Experiments in Novelty, Genes and Questions at The University of Iowa

David Eichmann,[1,2] Padmini Srinivasan,[1,3] Marc Light,[1,4]
Hudong Wang,[2] Xin Ying Qiu,[3] Robert J Arens[2] and Aditya Sehgal[2]
[1]School of Library and Information Science
[2]Computer Science Department
[3]Department of Management Sciences
[4]Department of Linguistics
The University of Iowa
{david-eichmann, padmini-srinivasan, marc-light}@uiowa.edu

The University of Iowa participated in the novelty, genomics and question answering tracks of TREC-2003.

1 – Novelty

Our system for novelty this year is a refinement of that used for last year. One of the challenges in preparing for the 2002 novelty track was the nature of the training data. Our experiments with using the 2002 evaluation data as training data for this year have shown that the novelty task can in fact be tuned to trade off precision and recall - at least across the range of what a given system can detect as novel. Our tuning involved establishing a similarity threshold for sentence relevance and an new entity threshold for novelty.

We decided to focus our development experiments for this year on a composite precondition of simple similarity matches between the topic definition and the candidate document and the topic and the candidate sentence. If both measures exceed the declared threshold, a sentence is declared relevant. Additionally, if the number of novel elements present in the sentence is above a declared number, the sentence is declared novel. 'Element' here can be a noun phrase or a named entity. For the available training topics, this proved to be remarkably responsive to tuning between precision-focused runs and recall-focused runs for novelty as well as the more predictable relevance decision.

Our official runs involved the following approaches for the defined tasks:

Task 1 (detect relevance and novelty). Proceed as described above, making a judgement on relevance based upon similarity, and given that as a guard, make a judgement on novelty based upon the existence of new entities.

Task 2 (given relevance, detect novelty). Load the given relevance judgements, and proceed as per task 1 for novelty.

Task 3 (given relevance and novelty for first 5, detect relevance and novelty for last 20). Load relevance judgements and entities present in the first five documents, and then proceed as per task 1 for both relevance and novelty.

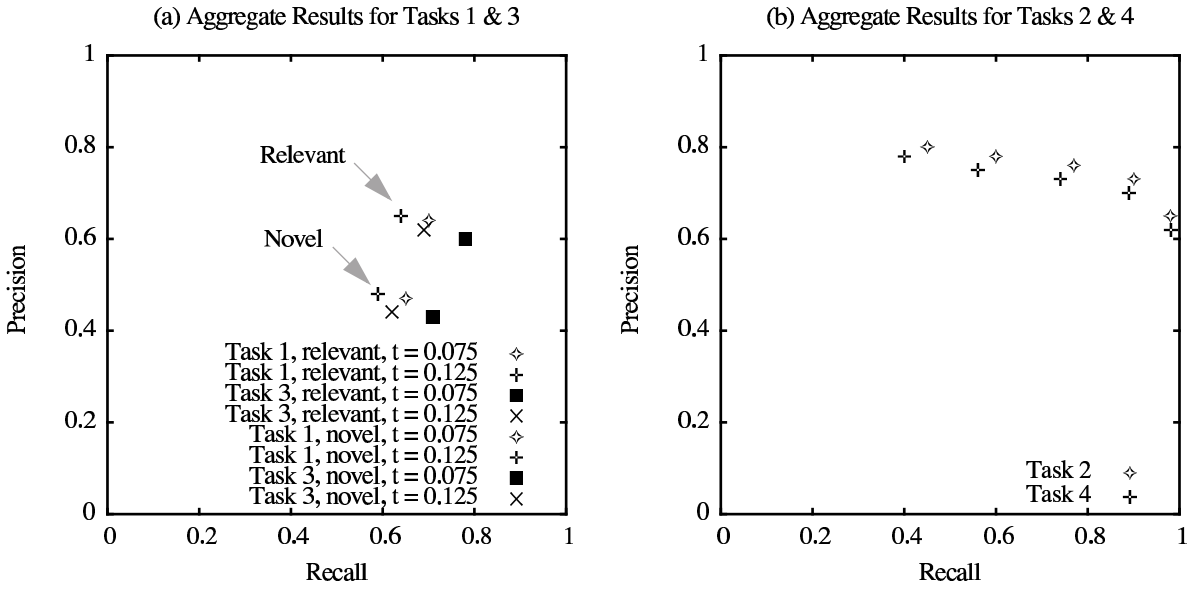

Figure 1: Novelty Task Results

Task 4 (given relevance for all and novelty for first 5, detect novelty for last 20). Load as per task 2 for relevance and task 3 for novelty, run as per task 2.

