UMass at TREC 2002:
Cross Language and Novelty Tracks

Leah S. Larkey, James Allan, Margaret E. Connell, Alvaro Bolivar, and Courtney Wade

Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA USA 01003

{larkey | allan | connell | alvarob | cwade}@cs.umass.edu

The University of Massachusetts participated in the cross-language and novelty tracks this year. The cross-language submission was characterized by combination of evidence to merge results from two different retrieval engines and a variety of different resources – stemmers, dictionaries, machine translation, and an acronym database. We found that proper names were extremely important in this year's queries. For the novelty track, we applied variants of techniques that have been employed for other problems. In addition, we created additional training data by manually annotating 48 additional topics.

1. Cross Language Track

We submitted one monolingual run and four cross-language runs. For the monolingual run, the technology was essentially the same as the system we used for TREC 2001. For the cross-language run, we integrated some new elements into the Arabic system - a light stemmer that was the result of extensive research [10], the standard probabilistic dictionary based on the UN bilingual lexicon, an expanded dictionary, acronym expansion [12], language modeling, and relevance modeling. In addition, we utilized combination of evidence extensively.

Because our submitted runs were the results of combination of evidence (combining ranked lists from multiple IR runs) we use the term *sub-run* to refer the individual component runs before combination. We first describe the resources, techniques, and models we used, then we describe each run in detail. After presenting our official results, we include some post-hoc runs which correct the errors made in preparing our official submissions, and which help interpret some of the results.

1.1. Arabic and English Resources

The Arabic resources consist of a corpus, normalization and stemming algorithms, and bilingual lexicons. Some of these were developed at UMass and others were provided for TREC participants this year. The English resources consist of a corpus, stemmers, and an acronym facility, as follows:

Arabic Corpus – was the same AFP ARB collection of 383,872 documents in Arabic for TREC-2001 [6] . It was converted to CP1256 encoding, and then indexed in two different ways (1) using UMass normalization and stemming, and (2) using TREC standard normalization and stemming. Only stemmed tokens longer than one byte in length were indexed, and stop words were removed.

UMass Normalization and Light Stemming – These are described in detail in SIGIR02 [10]. In brief, normalization consists of converting to Windows Arabic encoding (CP1256), if necessary, removing punctuation, diacritics, and non-letters, replacing أ, إ, and آ with bare alif ا, replacing final ى with ي, and replacing final ة with ه. Our light stemming algorithm, *light10_stop*, a slight modification of the light stemmer described in [10], consists of stop word removal, stripping definite articles (ال، وال، بال، كال، فال، لل) and و (*and*) from word beginnings and stripping 10 suffixes from word ends (ها، ان، ات، ون، ين، يه، ية، ه، ة، ي). Stopwords were removed using a list of 168 words from Shereen Khoja [8] .

UMass bilingual lexicon – This dictionary is an enhanced version of the dictionary we used for TREC-2001, which was gathered from several sources, as described in [11]. This year we collected more translations from online sources, by including the NMSU proper name dictionary, which used to be available from their Arabic tools page: http://crl.nmsu.edu/~ahmed/downloads.html. In addition, we submitted all the query words from our expanded English queries to two online translation systems, Al Misbar (http://www.almisbar.com/salam_trans.html) and

Ajeeb's Tarjim (http://tarjim.ajeeb.com/ajeeb/). These systems gave one translation (or transliteration) for each term. These were added to the dictionary.

TREC Normalization and Stemming – A Perl script written by Kareem Darwish and modified by Leah Larkey was distributed as the standard stemmer for TREC. It is also a light stemmer, but it is not as light as the UMass stemmer.

TREC bilingual lexicon – We used the bilingual lexicon with probabilities from BBN, derived from the UN parallel corpus. The English words in this dictionary are stemmed with the Porter stemmer, and the Arabic words are stemmed with the TREC standard stemmer. We did not use the bilingual dictionary from Tufts University.

English Corpus – We used AP news articles from 1994 through 1998 in the Linguistic Data Consortium's NA News corpus for an English background language model, and for English query expansion.

English stop words and stemming - English stop words are from INQUERY's standard list of 418 stop words. English stop phrases are defined by regular expressions in a script we have used before in TREC (in English). In order to use the BBN probabilistic dictionary we performed Porter stemming on the queries. In sub-runs that did not use the BBN dictionary, we left the English unstemmed, stemming with *kstem* [9] only when the English word was not found in the dictionary.

Acronym Expansion – In the cross-language runs, we looked up in the *Acrophile* system [12] any sequences of all-capital letters. The top-ranking expansion for each acronym was added to the query.

1.2. Information Retrieval

We used INQUERY [5] for the monolingual run and two of the cross-language sub-runs, and language modeling (LM) for the rest of the crosslingual sub-runs. For both English and Arabic, text was broken up into words at any white space or punctuation characters. For Arabic, there were five additional Arabic punctuation characters included in the definition of punctuation. Words of one-byte length (in CP1256 encoding) were not indexed.

INQUERY - The monolingual run and some of the cross-language runs used a version of INQUERY as the search engine. This version computes the belief function reported in UMass's TREC9 report [2]. The main difference between this version and "real" INQUERY is that proximity information is not stored in the index, so that INQUERY operators requiring proximity information are not implemented.

