Identifying Gene Function Descriptions by Probability-based Sentence Selection

Kazuhiro Seki, Nihar Sheth, and Javed Mostafa

Laboratory for Applied Informatics Research, Indiana University
1320 East Tenth Street, LI 011, Bloomington, Indiana 47405-3907, USA

{kseki,nisheth,jm}@indiana.edu

Abstract

This paper proposes an approach to the secondary task in the TREC Genomics Track. We regard the task as identification of the sentences describing gene functions (i.e., GeneRIFs) and propose a method considering two factors: topicality and relevance. The former refers to the topicality of a sentence and is measured based on location information and word frequencies in the article. The latter refers to the relevance as a GeneRIF based on the vocabulary used in the article. We formalize a probabilistic model combining these features. Our method is evaluated on the test set of 139 MEDLINE abstracts, and the results demonstrate that (a) function words in input could help to identify gene function descriptions and that (b) there is a vocabulary peculiar to GeneRIFs and that (c) location information shows the highest predictive power for this particular task despite its simplicity. Additionally, we examine some alternative methods in comparison with our method.

1 Introduction

The volume of publications in the biomedical domain has been rapidly growing, making it difficult for individual researchers to keep themselves updated. This resulted in a strong demand for information retrieval (IR) and information extraction (IE) techniques which could help us manage the information overload.

To foster the IR and IE research in the area of biomedicine, the Genomics Track was launched at the Text REtrival Conference (TREC) 2003 (Hersh, 2002). TREC is one of the major conferences targeting IR and has been contributing to the advance in IR research and related areas (e.g., question answering and filtering) since it first started in 1992.

The Genomics Track is aiming at IR and IE, reflecting the increasing interest in the practical applications of those techniques to the biomedical literature. This year, the Genomics Track offers two independent tasks for IR and IE, namely, the primary and secondary tasks. In short, the primary task aims at finding MEDLINE articles stating the functions associated with given gene names, and the secondary task aims at automatically generating concise descriptions of gene functions stated in given research articles. We are particularly interested in the great potential of IE in this field and therefore targeted the secondary task.

The rest of this paper is structured as follows: Section 2 overviews the secondary task. Section 3 summarizes the past research related to the task. Section 4 describes our proposed method for identifying gene function descriptions. Section 5 reports experiments carried out to evaluate our method. Section 6 compares our method with alternative approaches. Lastly, Section 7 concludes this paper with a brief summary and possible directions for future research.

2 The Secondary Task

The secondary task targets information extraction (IE) from the biomedical literature. Specifically, it aims at generating descriptions related to gene functions in an automated way. For this year, the Track Steering Committee decided to experimentally use GeneRIF (Gene References into Function) entries as the gold standard, which are described in the LocusLink database (Pruitt and Maglott, 2001) maintained by National Center for Biotechnology Information (NCBI).

GeneRIFs are functional annotations of genes and, according to the NCBI web page[1], is defined as "*a concise phrase describing a function or functions (less than 255 characters in length, preferably more than a restatement of the title of the paper).*" They have been mainly annotated by experts in the life sciences at National Library of Medicine (NLM). Figure 1 shows an example, where `GRIF` provides PubMed identifier (PMID) and the associated GeneRIF, separated by a vertical line (i.e., PMID is `12037388` and GeneRIF is "`NAT1 polymorphisms may be`...").

