Generating Gene Summaries from Biomedical Literature: A Study of Semi-Structured Summarization

Xu Ling, Jing Jiang, Xin He, Qiaozhu Mei
Chengxiang Zhai, Bruce Schatz

Department of Computer Science
Institute for Genomic Biology
University of Illinois at Urbana-Champaign, IL 61801
E-mail: {xuling,jiang4,xinhe2,qmei2,czhai,schatz}@uiuc.edu

Abstract

Most knowledge accumulated through scientific discoveries in genomics and related biomedical disciplines is buried in the vast amount of biomedical literature. Since understanding gene regulations is fundamental to biomedical research, summarizing all the existing knowledge about a gene based on literature is highly desirable to help biologists digest the literature. In this paper, we present a study of methods for automatically generating gene summaries from biomedical literature. Unlike most existing work on automatic text summarization, in which the generated summary is often a list of extracted sentences, we propose to generate a semi-structured summary which consists of sentences covering specific semantic aspects of a gene. Such a semi-structured summary is more appropriate for describing genes and poses special challenges for automatic text summarization. We propose a two-stage approach to generate such a summary for a given gene – first retrieving articles about a gene and then extracting sentences for each specified semantic aspect. We address the issue of gene name variation in the first stage and propose several different methods for sentence extraction in the second stage. We evaluate the proposed methods using a test set with 20 genes. Experiment results show that the proposed methods can generate useful semi-structured gene summaries automatically from biomedical literature, and our proposed methods outperform general purpose summarization methods. Among all the proposed methods for sentence extraction, a probabilistic language modeling approach that models gene context performs the best.

Key words: Summarization, Genomics, Probabilistic language model

1 Introduction

Biomedical literature has been playing a central role in the research activities of all biologists. The growing amount of scientific discoveries in genomics and related biomedical disciplines have led to a corresponding growth in the amount of literature information. Because of its daunting size and complexity, there have been increasing efforts devoted to integrate this huge resource for biologists to digest quickly.

Understanding gene functions is fundamental to biomedical research, and one fundamental task that biomedical researchers often have to perform is to find and summarize all the knowledge about a particular gene from the literature, a problem that we call gene summarization.

Because of the importance of genes, there has been much manual effort on constructing an informative summary of a gene based on literature information. For example, FlyBase [1] (R. A. Drysdale and Consortium, 2005) (one of the model organism genome database) provides a text summary for each Drosophila gene, including DNA sequence, functional description, mutant information *etc.*. Compressing and arranging all the knowledge from a huge amount of literature into different aspects enable biologists to quickly understand the target gene.

However, such gene summaries are currently generated by manually extracting information from literature, which is extremely labor-intensive and cannot keep up with the rapid growth of the literature information. As the growing amount of scientific discoveries in genomics and related biomedical disciplines, automatic summarization of gene descriptions in multiple aspects from biomedical literature has become an urgent task.

One characteristic of an informative gene summary is that the summary should ideally consists of sentences that cover several important semantic aspects such as sequence information, mutant phenotype, and gene product. That is, the summary is semi-structured. For example, Figure 1 shows a sample gene summary in FlyBase retrieved in 2005. Here we see that the summary consists of sentences covering the following aspects of a gene: (1) Gene products (*GP*); (2) Expression location (*EL*); (3) Sequence information (*SI*); (4) Wild-type function and phenotypic information (*WFPI*); (5) Mutant phenotype (*MP*); and (6) Genetical interaction (*GI*), as annotated. We thus propose to frame the gene summarization problem as to automatically generate a semi-structured summary consisting of sentences covering these six aspects of a gene. Such a summary not only is itself very useful, but also can serve as useful entry points to the literature through linking each aspect to the supporting evidence in the literature.