Aggregate Results for All Tasks

We submitted two sets of runs for tasks 1 and 3, with similarity thresholds of 0.075 and 0.125 and a new entity/noun phrase threshold of 1 and five runs each for tasks 2 nd 4, using new entity/noun phrase thresholds of 0-4. Table 1 shows the full results for all runs. As show in Figure 1a, there is a distinct performance differential between relevance and novelty detection, but relative performance among the four threshold/task conditions is comparably positioned for relevance and novelty. As might be expected, increasing the similarity threshold slightly improves precision at a slight cost to recall. More interestingly, precision of the task 1 configurations is similarly higher than their task 3 counterparts. Having the additional information regarding the first five documents for each topic slightly improves recall, but at the cost of precision. In other words, we can achieve better precision in both relevance and novelty by *not* looking at the initial pool of documents available in task 3.

As shown in Figure 1b, there is a very regular recall/precision trade-off achieved when varying the number of entities and/or noun phrases required to declare a sentence novel, given that it is relevant. It is also interesting to note that the oddity noted for tasks 1 and 3 is still present for tasks 2 and 4. Indeed, in this case, task 4 with relevance and novelty information available for the first five documents uniformly performs less well for both precision and recall for all thresholds. We find this intriguing and plan on further analyzing the cause of these results.

Conditioning by Topic Type

Figures 2 and 3 show the performance per topic for relevance and novelty, broken out by event and opinion topics. There appear to be no major trends to distinguish event topics from opinion

Table 1: Summary Results for Novelty Track

Run	Task	Sim. Thresh.	Entity/NP Thres.	Relevant			New		
				Prec.	Recall	F	Prec.	Recall	F
UIowa03Nov01	1	0.075	1	0.64	0.70	0.594	0.47	0.65	0.480
UIowa03Nov02	1	0.125	1	0.65	0.64	0.568	0.48	0.59	0.461
UIowa03Nov03	2	–	0	–	–	–	0.65	0.98	0.767
UIowa03Nov04	2	–	1	–	–	–	0.73	0.90	0.794
UIowa03Nov05	2	–	2	–	–	–	0.76	0.77	0.746
UIowa03Nov06	2	–	3	–	–	–	0.78	0.60	0.659
UIowa03Nov07	2	–	4	–	–	–	0.80	0.45	0.555
UIowa03Nov08	3	0.075	1	0.60	0.78	0.606	0.43	0.71	0.466
UIowa03Nov09	3	0.125	1	0.62	0.69	0.585	0.44	0.62	0.448
UIowa03Nov10	4	–	0	–	–	–	0.62	0.98	0.741
UIowa03Nov11	4	–	1	–	–	–	0.70	0.89	0.767
UIowa03Nov12	4	–	2	–	–	–	0.73	0.74	0.712
UIowa03Nov14	4	–	3	–	–	–	0.75	0.56	0.617
UIowa03Nov15	4	–	4	–	–	–	0.78	0.40	0.505