For cross-language INQUERY retrieval, English queries are translated to Arabic as follows. For each English word in a query, do the following: find the set of all translations in the dictionary. If the English word is not found, stem the English word using the *kstem* stemmer and look it up again. Stem the Arabic translations. If any of the translations consist of an Arabic phrase rather than a single word, enclose the phrase in a **`#filreq`** operator. Enclose all the alternative Arabic translations for a single English word under a **`#syn`** (synonym) operator. Finally, take all the **`#syn`** sets and build a weighted sum query out of all the stemmed translations of the query terms by subsumimg all the synonym sets under a **`#wsum`** (weighted sum) operator. Each synonym set was given the weight described in the query expansion section.

Crosslingual Language Modeling (LM) - In language modeling, documents are represented as probability distributions over a vocabulary. Documents are ranked by the probability of generating the query by randomly sampling the document model. The language models here are simple unigram models, similar to those of [14] . Unigram probabilities in our official run were estimated as a mixture of maximum likelihood probability estimates from the document and the corpus, as follows:

$$P(Q_e \mid D_a) = \prod_{e \in Q_e} \left((.7) \sum_{a \in Arabic} P(a \mid D_a) P(e \mid a) + (.3) P(e \mid GE) \right)$$

where $P(Q_e|D_a)$ is the probability of generating the query from the document model, the e's are the English words in the query, $P(a|D_a)$ is the probability of an Arabic word a in the document D_a, $P(e|a)$ is the translation probability of seeing the English query word e given the presence of Arabic word a, and $P(e|GE)$ is the probability of the English query word in the English background model. $P(a|D_a)$ is estimated as $tf_{a,Da}/|D_a|$ where $tf_{a,Da}$ is the number of occurrences of term a in Arabic document D_a and $|D_a|$ is the length of document, that is, the number of total term occurrences in the document. The translation probability $P(e|a)$ comes from the bilingual lexicon. If the lexicon was derived from a parallel corpus, these probabilities represent the proportion of the time that Arabic word a was aligned with English word e in the parallel corpus. If the lexicon is a dictionary, and Arabic word a has n different

translations into English $e_1,...e_n$, then $P(e|a)$ is estimated as $1/n$. The background probability $P(e|GE)$ is estimated as $P(e \mid GE) = df_{e,C} \Big/ \sum_{t \in C} df_{t,C}$ where $df_{e,C}$ is the number of English documents in C containing term e, and the summation is over all the terms in the English collection, as in [7].

Query Expansion - We expanded English queries in some of our cross-language sub-runs using the AP news articles from 1994 through 1998 in the Linguistic Data Consortium's NA News corpus. This corpus was indexed without stemming, but normalized to lower case. We retrieved the top 10 documents for each query. Terms from these documents received an expansion score which was the sum across the ten documents of the INQUERY belief score for the term in the document. The 5 terms with the highest expansion score were added to the query. Final term weights were set to $2w_o + w_e$ where w_o is the original weight in the unexpanded query and w_e=1.

Arabic query expansion was handled in different ways for INQUERY sub-runs and LM sub-runs. For INQUERY sub-runs, Arabic query expansion was just like English query expansion, except the top 10 documents were retrieved from the Arabic corpus, rather than the English corpus, and 50 terms, not 5, were added to the query.

In language model sub-runs, query expansion was carried out using relevance modeling [13]. The best matching fifty documents were retrieved and 500 words were selected as the new query. Associated with each word is an estimated probability of observing this word in the relevant documents.

Combination of Evidence – Sub-runs were combined into submitted runs by normalizing document scores in ranked lists, and then summing lists two at a time. Score normalization was a linear min-max normalization where $score_{norm}$=(*score-min*)/(*max-min*).

1.3. Monolingual run

Our one monolingual run was designated **UMassM**, and carried out as follows:

1. Convert queries to CP1256 encoding
2. Extract titles and descriptions from topics.
3. Remove stop phrases, using a script developed for TREC2001.
4. Stem the query with the UMass light stemmer and remove stop words.
5. Expand the query by adding the best 50 words from the top ranking 10 documents.
6. Retrieve the top 1000 documents using INQUERY.

Overall average precision on this run was .3619. The per-query comparison among the 18 submitted monolingual runs suggests that our approach favored recall over precision. We performed at or above the median average precision on 34 of the 50 queries, and below the median on 16 queries. In number of relevant documents retrieved in top 1000, we were at or above the median in 44 or the 50 queries, and at the highest number in 28 queries.

1.4. Cross-language runs

Our cross-language submissions relied heavily on combination of evidence this year. We used two different dictionaries that could not be combined because they were built with different stemmers. In our own dictionary, a composite of a variety of different sources, English was not stemmed, and Arabic words were stemmed using the UMass light stemmer. In the standard probabilistic dictionary, (the bilingual lexicon built at BBN using the UN parallel corpus), Arabic words were stemmed using the Darwish stemmer, and English was stemmed with the Porter stemmer. We ran several independent retrieval sub-runs and combined their ranked lists at the end.

Each sub-run used one of two different retrieval engines – INQUERY or LM, one of two different resource sets: UMass dictionary and stemmer, or TREC standard probabilistic dictionary and stemmer, one of 3 different query expansion options: English only, Arabic only, or English and Arabic, and one of two different selections from the topic: title and description, or title, description, and narrative.

The steps were as follows: (For sub-runs)

1. Select text from titles and descriptions, or titles, descriptions, and (in X2n and X6n conditions) select capitalized words from narrative field.
2. Remove stop phrases from English queries.
3. Expand acronyms
4. Lower case the query

5. Expand the English query (optional – depends on condition)
6. Expand the Arabic query (optional – depends on condition)
7. Retrieve top 2000 documents

Twelve different crosslingual conditions were run. These sub-runs were combined into four cross language runs that can be briefly described as follows:

1. **UMassX2** – combination of 2 INQUERY cross language sub-runs both using title and description fields.
2. **UMassX6** – combination of all 6 cross language sub-runs using title and description fields.
3. **UMassX2**n – combination of 2 INQUERY sub-runs using title, description, and narrative fields.
4. **UMassX6n** – combination of all 6 cross language sub-runs using title, description, and narrative fields.