[1] http://flybase.bio.indiana.edu/

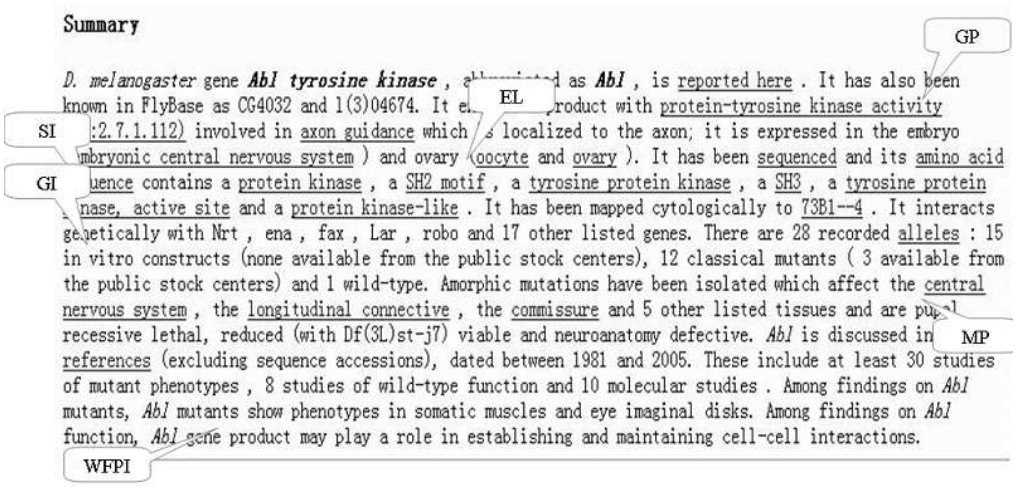

Fig. 1. Example Gene Summary In FlyBase.

Most existing work on automatic text summarization has focused on news summarization and the generated summary is generally unstructured, consisting of a list of sentences. The existing summarization methods are thus inadequate for generating a semi-structured summary. In this paper, we present a study of methods for automatically generating semi-structured gene summaries from biomedical literature. Although our studies mainly focus in the biomedical literature domain, the approaches we proposed are generally applicable to semi-structured summarization in other applications, such as product reviews. Under the assumption that we have some training sentences for each aspect, generalizing our methods for applying to other applications is very straightforward.

We propose a two-stage approach to generate such a summary for a given gene, in which we would first retrieve articles about a gene and then extract sentences for each of six specified semantic aspects. While the first stage can be implemented using any standard information retrieval techniques, a standard IR technique generally cannot handle gene name variations well. We address this issue through adding some heuristic methods on top of regular keyword matching. For the second stage, we leverage some existing training resources and propose several different methods to learn from the training data and extract sentences in each semantic aspect.

We evaluate the proposed methods using a test set with 20 randomly selected genes. Experiment results show that the proposed methods are potentially useful in automatically generating informative semi-structured gene summaries from biomedical literature and outperform general purpose summarization methods. Among our proposed methods, although vector space model generally performs comparably with most probabilistic approaches, the probabilistic model with gene context analysis is the best by most of the evaluation metrics.

With the proposed methods, we would be able to generate summaries such as those in FlyBase automatically. The generated summaries would allow biologists to more

easily keep track of new discoveries recently occurring in the literature. Compared with the currently available GeneRIF [2] used in PubMed, our summaries not only are generated automatically, but also can organize information into aspects. The proposed methods can also be exploited to generate candidate summaries to assist a human expert in curating resources such as FlyBase.

The remainder of this paper is organized as follows. Section 2 discusses the related work. We present the proposed summarization method in Section 3, followed by a detailed discussion of sentence extraction methods in Section 4. The evaluation results are discussed in Section 5. We discuss the generality of the proposed approaches in Section 6 and conclude our study in Section 7.

2 Related Work

To the best of our knowledge, this is the first attempt to automatically generate a structured summary of a gene from biomedical literature. Although, automatic text summarization has been extensively studied before (Luhn, 1958; Kupiec et al., 1995), a distinctive feature of our work is that the generated summary has explicitly defined semantic aspects, whereas most news summaries are simply a list of extracted sentences (Goldstein et al., 1999; Kraaij et al., 2001). In our task, we also consider the special characteristics of the biomedical literature.

Our work is very much related to the recent work on summarizing/clustering search results (Kummamuru et al., 2004), especially work such as Scatter/Gather (Hearst and Pedersen, 1996), Grouper (Zamir and Etzioni, 1999). Clustering web search results was studied in (Zeng et al., 2004), which attempts to organize the search results of each query into clusters labeled with key phrases. Their work tries to discover the latent clusters of ad hoc retrieval results, which does not predefine a structure. Differently, we are generating semi-structured summaries to satisfy people's specific information needs which are well defined (i.e., aspects). Therefore, we have fixed semantic meaning in each dimension and aim at generating a sentence-based summary whereas existing work leaves the definition of each dimension open and relies on clustering algorithms to discover meaningful dimensions.