Our goal is to automatically generate a GeneRIF, given an abstract or a full text associated with the correspond-

[1] http://www.ncbi.nlm.nih.gov/LocusLink/GeneRIFhelp.html

```
LOCUSID:             9
LOCUS_CONFIRMED:     yes
LOCUS_TYPE:          gene with protein
                     product, function known
                     or inferred
                     .....
OFFICIAL_SYMBOL:     NAT1
OFFICIAL_GENE_NAME:  N-acetyltransferase
                     1 (arylamine
                     N-acetyltransferase)
ALIAS_SYMBOL:        AAC1
                     .....
GRIF:                12037388|NAT1
                     polymorphisms may be
                     correlated with an
                     increased risk of larynx
                     cancer
```

Figure 1: A fragment of a LocusLink record. GRIF gives a PMID and a gene function description (i.e., GeneRIF).

ing PMID as input. In addition, we may use official gene names and aliases provided by LocusLink (e.g., OFFICIAL_GENE_NAME) in generating a GeneRIF. (However, as described later, we use only abstracts and basically do not use gene names in this study.)

The generated GeneRIF candidates are to be evaluated using the Dice coefficient and its variants, which measure the extent of word overlap between the generated GeneRIF candidate and the actual GeneRIF. Section 5 will formally describe the evaluation measures.

3 Related Works

Given that 95% of actual GeneRIFs are reported to contain some text from titles or abstracts (Hersh, 2003), we view the secondary task as sentence selection, that is, we simplify the task to identifying those sentences which are likely to describe gene functions. Sentence selection (or passage retrieval) is one of core components of automatic summarization and question answering (QA) systems and has been widely explored.

Text summaries are typically generated by extracting text segments based on several features, such as existence of title words, locations of text segments, and similarities between the text segments and the entire text (Gong and Liu, 2001; Chuang and Yang, 2000; McDonald and Chen, 2002). By using these features, each text segment is given a score indicating the extent to which the segment would be included in a summary. We will utilize some of these features to identify topical descriptions.

QA aims at providing the information that directly answers users' questions, as opposed to a ranked list of documents usually returned by conventional IR systems. Most of current QA systems are composed of four basic modules (Tellex et al., 2003): question analysis, document retrieval, passage retrieval, and answer extraction. Here, let us focus on passage retrieval, which breaks down documents retrieved in the preceding module into smaller units, such as sentences, and returns only passages potentially relevant to the query. Passage retrieval is often treated on an analogy to IR, where each passage is regarded as a document and relevant passages are retrieved based on their similarities to a query. In the secondary task, the query is "gene functions," which will be too general to find relevant passages. Instead, we use vocabulary related to gene functions for measuring relevancy of passages, which will be described next.

4 Our Method

4.1 Probabilistic Sentence Selection

The secondary task can be performed by identifying those sentences which describe gene functions, assuming that such sentences exist in an input article. We propose a probabilistic model incorporating two measures: relevance and topicality.

For the relevance as GeneRIFs, we make use of word frequencies in GeneRIFs; higher scores are given to those sentences which contain more words frequently appearing in GeneRIFs. Our assumption is that there is a typical vocabulary used for gene functions frequently. For example, "activate" or "bind" may be often used in describing gene functions and then a sentence containing those words could receive higher scores. For sentence s composed of a sequence of words $w_1\ w_2 \cdots w_n$, the probability of being GeneRIF can be formalized as a product of the relative frequencies of the words composing the sentence as in Equation (1), where $F_G(w_j)$ and N_G denote the frequency of word w_j and the total number of words in GeneRIFs, respectively.

$$P_G(s) = \prod_{j=1}^{n} P_G(w_j) = \prod_{j=1}^{n} \frac{F_G(w_j)}{N_G} \qquad (1)$$

Incidentally, this can be regarded as a unigram language model; that is, it models GeneRIF descriptions by word unigrams.

For the topicality of sentences, we use word frequencies in a given article; higher scores are given to those sentences which contain more words frequent in the article itself. Note that word frequencies here are based on the *input*, as opposed to P_G based on word frequencies in GeneRIFs. The rationale behind it is that the topic of the article is likely to be repeatedly stated. Given this assumption, the probability of being a topical sentence can be expressed as in Equation (2), where N_T denotes the total number of word tokens in the given article and $F_T(w_j)$ is a frequency of word w_j in the given article.

$$P_T(s) = \prod_{j=1}^{n} P_T(w_j) = \prod_{j=1}^{n} \frac{F_T(w_j)}{N_T} \tag{2}$$

As with P_G, P_T can be regarded as a unigram language model; that is, it models an input article by word unigrams.