Table 1 lists the resources used in each of the 12 sub-runs, and indicates which sub-runs composed each submitted run. Names of submitted runs are abbreviated e.g. *UMassX2* as *X2*, etc.

Table 1: Definition of sub-runs

Sub-run	Compo-nent of	Engine	Dictionary + Stemmer	Expansion	Narrative Included?
Inq-UM-EngAr	X2, X6	INQUERY	UMass	English+Arabic	no
Inq-TREC-EngAr	X2, X6	INQUERY	TREC	English+Arabic	no
LM-UM-Ar	X6	LM	UMass	Arabic	no
LM-UM-Eng	X6	LM	UMass	English	no
LM-TREC-Ar	X6	LM	TREC	Arabic	no
LM-TREC-Eng	X6	LM	TREC	English	no
Inq-UM-EngAr-Nar	X2n, X6n	INQUERY	UMass	English+Arabic	yes
Inq-TREC-EngAr-Nar	X2n, X6n	INQUERY	TREC	English+Arabic	yes
LM-UM-Ar-Nar	X6n	LM	UMass	Arabic	yes
LM-UM-Eng-Nar	X6n	LM	UMass	English	yes
LM-TREC-Ar-Nar	X6n	LM	TREC	Arabic	yes
LM-TREC-Eng-Nar	X6n	LM	TREC	English	yes

1.5. Cross-Language Results

Table 2 summarizes the results of our official submissions. Combination of results based on different resources improved performance. Inclusion of narrative words also improved performance a great deal. UMassX6n had the highest mean average precision of all the officially submitted cross language runs in TREC 2002.

Table 2: Monolingual and Cross-language Results: official runs

Name of Run	Official Mean Average Precision	Number of Queries at or Above Median Average Precision	Rel Ret in top 1000	Post hoc Average Precision
UMassM	.3619	34/50	44/50	.3619
UMassX2	.3538	30/50	40/50	.3589
UMassX6	.3658	36/50	42/50	.3801
UMassX2n	.3900	35/50	37/50	.3941
UMassX6n	.3996	39/50	41/50	.4107

1.6. Post hoc Cross-Language Experiments

Post hoc experiments were performed first, to correct some errors in our official submissions, second, to provide a "standard resources" run, and third, to explore the role of acronyms, proper names, and stemming of the UN parallel corpus.

1.6.1. Fixing Errors

We discovered two problems after submitting our results. First, the Porter stemmer used in processing the BBN bilingual corpus was different from our version of the Porter stemmer, so that many English words were not found in the probabilistic dictionary. For example, *contrary* was stemmed to *contrari* by the BBN Porter stemmer, but to

contrar by our Porter stemmer. *Money* became *monei* under the BBN Porter stemmer, but remained *money* under our Porter stemmer. When we reran the sub-runs with compatible Porter stemming, results on those sub-runs improved, as did the combinations that included them.

A second problem resulted from a procedural error, in which the language model runs (but not the INQUERY runs) using the UMass resources were run with an older, smaller version of the dictionary. We therefore reran the affected conditions with the correct dictionary, and obtained the results shown in the last column of Table 2.

Overall, the greater coverage of the dictionaries and the use of compatible versions of the Porter stemmer improved performance. The overall patterns remained the same - combination of resources improved performance, and retrieval was more effective when the narrative portion was included in the query.

1.6.2. Standard Resources Run

Although we did not submit an official standard resources run we ran one later for this report. Recall that one of the two sub-runs that made up **UMassX2 and UMassX2n,** and three of the six sub-runs that made up **UMassX6** and **UMassX6**n, used the standard parallel corpus dictionary and stemmer. The ***Standard Resources*** column of Table 3 shows the results when the sub-runs based on the UMass resources and acronym expansion were excluded. Only the sub-runs based on the standard resources were included. Thus, in the ***Standard Resources*** column, the **UMassX2** and **UMassX2n** rows show the results of a single sub-run, and **UMassX6** and **UMassX6**n rows each show results based on a combination of three, rather than six, sub-runs. Relative to these three way combinations, the additional resources increased average precision 3 percentage points for title+description queries, and 4 points for title+description+narrative queries.

1.6.3. Acronym expansion

Because this was the first time acronym expansion was used in the TREC cross-lingual track, we assessed its contribution separately. We reran the **UMassX6** and **UMassX6n** runs without expanding acronyms. The results, shown in the ***No Acro*** column of Table 3, revealed that acronym expansion added almost nothing. Analysis of individual queries containing acronyms revealed that while acronym expansion helped on some queries, it hurt on others.

1.6.4. Importance of Proper Names

It had struck us in informal query analyses that proper names were extremely important in these queries. We hypothesized that successful retrieval depended upon having these proper names in the lexicon. In order to test this, we identified all the proper names (people, places, organizations, acronyms) in our queries and in our expanded English queries, and made special versions of the UMass and BBN dictionaries from which these names had been removed. The column labeled ***No Names*** in Table 3 gives the results. Performance dropped more than 50% when names were not available in the dictionaries. We speculate that one reason dictionaries derived from parallel corpora work so well is that they cover so many more proper names than do static dictionaries.

