# UB at TREC-12: HARD and Genomics Tracks

Munirathnam Srikanth
Computer Science and Engineering
SUNY at Buffalo
Buffalo, NY, 14228
srikanth@cedar.buffalo.edu

Miguel E. Ruiz
Library and Information Studies
SUNY at Buffalo
Buffalo, NY, 14228
meruiz@buffalo.edu

Rohini Srihari
Computer Science and Engineering
SUNY at Buffalo
Buffalo, NY, 14228
rohini@cedar.buffalo.edu

## 1. INTRODUCTION

University at Buffalo (UB) participated in TREC-12 in Genomics and High Accuracy Retrieval from Documents (HARD) tracks. We explored some techniques that combine Information Retrieval and Information Extraction to perform the TREC tasks. We used an Information Extraction engine - InfoXtract [3] from Cymfony Inc.[1] - to enhance retrieval results.

For the Genomics primary task, documents retrieved using a vector space model with relevance feedback are reweighted based on the biomedical named entities discovered by InfoXtract. For the secondary task, extracted information along with cue words for text snippets that describe functionality is used for generating GeneRIFs for given Gene name and PubMed abstract. A language modeling approach that incorporates keyword and non-keyword features are used for the HARD task. Features extracted by InfoXtract from the HARD corpus are used to rank documents and/or passages as answers to the HARD queries.

Cymfony's InfoXtract [3] is a customizable Information Extraction engine that performs syntactic and semantic parsing of a document to identify features like named entities, relationships and events in them. The baseline InfoXtract engine has been trained for the general English and news domain, It can be customized to recognize new named entities like Gene Names and Gene Function. Biomedical Customization of InfoXtract is briefly presented in Section 2.2.

## 2. GENOMICS TRACK

UB participated in both primary and secondary task of TREC 12 Genomics track. Our efforts concentrated on trying to apply Information extraction to improve retrieval performance in task 1 and to combine support vector machines and information extraction for selecting sentences as GeneRIFs annotations.

### 2.1 Relevance Feedback Model for Genomics Retrieval

We used the SMART system as a baseline as the search engine for the Genomics track. Documents are represented using title (TI), abstract (AB), MeSH terms (MH) and EC/RN Numbers (RN). We tried several models using separate *ctypes* for each of the previously mentioned fields and measure the contribution of each part to retrieval performance. We also experimented with a version that used all the information in a single *ctype*. Table 1 shows the contribution of each part to the retrieval performance on the training topics. It is interesting to note that these runs show that the MeSH terms have a very small contribution in the retrieval results of the training topics (0.0326 Avg-Prec). We believe that this is due to two factors: a) low coverage of domain specific Genomics concepts in MeSH, and b) we did not attempt to do a mapping of topics against the MeSH vocabulary (only single word matching is used for this particular run). We were also surprised to see that the contribution of EC/RN numbers does contribute significantly to retrieval performance. The combination of all fields into a single *ctype* outperforms all runs that use a single field. We also tried two different stemming algorithms since we were not sure whether a simple stemming algorithm that takes care of plurals only would work better for our experiments (as reported by Jacques Savoy in his preliminary experiments on this collection). Additionally we used a heuristic method to try to capture phrases and proper nouns. For this purpose we preprocessed the documents to identify fragments delimited by punctuation symbols and extract bigrams (groups of two consecutive words) that don't include stopwords. The heuristic process takes into account exceptions that allow a limited number of stop words to be part of the bigram term ("of", and "for"), i.e. "alignment of proteins". The best results in the baseline runs were obtained using this heuristic method (0.3200 Avg-Prec).

Topics were processed by extracting the information available in each field and then representing it in the corresponding *ctype*. For runs that use bigrams we added the corresponding bigrams and phrases to each training topic using the previously described heuristic. We also tried several weighting schemes (*atn, atc, ann,* and *Lnu* for documents and *ntc, ltc, lnc, atc, atn, ann, anc,* and *ltu* for queries).

