Novelty, Question Answering and Genomics: The University of Iowa Response

David Eichmann[ac], Yi Zhang[b], Shannon Bradshaw[bc], Xin Ying Qiu[b], Li Zhou[a], Padmini Srinivasan[abc], Aditya Kumar Sehgal[c], Hudon Wong[c]

[a]School of Library and Information Science
[b]Department of Management Sciences
[c]Department of Computer Science
The University of Iowa
Iowa City, IA, 52242

Novelty

Our system for novelty this year comprises three distinct variations. The first is a refinement of that used for last year involving named entity occurrences and functions as a comparative baseline. The second variation extends the baseline system in an exploration of the connection between word sense and novelty. The third variation involves more statistical similarity schemes in the positive sense for relevance and the negative sense for novelty.

Variations 1 and 2

Our general approach involves establishing a similarity threshold for sentence relevance and an new entity threshold for novelty. If this exceeded a 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.

Our first two variations for this year operate 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. For the available training topics, our relevance and novelty strategies 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 both noun phrases and named entities. The second variation has two alternatives. The first attempts to address the semantics of novelty by expanding all noun phrases (and contained nouns) to their corresponding synset IDs, and subsequently using synset IDs for novelty comparisons. Our conjecture here is that it is possible to conflate variations in wording and hence improve novelty precision without compromising recall. The risk of over-expansion of word-sense is likely to be minimal within the confines of a single topic and a limited number of sentences. The second alternative performs word sense disambiguation using an ensemble scheme to establish whether the additional computational overhead is warranted by an increase in performance over simple sense expansion. The runs submitted within these variations are UIowa04Nov11, UIowa04Nov12, UIowa04Nov13 for task 1, UIowa04Nov21, UIowa04Nov22 for task 2, UIowa04Nov31, UIowa04Nov32 for task 3 and UIowa04Nov41, UIowa04Nov42 for task 4.

Variation 3

The third variation employs SMART for vector similarity based decisions. First, the SMART system then generated one term vector for each sentence. The topic documents were also fed into the SMART system. For each topic, the title, description and narrative fields were included to generate the topic vector. Specifically for the narrative field, the paragraph was firstly split into sentences. The sentences containing terms like irrelevant or not relevant were discarded. The weighting scheme for both sentence and topic vectors is nnn.

Task 1

If the total number of documents of a topic exceeds 25, the top 25 documents with highest similarity to the topic were retrieved. Only sentences within these 25 documents were considered as relevant candidates. We set up two relevant sentence retrieval thresholds for two runs: $mean(STSIM - 2 * std(STSIM)$ and $mean(STSIM - std(STSIM)$, here $STSIM$ is the vector of cosine similarity between the candidate sentences and the topic. After retrieving all the relevant sentences, we extracted the novel sentences by the following procedure:
1. Mark the first relevant sentence as novel. Set the Current Knowledge vector as the term vector of the first sentence.
2. Get the next relevant sentence. Compute the cosine similarity between

the sentence and the Current Knowledge $SKSIM(i)$.
3. Expand the Current Knowledge vector by adding in the sentence vector.
4. Go to step 2 if there are more relevant sentences left unprocessed. Stop otherwise.
5. Set novelty threshold as $mean(STSIM - 2 * std(STSIM)$ for run UIowa04Nov14 $mean(STSIM - std(STSIM)$ for UIowa04Nov15 and sentences with $SKSIM(i)$ below the thresholds are retrieved as novel sentences.

Task 2

The same novel retrieval strategy as task 1 was applied on the provided relevant sentence set to get novel sentences. Three runs were preformed on this task (UIowa04Nov23, UIowa04Nov24, UIowa04Nov25). The UIowa04Nov25 run used threshold $mean(STSIM)$.