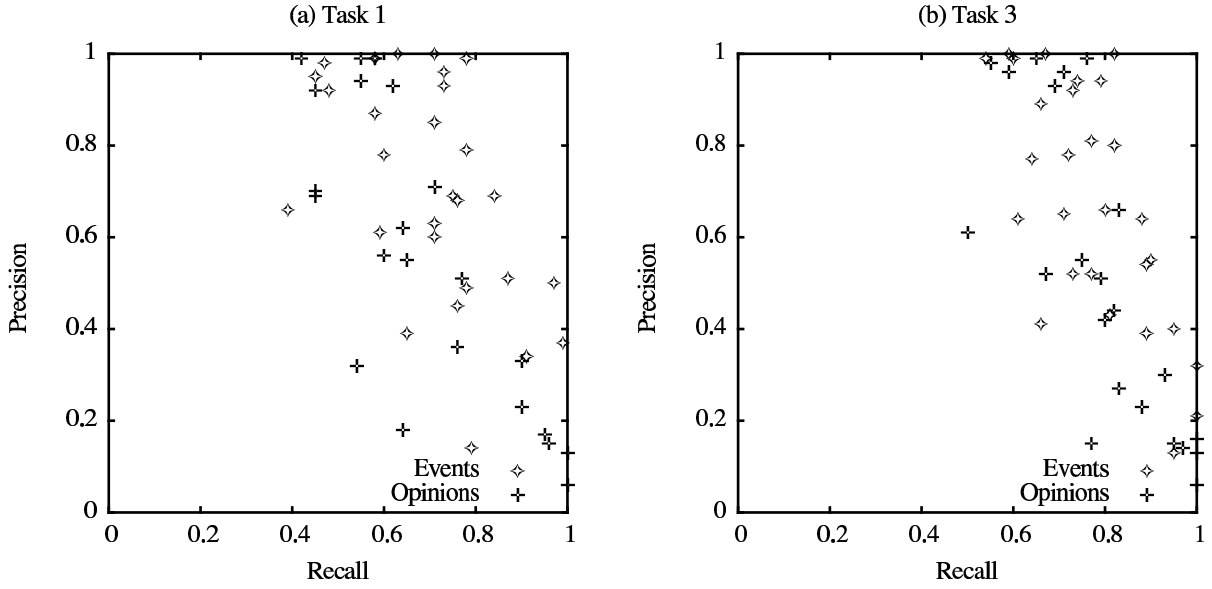

Figure 2: Novelty Task, Relevant by Topic Type

topics, although events do seem to edge opinions out in general. It does appear that the additional information available in task 3 results in a 'tightening' of the topic clouds for both relevant and novelty over the topic clouds for task 1.