A problem closely related to ours was addressed in the Genomics Track in the Text REtrieval Conference (TREC) 2003 (Hersh and Bhupatiraju, 2003), where the task was to generate descriptions about genes from MedLine records. The major difference between this task and ours is that the generated descriptions do not organize the information into clearly defined aspects. In contrast, we define six reasonable aspects of genes and propose new methods for selecting sentences for specific aspects.

[2] http://www.ncbi.nlm.nih.gov/projects/GeneRIF/GeneRIFhelp.html

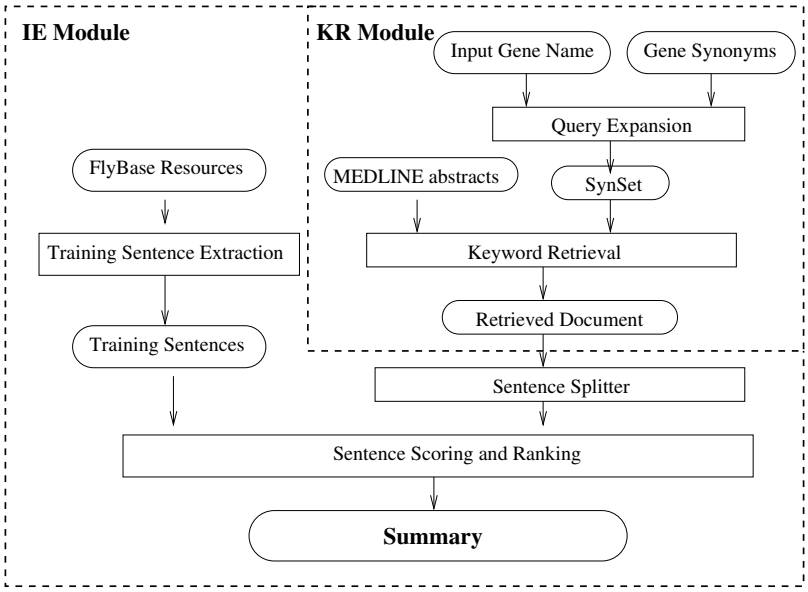

Fig. 2. System Overview.

Most existing studies of biomedical literature mining focus on automated information extraction, using natural language processing techniques to identify relevant phrases and relations in text, such as protein-protein interactions (Iliopoulos et al., 2001) (see (Hirschman et al., 2002; Shatkay and Feldman, 2003) for reviews of these works). The information we extract is at the sentence level, which allows us to cover many different aspects of a gene and extract information in a more robust manner.

Our work was initially published in the Proceedings of Pacific Symposium of Biocomputing 2006 (Ling et al., 2006). This previous work has been significantly extended here with several new probabilistic approaches for sentence extraction and a comprehensive evaluation of all the methods for generating a semi-structured summary.

3 Automatic Gene Summarization

3.1 Overview

The proposed automatic gene summarization system mainly consists of two components: a Keyword Retrieval module that retrieves sentences about a target gene, and an Information Extraction module that extracts retrieved sentences to summarize the target gene. The Information Extraction module itself consists of two components, one for training data generation, and the other for sentence extraction. The whole system is illustrated in Figure 2.

3.2 Keyword Retrieval Module

First, to identify documents that may contain useful information for the target gene, we use a dictionary-based keyword retrieval approach to retrieve all documents containing any synonym of the target gene.

*3.2.1 Gene synonym set (*SynSet*) Construction*

Gene synonyms are very common in biomedical literature. It is important to consider all the synonyms of a target gene when searching for relevant documents about the gene. We used the synonym list for fly genes provided by BioCreAtIvE 2003 Task 1B (Hirschman et al., 2005) and extended it by adding names or functional information of proteins encoded by each gene from FlyBase's annotation. In the end, we constructed a set of synonyms and protein names (called *SynSet* here) for each known Drosophila gene.

Because of variations in gene name spelling, we use a special tokenizer for both MedLine abstracts and *SynSet* entries, to normalize the gene name. The tokenizer converts the input text into a sequence of tokens, where each token is either a sequence of lowercase letters or a sequence of numbers. White spaces and all other symbols are treated as token delimiters. For instance, the different synonyms for gene *cAMP dependent protein kinase 2*, "PKA C2", "Pka C2", and "Pka-C2", are all normalized to the same token sequence "pka c 2" to allow them to match each other. A MedLine abstract is considered as being relevant only if it matches the token sequence of a synonym *exactly*.