Additionally, we use location information as an indicator of topicality, given the fact that there is a conventional structure of where topics of articles appear at some typical locations. We define $P_L(L(s))$, which is a probability that sentence s is a topic sentence, based on its location $L(s)$. The function $L(s)$ returns a location of s and its possible values are *title*, *abstract_last*, *abstract_body*, defined based on our preliminary study having suggested that actual GeneRIFs occur in many cases in titles or the end of abstracts. It returns *title* if s is a (part of) title; *abstract_last* if s is the last sentence of abstracts; and *abstract_body* otherwise.

We combine these probabilities introduced above, i.e., P_G, P_L, and P_T, assuming their mutual independency, as in Equation (3).

$$P(s) = P_T(s) \cdot P_L(L(s)) \cdot P_G(s) \tag{3}$$

Since $P(s)$ is influenced by sentence lengths (longer sentences are generally result in smaller values), we normalize it by the number of words n in sentence s and take a logarithm for computational efficiency, forming a score indicating the extent to which s is likely to be a GeneRIF.

$$\begin{aligned} S_{GRIF}(s) &= \log P(s)^{\frac{1}{n}} \\ &= \log \left(P_T(s) \cdot P_L(L(s)) \cdot P_G(s)\right)^{\frac{1}{n}} \\ &= \frac{1}{n} \left(\log P_T(s) + \log P_L(L(s)) + \log P_G(s)\right) \end{aligned} \tag{4}$$

We select the sentence which maximizes $S(\cdot)$ as output (GeneRIF), that is:

$$\hat{s} = \arg\max_{s_i} S_{GRIF}(s_i) \tag{5}$$

4.1.1 Probability estimation

To compute S_{GRIF}, we estimate the probabilities, P_G, P_T, and P_L, as follows.

Firstly, as defined in Equation (1), $P_G(s)$ is a product of relative frequencies of words composing sentence s in GeneRIFs. This can be estimated based on word frequencies in a training set of GeneRIFs. However there are two things to take into account, that is, function words (e.g., *the* and *to*) and word inflection (e.g., *activate* and *activation*). We use a stop word list[2] containing 571 words so as to exclude function words, and use the Porter stemmer (1980) so as to eliminate inflectional variations. To observe their effect on this task, we create four different models of P_G with/without applying the stop word list and the stemmer. In addition, we employ a discounting (smoothing) method, since a significant number of words in input texts would never appear in GeneRIFs and thus we will encounter unknown words in estimating their probabilities, i.e., the zero frequency problem. The absolute discounting method (Ney et al., 1994) is experimentally used to remedy the problem. Absolute discounting takes out a constant proportion from the probability mass and uniformly distributes it to unknown words.

Secondly, $P_T(s)$ can be calculated by simply counting word frequencies in the input. As with P_G, again we make use of the stop word list and the Porter stemmer in order to deal with function words and word inflection, and create four different models of P_T by using/not using the stop word list and the stemmer. Notice that estimating probabilities P_T does not require to train the model in advance, as opposed to P_G.

Lastly, P_L can be estimated by counting where GeneRIFs appear in their corresponding articles, given pairs of articles and GeneRIFs. For example, if most GeneRIFs are taken from titles, $P_L(title)$ would have a high probability. However, this cannot be automatically done because GeneRIFs are not marked in articles and they are not even guaranteed to literally appear in articles since they are generated by human; words, phrases, or word orders may be changed in abstracting GeneRIFs from articles. Thus, it is ideal to use human in order to accurately identify where GeneRIFs (or similar sentences) appear, which is however costly. Instead, we use *bigram phrase* Dice coefficient (see Section 5.1) to measure how similar each sentence is to the corresponding GeneRIF, and consider the computed similarity as the number of occurrences of the GeneRIF. To put it differently, given an article (sentences) and GeneRIF, we compute a similarity score between each sentence and the GeneRIF, and the similarity scores are summed up within each category of location (i.e., *title*, *abstract_last*, and *abstract_body*), which is regarded as a frequency of the GeneRIF occurrences in the location.