We also performed pseudo-relevance feedback using the top $n$ documents as relevant and selecting the top $m$ terms to expand each query. The results of our experiments on the training set are presented in Table 2. Pseudo Relevance feedback combined with bigrams is the one that yields the best performance in the training topics (0.3702 Avg-Prec with a 16% improvement over the corresponding baseline).

### 2.2 Information Extraction in BioMedical Domain

The InfoXtract engine is customized for the BioMedical domain before using it to process the Genomics document collection. Domain knowledge is essential for an Information Extraction engine to tag documents with named entities in the domain of interest. We used part of UMLS hierarchy

[1] www.cymfony.com

| | weights | P@10 | Avg. Prec. | R-Prec. |
|---|---|---|---|---|
| TI | *atn.ntc* | 0.1040 | 0.1961 | 0.1757 |
| AB | *atn.atc* | 0.1380 | 0.2604 | 0.2115 |
| MH | *atn.lnc* | 0.0180 | 0.0326 | 0.0183 |
| RN | *atn.ntc* | 0.0620 | 0.1777 | 0.1615 |
| TI+AB+ MH+RN (one ctype) Lovins' Stemmer | *atn.ltu* | 0.1320 | 0.3028 | 0.2683 |
| TI+AB+ MH+RN (one ctype) Rem-s Stemmer | *atn.ltu* | 0.1340 | 0.3044 | 0.2706 |
| TI+AB+ MH+RN+ Bigrams (one ctype) Rem-s Stemmer | *atn.ltc* | 0.1320 | 0.3200 | 0.2743 |

**Table 1: Baseline runs on Training Topics**

as domain knowledge for InfoXtract. We restricted our customization effort to identifying domain-specific terminology through lexicons.

We selected several subtrees from the UMLS that are most related to the Genomics sub domain. For this purpose we selected the following concepts:

- C1136351: Genetic Phenomena.
- C1136308: Genetic Processes
- C1136352: Genetic Structures
- C0017398: Science of Genetics
- C0002526: Amino acids, Peptides and Proteins
- C0019934: Hormones, Hormone Substitutes and Hormone Antagonists
- C0018285: Growth Substances
- C0014443: Enzymes, Co-enzymes and Enzymes Inhibitors

We use the parent–child relationship in the UMLS metathesaurus to select all terms related (as a narrower concept) for each of these general concepts. This produced a set of 21,070 concepts and 51,571 unique terms (after normalization). We also add the related terms to the species of interest in this task ( Homo sapiens, Mus musculus, Rattus norvegicus, and Drosophila melanogaster) for 31 extra terms.

In addition, InfoXtract is customized to identify and tag named entities of type *Gene Name* and *Gene Functionality*. Gene names and their synonyms are collected from LocusLink and assigned a unique name (typically the preferred product name). Automatic candidate selection followed by manual truthing is adopted for generating the Gene Functionality lexicon. The InfoXtract engine is customized to detect Gene Names. Some training documents are processed by InfoXtract and term statistics is used to generate candidate functionality terms. Some heuristics like ignoring terms that appear in a named entity like Gene Name are used to filter out terms and construct a candidate set for Gene Functionality. Researchers in Biology were asked to manually go through the functionality term list and the context of their usage in the training documents subset to identify Gene functionality terms.

| | weights | P@10 | Avg. Prec. | R-Prec. |
|---|---|---|---|---|
| TI+AB+ MH+RN ($\alpha = 8; \beta = 64;$ $\gamma = 16$) ($n = 3; m = 10$) Lovins' Stemmer | *atn.ltc* | 0.1420 | 0.3316 | 0.3044 |
| TI+AB+ MH+RN ($\alpha = 32; \beta = 16;$ $\gamma = 8$) ($n = 3; m = 5$) Rem-s Stemmer | *atn.ltc* | 0.1380 | 0.3300 | 0.2907 |
| TI+AB+ MH+RN+ Bigrams ($\alpha = 32; \beta = 16;$ $\gamma = 8$) ($n = 3; m = 5$) Rem-s Stemmer | *atn.ltc* | 0.1380 | 0.3702 | 0.3291 |

**Table 2: Pseudo-Relevance Feedback runs on Training Collection**

Genomics document collection is processed by the customized InfoXtract engine. It extracts the 12 different named entities mentioned above.

