Automatically Generating Gene Summaries from Biomedical Literature

Xu Ling, Jing Jiang

May 8, 2005

Abstract

In this project, we address the problem of automatic gene summarization from biomedical literature. The ultimate goal is to automatically extract relevant statements from biomedical literature to summarize the already-discovered knowledge about a target gene, such as sequence information, mutant phenotypes, and molecular interaction with other genes, *etc.* The problem was divided into two sub-tasks: ad hoc retrieval of relevant documents, and information extraction of knowledge about the target gene from the retrieved documents. In order to help summarize the target gene in several aspects, we tried different clustering approaches to group similar documents together. We also attempted 3 heuristics to extract a number of sentences that can best represent the key information related to the target gene conveyed by these documents. Two experiments were conducted on the clustering and the sentence extraction task. The preliminary results indicated that those approaches are effective in summarizing the gene from literature.

1 Introduction

The growing amount of scientific discoveries in genomics and related biomedical disciplines have led to a corresponding growth in the amount of data and information. Meanwhile, with the fast growth of digital libraries and the Internet, many biomedical articles are now easily accessible in electronic format. Biomedical literature has thus become an important source of information as more and more researchers turn to the literature to search for knowledge that can drive new hypotheses and research. Because of the daunting size and complexity of the biomedical literature, there have been increasing efforts devoted to integrate this huge amount of information for the biologists to digest quickly.

Of particular note are the model organism genome databases such as FlyBase, Mouse Genome Informatics, and Saccharomyces Genome Database. These databases bring together all kinds of information in an easy-to-use format. For example, in FlyBase, a report consisting of a paragraph of text that summarizes the various information attributed to particular references is provided for each

Drosophila gene. Information in this summary includes sequence structure, cytogenetic map, phenotypic information, gene ontology, wild-type functions, *etc.* By compressing all the knowledge from a huge amount of literature into a short paragraph and arranging it in an easy-to-read format, such a gene summary can help biologists to quickly understand the target gene and to further study the gene. However, such gene summaries require extensive human annotation or curation, which is usually done by PhD-level researchers. With the rapid growth of scientific discoveries in the biomedicine field, it seems too inefficient and probably unrealistic any more for a group of annotators to construct such curated databases by reading the literature one article by one article. Therefore, many IR and NLP researchers have started developing high quality information tools to aid such curation.

The goal of our work is to use Information Retrieval (IR) and Information Extraction (IE) techniques to automate gene summarization from biomedical literature. If gene summaries can be automatically generated with decent accuracy, curation of databases such as FlyBase can be done much faster and with less human effort.

2 Related Work

Most efforts concerned with biomedical literature mining to date focus on automated information extraction, using natural language processing techniques to identify relevant phrases and facts in text (see [6] for a review). Existing work typically aims at assisting in finding the information in a specific aspect about a given gene or about relationships between specific genes, e.g., gene location on chromosomes [3], and protein-protein interactions [2].

The previous work that is the closest to ours may be the Genomics Track in the Text REtrieval Conference (TREC), first launched in 2003. The first task in Genomics Track 2003 was to find articles about the query gene that describe some aspect of its function. The second task was to generate descriptions related to gene functions from MEDLINE records. A major difference between Genomics Track and our work is that Genomics Track provided the participants with comparatively clean data to start with, while we are dealing with real world data. We also propose to relax the precision of the document retrieval sub-task in an attempt to achieve higher recall. We then rely on the information extraction sub-task to filter out irrelevant documents.

There has been extensive study on automatic text summarization. [5] surveyed different genres of summaries and techniques developed for automatic summarization, as well as evaluation metrics. According to their classification, our gene summaries are a type of informative, query-oriented, multi-document extracts. Thus, we can adopt some typical methods for this kind of summarization, such as location-based method, title-based method, and word-frequency method, although we should take into consideration the special characteristics of biomedical literature.