5 Evaluation

5.1 Methodology

We evaluate our proposed method on the 139 MEDLINE abstracts provided for the secondary task, where each of the abstracts is associated with an actual GeneRIF. Full-text articles are also available for this task, but we use only abstracts (and titles) given that 95% of actual GeneRIFs contained some text from titles and abstracts (Hersh, 2003). In addition, gene names associated with each GeneRIF can be utilized, but our framework does not incorporate them because, in the actual GeneRIF annotation, indexers do not have specific gene names in mind in advance.

[2] ftp://ftp.cs.cornell.edu/pub/smart/english.stop

As evaluation metrics, the secondary task uses the Dice coefficient to measure similarities between actual GeneRIFs and generated GeneRIF candidates. Given two strings s_x and s_y, the Dice coefficient σ between s_x and s_y is defined as in Equation (6), where N_x, N_y, and N_{xy} are the numbers of words in s_x, in s_y, and in both s_x and s_y, respectively.

$$\sigma(s_x, s_y) = \frac{2 \cdot N_{xy}}{N_x + N_y} \tag{6}$$

However, the Dice coefficient has several limitations as an evaluation metric for this task. Most of them result from the fact that it treats strings as *bags of words* and treats words just as symbols. To compensate for the problems to some extent, the secondary task uses four variants of the Dice coefficient below.

- Classic Dice (CD):
  Uses the Dice coefficient after removing stop words and stemming suffixes. This enables an evaluation based on normalized contents words.
- Modified Unigram Dice (MD):
  Similar to CD but considers word frequencies to give additional scores to words appearing multiple times in both strings compared.
- Bigram Dice (BD):
  Regards two adjacent words (bigrams) as a unit for comparison and applies the Dice coefficient. This allows us to take word order into account to some extent.
- Bigram Phrases (BP):
  Same as BD but excludes bigrams containing stop words. This metric has more focus on noun phrases.

5.2 Results

5.2.1 Exploring Word Frequencies in Input

We examined the effects of stemming word suffixes and removing stop words on $P_T(s)$, which indicates the topicality of sentence s based on word frequencies in an input text. We applied the model with/without stemming and removing stop words, and output the predicted GeneRIFs with the highest probabilities. Table 1 shows the result, where bold figures indicate the highest similarities for each evaluation metric.

Somewhat unexpectedly, the result indicates that the stemmer and the stop word list did *not* contribute to predicting actual GeneRIFs. Especially, when stop words were removed, Dice coefficients radically dropped by more than 10 points, irrespective of evaluation criteria. In IR and related areas, stop words are commonly thought to be less (or not at all) informative and removed, but for this particular task, stop words appear to play a certain role to characterize GeneRIFs.

In the remainder, we do not use the stemmer nor exclude stop words for P_T estimation.

Table 1: Effects of stemming and removing stop words in estimating $P_T(s)$. CD, MD, BD, and BP denote classic Dice, modified unigram Dice, bigram Dice, and bigram phrase, respectively.

		Stop words			
		remained		excluded	
Stemmer	off	CD	**38.37**	CD	26.36
		MD	**39.04**	MD	26.37
		BD	**21.45**	BD	11.38
		BP	**24.55**	BP	13.59
	on	CD	36.69	CD	25.86
		MD	37.38	MD	25.24
		BD	20.27	BD	10.27
		BP	23.42	BP	12.43

5.2.2 Exploring Word Frequencies in GeneRIF

As with $P_T(s)$ above, we examined the effects of stemming word suffixes and removing stop words on $P_G(s)$, which is a probability that a given sentence s is relevant to gene functions based on the vocabulary used in GeneRIFs.