3.2.2 Synonym Filtering

Some gene synonyms are ambiguous, for example, the gene name "PKA" is also a chemical term with a different meaning. In these situations, a document containing the synonym with an alternative meaning would be retrieved. Our strategy of alleviating this problem is based on the observations that (1) the longer or full name of a gene is often unambiguous; (2) when a gene's short abbreviation is mentioned in a document, its full or longer name is often present as well. Therefore, we force all retrieved documents to contain at least one synonym of the target gene that is at least 5-character long.

3.3 Information Extraction Module

The information extraction module extracts sentences containing useful factual information about the target gene from the documents returned by the keyword re-

trieval module. To ensure the precision of extraction, we only consider sentences containing the target gene, which are further organized into the six general aspects listed in Table 1, which we believe are important for gene summaries.

Table 1
Aspects for Gene Summary

GP	Gene Product, describing the product (protein, rRNA, *etc.*) of the target gene.
EL	Expression Location, describing where the target gene is mainly expressed.
SI	Sequence Information, describing the sequence information of the target gene and its product.
WFPI	Wild-type Function & Phenotypic Information, describing the wild-type functions and the phenotypic information about the target gene and its product.
MP	Mutant Phenotype, describing the information about the mutant phenotypes of the target gene.
GI	Genetical Interaction, describing the genetical interactions of the target gene with other molecules.

Our main idea for sentence extraction is to leverage the existing training resources (such as FlyBase) to learn a model for each semantic aspect and use such models to categorize the top ranked sentences into appropriate semantic aspects.

3.3.1 Training Data Generation

To help identify informative sentences related to each aspect, we construct a training data set consisting of "typical" sentences for describing each of the six aspects using three resources: the *Summary* pages, the *Attributed data* pages, and the *references* of each gene in FlyBase.

The "Summary" Paragraph: FlyBase curators have compressed all the relevant information about a gene into a short paragraph, the text *Summary* in the FlyBase report. This paragraph contains good example sentences for each aspect of a gene. A typical paragraph contains information related to gene product, sequence information, genetical interaction, *etc.* More importantly, verbs such as "encode", "sequence" and "interact" in the text are very indicative of which aspect the sentence is related to. Based on the regular structure of these text summaries, we decompose each paragraph into our six aspects with non-relevant sentences discarded.

However, since these sentences are automatically generated by filling the information in FlyBase databases into a common template, they are not good examples of typical sentences that appear in real literature. For instance, genetical interaction can be described in many different ways using verbs such as "regulate", "inhibit", "promote" and "enhance". In the "summary" paragraph, it is always described us-

ing the template "It interacts genetically with ...". Thus we also want to obtain good examples of original sentences from the literature.

The "Attributed Data" Report: One resource of original sentences is the "attributed data" report for each Drosophila gene provided by FlyBase. For some attributes such as "molecular data", "phenotypic info." and"wild-type function", the original sentences from literature are listed. These sentences seem to be good complements of the training data from the "summary" paragraph. In our system, we collect the sentences from "phenotypic info." and "wild-type function" as training sentences for the aspect *WFPI*.

The References: For aspects such as "gene product" and "interacts genetically with", the "attributed data" reports only list the noun phrases related to the target gene, but do not show any complete sentences. In order to find the patterns of sentences containing such information, we exploit the links to the corresponding references given in the "attributed data" reports to find the PubMed ID of the reference. We then look for occurrences of the item, *i.e.,* a protein name in "gene product" or another gene name "interacts genetically with", in the abstract of the reference. We add the sentence containing both the item and the target gene to our training data. Inclusion of these sentences is useful because verbs such as "enhance" and "suppress" now appear in the training data.

3.3.2 Sentence Extraction

Our general idea for sentence extraction is the following: We first exploit any proposed method (see Section 4) to compute the relevance score S for each sentence-aspect pair. To ensure reliable association between sentences and aspects, for each sentence, we rank all the aspects based on S and keep only the top two aspects. The rationale behind it is that a sentence may contain more than one aspect of a gene. For instance, a sentence describing the mutant phenotype of a gene may have information about the molecular function of this gene. Therefore we empirically only consider the two most dominant aspects of information in a sentence.

To generate a structured, aspect-based summary, for each aspect, we rank all the kept sentences according to S and pick the top-k sentences. Such an aspect-based summary is similar to the "attributed data" report in FlyBase.