3 System Overview

Our automatic gene summarization system mainly consists of two components: an Information Retrieval module that retrieves documents useful for a target gene summarization from a collection of documents, and an Information Extraction module that extracts sentences from the retrieved documents to summarize the target gene. The Information Extraction module itself consists of two components, one for document clustering, and the other for sentence extraction. The whole system is illustrated in Figure 1.

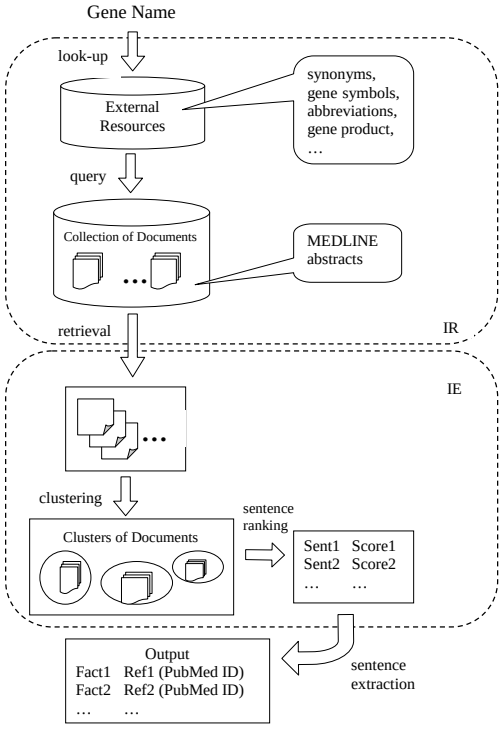

Figure 1: System Overview.

In the following two sections, we describe the techniques we employed in our system in detail.

4 Information Retrieval Module

The first step of gene summarization is to identify documents that may contain useful information for summarizing the target gene. We took the simplest approach by retrieving all documents that contain the name or any synonym of the target gene. This set of documents will be further processed for summarization.

4.1 Gene *SynSet* Construction

The major IR technique we applied to document retrieval is query expansion. In biomedical literature, since there is no standard gene nomenclature, gene synonyms are very common. When the user asks for summarization of a certain gene, it is therefore important to consider all synonyms of this gene when looking for relevant documents. For model organisms such as Drosophila, there are compiled gene synonym lists that can be used for this purpose. In our experiments, we used a Drosophila gene synonym list provided in BioCreAtIvE Task 1B. We extended this list by adding names of proteins encoded by each gene, where the information is obtained from FlyBase's gene annotation. Our belief is that mentions of these proteins are also strong indications of reference to the corresponding gene. In the end, for each known Drosophila gene, we got a set of synonyms and protein names, which we call a gene *SynSet*.

When user specifies a gene name, we look for documents that contain at least one occurrence of any synonym in the SynSet of this gene. When a gene synonym contains multiple tokens, we search for exact matches of this multi-token phrase.

For Drosophila and for many other organisms, some gene synonyms can be also common English words. For example, gene *foraging* is abbreviated as *for*, which is a common proposition in English. Searching for matches of such common English words would introduce too many false positives. We took a heuristic solution by removing stop words from the SynSets.

4.2 No *TF-IDF* Ranking

Note that we did not use any standard TF-IDF ranking or length normalization of the documents. The reasons are fourfold. First, we observed that for most genes, the number of documents retrieved by this simple approach is small. For example, out of the 35971 Drosophila genes that currently our system supports, 31675 (or 88.1%) of them have only less than or equal to 20 documents that mention the gene name. For such small set of documents, ranking becomes unnecessary. Second, even for genes with a large number of matching documents, we found that TF-IDF ranking does not necessarily help find the more relevant documents. For many genes that have a large number of matching documents, the false positive documents result from ambiguity of the gene name or synonym. For example, Drosophila gene *alphaTry* (FBgn0003863) has 1070 retrieved documents, but most of them are false positives because one of the synonyms of *alphaTry* is *alpha*, which is a common word in biomedical literature. Another example is gene *Bar*, which has a synonym *B*. Although *B* itself is not an English word, it is also a synonym of 24 other Drosophila genes. For this kind of genes, term frequency of the query word cannot be used to discriminate against false positive documents because false positive documents can also have a high frequency of the ambiguous gene synonym. Third, TF-IDF ranking may still be useful to differentiate between documents that focus on the target gene and documents that only mention the target gene as a side topic. However, we

believe that even if the target gene is only a subtopic, that piece of information may still be useful for our gene summarization. Fourth, length normalization is not important in our case because the documents we consider are PubMed abstracts, which are all short documents of about 10 sentences.