## 2.3 Using IE in Genomics IR

Natural Language Processing (NLP) techniques have been used in document retrieval to select index terms. Prior use of Information Extraction output has been restricted to narrow search problems like question answering. We used Information Extraction as a filter to improve or re-rank the retrieval results. Documents are processed by InfoXtract customized for the biomedical domain. The output of InfoXtract for a document, in addition tagged biomedical named-entities, includes part-of-speech, shallow parsing results and relations between named entities. A subset of the extracted information along with the terms in a document are indexed using the TAPIR toolkit. TAPIR toolkit is a library of software tools that facilitate a number of IR tasks and supports different IR models including language models. The position of the index term in a document is also used as the position of tags representing the extracted information. This index is used to re-rank the document retrieval results corresponding to the run *UBgenomRFB1*.

Given a query, each document deemed relevant in the run *UBgenomRFB1* to the query is weighted for the co-occurrence of query terms with named entities in the biomedical domain. The co-occurrence frequency of tags with query terms is used to reweight documents. Ad-hoc weights are assigned to different named entity tags with the highest weight given to co-occurrence with gene function words, terms related to genetic process and other gene name. Reweighted documents set is normalized and re-ranked to generate the result named *UBgenomBGNE*.

## 2.4 GeneRIF Extraction

As noted in the guidelines for this task Mork and Aronson have found that 95% of the GeneRIF snippets contain text from the title and abstract of the articles. For this reason we decided to concentrate our approach on selecting sentences from the title and abstract as a first approach. For GeneRIF extraction we propose a solution that uses text categorization to select the sentence from the MEDLINE document (Title and Abstract) that is the best candidate to be a GeneRIF descriptor.

Documents are processed using InfoXtract to detect sentences, as well as important information such as the name of the gene of interest, or description of the functionality related to the gene of interest.

Our baseline system for this task is a trivial procedure that assigns the title as the candidate for GeneRIF description. We also tried to find what would be the upper bound of performance for a method based on sentence selection. For this purpose we found the sentence that would give the highest Dice coefficient value for each GeneRIF.

The approach that we use for this task uses a support vector machine (SVM-light) to learn the sentences that should be selected as GeneRIF using as input features the vector representation of the document using the smoothed unigram language model. We also included other features such as the position of then sentence in the document, whether the gene of interest is mentioned in the sentence, and whether biomedical terms (that were extracted by InfoXtract) appear in the sentence. For training the SVM we collected 5496 GeneRIFs annotations from LocusLink and gather the respective MEDLINE documents (making sure that these GeneRIFs where not in the test set for the secondary task). This set was randomly divided into $3,676$ documents for training and $1,820$ documents for validation.We also determined the sentence that had the best Dice score in the document to be the "correct GeneRIFs" and mark it as a positive example while the rest of the sentences in the document were marked as negative examples. This process gave us a total of $19,658$ sentences in the training set and $9,947$ sentences in the validation set. The sentence with the highest SVM classification score was selected as the GeneRIF for each document.

| | Baseline | Upper bound | Features only | Feature+ cues+ position |
|---|---|---|---|---|
| Classic Dice | 57.02 | 76.18 | 45.54 | 56.94 |
| Mod unigram Dice | 57.70 | 76.51 | 45.77 | 57.68 |
| Mod bigram Dice | 42.79 | 67.59 | 30.87 | 43.23 |
| Mod bigram Dice phrases | 46.00 | 69.93 | 34.01 | 46.52 |

**Table 3: Results in the Validation Set.**

Table 3 shows the results of our experiments on the validation set of $1,820$ GeneRIFs. The upper-bound indicates that the best we can performance of a method that selects full sentences from the MEDLINE article would be at 76.18. This corroborates that most of the GeneRIFs come from "cut and paste" text from the Title and abstract. Our baseline system that selects the title as the GeneRIF annotation performs at 57.02% which indicates that the simplest algorithm for selecting the GeneRIFs annotations is obtaining about 73% of the Upper-bound performance. This is not surprising since a significant number of GeneRIFs are just the title of the article. The first attempt to use SVM only used the features extracted from the unigram language model and performs significantly below our baseline (45.54% Classical Dice and about 20% below the baseline). when we added the cues (gene name, and functionality, words, etc) and the relative position of the sentence in the document the SVM was able to achieve a performance that is about the same as the baseline (56.94% Classical Dice). We were disappointed to realize that after all the training process and information extraction our system wasn't better than our simple baseline.