Estimating P_G requires training data. However, since there are no training data besides the test data of 139 GeneRIFs, we trained the model by a leave-one-out cross-validation using the test data, where each GeneRIF was predicted based on the model trained on the other 138 GeneRIFs; that is, training data and test data are always mutually exclusive. We created four different models with/without stemming and removing stop words, and evaluated their effectiveness. The result is shown in Table 2, where bold figures indicate the highest similarities for each metric.

Table 2: Effects of stemming and removing stop words in estimating $P_G(s)$.

		Stop words			
		remained		excluded	
Stemmer	off	CD	32.68	CD	35.55
		MD	31.30	MD	36.83
		BD	15.94	BD	20.80
		BP	18.75	BP	23.41
	on	CD	31.72	CD	**36.68**
		MD	29.25	MD	**37.55**
		BD	14.59	BD	**22.02**
		BP	16.77	BP	**24.88**

As opposed to the case with P_T, stemming and removing stop words resulted in the best result. Especially, removing stop words improved the similarity scores by 3–8 points (9–50%). It proves that there exists a vocabulary particularly used for describing GeneRIFs (or gene functions). Incidentally, it was found that when only the stemmer was applied without removing stop words, it decreased the similarities.

As an illustration, Table 3 shows the 12 stems, excluding stop words, which most frequently appeared in the test data set of 139 GeneRIFs, where one can find a number of stems related to gene functions, such as “activ”, “regul”, and “role”.

Table 3: The most frequent 20 stems in the test data of 139 GeneRIFs. The figures on their right show the logarithms of their relative frequencies.

Rank	Stem	$\log P_G(w)$	Rank	Stem	$\log P_G(w)$
1	activ	−1.4838	7	express	−1.7137
2	cell	−1.4838	8	gene	−1.8031
3	regul	−1.5342	9	induc	−1.8031
4	protein	−1.6396	10	signal	−1.8031
5	role	−1.6569	11	mediat	−1.8286
6	1	−1.7137	12	receptor	−1.8286

In the remainder, we use the stemmer and remove stop words for P_G estimation.

5.2.3 Exploring Optimal Combinations of Models

As defined in Equation 3, our final model combines three independent estimations: $P_T(\cdot)$, $P_G(\cdot)$, and $P_L(\cdot)$. To demonstrate the contribution of each model and to explore their optimal combination, we evaluated each model and every combination on the test data set. Table 4 summarizes the results of the different models, where the top row indicates the combinations of models applied. The right most column ($P_T \cdot P_G \cdot P_L$) shows our submitted official run.

From the results in Table 4, it is apparent that, despite its simplicity, location information (P_L) dominantly contributed to the result and the other two models hardly had effect on the outcome when combined with P_L. This is mainly because the actual GeneRIFs are more or less taken from titles in many cases.

6 Discussions

The evaluation in Section 5 revealed that location information impacts the most in identifying GeneRIFs. However, it does not mean that we can ignore the contents of input sentences because whether each sentence describes gene functions depends on its contents. We explore an alternative method making use of contents (word frequencies) from a viewpoint of classification.

The secondary task can be seen as classification, assigning a class (c_{GRIF} or $c_{nonGRIF}$) to each sentence and selects the one which is most likely to be a GeneRIF. There is a number of methods that can be applied, e.g., naive Bayes classifiers, decision trees, support vector machines. These methods have been compared for their effectiveness and, to our knowledge, there is no clear evidence about which performs best; it depends on tasks applied to, training data size, the number of classes, and so on (Chuang and Yang, 2000; Yang and Liu, 1999). For comparison, we experimentally implemented a naive Bayes classifier, which has been widely used in past research.