5 Information Extraction Module

The information extraction module takes a set of documents returned from the information retrieval module, and extracts statements that contain useful factual information about the target gene. Because information about a certain gene can touch several different aspects, the statements are also organized into several categories.

To identify such kind of different aspects of information and sentences that are most informative in representing such aspects, the information extraction module is separated into two components: a document clustering component and a sentence extraction component.

5.1 Document Clustering

The purpose of the document clustering component is to group the retrieved documents into several clusters where each cluster focuses on one aspect of the knowledge about the target gene.

We tried two clustering methods: K-means clustering and a probabilistic theme clustering method based on PLSA.

5.1.1 K-Means Clustering

K-Means is a commonly used method for document clustering. [7] found that regular K-means is more effective than agglomerative clustering. In our experiments, each document d_i is represented as a vector $(w_{i1}, w_{i2}, \ldots, w_{i|V|})$, where V is the vocabulary. The weight w_{ij} for word v_j in d_i is defined as $w_{ij} = \mathrm{TF}_{ij} \times \mathrm{IDF}_j$, where TF_{ij} is the term frequency of word v_j in d_i, $\mathrm{IDF}_j = 1 + \log(\frac{N}{n_j})$, n_j is the number of documents containing word v_j, and N is the total number of documents in the collection. Cosine similarity is used to measure the distance between two document vectors.

The results of K-means clustering depend on the selection of the starting cluster centroids. We randomly pick the starting cluster centroids among the documents, and repeat 5 times to pick the run that gives the highest overall similarity, The similarity of a cluster is defined in [7] as

$$\frac{1}{|S|^2} \sum_{d \in S, d' \in S} \cos(d, d'),$$

where S is the cluster of documents.

5.1.2 Probabilistic Theme Clustering

We extract common k themes from each document d (*i.e.*, PubMed abstract) using the simple probabilistic mixture model presented in [4]. The basic idea of this method is to treat the words in each document as observations from a mixture model where the component models are the theme word distributions and a background word distribution. Hence, for each document d, there is a mixing weight $\pi_{d,j}$ for choosing the j-th theme θ_j. Words in the same document share the same mixing weights. In this approach, we use the estimated $\pi_{d,j}$ to measure the strength of the theme j in d. Then we assign each document d to the cluster j such that theme j is the strongest one in d, *i.e.*, cluster $C_j = \{d|d \in C, \pi_{d,j} \geq \pi_{d,i}, 1 \leq i \leq k\}$.

The model can be estimated using the Expectation Maximization (EM) algorithm to obtain the theme word distributions. Specifically, let $\theta_1, \ldots, \theta_k$ be k theme unigram language models (*i.e.*, word distributions) and θ_B be a background model for the whole collection C. A document d is regarded as a sample of the following mixture model:

$$p(w, d) = \lambda_B p(w|\theta_B) + (1 - \lambda_B) \sum_{j=1}^{k} [\pi_{d,j} p(w|\theta_j)],$$

where w is a word in document d, $\pi_{d,j}$ satisfies that $\sum_{j=1}^{k} \pi_{d,j} = 1$, and λ_B is the mixing weight for θ_B.

The algorithm is going to achieve the local Maximum A Posteriori (MAP): $(1-\lambda_p) \log p(C|\Lambda) + \lambda_p \log p(\Lambda)$. As here we do not have any prior, λ_p is set to 0. Hence the objective is to maximize $\log p(C|\Lambda)$, the log likelihood of seeing all the documents under the estimated parameters, where the log-likelihood of each d is $\log p(d|\Lambda) = \sum_{w \in V} [c(w, d) \times \log \Big(\lambda_B p(w|\theta_B) + (1 - \lambda_B) \sum_{j=1}^{k} (\pi_{d,j} p(w|\theta_j))\Big)]$. In our experiments, we use multiple trials to improve the local maximum we obtain.