## 2.5 Results and Analysis

Our official results in task 1 are presented in Table 4. *UBgenomRFB1* uses pseudo-relevance feedback, *atn.ltc* weighting scheme, and the top 3 retrieved documents from which we get the top 5 terms for query expansion, and $\alpha = 32.\beta = 16.\gamma = 8$ as the parameter in Rocchio's formula. *UBgenomRFB2* uses pseudo-relevance feedback, *atn.ltu* weighting scheme, and the top 3 retrieved documents from which we get the top 5 terms for query expansion, and $\alpha = 8, \beta = 64, \gamma = 16$ as the parameter in Rocchio's formula. *UBgenomeBGNE* is the re-ranked output of *UBgenomRFB1* using the filtering process explained previously. In general our results are slightly below the average performance. We suspect that this could be a consequence of the way topics were converted into queries in the system but we still need to do a more detailed analysis.

| | P@10 | Avg. Prec. | R-Prec. |
|---|---|---|---|
| UBgenomRFB1 | 0.1160 | 0.1511 | 0.1232 |
| UBgenomRFB2 | 0.1120 | 0.1493 | 0.1141 |
| UBgenomeBGNE | 0.1440 | 0.1867 | 0.1603 |

**Table 4: Pseudo-Relevance Feedback runs on Training Collection**

| | Best | ≥Median | <Median | Worst |
|---|---|---|---|---|
| UBgenomRFB1 | 1 | 15 | 34 | 6 |
| UBgenomRFB2 | 1 | 14 | 35 | 7 |
| UBgenomeBGNE | 1 | 18 | 31 | 0 |

**Table 5: Relative performance of Official Task 1 Genomics runs**

Results for the secondary task in Genomics are presented in Table 6. As suspected from our results in the training and validation set the SVM based approach did not perform significantly better than our base line system. All our runs are fairly close to the median system performance (49.31% Classic dice coefficient) in this task.

| | Baseline | SVM 1 | SVM 2 |
|---|---|---|---|
| Classic Dice | 49.28 | 49.03 | 49.40 |
| Mod unigram Dice | 51.25 | 51.16 | 51.30 |
| Mod bigram Dice | 33.59 | 33.94 | 33.59 |
| Mod bigram Dice phrases | 36.99 | 37.35 | 36.99 |

**Table 6: Officail results for Secondary Task.**

# 3. HARD TRACK

In the HARD track user queries are qualified by metadata that provide additional information on the user's information need such as purpose, genre and granularity of query. Of the three steps in the track - baseline retrieval, clarification forms and final run to use all the information available about the user query. The second step is optional and we did not generate any clarification forms for the queries. We used the metadata provided along with the query in the final run. Language Modeling approach [1] is adopted for both the baseline and final HARD tasks.

HARD document corpus is processed by Cymfony's InfoXtract engine which tags part-of-speech, named entities and associated profiles to the entities and events discovered in the document. A subset of information extracted by InfoXtract is used in our HARD solution. However, the text and all extracted information from a document are indexed by TAPIR - an IR toolkit that supports different IR models including Language Modeling approach to IR. A tag of a term in a document is indexed as though it is embedded in the document at the position(offset) corresponding to the term. This method of indexing has been explored for question answering in earlier TREC.

## 3.1 Baseline System

We submitted three different runs for baseline. Our baseline solution is based on the textual part of the user query - title and description - and perform document retrieval. No effort is made to predict any metadata for the user queries as well as no query expansion or relevance feedback is experimented with. A brief description of the three runs are given below:

- *ub03sugT* Run based on smoothed unigram language model that uses Dirichlet smoothing. The Dirichlet parameter is set to 1000. This run used only the title of the query.