Table 3: Post hoc average precision: Cross Language runs

Name of Run	Official	Errors Corrected	Standard Resources	No Acro	No Names	Reprocessed UN Corpus
UMassX2	.3538	.3589	.3141	.3577	.1561	.3668
UMassX6	.3658	.3801	.3508	.3795	.1629	.3948
UMassX2n	.3900	.3941	.3337	.3936		.3983
UMassX6n	.3996	.4107	.3724	.4104		.4189

1.6.5. Reprocessing the UN Corpus

Although the standard parallel corpus dictionary as processed by BBN was a very valuable resource, we found it awkward to fit into the rest of our system because of the Porter stemming used on the English words, and because of the small differences between the standard Arabic stemmer and the UMass light stemmer. We reprocessed the UN corpus, obtained from LDC, using the same configuration Alex Fraser used at BBN, except we used different preprocessing in preparing the English and Arabic input to GIZA++. English words were lower cased, and stop words were removed. Arabic words were stemmed and stop words removed using the UMass (*light10_stop*) stemmer. Retrieval results comparing the two versions of the parallel corpus dictionary can be seen in Table 4. Table 4 contains only sub-runs and combinations that used the parallel corpus dictionary. It does include any runs

that used the UMass dictionary. The improvement in the official runs that include all the resources can be seen in Table 3, above, in the ***Reprocessed UN Corpus*** column.

Table 4: Mean average precision using two dictionaries made from the UN parallel corpus.

Type of Run		TREC Standard Dictionary	Reprocessed UN Corpus
INQUERY	title+description	.3161	.3413
LM	title+description	.3464	.3637
INQ + LM	title+description	.3520	.3678
INQUERY	title+desc+narrative	.3350	.3614
LM	title+desc+narrative	.3605	.3804
INQ+LM	title+desc+narrative	.3734	.3880

2. Novelty Track

Our attempts to find relevant and novel information focused on variants of techniques that have been employed for other problems [1] [3] [4]. Basically, we looked for relevant sentences by comparing them to the query, and we looked for redundancy by estimating whether a sentence was dissimilar from all prior (relevant) sentences. We found that the task of recognizing relevant sentences was the major challenge and that our errors there account for poor overall performance.

One notable feature of the CIIR's participation in the novelty track is our creation of additional training data. We hired students to annotate 48 topics in addition to the handful provided by NIST. Details on that process are discussed below.

2.1. Creating additional training topics

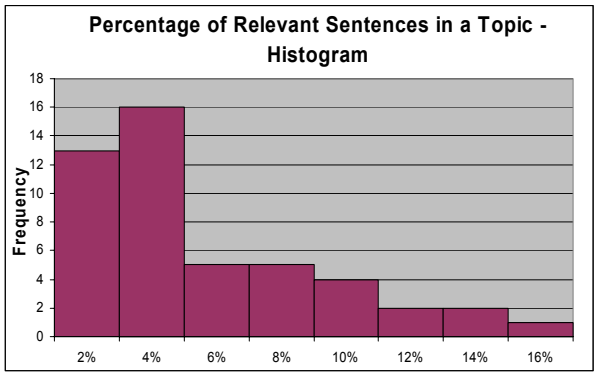
For the purpose of this evaluation we built our own collection of documents and sentence level relevance judgments. We randomly chose 48 topics from the TREC-7 and TREC-8 ad-hoc retrieval tracks (topics 300-450)—though we were careful not to use the topics set aside for the novelty track evaluation. For each of the 48 topics, we used the INQUERY search engine to find the top-ranked documents in a subset of TREC volumes 4 and 5 (Federal register 1994, LA Times 1989-90, and FBIS 1996). We selected the top 25 known-relevant documents (based on TREC judgments) for each topic.

We followed the same methodology defined by the novelty track to collect relevance and novelty assessments. We hired undergraduate students, who were otherwise unaffiliated with our research, to read the relevant documents for each topic in rank order. They extracted relevant and novel sentences from each topic in a two step process. First, the assessors were told to read a printed copy of the relevant documents and highlight the relevant sentences. Second, they read the sentences marked as relevant again and flagged the ones that contained novel information. For this process, order is very important. By definition, a sentence is novel if it provides totally new information or further details on previously seen information. Instances that summarize details seen earlier in the document stream were not considered novel.

Some statistics for the constructed training data are presented in Figure 1 and Table 5. These statistics are consistent with the NIST-provided training topics as well as the evaluation topics. Interestingly, for the average topic less than 5% of the sentences contain relevant material. Further, more than 80% of the relevant sentences on average contain novel information. That is, most of the material is non-relevant and most of the relevant material is novel. The material for reconstructing our data and some additional statistics about the data are available at [http://ciir.cs.umass.edu/downloads/access.html]. That material assumes that TREC volumes 4 and 5 are available to the user.

Table 5: Collection statistics summary for training set

TOPICS	# ASSESSORS	DOCS	SENT	NOVEL	REL	REL/SENT	NOVEL/REL
48	6	1122	84588	2759	3400		
	AVERAGE/TOPIC	23.38	1762	57.48	70.83	4.6%	81.6%
	σ	3.923	797.22	46.217	55.63	3.64%	12.74%



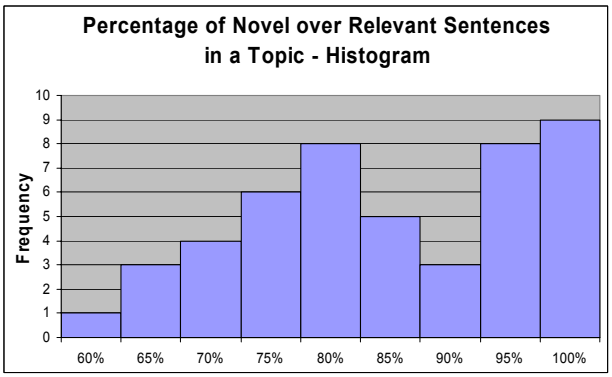


Figure 1: Histograms of distribution of (a) the percentage of relevant sentences over the total number of sentences and, (b) the percentage of novel sentences in terms of the number of relevant sentences.