Task 3

In task 3, the relevant and novel sentences in the first 5 documents are given. Firstly, the topic vectors were expanded by including the provided relevant sentences. Secondly, the number of relevant documents among the first 5 documents DREL5 can be figured out and hence there will be exactly 25-DREL5 relevant documents in the remaining ones. The top 25-DREL5 documents with highest similarity to the expanded topic were retrieved. Thirdly, we tried to get the optimal relevant sentence threshold for each topic by enumerating different thresholds ranging from 0.05 to 0.25 on the relevant documents in the first 5 documents. The threshold producing highest F score in the first 5 documents was applied to the remaining documents. Note that here the original topic vector was used to compute sentence topic similarity. Finally the sentences in the remaining relevant documents with higher-than-threshold similarity score were retrieved as relevant sentences. Similar threshold optimization was performed on novel retrieval. For each topic, the threshold with highest novel F score for the first 5 documents was picked. The search range for the best novel threshold is from 0.2 to 0.4 with step 0.05. However, there are cases that no relevant sentences are found in the first 5 documents. If that happens, the same strategies as for task 1 were applied on the remaining documents. Runs UIowa04Nov33, UIowa04Nov34 and UIowa04Nov35 represent these strategies.

Task 4

Use the same novel threshold search technique as task 3 and novel sentences were retrieved by task 1 procedure using the obtained best threshold. Runs

UIowa04Nov43, UIowa04Nov44 and UIowa04Nov45 represent these strategies. Our results are presented in Table 1. It may be observed that our first two variations perform the best. However, it is interesting to observe that a simple, word based similarity approach is not too far behind our more sophisticated approaches.

Table 1: Novelty Track Results

	Relevant			Relevant		
	Precision	Recall	F	Precision	Recall	F
Task 1						
UIowa04Nov11	0.31	0.82	0.42	0.15	0.71	0.229
UIowa04Nov12	0.31	0.82	0.42	0.15	0.67	0.23
UIowa04Nov13	0.31	0.82	0.42	0.15	0.71	0.227
UIowa04Nov14	0.29	0.74	0.392	0.12	0.68	0.188
UIowa04Nov15	0.29	0.74	0.392	0.11	0.58	0.175
Task 2						
UIowa04Nov21				0.48	0.9	0.609
UIowa04Nov22				0.46	0.92	0.604
UIowa04Nov23				0.42	0.96	0.569
UIowa04Nov24				0.42	0.87	0.553
UIowa04Nov25				0.44	0.57	0.482
Task 3						
UIowa04Nov31	0.29	0.91	0.407	0.13	0.74	0.208
UIowa04Nov32	0.29	0.91	0.407	0.13	0.79	0.207
UIowa04Nov33	0.32	0.64	0.398	0.12	0.62	0.188
UIowa04Nov34	0.32	0.64	0.398	0.12	0.61	0.187
UIowa04Nov35	0.33	0.62	0.396	0.12	0.55	0.178
Task 4						
UIowa04Nov31				0.44	0.9	0.57
UIowa04Nov32				0.43	0.92	0.567
UIowa04Nov33				0.39	0.97	0.538
UIowa04Nov34				0.39	0.96	0.536
UIowa04Nov35				0.39	0.96	0.535

Table 2: Genomics Track - Task 1 Results (UIowaGN1)

	Number of Topics		
Performance	Precision at 10	Precision at 100	Average Precision
Best score	10	3	0
Above median score	15	18	18
At median score	9	10	3
Below median score	7	17	28
Worst score	9	2	1
Overall	Above median	At median	Below median

Genomics Task 1

We used the Lucy/Zettair search engine as the retrieval system for the Ad Hoc task because of its ability to handle the large size of the collection. Lucy/Zettair supports Boolean, ranked and phrase querying. To construct an inverted index using Lucy/Zettair, the dataset was first converted to TREC format. In addition to AB, TI, MH Medline fields, RN and GS fields were included in indexing. All the fields in the query topics were included. Gene names were automatically expanded using synonyms from the LocusLink database. Then the queries were processed to produce segments of phrases and words. All gene names were regarded as phrases. We also tried Boolean search using only the title but the results were not good on the five samples. We were interested in the combination of the Boolean search and ranked search but found it difficult to automatically construct a satisfying Boolean query using all the fields in the topics. Our final strategy consisted of ranked searches using the extracted phrases and words. Table 2 below gives the results for our one submission which was called UIowaGN1. The table essentially compares our performance with the summary of the performance for all submissions for this task. Our results indicate that we did fairly well for precision at top 10. However, since we essentially did not consider thresholding strategies, and instead submitted upto 1000 documents for each topic, our performance in the other measures has suffered.

Genomics Task 2

The categorization task requires us to first decide if a test document is about mouse genomics biology. If it is, we need to categorize the document into one or more of the three Gene Ontology categories: biological processes, celluar components, and molecular funtions.