- *ub03cugTD* This run is based on the Concept Unigram Language Models (CULM) [4] that have been shown to perform better than smoothed unigram and bigram language models. In Concept Language Model a query is viewed as a sequence of concepts and concepts, in turn, are viewed as a sequence of terms. Consecutive terms typically constitute concepts that can be single terms, bigrams or n-grams. In CULM, concept independence is assumed and query probability is computed as a unigram model on concepts. Concept probability is approximated to smoothed bigram probabilities. The InfoXtract question parser [2] and its shallow parsing results are used to identify concepts of interest in the query.

- *ub03ugTcugTD* This run corresponds to a linear combination of the above two methods. It provided slight improvement in mean average precision on the training set.

The results of *ub03ugTcugTD* are used for the final run (*ub03smfugTD*) as the document retrieval system. Its results are further refined to satisfy the metadata values of the user query.

## 3.2 Passage Retrieval

Except for the queries with granularity value of document, passage retrieval is performed for other queries. Relevant passages are selected based on the query keywords. The granularity of the query determines the length of the passage. Relevant sentences (i.e. passages of length 1) are identified for queries with granularity of sentence or phrase. The granularity of passage resulted in the system selecting text snippets with 3 to 6 sentences. The passages are not overlapped. The coverage of query terms is used as a criteria to determine the passage length. All candidate passages thus selected are considered for final ranking.

## 3.3 MetaData Modeling and HARD

We modeled Purpose and Genre metadata for the HARD task. For a given query, text snippets are short-listed based on their satisfying the user's query and granularity requirements. A number of keyword- and non-keyword based features are used to weight the snippets and rank them. A text snippet is viewed as a sequence of keyword and non-keyword features and a model is associated with it in the language modeling sense. Text snippets are weighted based on the probability of generating a given feature. Smoothed probabilities are estimated for keyword features and empirical probabilities are used for non-keyword features. Ad-hoc query weights are assigned for these features based on the metadata values.

Genre metadata is handled as follows: *Reaction* or *I-Reaction* typically involve entities that we group as *actors*. This includes persons, organizations, Government entities. The occurrence of such entities triggers Reaction or I-Reaction genre type. If the origin of such an actor is a US location, the text snippet is most likely to be a Reaction than I-Reaction. InfoXtract engine tags associations or relations that can link entities of one type with another. To determine if the text snippet is Reaction or I-Reaction, entities associated with *actors* are searched to see if they include location entities. Location names are checked against a lexicon of US cities, counties and states to determine if the origin of the *actor*. A number of US cities share names with non-US locations - e.g. *Moscow* and *China*. In such instances, the context of the location name is checked. If the next word/phrase is a US location or the phrase "United States" or its variations, the location is classified as a US location. Document-source based features are used to weight text snippets against genre value of *Administration*. We did not eliminate all non-government sourced documents but assigned lower weights than government sourced documents.

The location of the text snippet in a document, presence or absence of *details* (described below) information and the query term coverage were used as triggers for the different values of purpose metadata. We assumed that a text snippet provides details on a particular topic of interest if it contains numeric, percentage, frequency and time information. Such

information, extracted by InfoXtract, are grouped together as *details* information and snippets are weighted based on the frequency of such information. The absence of such information is also used as a feature. The snippet location feature gives more weights to snippets closer to the center of the document. It is based on the heuristic that details and background information are typically found in the middle rather than at the start or end of the document. Queries with metadata background or details assign more weights to this feature than queries with metadata or answer or any.

With the absence of significant training data for the different metadata values, we used ad-hoc query weighting. Expectation Maximization or maximum entropy models can be used, in the presence of training data, to weight these features against the metadata values in a query.