We were curious about the impact of time on a person's conception of novelty. For instance, we were interested in answering question like the following: How far could an assessor go without losing track of the mental representation of a novel sentence with respect to a particular topic? What is the behavior of the novelty rate as more documents are added to the knowledge base? To answer this questions, one of the assessors was told to read through a bigger set of documents (75 documents) carrying out the relevance and novelty judgments steps as they were explained before. The topic chosen for this experiment was 422 (Figure 2). The results are presented in Table 6.

```
<top>
  <num> Number: 422
  <title> art, stolen, forged
  <desc> Description:
  What incidents have there been of stolen or forged art?
  <narr> Narrative:
  Instances of stolen or forged art in any media are relevant.  Stolen mass-  produced
things, even though they might be decorative, are not relevant (unless they are mass-
produced art reproductions). Pirated software, music, movies, etc. are not relevant.
</top>
```

Figure 2: Topic 422's description.

Table 6: Annotation information for topic 422. The first row presents statistics for an assessor who was asked to annotate 75 documents rather than just 25.

TOPICID	ANNOT	DOCS	SENT	NEW	REL	REL/SENTS	NEW/REL
422	D	75	3593	164	181	5.0%	90.6%
422	D	25	2065	89	94	4.6%	94.7%
422	C	25	2065	75	77	3.7%	97.4%

The assessor in charge of this task reported that it was not difficult to assess the relevance or novelty of a sentence even with this many documents, because it was fairly easy to maintain a clear sense of the topic definition and when new information about the topic was appearing.

Table 6 shows an increase for the percent of relevant sentences across different assessors and different evaluation set sizes. The difference is presumably accounted for by normal inter-annotator disagreement. Interestingly, though, the proportion of new material decreases with more sentences judged. Our intuition tells us that as more

documents are processed and our knowledge base about the topic increases, the tendency to find new sentences should be greatly reduced: the topic should be fully covered. The results from trying one topic in more detail might be explained by a lack of redundancy in the collection or by the intrinsic property of the topic to constantly generate new events related to the same topic (and thus novel sentences). Topics defined the way topic 422 is defined admit constant generation of new events that must create new relevant and novel sentences.

We were also concerned by the low number of relevant sentences and wondered if this was the result of the annotation instructions. We told one of our assessors to read the relevant documents for topic 327 and topic 417 (Figure 3) and identify the sentences that do *not* provide any relevant information whatsoever in relation to the topic. The results are presented in Tables 7 and 8.

```
<top>
<num> Number: 327
<title> Modern Slavery
<desc> Description:
Identify a country or a city where there
is evidence of human slavery being
practiced in the eighties or nineties.
<narr> Narrative:
A relevant document would present
evidence of current slavery practices
being carried out.  It would identify a
specific country or city and give some
information on who the slaves are, or who
was buying or selling them, and for what
purposes they were being used.
References to slavery being carried out
several years ago would not be relevant.
</top>
```

```
<top>
<num> Number: 417
<title> creativity
<desc> Description:
Find ways of measuring creativity.
<narr> Narrative:
Relevant items include definitions of
creativity, descriptions of
characteristics associated with
creativity, and factors linked to
creativity.
</top>
```

Figure 3: Topic description for topic 327 and topic 417

Table 7: Comparison of assessor results when one assessor located relevant sentences and the other locates non-relevant sentences. Upper right and lower left portions of each table represent disagreement

	TOPIC 327	ASSESSOR D			
		NOT JUDGED		NOT REL	
ASSES. F	RELEVANT	19	4.2%	40	8.8%
	NOT JUDGED	8	1.8%	389	85.3%

	TOPIC 417	ASSESSOR D			
		NOT JUDGED		NOT REL	
ASSES. A	RELEVANT	20	0.9%	9	0.4%
	NOT JUDGED	103	4.4%	2215	94.4%

Table 8: Comparison of sentences judged by two assessors in selected topics

TOPICID	ASSESSOR	*		*/S
327	A	59	Relevant	12.9%
	D	429	Not Relevant	94.1%

Documents: 11 Sentences: 456

TOPICID	ASSESSOR	*		*/S
417	F	29	Relevant	1.2%
	D	2224	Not Relevant	94.8%

Documents: 25 Sentences: 2347

Table 7 shows the inconsistencies between two assessors, one finding relevant and one finding non-relevant. For topic 327, for example, the upper-right cells indicate that 40 sentences were explicitly identified as relevant *and* as non-relevant. Presumably those reflect inter-annotator disagreement, though we have not adjudicated the results to see whether there are errors. Nineteen of the sentences were judged relevant by both approaches, and almost 86% of the sentences were consistently judged non-relevant.

Table 8 shows for topic 327 that an assessor finding relevant sentences found that 12.9% of the sentences were relevant, whereas an assessor looking for non-relevant implicitly found only 5% of the sentences relevant. The difference is surprising, particularly since we were expecting that the "find non-relevance" assessor would "find"

substantially more relevant material. On the other hand, for topic 417 we obtained a more reasonable result with a very low 0.4% of inter-annotator disagreement. These differences among topics may be explained by the information need that the topic describes, which for topic 327 seems to be pretty ambiguous.