5.1.3 Probabilistic Theme Clustering with Prior Theme Language Models

We observed that much annotation work has been done to summarize the knowledge about genes. For instance, the full reports for each gene in FlyBase arrange various information attributed to particular references. From these reports, we know that the information of a gene usually includes sequence information, phenotypic information, wild-type functions, genetic interactions, *etc.* We thus expect that all the abstracts could be classified into several predefined categories, such as information about function, sequence, interactions with other genes, and behavior studies. For some categories (*e.g.*, function), we could get a collection of documents from FlyBase, which have been annotated as containing the relevant information. We expect this prior knowledge to help us detect the hidden themes.

In out experiments, we collect all the PubMed abstracts from FlyBase which were annotated as containing information of *GO functional annotation, Phenotypic Info., Wild-type function* into the category called *Function*, and all the abstracts containing information of *Interacts genetically with* into the category called *Regulation and Interaction*. we build two prior theme language models according to these two collections respectively, and set $\lambda_p = 0.2$ in the objective function of EM algorithm. Then when we employ EM algorithm to extract the hidden themes, we use the above two prior theme language model as two of the initial points and generate the other initial points randomly.

5.2 Sentence Extraction

Given a cluster of documents related to the same aspect of information of the target gene, we want to extract a number of sentences that can best represent the key information related to the target gene conveyed by these documents. There are many heuristics for selecting such sentences proposed by previous work on text summarization. In our system, we used the following methods to rank sentences in a cluster, and then pick the top k sentences as summary of the cluster.

5.2.1 Ranking by Similarity to Centroid

The centroid method looks for sentences that are most similar to the centroid of a cluster of documents. We explored two methods to measure the similarity between a sentence and the centroid of a cluster.

In the first method, we directly computed the cosine similarity between the sentence vector and the centroid vector.

The second method we used is a modified version of the scoring function in [4]. For each cluster, we first compute the cluster centroid vectors, which are means of the document vectors of each cluster. The document vector is defined the same as in Section 5.1.1. Once we get the cluster centroids, we pick the top m words with the highest TF-IDF weight for each cluster. In our experiments, we set m to 10 as in [4]. Let $k_1, k_2, \ldots, k_m$ be the m words we pick. We call this the set of keywords. Let $w_1, w_2, \ldots, w_m$ be the TF-IDF weights of these m keywords. For each sentence, we find the first occurrence of any keyword in the sentence, and the last occurrence of any keyword in the sentence. Let N be the number of words between these two positions. Let n be the number of keywords in this sentence. The the score of the sentence is

$$s = \frac{n^2}{N} \cdot \sum_{l=1}^{m} w_l \cdot I(k_l),$$

where $I(k_l)$ is 1 if k_l occurs in the sentence and 0 otherwise.

5.2.2 Ranking by Sentence Probability

From the above probabilistic theme clustering approach, we have estimated the words distribution model for each theme. So basically for each sentence s in a document d, we could calculate the probability of observing s under theme j:

$$p(s|\theta_j) \propto \pi_{d,j} \frac{|s|!}{\prod_{w \in s} c(w,s)!} \prod_{w \in s} p(w|\theta_j)^{c(w,s)},$$

where $c(w,d)$ is the count of word w in sentence d, and $|s| = \sum_{w \in s} c(w,s)$ is the length of the sentence. Here, we use multinomial distribution to eliminate the affect of the sentence length. In this way, we can use this probability to rank the sentences in each cluster.

5.2.3 Other Heuristics

We also considered some other heuristics.

First, we considered the difference between sentences containing the target gene and sentences not containing the target gene. Although sentences not containing the target gene may also contain useful contextual information, we should put strong prior on sentences containing the target gene. In our experiments, we tried filtering out sentences not containing the target gene before we ranked the sentences, and compared this result with the result without filtering.