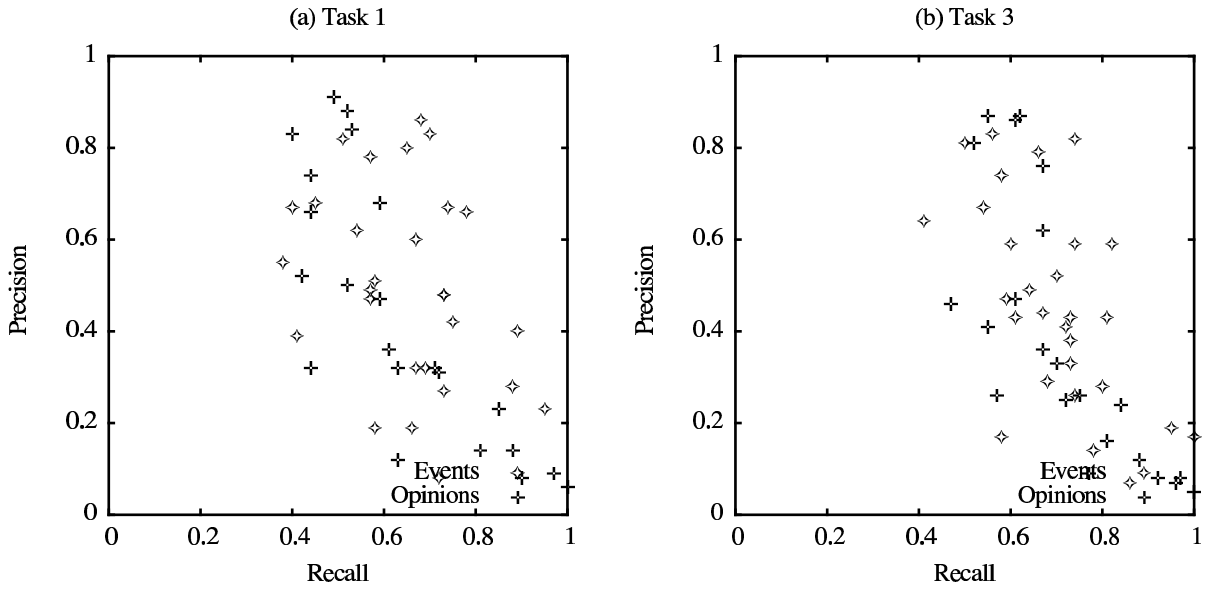

Figure 3: Novelty Task, New by Topic Type

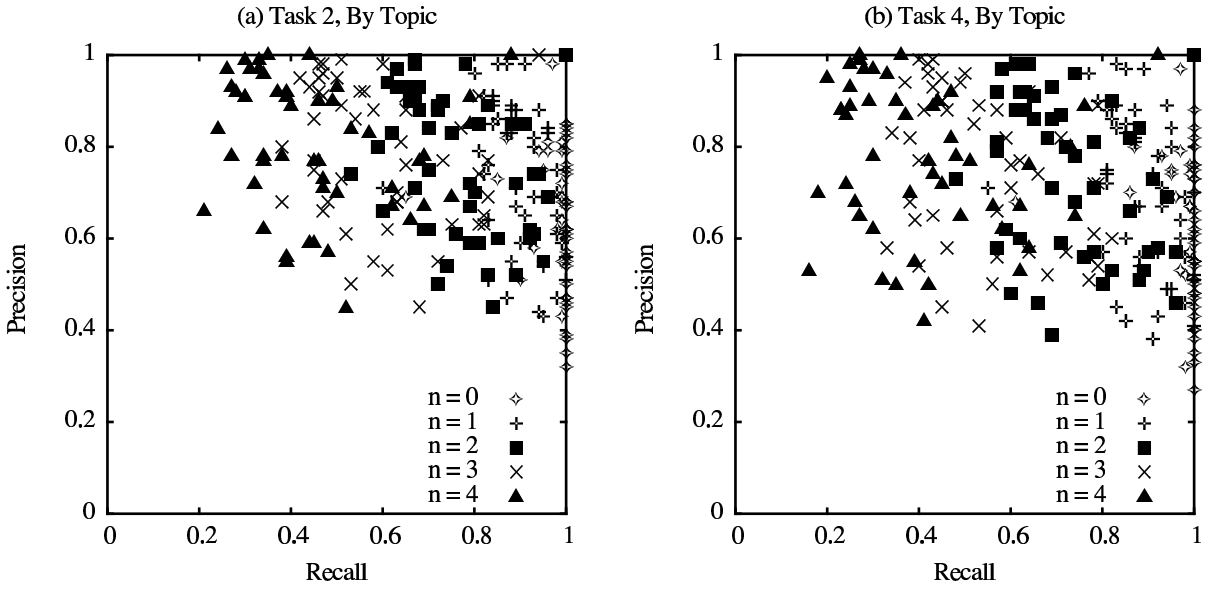

Figure 4: Novelty Task, New by Entity/NP Threshold

Effect of New Entities and Noun Phrases

Figure 4 shows a topic-level breakout of performance for each run for tasks 2 and 4. For the degenerate condition, n = 0, we see perfect recall generally, but our dual guard of both sentence and document level similarity does lower recall for some topics. Increasing the threshold to n = 1 improves precision overall, as seen in Figure 1b in aggregate. Further increases in the threshold generate little benefit with respect to precision and seriously erodes recall. Based upon this we have concluded that a single new entity or noun phrase can serve as an indicator of novelty. Our future work will focus on the analysis of poorly performing topics for n=1.

2 – Genomics

We participated in both the primary and secondary tasks in this track.

Primary task:

This is a baseline run where we used SMART as the retrieval system with atc weighting on queries and documents. Queries were generated from the different fields provided to us including gene name, symbols, product names. A few low level experiments were conducted with different weighting schemes and stemming options. We also tried using document classifiers (SVM) to limit the document set, but the results were not good.

Secondary task:

Each abstract sentence was classified to gauge its likelihood as a source of a GeneRIF. A sentence classifier was built using GeneRIF entries in LocusLink excluding those that were in the secondary.txt file and their abstracts. For feature selection an in house tokenizer was used and idf weights computed against a reference subset of 211,457 MEDLINE abstracts selected independent of this track.

Training Set:

GeneRIF entries (excluding the 'test' set as described above) were used to identify abstracts. 90% of the abstracts were used as training and 10% as testing for model/parameter selection.

