Table 11: Anchors only run results.

Run	Index	MRR	Prec. at 10
UAmsT02WnA'	Anchors 1.	0.1391	
UAmsT02WnA	Anchors 2.	**0.3279**	
UAmsT02WnA"	Anchors 3.	0.3098	
UAmsT02WtA'	Anchors 1.		0.0673
UAmsT02WtA	Anchors 2.		**0.1000**
UAmsT02WtA"	Anchors 3.		0.0837

The post-submission experiments shown in Table 11 show the performance of anchor-text only runs using the three anchor-text indexes. The second anchor-text index, which was used for our official runs, shows the best performance.

We carried out pre-submission experiments using Kleinberg's HITS [10] in order to retrieve key resources for the topic distillation task. Table 12 shows the results for the test topic '*obesity in the U.S.*': the 'Base top 10' are the top 10 results of the text base run; and 'HITS 100' and 'HITS 200' show the top 10 authorities over the top 100 and top 200 documents respectively. Although HITS is successful at isolating key resources, there is a considerable topic drift towards generally good 'authorities.' As is well-known, good authorities and the number of inlinks show considerable correlation [10, 1]. Thus, one can easily image how a loosely-related site with a high indegree can infiltrate in the HITS method. We experimented with a link-based method that tries to avoid such topic drift, by looking at the proportion of inlinks that is in the local set of documents. The top 10 results are also shown in Table 12: 'Realized indegree 100' and 'Realized indegree 200' show the results over the top 100 and top 200 documents of the initial text base run. Informal evaluation shows that our combined approach is much more robust than HITS (by comparing results over different numbers of top documents), for example, when considering the top 500 initially retrieved documents HITS authorities appear almost unrelated to the topics, where as the 'realized indegree' method is still on topic.

Table 12: Test Topic "obesity in the U.S."

Base Top 10
www.surgeongeneral.gov/topics/obesity/calltoaction/4_2.htm
4woman.gov/faq/easyread/obesity-etr.htm
whi.nih.gov/guidelines/obesity/e_txtbk/intro/intro.htm
www.surgeongeneral.gov/topics/obesity/calltoaction/2_0.htm
www.surgeongeneral.gov/topics/obesity/calltoaction/fact_glance.htm
www.surgeongeneral.gov/topics/obesity/calltoaction/principles.htm
www.cdc.gov/nccdphp/dnpa/obesity/trend/maps/
www.nalusda.gov/ttic/tektran/data/000010/76/0000107699.html
www.nalusda.gov/ttic/tektran/data/000010/09/0000100959.html
www.surgeongeneral.gov/topics/obesity/calltoaction/2_2.htm
HITS Top 100
www.nih.gov/icd/od/foia/
www.nlm.nih.gov/
www.nlm.nih.gov/medlineplus/obesity.html
www.nlm.nih.gov/accessibility.html
www.nlm.nih.gov/contacts/
www.nlm.nih.gov/disclaimer.html
www.nichd.nih.gov/
www.nlm.nih.gov/medlineplus/diabetes.html
www.nlm.nih.gov/medlineplus/highbloodpressure.html
www.nlm.nih.gov/medlineplus/sleepdisorders.html
HITS Top 200
www.nih.gov/icd/od/foia/
www.nlm.nih.gov/
www.nlm.nih.gov/medlineplus/obesity.html
www.nichd.nih.gov/
www.nlm.nih.gov/disclaimer.html
www.nlm.nih.gov/accessibility.html
www.nlm.nih.gov/contacts/
www.nlm.nih.gov/medlineplus/diabetes.html
www.nlm.nih.gov/medlineplus/highbloodpressure.html
www.nlm.nih.gov/medlineplus/respiratorydiseasesgeneral.html
Realized Indegree Top 100
www.niddk.nih.gov/health/nutrit/pubs/unders.htm
www.nlm.nih.gov/medlineplus/obesity.html
hin.nhlbi.nih.gov/bmi_palm.htm
www.ahcpr.gov/research/may00/0500RA6.htm
www.nhlbi.nih.gov/guidelines/obesity/bmi_tbl.htm
www.nlm.nih.gov/medlineplus/diabetes.html
www.fitness.gov/Reading_Room/reading_room.html
www.cdc.gov/nccdphp/dnpa/dnpalink.htm
response.restoration.noaa.gov/photos/dispers/dispers.html
www.fda.gov/bbs/topics/NEWS/NEW00575.html
Realized Indegree Top 200
www.nhlbi.nih.gov/health/public/heart/obesity/lose_wt/patmats.htm
www.nlm.nih.gov/medlineplus/obesity.html
hin.nhlbi.nih.gov/bmi_palm.htm
www.niddk.nih.gov/health/nutrit/pubs/unders.htm
www.ftc.gov/bcp/conline/pubs/health/setgoals.htm
www.cdc.gov/nccdphp/dnpa/
www.cdc.gov/health/obesity.htm
whi.nih.gov/health/prof/heart/
www.ahcpr.gov/research/may00/0500RA6.htm
www.ftc.gov/bcp/conline/pubs/health/setgoals.pdf