Similar experiments on a larger collection of topics might make clearer what is happening.

2.2. Experiments finding relevant sentences

In order to extract the sentences with relevant information, we used the traditional IR ranking approach with three different retrieval models. Pseudo-relevance feedback (PRF) was used with each approach and executed depending on the retrieval model being used.

1. ***TFIDF***. Here, we tried the traditional TFIDF approach. Given a query *q* and sentence *s*:

$$score(s) = \sum_{t \in q} \log(tf_{t,q} + 1) \log(tf_{t,s} + 1) \log\left(\frac{n+1}{.5 + sf_t}\right)$$

   $tf_{t,q}$ and $tf_{t,s}$ are the number of times term t occurs in the query and sentence, respectively, sf_t is the number of sentences in which term t appears, and n is the number of sentences in the collection.
2. ***Simple Language Modeling (KLD).*** Given smoothed language models of a query *q* and a sentence *s*, the score is given by the Kullback-Leibler divergence (KLD) between the two mass distributions:

$$score(s) = KLD(q \,||\, s) = \sum_{w \in V} p(w \,|\, q) \cdot \log \frac{p(w \,|\, q)}{p(w \,|\, s)}$$

3. ***Two-Stage Smoothing in Language Modeling.*** The two-stage smoothing method allows the use of different smoothing techniques for the query and the sentence language models. This is used in order to differentiate the two roles that smoothing plays in the retrieval process. For the sentence, smoothing assigns non-zero probabilities to words not present in the sentence. For the query, smoothing "explains away" the common non-topic words in the query. This approach is extensively explained in [15].

Multiple runs were carried out on our training topics in order to tune the multiple parameters in the different retrieval models. Results from the best runs are presented in Figure 4. Although TFIDF with PRF had the best average precision, its performance is not significantly different (student's t-test) from the other models. For all models, performance of the retrieval process at sentence level was poor and very hard to improve.

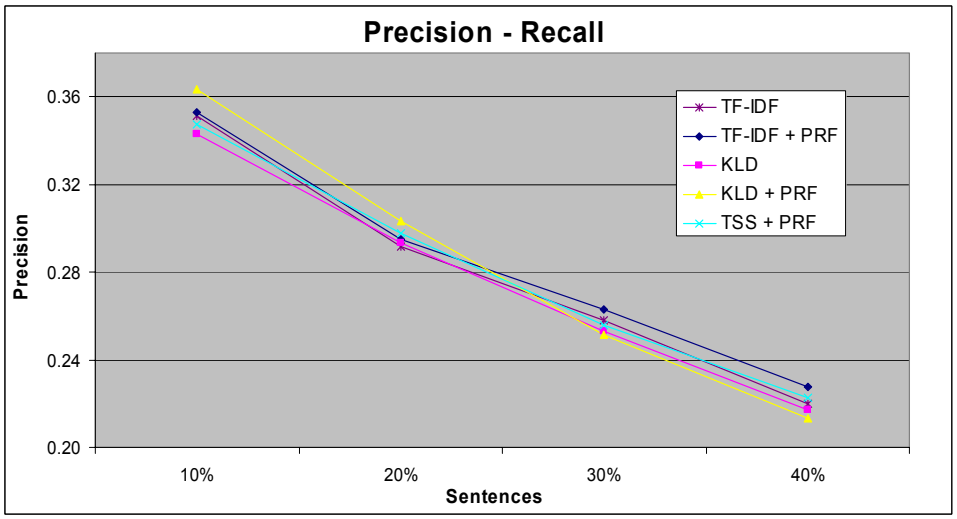


Figure 4: Results for sentence relevance retrieval on the training topics

We also explored the influence of query length on performance. We used "short" queries and "long" queries. For every case, we tried using only some portion of the topic description. For short queries we tried using only the topic title or the topic title and description. As long queries, we tried using the topic title, description, and narrative. Our

results were consistent with the results reported in [15]. On average, it is better to use the complete topic text as opposed to using only portions of it.

Some additional experiments were carried out, without much success, to try to improve performance by means of query preprocessing. A standard ad-hoc query algorithm [2] was run on the query text (topic text) for the different topics. The goal of this algorithm was to get rid of what we believed were query stop words and stop phrases. We believed that words like *"narrative"* and *"description"* as well as phrase patterns such as *"A document that discusses word [word ... word] is considered non-relevant"* should not be present in the query. Contrary to our intuition, the use of this algorithm did not improve the retrieval performance significantly.

2.3. Looking for novelty

Our novelty detection systems take as input the ranked list of relevant sentences for each topic (i.e., the output of the previous step). We decided to use the vector space retrieval model with TFIDF weighting and pseudo-relevance feedback as we found that it was the method that worked best with our training data. Both of our systems (CIIR02tfkl and CIIR02tfnew) use this method to identify the relevant sentences. We determined through trial and error on the training data that our results were best if we used the top 10 percent of relevant sentences, even though we know from the training topics that only about 5% of the sentences are likely to be relevant. These top relevant sentences are re-indexed using Lemur.[1] No stopping or stemming is used at this stage.