Selection of positive and negative sentence samples.

Several methods were tried. But first, sentences in title, last sentence and first sentence are found to be most relevant, thus other sentences are discarded. Our methods involve a measure called pDice. This measures the percentage of words in a sentence that are in the GeneRIF entry corresponding to the abstract in which the sentence occurs.

Method 1: GeneRIF sentences are positive samples, low pDice sentences are negative samples.

Method 2: High pDice sentences are positive samples, low pDice ones are negative samples.

Method 3: GeneRIF and high pDice sentences are positive samples, low pDice ones are negative samples.

We used SVM classifier technology, specifically LIBSVM java classifier, with most parameters at default value. We also used EPSILON_SVR SVM, RBF kernel function. Positive/Negative class ratio is not used because it doesn't help. The best model found uses GeneRIF statements as positive samples and sentences with pDice<0.25 as negative samples. SVM gives each sentence score, the larger the score the more likely it is to be a GeneRIF. A weighting scheme was also used to emphasize titles, first sentences, and last sentences. The best weighting scheme on the test set was 5:0:1 respectively which is almost the same as saying select titles only. The best model and parameters was selected for use on secondary.txt and corresponding abstracts to generate result, for the official TREC run.

3 – Question Answering

This year marks the first evaluation of our complete implementation of an extraction-based QA system. By shifting natural language parsing forward in the process, we can amortize this very expensive step against a number of downstream extraction processes that mine the text for named entities, relationships, etc. Redefinition of extraction specifications hence does not require reparsing of the source text. We have implemented tgrep-like extraction grammar designed for predicate-based extensibility using it in mapping sentence parse trees to relational structure. This overall approach handles not only factoid answers, but definitional answers and those requiring inference across multiple extracted relationships.

Each document in the corpus is decomposed into doc-id / sentence pairs, with the sentence being the unit of analysis from that point. Each sentence is then POS-tagged and fed to the CMU link grammar parser. The parse tree for the sentence is then attributed with the POS tags for each word. Processing both queries and documents using this scheme allows us to establish both the nature of the query (using a fairly typical taxonomy) and the nature of the needed answer. This is particularly useful with respect to identification of candidate phrases in sentences and scoring of these phrases against the goal of the query. Sentences are then matched against the set of extraction patterns, populating a set of relations used to answer queries derived from the questions.

The availability of the parse tree for the phrase allows for elision of subordinate clauses that can cause answers to span too long a string and for extraction of likely answers through heuristic matching of, for example, a subordinate clause immediately trailing a mention of a candidate named entity.

We view our results for this year as very preliminary for two key reasons. The first is operational – a few days before the deadline a database failure cost us the full parse of the corpus and we were only able to reparse the top fifty documents for each question in the time remaining. The second is a developmental one – we have only begun the specification of our extraction pattern framework, and coverage is limited to

- persons' titles, ages and a minimal set of interpersonal relationships;
- location of organizations (e.g., "Seattle-based Microsoft"); and
- relative location of place names (e.g., "the resort, five miles east of Seattle").

Factoid Questions

The preliminary nature of our extraction patterns is probably most evident for factoid questions. Our pattern sets are insufficiently rich to provide sufficient coverage of potential questions, and hence the number of correct answers we generate is modest. As shown in Table 2, there is interesting potential in the low levels of unsupported and inexact answers relative to correct answers. We also have a comparatively high level of NIL answer recall, particularly given our level of correct answers. This is easily explained when the number of NIL answers returned is considered - ~20% of all questions. This is directly attributable to failure to extract sufficient information with the available patterns – we are returning so many NILs that we are catching those questions that actually have no answer in the corpus.

Table 2: QA Track, Factoids

Run	U	X	R	Accuracy	# NIL returned	NIL P	NIL R
UIowaQA0301	3	4	14	0.034	100	0.100	0.333
UIowaQA0302	2	2	17	0.041	173	0.087	0.500
UIowaQA0303	3	2	17	0.041	98	0.102	0.333

List Questions

Our implementation for this year had no support for list identification or extraction. Any coverage of answers in this category was purely accidental...