We also considered the title method. If the sentence is the title of an abstract, presumably it is more concise and more informative. We defined the title score as follows: $s = 1$ if the sentence is a title sentence, and 0 otherwise. We combined this title score with the basic centroid score, and compared its performance with using centroid score alone.

6 Experiments and Evaluation

6.1 Data Set

Our experiments are focused on fly genes, although the methodology should apply to other organisms, too. We downloaded 22092 PubMed abstracts that are related to Drosophila as our document collection. We used Lemur Toolkit to index this collection, and implemented most of our functions using Lemur Toolkit.

6.2 Results and Evaluation

Since it is very time consuming to manually generate evaluatioin standards, we randomly picked two genes, *ether-a-go-go* (EAG) and *spineless* (SS), to conduct our evaluation on these two genes.

First, to see whether the non-relevant documents retrieved by the IR module due to ambiguity of gene synonyms could be detected with certain accuracy by document clustering, we manually clustered the documents into several groups,

	gene EAG			gene SS	
	Std 1	Std 2	Std 3	Std2	Std 3
K-means	0.835	0.880	0.411	0.594	0.317
PTC w/ Prior	0.910	1.092	N/A	0.694	N/A
PTC w/o Prior	1.094	1.197	0.790	0.992	0.773
Random	1.228	1.359	0.834	1.261	1.329

Table 1: Clustering results.

so that the matched "gene name" refers to the same thing within each cluster. We got 9 clusters for *SS* and 2 for *EAG* respectively. For instance, besides spineless, the symbol *SS* is frequently used as abbreviation for *single stranded*, *stable strain*, *homozygote SS*, *etc.* We call this clustering standard based on the meaning of the gene symbol "Std 3". This standard is compared with K-means clustering results and probabilistic theme clustering results.

To group documents into clusters based on the types of information they contain, we asked two people with certain biology background to manually classify the retrieved documents of each target gene into 6 predefined categories: Function, Sequence analysis, Regulation and Interaction, Homolog, Behavior Study, and False Positives. Using this human judgment as the standard, we measured the performance of K-means and the probabilistic theme clustering with and without prior theme language models. These standards are referred to as "Std 1" and "Std 2", where "Std 1" is from annotator A and "Std 2" from annotator B.

To measure how much the K-means and probabilistic theme clustering results deviate from the standard clusters, we used the metrics proposed in [1], *i.e.*, class entropy and cluster entropy. Because of the tradeoff between these two metrics, we used the average of class entropy and cluster entropy to measure the overall similarity between the clustering result and the standard. The lower this average value is, the better the clustering result is. The values are shown in Table 1. A method that randomly assigns a document to a cluster is used as the baseline.

k	Gene	w/ Name Constraint						All					
		C1	C2	T1	T2	P	M	C1	C2	T1	T2	P	M
Top-1	EAG	1	0.6	1	0.8	0.6	0.8	0.6	0.4	0.6	0.4	0	0.6
	SS	1	0.8	1	0.6	0.6	0.8	1	0.6	1	0.6	0.6	0.6
Top-2	EAG	0.6	0.6	0.6	0.7	0.7	0.6	0.3	0.4	0.4	0.5	0.3	0.4
	SS	0.9	0.8	0.9	0.7	0.7	0.8	1	0.6	1	0.7	0.7	0.8
Top-3	EAG	0.7	0.7	0.7	0.7	0.7	0.7	0.4	0.5	0.4	0.5	0.2	0.4
	SS	0.9	0.8	0.9	0.7	0.7	0.9	0.9	0.7	0.9	0.6	0.6	0.8

Table 2: Sentence extraction results.

The evaluation of sentence extraction is isolated from document clustering

by extracting sentences from standard clusters of documents. We combined top-k sentences extracted by different methods into a single judgement pool, and judged by a human with domain knowledge. The results are measured by precision at the top-k ranked sentences as showed in Table 2.