Our basic idea to tackle the categorization task is to utilize the GO terms that have been used to annotate mouse genomics biology documents, and the MeSH terms from all the training documents, to perform retrieval from the test documents.

GO Terms Retrieval

We collected a complete set of GO terms from each of the three GO vocabularies. These GO terms are the ones that have been used to annotate mouse genomics biology documents in MGI. We indexed all test documents with all individual words using our own information indexing/retrieval system created by Shannon Bradshaw at University of Iowa. We calculated the proportion of each categories' document in the training data set and estimated the number of testing documents that could belong to each of the three categories. Then we used each of the three categories' GO terms to perform retrieval on all testing documents, and used the estimated proportion as a threshold to select the top set of documents for each category.

Mesh Terms Retrieval

From the training documents, we picked the documents that are annotated to belong to only one of the three GO categores. We then collected the mesh terms from these three sets of documents' MEDLINE records. We used these three sets of mesh terms as the three GO categories' mesh term vocabulary, and perform retrieval on the testing documents using our own indexing/retrieval system. We used the same estimated proportion as in the GO terms retrieval as thresholds to select the top set of documents for each category.

Combining the GO terms retrieval results and the Mesh terms retrieval results

For both retrieval experiments, we used the given assignement of the gene names to the testing documents as the gene name assignment to our retrieved documents. So, if a testing document is assigned with two genes, and our retrieval puts this document into two categories, then we assigned both genes to this document as triaged to two categoreis. We realized this strategy is problematic. We had planned to investigate a better way to associate genes with document categories but we ran out of time in the end.

In our experiments using the training data, we observed that a certain way of combining the results from the GO-terms based retrieval and the MeSH-terms based retrieval could out-perform the individual retrieval using

GO terms or MeSH terms alone. After a series of experiments, we decided on the following combination strategies:

- **iowarun1:** In our experiments using the training data, using the GO terms retrieval result for cellular component category was better than when using the combination of GO terms and Mesh terms retrieval for the cellular component category. So in our submitted run one on the test data, we use GO terms retrieval result for cellular component category. For the biological processes and the molecular functions categories, we get the retrieval results from both the GO-terms based retrieval and the MeSH-terms based retrieval given the same threshold, and then we take the union of these two sets of results as our final result for these two categories.

- **iowarun2:** Using the same GO terms and MeSH terms based retrieval strategies, here we only vary our thresholds for selecting the top sets of documents for each category. We set these to be 75 percent of our estimated proportion. Then we combine these two retrieval results the same way as designed in iowarun1. So the difference between iowarun1 and iowarun2 is only in the thresholds set for the GO-terms retrieval and the MeSH-terms retrieval.

- **iowarun3:** Using the same retrieval thresholds as in iowarun1, we use the union of the GO term retrieval and the MeSH term retrieval results for each category as our final sumbitted results for these categories. So the difference between iowarun1 and iowarun3 is only in the cellular component category. In iowarun3, the results for cellular component categories are also a union of results from the two retrieval strategies, instead of from GO-terms retrieval only.

- **iowarun4** Here we decided to explore a completely different approach using a simple strategy based on citation sentences. We first used PubMed and the publisher sites linked from PubMed to gather articles citing the documents in both the training and test collections. We then extracted the paragraph within each citing article containing a citation to an article in the training collection. The paragraphs we extracted contained approximately three sentences on average. We indexed each training document in a retrieval system using individual words found in paragraphs citing that document. For each of the three categories BP, MF, and CC we identified the most strongly associated index terms. To identify these "category terms" we chose terms that were used at least twice as frequently to cite articles in one category as they were to cite articles in either of the other two categories.

  We then indexed all test documents in our retrieval system using the same technique as for training documents using a standard vector space

approach. Using the category terms identified from the training collection, we queried the test collection. For each category we choose the top k documents so that the distribution of documents falling into each category matched the distribution found in the training collection. Unfortunately, this technique did not perform well. However, the citation sentences we selected did identify some useful discriminating terms for each category. We believe that further efforts employing more robust text classification techniques based on citation sentences would yield better results.

Table 3 provides a summary of our results for our Task 2 runs.