## 3.4 Results and Analysis

Table 7 presents the performance of the different runs submitted for the HARD track. These performance are judged for soft document relevance. Documents that satisfy the query and not necessarily the metadata requirements are identified to be soft relevant to the query. The Best, Worst and Median values for these measures are also given. Concept unigram language model performs better than the rest of our submissions. The use of syntactic information in query modeling provides around 13% improvement over smoothed unigram language model.

| | Rel. Ret. @10 | Avg. Prec. | R-Prec. |
|---|---|---|---|
| ub03sugT | 5.375 | 0.3126 | 0.3335 |
| ub03cugTD | 5.854 | 0.3540 | 0.3784 |
| ub03ugTcugTD | 5.729 | 0.3495 | 0.3663 |
| ub03smfugTD | 3.583 | 0.2073 | 0.2562 |
| Best | 6.5 | 0.4069 | 0.4250 |
| Median | 4.729 | 0.2841 | 0.2994 |
| Worst | 0.417 | 0.0026 | 0.0038 |

**Table 7: Soft Document Relevance Comparison**

Hard document relevance corresponds to documents that are relevant to the query as well as satisfy the metadata requirements of the query. The Hard-relevance performance comparison is given in Table 8.

| | Rel. Ret. @10 | Avg. Prec. | R-Prec. |
|---|---|---|---|
| ub03sugT | 4.042 | 0.2543 | 0.2764 |
| ub03cugTD | 4.542 | 0.2981 | 0.3286 |
| ub03ugTcugTD | 4.479 | 0.2896 | 0.3075 |
| ub03smfugTD | 2.771 | 0.1726 | 0.2078 |
| Best | 5.271 | 0.3875 | 0.3604 |
| Median | 3.792 | 0.2673 | 0.2490 |
| Worst | 0.312 | 0.0038 | 0.0024 |

**Table 8: Hard Document Relevance Comparison**

The *ub03ugTcugTD* run on training topics performed better than the other two baseline runs. While the combination of language models did improve the average precision values for 28 topics over the CULM, the average performance over all topics of CULM is better than the combination model. The performance reduction is significant - as much as 300% - for some topics. The *ub03smfugTD* run that used the metadata to re-rank the results of *ub03ugTcugTD* did not results in any improvements at document level performance measures. All text snippets from a document were selected and ranked for HARD retrieval. The ad-hoc weighting of the features and no cutoff on the number of snippets selected from a relevant document are possible reasons for the decrease in performance.

| | Psg. Prec. @10 | R-Prec. | F @ 30 |
|---|---|---|---|
| ub03sugT | 0.2399 | 0.1594 | 0.1178 |
| ub03cugTD | 0.2987 | 0.1997 | 0.1528 |
| ub03ugTcugTD | 0.2763 | 0.1825 | 0.1368 |
| ub03smfugTD | 0.2313 | 0.1699 | 0.0798 |
| Best | 0.3973 | 0.3195 | 0.1738 |
| Median | 0.2574 | 0.1794 | 0.1000 |
| Worst | 0.0136 | 0.0046 | 0.0010 |

**Table 9: Passage Relevance Comparison**

Table 9 gives the performance measures for passage relevance. The first three runs were evaluated with the whole document being the retrieved passage. *Ub03smfugTD* included passage level results corresponding to the granularity metadata.

# 4. CONCLUSION

This section discusses future directions of our work for the two tracks we participated in. Better query representation, the use of all extracted information and incorporating more domain knowledge in document and query processing are some of the avenues of improvement for the Genomics track. For the HARD track, we have only used a subset of information extracted by the InfoXtract engine to represent queries and model metadata. Ad-hoc weights were assigned to extracted features in ranking documents for a given query. This was partly due to the absence of sufficient training data. We plan to explore some formal methods for modeling metadata - specifically identifying and weighting features that satisfy the metadata requirements of queries.

# REFERENCES


[1] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR'98*, pages 275–281. ACM, New York, 1998.

[2] Rohini Srihari and Wei Li. A Question Answering System Supported by Information Extraction. In *Proceedings of ANLP'00*, pages 166–172, 2000.

[3] Rohini K Srihari, W Li, C Niu, and T Cornell. Infoxtract: A customizable intermediate level information extraction engine. In *Proceedings of the NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*, Edmonton, Canada, 2003.

[4] M. Srikanth and R. Srihari. Exploiting Syntactic Structure of Queries in a Language Modeling Approach to IR. In *to appear in Proceedings of CIKM'03*, 2003.