Our two systems assign novelty scores to sentences in different ways:

CIIR02tfkl. The CIIR02tfkl system uses the KL-divergence between two language models as its scoring method. For each topic, the documents are considered in the order given by the task and novelty scores are assigned using the following method. The first relevant sentence of the first relevant document is assigned a maximum score. For the remaining sentences we calculate a collection language model and a sentence language model in Lemur:

collLM(i) (the collection language model for sentence i) is a maximum likelihood model built on sentences 1 to (i-1), smoothed using linear interpolation against a maximum likelihood model built on sentences 1 to i.

sentLM(i) (the sentence language model for sentence i) is a maximum likelihood model built on sentence i, smoothed using linear interpolation against a maximum likelihood model built on sentences 1 to i.

Sentence i's score is the KL-divergence between its collection and sentence language models, KLD(sentLM(i) || collLM(i)).

We set the smoothing parameters for both language models so that almost no smoothing occurs. These parameter settings were chosen because they achieved the best results on our training data.

CIIR02tfnew. The CIIR02tfnew system assigns novelty scores in a very simple way. For each topic, it considers the documents in the order specified by the task. Each sentence is treated as a set of words and a sentence's score is equal to the number of new words in the set (i.e., words that have not appeared so far in the sentences for that topic).

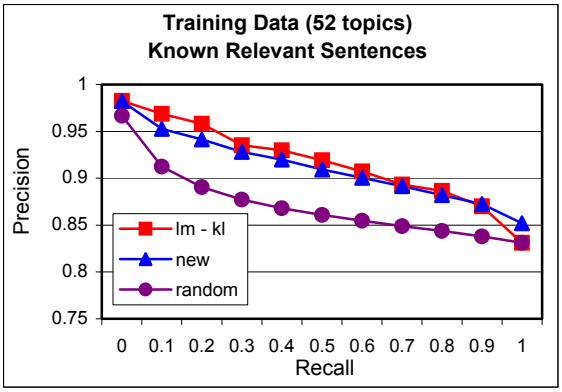
We observed that in our training data, approximately 80 percent of the sentences judged relevant by the annotators were also judged new. Therefore, both of our systems return the top 80 percent of the ranked list of novelty scores as new.

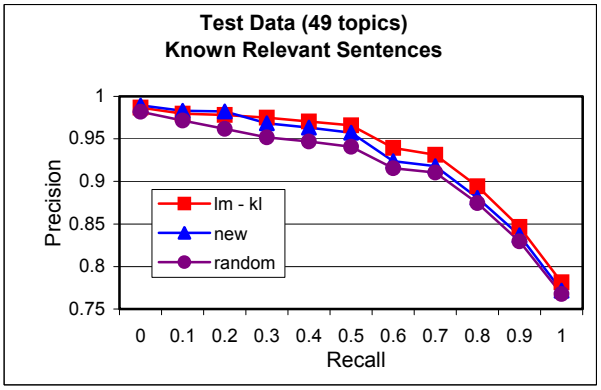
Interestingly, for both the training and test data, when we ran our two systems on the collection of *known* relevant sentences (i.e., we cheated), CIIR02tfkl performed better than CIIR02tfnew. However, when we ran these systems on our own relevance results (i.e., relevance results with errors), CIIR02tfnew performed better than CIIR02tfkl. We hypothesize that non-relevant sentences are likely to be identified as novel and that CIIR02tfkl models novelty better than CIIR02tfnew and therefore tends to pull those non-relevant sentences towards the top of the novelty rankings. We have not sufficiently investigated this issue, however. It may, for example, be nothing more than a statistical anomaly.

The following graphs show the effectiveness of the two techniques when only relevant sentences are ranked. Note that even a random ranking of these sentences does quite well, because about 80% of them are novel.

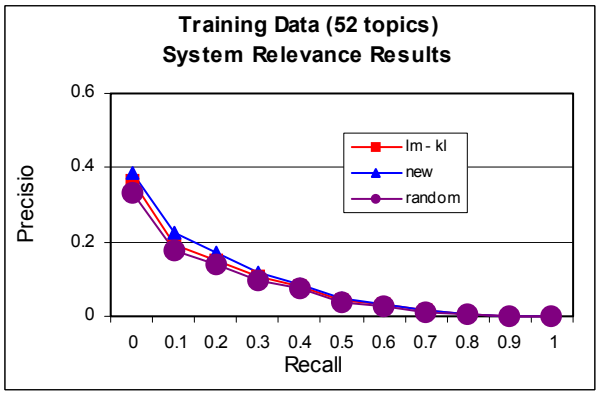
---

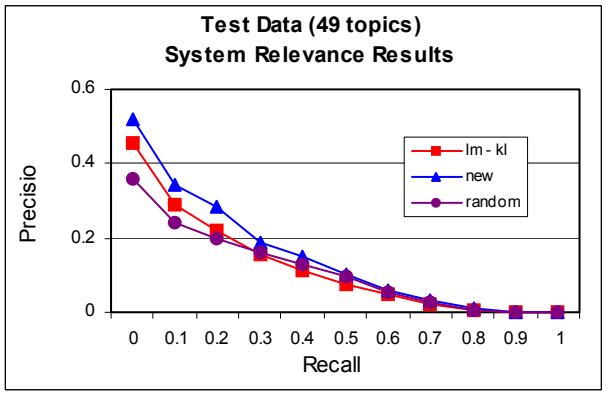
[1] Lemur is a toolkit from CMU and UMass Amherst intended to support the use of language modeling techniques for information retrieval. It is freely available for research purposes at http://www.cs.cmu.edu/~lemur.