Table 3: QA Track, Lists

Run	Ave. F
UIowaQA0301	0.002
UIowaQA0302	0.002
UIowaQA0303	0.004

Definition Questions

We do believe that the approach that we are taking with extraction holds good promise for definition questions. As shown in Table 4, performance for this category of question is very different than that for factoids and lists.

Table 4: QA Track, Definitions

Run	Ave. F
UIowaQA0301	0.214
UIowaQA0302	0.231
UIowaQA0303	0.048

Breaking out performance of individual questions, as shown in Figure 5, we see that there is a broad spread of performance, but there are a large number of questions with no answers provided.

Figure 6 shows our performance in relation to the number of vital and total facts connected to a question. For questions where our system is performing well, there are a relatively small number (~2-5) of vital facts and a modest number (~10) of total facts.

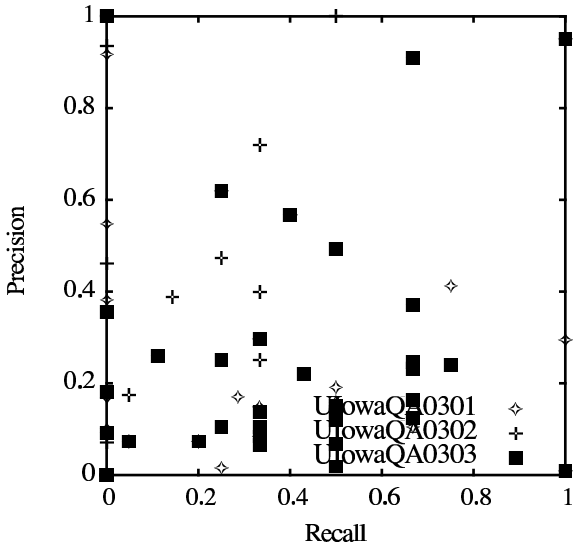

Figure 5: QA Task, Definition Questions

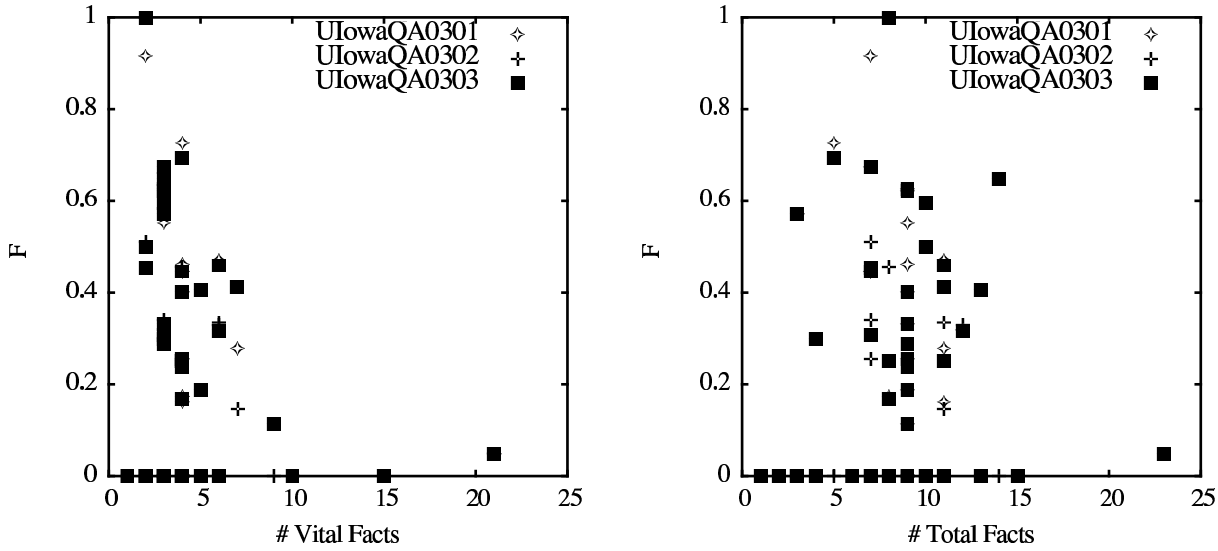

Figure 6: QA Task, Definition Richness