Table 3: Genomics Track - Task 2 Results

Run	iowarun1	iowarun2	iowarun3	iowarun4
True positive	269	223	297	66
False positive	529	362	629	324
False negative	226	272	198	426
Precision	0.3371	0.3812	0.3207	0.1692
Recall	0.5434	0.4505	0.6000	0.1333
F-score	0.4161	0.4130	0.4180	0.1492
Utility Factor	20	20	20	20
Raw Utility	4851	4098	5311	996
Max Utility	9900	9900	9900	9900
Normalized Utility	0.4900	0.4139	0.5365	0.1006

Before we decided on the strategies for the above runs (especially the first 3 task 2 runs), we had conducted numerous experiments exploring the best way to tackle the triage subtask. Unfortunately, due to a mistake made in evaluating our performance which we only found out the day before submission, we wrongfully underestimated all our previous experiments and therefore possibly failed to identify the best strategy from our studies. Specifically, when we used 2/3 of the given training data as our 'training data' and 1/3 for testing, instead of limiting the gold standard data to these 1/3 data points we used the complete training data as our answer file. Therefore, we found for example that our very first strategy achieved an F-score of only 0.1386, while the true performance if using the correct answer file is 0.3674. Although we did not have enough time to reevaluate all our previous tests we provide a summary of our experiments which may merit further investigation if we tackle a similar problem in the future:

- Test 1: We used the abstracts of the documents to implement a decision-tree kind of 4 steps of classification: Step 1, classify if a document is

positive or negative document. (By positive, we mean that the document is about some GO category of a certain gene. By negative, we mean that the document is not about any GO category). Step 2, if a document is positive, we classify if this document is about "cellular component" (CC) or not. Step 3, For the same document, we classify if this document is about "molecular funtion" (MF) or not. Step 4, For the same document, we classify if this document is about "biological processes" (BP) or not.

This 4-stage decision tree will provide eight leaves which are the eight possible categorizations for a given document: all 3 categories, CC and MF, CC and BP, CC only, MF and BP, MF only, BP only, and negative.

For this test, we used all words from the document set to perform indexing and retrieval. The retrieval step gives us a ranked list of the test documents. We used the proportion of the positive training data as a threshold to classify the test documents. The step 1 classification is trained on positive and negative documents in the training set. The step 2 to 4 classification are trained on the training documents that belong to only one category: CC or MF or BP. This strategy in our testing has a performance of 0.3674 F-score.

- Test 2: The same as test 1 but we omitted step 1, which can still produce eight possible categoriztions for a given document. We estimated the performance would not surpass that of test 1.

- Test 3: The same decision-tree classification strategy as in test 1, but in the indexing and retrieval step, we used GO terms to index and all words from documents to retrieve. The performance seemed to be worse than test 1.

- Test 4: The same decision-tree classification strategy and the same indexing and retrieval method as in test 1, but we worked on the methodology sections extracted from the full text of all the training documents. The performance seemed to be comparable to test 1.

- Test 5: The same implementation as in test 2, but we worked on the methodology sections instead. The performance seemed to be comparable to test 2.

- Test 6: Similar to test 5, the only difference is that we used GO terms for indexing but all words from documents for retrieval. The performance is estimated to be worse that test 5.

- Test 7: The above tests are basically document-level retrieval. We also performed sentence-level retrieval as a strategy to classify documents.

The rational of this strategy is similar to test 2, but we worked on the sentence level indexing/retrieval instead. We first index each sentence in our testing set of documents with all its words. Then for each of the three GO cateogory, we collect the training documents that belong to only one category, and used all the words in the training documents sentences to perform retrieval on the sentences on the testing set of documents. Since each sentence is associated with a document, the ranked list of sentence can also provide us with a ranked list of documents. Then we used a threshold (estimated proportion of documents belonging to a category) to classify a top set of testing documents as belonging to a category.

We also tried some variations on this strategy by filtering the terms used for retrieval. Instead of using all terms from the training documents, we tried using the top 30 terms of a certain frequecy, and the top 100 terms of a certain frequcy, for retrieval. The top 30 terms of frequency 10 seems to perform better but not significantly better among all our previous tests.

Again, due to the evaluation mistake, we were not able to investigate the true performance of these tests before we submitted our official runs. Given these problems we are pleased with the final results that we have obtained.