The naive Bayes classifier predicts class $\hat{c}$ for input s, where $\hat{c}$ maximizes the probability $P(c_k|s)$ and c_k can be either c_{GRIF} or $c_{nonGRIF}$.

$$\begin{aligned}\hat{c} &= \arg\max_{c_k} P(c_k|s) \\ &= \arg\max_{c_k} P(s|c_k)P(c_k)\end{aligned} \tag{7}$$

For each sentence s_i, we compute a likelihood ratio of a probability associated with class c_{GRIF} to one associated with class $c_{nonGRIF}$, and select a sentence as a GeneRIF which produces the highest ratio.

$$\begin{aligned}\hat{s} &= \arg\max_{s_i} \frac{P(s_i|c_{GRIF})P(c_{GRIF})}{P(s_i|c_{nonGRIF})P(c_{nonGRIF})} \\ &\approx \arg\max_{s_i=w_1\ldots w_n} \prod_{w_j} \frac{P(w_j|c_{GRIF})}{P(w_j|c_{nonGRIF})}\end{aligned} \tag{8}$$

We used the GeneRIFs in the test set to train the classifier for class c_{GRIF} (i.e., the numerator) and used the abstracts to train it for class $c_{nonGRIF}$ (i.e., the denominator). Although most abstracts would include GeneRIFs, it should not be harmful as long as there are more non-GeneRIFs than GeneRIFs in the training data. We evaluated the method on the test set; Table 5 compares the results produced by our model ($P_T \cdot P_G$) and the naive Bayes classifier.

Table 5: Comparison between our model based on word frequencies ($P_T \cdot P_G$) and the naive Bayes classifier for identifying GeneRIFs.

	$P_T \cdot P_G$	Bayes
CD	39.11	34.70 (−11.2%)
MD	40.62	34.66 (−14.7%)
BD	22.42	19.64 (−12.4%)
BP	25.78	22.18 (−13.4%)

Our method outperformed the naive Bayes classifier in all evaluation criteria. The result demonstrates the effectiveness of our method but, at the same time, it implies the limitation of the methods based solely on word distributions, as location information alone results in much higher similarity scores.

In order to combine multiple information sources, our model multiplies the resulting probability estimates (i.e., P_T, P_G, and P_L). This can be regarded as probability voting. On the other hand, one of voting algorithms often used is *majority voting* where each information source gives a vote to its best candidate and the one which received the majority of votes wins. We implemented a (modified) majority voting method for comparison. The voting scheme considers every candidate and cast $1/n$ votes for n-th ranked candidate, so as

Table 4: Results for different combinations of models. Bold characters indicate the best score for each row across the combinations. The right most column ($P_T \cdot P_G \cdot P_L$) is our submitted official run.

	P_T	P_G	P_L	$P_T \cdot P_G$	$P_T \cdot P_L$	$P_G \cdot P_L$	$P_T \cdot P_G \cdot P_L$
CD	38.37	36.68	**50.47**	39.11	50.25	50.44	50.40
MD	39.04	37.55	**52.60**	40.62	52.36	52.52	52.56
BD	21.45	22.02	34.82	22.42	34.66	**34.93**	34.83
BP	24.55	24.88	37.91	25.78	37.92	**38.05**	37.97

to avoid the case where no candidate receives the majority. Equation (9) shows the formula.

$$\hat{s} = \arg\max_{s_i} \sum_{P \in \{P_T, P_G, P_L\}} \frac{1}{rank(P(s_i))} \tag{9}$$

where, $rank(P(s_i))$ is a rank of candidate (sentence) s_i based on probability $P(\cdot)$. Table 6 compares two voting schemata, i.e., probability voting (our model) and majority voting.

Table 6: Results for different voting schemata: probability voting (our model) vs. majority voting.