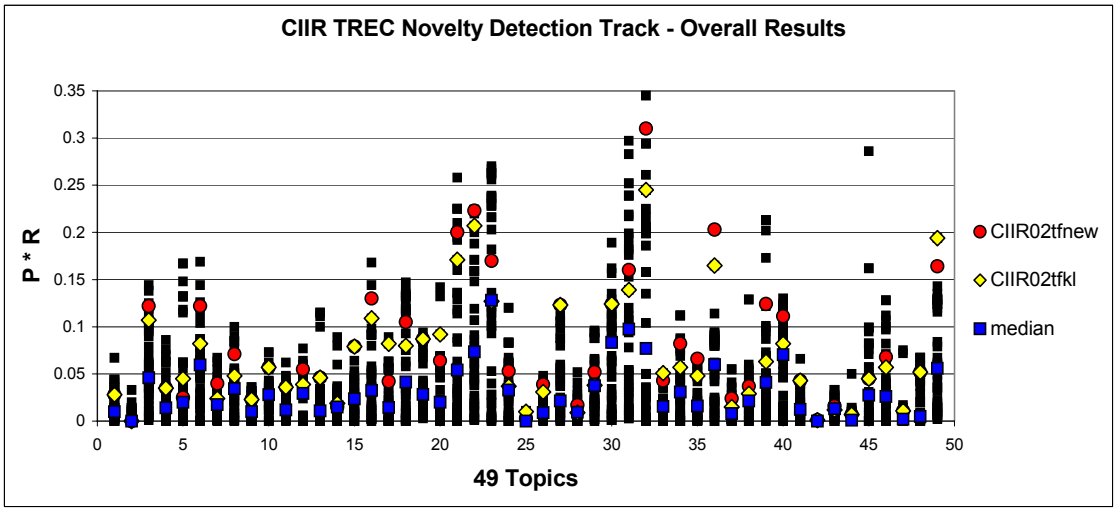


The next graphs provide the same information for when the "relevant" sentences are chosen using one of the approaches described above—i.e., generated by a system. Overall performance drops substantially because the quality of the initial retrieval is poor (the scale of the axes is dramatically different).





This final graph shows how the results of our official submissions compared to other submissions of the TREC and shows on a query-by-query basis how our two approaches stacked up.[2]



[2] This graph compares systems based on precision * recall, the evaluation measure used at the TREC workshop. After the workshop, the evaluation metric was changed to the F measure.

3. Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903, and in part by Advanced Research and Development Activity under contract number MDA904-01-C-0984. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

4. References

[1] Allan, J. Introduction to topic detection and tracking. In *Topic detection and tracking: Event-based information organization*, J. Allan, Ed.: Kluwer Academic Publishers, pp. 1-16, 2002.

[2] Allan, J., Connell, M. E., Croft, W. B., Feng, F.-f., Fisher, D., and Li, X. INQUERY and TREC-9. Presented at The Ninth Text REtrieval Conference (TREC-9), Gaithersburg, Maryland, 2000.

[3] Allan, J., Gupta, R., and Khandelwal, V. Temporal summaries of news topics. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 10-18, 2001.

[4] Allan, J., Jin, H., Rajman, M., Wayne, C., Gildea, D., Lavrenko, V., Hoberman, R., and Caputo, D. *Topic-based novelty detection: 1999 summer workshop at clsp*., Final Report available on-line at http://www.clsp.jhu.edu., 1999.

[5] Callan, J. P., Croft, W. B., and Broglio, J. TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, 31 (3), pp. 327-343, 1995.

[6] Gey, F. C. and Oard, D. W. The TREC-2001 cross-language information retrieval track: Searching Arabic using English, French, or Arabic queries. In *TREC 2001*. Gaithersburg: NIST, 2002.

[7] Hiemstra, D. and de Vries, A. *Relating the new language models of information retrieval to the traditional retrieval models*. University of Twente, Enschede, The Netherlands, CTIT Technical Report TR-CTIT-00-09, May 2000 2000.

[8] Khoja, S. and Garside, R. *Stemming Arabic text*. Computing Department, Lancaster University, Lancaster, U.K. http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps, 1999.

[9] Krovetz, R. Viewing morphology as an inference process. In *Proceedings of the sixteenth annual international ACM SIGIR conference on research and development in information retrieval*, pp. 191-203, 1993.

[10] Larkey, L. S., Ballesteros, L., and Connell, M. E. Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. In *SIGIR 2002: The twenty-fifth annual international ACM SIGIR conference on research and development in information retrieval*. Tampere, Finland: ACM, pp. 275-282, 2002.

[11] Larkey, L. S. and Connell, M. E. Arabic information retrieval at UMass in TREC-10. In *TREC 2001*. Gaithersburg: NIST, 2001.

[12] Larkey, L. S., Ogilvie, P., Price, M. A., and Tamilio, B. Acrophile: An automated acronym extractor and server. In *Digital libraries '00 - the fifth ACM conference on digital libraries*. San Antonio, TX: ACM Press, pp. 205-214, 2000.

[13] Lavrenko, V., Choquette, M., and Croft, W. B. Cross-lingual relevance models. In *SIGIR 2002: The twenty-fifth annual international ACM SIGIR conference on research and development in information retrieval*. Tampere, Finland: ACM, pp. 175-182, 2002.

[14] Xu, J., Weischedel, R., and Nguyen, C. Evaluating a probabilistic model for cross-lingual information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval.* New Orleans: ACM Press, pp. 105-110, 2001.

[15] Zhai, C. and Lafferty, J. Two-stage language models for information retrieval. In *SIGIR 2002: The twenty-fifth annual international ACM SIGIR conference on research and development in information retrieval*. Tampere, Finland: ACM, pp. 49-56, 2002.