Earlier attempts at exploiting link structure (in the ad hoc task) failed to show an improvement of retrieval effectiveness [9]. Our experiments with HITS and with the 'realized indegree' method show a decrease in precision at 10 (see Table 10). A possible explanation could be the topics used for the distillation task. These are more specific than the very general topics used in [10], such as '*java*,' '*censorship*,' '*search engines*,' and '*Gates*.' Also, after stopping, the test topic '*obesity in the U.S.*' results in the one-word query '*obesity*.' For such general queries, relevant documents will dominate the top 10, top 100, or even top 200 of initially retrieved documents. Under this assumption, link-based approaches, which ignore the content of documents and solely

consider the link topology, can be effective. If non-relevant documents dominate the initially retrieved set of documents, one cannot expect link-based methods to deliver.

5 Conclusions

In this paper we described our participation in the TREC 2002 Novelty, Question answering, and Web tracks. We set up a baseline system for the Novelty track, and showed that both lemmatizing and combining yield significant improvements for the relevance as well as the novely part. We can look at the novelty part of our system in isolation by assuming perfect output from the relevance part of our system. As it turns out, our system's recall scores for the novelty part are very close to the maximal performance. Our results for the relevance part of the task are less impressive. It seems that the relevance part is the hardest and most important part of the task.

For the question answering track, we experimented with a revised version of our Tequesta system. The main innovation was to introduce document retrieval techniques that were tuned for question answering purposes; in particular, we used high precision settings, together with minimal span matching for each document. In a later stage of the question answering process, MSMs are used to restrict documents to passages which are likely to contain the answer. Our results show considerable differences across question types, which is probably due to quality of the extraction components.

For the web track, we set up a baseline system using separate text and anchor-text indexes. We experimented with the use of non-content features, such as the URL and link structure in the collection for the topic distillation task. Our results failed to show a positive effect on retrieval effectiveness. For the named page finding task, a genuine needle-in-a-haystack task, we experimented with text-only and anchor-text only runs, and their combinations. Here, the combined text/anchor-text run slightly improves the mean reciprocal rank, but significantlty improves the number of topics with the named page in the top 10.

Acknowledgments

Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO), grant # 400-20-036. Christof Monz was supported by the Physical Sciences Council with financial support from NWO, project 612-13-001. Maarten de Rijke was supported by grants from NWO, under project numbers 612-13-001, 365-20-005, 612.069.006, 612.000.106, 220-80-001, and 612.000.207.

References

[1] B. Amento, L. Terveen, and W. Hill. Does 'authority' mean quality? predicting expert quality ratings of web documents. In E. Yannakoudakis, N.J. Belkin, M.-K. Leong, and P. Ingwersen, editors, *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 296–303, 2000.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 107–117, 1998.

[3] C. Buckley, A. Singhal, and M. Mitra. New retrieval approaches using SMART: TREC 4. In D. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 25–48. NIST Special Publication 500-236, 1995.

[4] C. Clarke, G. Cormack, and T. Lynam. Exploiting redundancy in question answering. In Kraft et al. [12], pages 358–365.

[5] N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In Kraft et al. [12], pages 250–257.

[6] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM*, pages 251–262, 1999.

[7] E.A. Fox and J.A. Shaw. Combination of multiple searches. In D.K. Harman, editor, *The Second Text Retrieval Conference (TREC-2)*, pages 243–252. National Institute for Standards and Technology. NIST Special Publication 500-215, 1994.

[8] D.K. Harman. Overview of the TREC 2002 Novelty Track. In *This Volume*.

[9] D. Hawking and N. Craswell. Overview of the TREC-2001 web track. In Voorhees and Harman [21], pages 25–31.

[10] J.M. Kleinberg. Authoritative structures in a hyperlinked environment. *Journal of the ACM*, 46:604–632, 1999.

[11] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In K. Järvelin, M. Beaulieu, R. Baeza-Yates, and S.H. Myaeng, editors, *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and development in information retrieval*, pages 27–34, 2002.

[12] D.H. Kraft, W.B. Croft, D.J. Harper, and J. Zobel, editors. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

[13] J.H. Lee. Combining multiple evidence from different properties of weighting schemes. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 180–188, 1995.

[14] D. Lin. PRINCIPAR—an efficient, broad-coverage, principle-based parser. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, pages 42–48, 1994.

[15] C. Monz and M. de Rijke. Shallow morphological analysis in monolingual information retrieval for Dutch, German and Italian. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Proceedings CLEF 2001*, LNCS 2406, pages 262–277. Springer Verlag, 2002.

[16] C. Monz and M. de Rijke. Tequesta: The University of Amsterdam's texual question answering system. In Voorhees and Harman [21], pages 519–528.

[17] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[18] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[19] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, 1994.

[20] The TREC 2002 novelty track guidelines. `http://trec.nist.gov/act_part/guidelines/novelty_guidelines.html`.

[21] E.M. Voorhees and D.K. Harman, editors. *The Tenth Text Retrieval Conference (TREC 2001)*. National Institute for Standards and Technology. NIST Special Publication 500-250, 2002.