7 Conclusion and Future Work

From Table 1, it is quite obvious that all the approaches are effective when compared with the random clustering method. Among them, K-means is more effective than probabilistic theme clustering for this task. And for both genes, probabilistic theme clustering with prior theme language models seems better than that without prior model. Although this experiment is only conducted on two genes, the consistency of the trend gives us much confidence to draw the conclusion that the prior language models built from the documents curated by FlyBase can help cluster the documents into different relevant aspects of a gene, thus help summarize the knowledge of a gene in those aspects. In our future work, we will apply similar prior information to K-means clustering. We expect it would further improve the performance.

We also noticed that probabilistic theme clustering performed worse than K-means, although intuitively we would think that PLSA should outperform K-means. The assumption of probabilistic theme clustering seems closer to the real property of the biological abstracts. By looking into the experiments we have run, it is not hard to see that actually we did not take any advantage of the theme clustering approach at all. By assigning each document to the strongest theme cluster, we did not use the information that the document is generated by a mixture of multiple themes. In each abstract, a variety of knowledge is put together into a short paragraph. When a human is asked to classify those documents, he may not always assign it to the strongest theme. So even between the two standards curated by two people, the average of cluster entropy and class entropy is 0.888 for *EAG*. In the future work, we will explore how to take advantage of the theme information by the probabilistic theme clustering approach instead of clustering the abstracts directly. One possibility is to cluster the sentences according to the theme language models, then we need to put more efforts to figuring out a good way of evaluation.

The results in column "Std 3" showed that K-means is very promising in detecting the non-relevant documents retried. Based on these clustering, we could use some heuristics to automatically filter out false positives. For instance, the false positives are usually retrieved because they contain some synonym of the target gene that is short and frequently used as abbreviation of other names. The content of those documents containing the same symbol are topically very similar, thus easy to be clustered together. Usually the false cluster of documents only includes one synonym of the target gene, while the true cluster of documents would include several of them. However, we still have the risk of eliminating relevant documents by the automated approach, unless a novel technique with guarantee of high accuracy is achieved for this task. Therefore,

in order not to lose relevant information, we did not remove the false positives in our experiments. We believe there is still large room for future work in that direction.

From the evaluation of sentence extraction (Table 2), we can see that the top ranked sentences can represent the document cluster to some extent and all these approaches (see columns *C1, C2, T1, T2, P*) are comparatively good. We also summed up the ranks from the above 5 heuristics to re-rank those sentences (see results in column *M*). No improvement by this heuristic was observed. We noticed that the left half of the table, for which only the sentences containing the target gene name or synonym are considered in ranking, are generally better than the right half, which rank all the sentences for each cluster. This indicates that the sentences containing the target gene name would be more informative than those without the name.

8 Acknowledgement

We would like to thank Qiaozhu Mei for his help with the probabilistic theme clustering method. We would also like to thank Xin He for his help with the human annotation.

References

[1] J. He, A.-H. Tan, C. L. Tan, and S. Y. Sung. On quantitative evaluation of clustering systems. *Clustering and Information Retrieval*, pages 105–134, 2003.

[2] I. Iliopoulos, A. Enright, and C. Ouzounis. Textquest: document clustering of medline abstracts for concept discovery in molecular biology. *Pac Symp Biocomput*, pages 384–395, 2001.

[3] T. R. Leek. Information extraction using hidden Markov models. *Master's thesis, UC San Diego*, 1997.

[4] M. J. Mana-Lopez, M. D. Buenaga, and J. M. Gomez-Hidalgo. Multidocument summarizatioin: An added value to clustering in interactive retrieval. *ACM Transactions on Information Systems*, 22(2):214–241, April 2004.

[5] D. Marcu. Automatic abstracting. *Encyclopedia of Library and Information Science*, pages 245–256, 2003.

[6] H. Shatkay and R. Feldman. Mining the biomedical literature in the genomic era: An overview. *Journal of Computational Biology (JCB)*, 10(6):821–856, December 2003.

[7] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. *KDD Workshop on Text Mining*, 2000.