It is clear that among all the components in the proposed gene summarization method, the main challenge is sentence extraction. More specifically, the challenge is to design an appropriate scoring function for scoring a sentence w.r.t. a semantic dimension. In the following section, we present several different methods for solving this problem.

4 Sentence Extraction Methods

As discussed in the previous section, our general idea for sentence extraction is to first compute aspect models based on training data, then compute the relevance score of each sentence with respect to each aspect, and finally extract sentences for each aspect of the target gene. We present different methods for modeling term usages based on the training data and scoring sentences for each semantic aspect: vector space model and probabilistic language model. We now discuss each in detail.

4.1 Vector Space Model

We can use the vector space model and cosine similarity function from information retrieval to assign a relevance score to each sentence *w.r.t.* each aspect. Specifically, For each aspect, we construct a corresponding term vector V_c using the training sentences for the aspect. Following a commonly used information retrieval heuristic, we define the weight of a term t_i in the aspect term vector for aspect j as $w_{i,j} = \mathrm{TF}_{i,j} * \mathrm{IDF}_i$, where $\mathrm{TF}_{i,j}$ is the term frequency, *i.e.*, the number of times term t_i occurs in all the training sentences of aspect j, and IDF_i is the inverse document frequency. IDF_i is computed as $\mathrm{IDF}_i = 1 + \log \frac{N}{n_i}$, where N is the total number of documents in our document collection, and n_i is the number of documents containing term t_i. Intuitively, V_c reflects the usage of different words in sentences describing the corresponding aspect.

Similarly, for each sentence we can construct a sentence term vector V_s, with the same IDF and the TF being the number of times a term occurs in the sentence. The aspect relevance score is then the cosine of the angle between the aspect term vector and the sentence term vector: $S = \cos(V_c, V_s)$.

4.2 Probabilistic Methods: Language Modeling Approaches

Alternatively, we may also use language models to score a sentence for each aspect. Specifically, different language modeling approaches can be used to estimate a language model for each aspect. Then to compute the relevance score S between each sentence-aspect pair, we can use the negative KL-divergence function to measure the similarity between the retrieved sentence and the aspect language model.

$$S = -D(\theta_s||\theta_m) = \sum_w p(w|\theta_s) \log p(w|\theta_m) - \sum_w p(w|\theta_s) \log p(w|\theta_s),$$

where θ_s, θ_m represents the language model of the sentence and the aspect respectively.

Note that using KL-divergence for scoring is equivalent to using the Naive Bayes classifier to classify a sentence into a semantic aspect as the entropy term in the KL-divergence formula does not affect ranking of aspects whereas the cross entropy term is equivalent to the log-likelihood of the sentence given a model.

$p(w|\theta_s)$ can be computed using relative frequency of words in sentence s. The main challenge is thus how to compute $p(w|\theta_m)$. One simple method for estimating the aspect language model could be based on the relative word frequency in the training data of each aspect smoothed by the word frequency of the whole collection. That is, we can simply compute the aspect language model by $p(w|\theta_i) = \lambda_B p(w|\theta_B) + (1 - \lambda_B)\frac{c(w,C_i)}{|C_i|}$, where $p(w|\theta_B) = \frac{1+c(w,\mathcal{C})}{|\mathcal{C}|+|V|}$), $c(w, C_i)$ is the count of word w in the training documents C_i of aspect i, and V is the joint vocabulary of the training collection and test collection. We denote this method as the baseline language model method (***baseLM***).

The baseline language model method can be further improved in several ways. First, common English words and those domain/gene-specific words in the training data are both generally noise when we try to extract the language models of specific information. For example, in our task, the words like "gene, protein, biology, experiment" are not informative words for any aspect. Clearly, the above simple language model estimation can not filter out this kind of noise. Hence, we propose two more sophisticated generative probabilistic mixture models to remove the noise and expect to extract more general and robust language models for modeling the word distribution of each aspect. The first approach is a variant of the cross-collection mixture model (CTM) proposed in (Zhai et al., 2004), which intends to extract the discriminative aspect models by taking into consideration the hidden background model of the whole collection. The second one is a variant of the contextual probabilistic latent semantic analysis (CPLSA) model proposed in (Mei and Zhai, 2006), which attempts to further factor out training-gene-specific language models (i.e., noise) embedded in the training sentences.