	$P_T \cdot P_G$		$P_T \cdot P_G \cdot P_L$	
	prob	majority	prob	majority
CD	39.11	39.67 (+1.4%)	50.40	42.43 (−15.6%)
MD	40.62	41.06 (+1.1%)	52.56	44.20 (−15.9%)
BD	22.42	24.06 (+7.3%)	34.83	26.75 (−23.2%)
BP	25.78	27.74 (+7.6%)	37.97	30.59 (−19.4%)

When used for combining two probabilities ($P_T \cdot P_G$), majority voting improved the result, especially for bigram-based evaluation criteria (BD and BP). On the other hand, when applied to combine P_T, P_G, and P_L, it significantly decreased the similarity scores by 15%–23%. This is presumably because the probability P_L has much more predictive power than the others. Weighted voting, which gives certain weights to each source, could work better for this model.

Lastly, we report the result when gene names are used as a filter. Each MEDLINE article in the test set is associated with specific gene names, thus it is very likely that gene function descriptions (GeneRIFs) would contain those gene names in them. Based on this assumption, we restricted the system output to those containing the associated gene names. In cases where no gene name appeared in input sentences, the highest ranked sentence was outputted. The experimental result is presented in Table 7.

The filter using gene names did not raise the result. This implies that (exact) gene names do not necessarily appear in GeneRIFs.

7 Conclusions and Future Directions

Table 7: Results when gene names are used/not used as a filter. The columns labeled “not used” are the results of our proposed method.

	$P_T \cdot P_G$		$P_T \cdot P_G \cdot P_L$	
	not used	used	not used	used
CD	39.11	36.18 (−7.5%)	50.40	48.41 (−3.9%)
MD	40.62	36.64 (−9.8%)	52.56	50.28 (−4.3%)
BD	22.42	21.51 (−4.1%)	34.83	32.98 (−5.3%)
BP	25.78	24.44 (−5.2%)	37.97	36.32 (−4.3%)

This paper presented a method for identifying gene function descriptions (GeneRIFs) in biomedical articles. We regarded the task as sentence selection assuming that input articles do contain actual GeneRIFs. Our method exploits location information and word frequencies both in input and GeneRIFs and, given an input text, identifies a sentence which is most likely a GeneRIF using a probabilistic model. We evaluated our method on the test set of 139 MEDLINE abstracts, and the results indicated that (a) function words in input can be used for identifying GeneRIFs, that (b) there exists a vocabulary peculiar to gene function descriptions, and that (c) location information has the most impact in identifying GeneRIFs.

Future directions would include the use of a larger training set and wider contexts for modeling and probability estimation, rather than independent word occurrences. In addition, the effect of using full text articles in estimating P_T (which is based on word frequencies in input) should be investigated.

Acknowledgment

This project is partially supported by the NSF ITR grant #9817572.

References

Chuang, W. T. and Yang, J. (2000). Extracting sentence segments for text summarization: a machine learning approach. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 152–159.

Gong, Y. and Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR* *conference on Research and development in information retrieval*, pages 19–25.

Hersh, W. (2002). Text retrieval conference (TREC) genomics pre-track workshop. In *Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries*, pages 428–428.

Hersh, W. (2003). TREC genomics track overview. In *The Twelfth Text REtrieval Conference (TREC 2003) Notebook*. National Institute of Standards and Technology. Available at `http://medir.ohsu.edu/˜genomics/overview.pdf`.

McDonald, D. and Chen, H. (2002). Using sentence-selection heuristics to rank text segments in TXTRACTOR. In *Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries*, pages 28–35.

Ney, H., Essen, U., and Kneser, R. (1994). On structuring probabilistic dependences in stochastic language modeling. *Computer Speech and Language*, 8(1):1–38.

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

Pruitt, K. D. and Maglott, D. R. (2001). RefSeq and LocusLink: NCBI gene-centered resources. *Nucleic Acids Research*, 29(1):137–140.

Tellex, S., Katz, B., Lin, J., Fernandes, A., and Marton, G. (2003). Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–47.

Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49.