As mixture models, both approaches explicitly distinguish a common background model that characterizes common information over a collection of documents from special topic models that characterize topic-specific information in the text. They also distinguish between different topic models that characterize different information in different context. These approaches involve common background model as well as multiple topic-specific models. The underlying basic idea is to treat the words as observations from a mixture model where the component models are the topic-specific word distributions and the background word distributions across different document collections. The Expectation Maximization (EM) algorithm is used to estimate the topic-specific models which people are mostly interested in. In our task, we aim to apply these mixture model methods to obtain the aspect-specific word distributions $p(w|\theta_m)$. The EM algorithm will terminate when it achieves a local maximum of the data likelihood.

In our experiments, we use multiple trials to improve the local maximum we obtain. In the following, we discuss each approach in detail, and give corresponding EM updating formulas.

4.2.1 Discriminative Aspect Model

In order to remove from the estimated aspect models the noise of the background words over the whole collection of the training data, thus also make the estimated models more discriminative, we adopt the cross-collection mixture model (CTM) proposed in (Zhai et al., 2004). Figure. 3 illustrates the idea of explicitly distinguishing common background model that characterizes common words across all aspects from special aspect models that characterize aspect-specific information. By filtering the background model from the aspect models, we expect to extract more discriminative language models that can effectively differentiate word usages of different aspects.

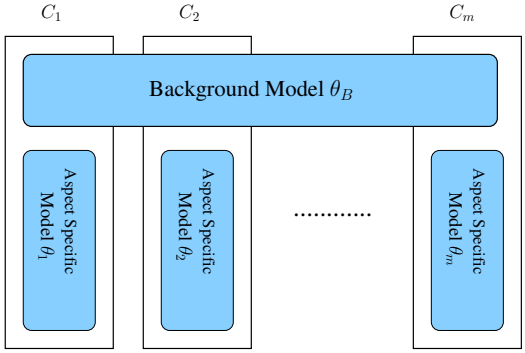

Fig. 3. Discriminative Aspect Model

The training data for each aspect i is collected as sub-collection C_i. Specifically, let $\mathcal{C} = \{C_1, C_2, \cdots, C_m\}$ be m sub-collections for m aspects, respectively. Let $\theta_1, \cdots, \theta_m$ be m aspect unigram language models (*i.e.*, word distributions) and θ_B be a background model for the whole collection $\mathcal{C}$. A document d is regarded as a sample of the following mixture model.

$$p_d(w) = \lambda_B p(w|\theta_B) + (1 - \lambda_B)p(w|\theta_i),$$

where w is a word, $d \in C_i$, λ_B is the mixing weight of the background model θ_B. The log-likelihood of the entire collection $\mathcal{C}$ is

$$\log p(\mathcal{C}) = \sum_{i=1}^{m} \sum_{d \in C_i} \sum_{w \in V} \{c(w, d) \log[\lambda_B p(w|\theta_B) + (1 - \lambda_B)p(w|\theta_i)]\}$$

where $c(w, d)$ is the count of word w in document d.

According to the EM algorithm, the following iterative updating formulas are used to estimate all the parameters. $\{z_{d,w}\}$ is a hidden variable and $p(z_{d,w} = i)$ indicates that the word w in document d is generated by the aspect model i.

$$p(z_{d,w} = B) = \frac{\lambda_B p^{(n)}(w|\theta_B)}{\lambda_B p^{(n)}(w|\theta_B) + (1-\lambda_B)p^{(n)}(w|\theta_i)}$$

$$p(z_{d,w} = i) = \frac{(1-\lambda_B)p^{(n)}(w|\theta_i)}{\lambda_B p^{(n)}(w|\theta_B) + (1-\lambda_B)p^{(n)}(w|\theta_i)}$$

$$p^{(n+1)}(w|\theta_B) = \frac{\sum_{i=1}^{m}\sum_{d\in C_i} c(w,d)p(z_{d,w} = B)}{\sum_{w'\in V}\sum_{i=1}^{m}\sum_{d\in C_i} c(w',d)p(z_{d,w'} = B)}$$

$$p^{(n+1)}(w|\theta_i) = \frac{\sum_{d\in C_i} c(w,d)(1-p(z_{d,w} = B))p(z_{d,w} = i)}{\sum_{w'\in V}\sum_{d\in C_i} c(w',d)(1-p(z_{d,w'} = B))p(z_{d,w'} = i)}$$

The estimated θ_i can then be assumed to be our semantic aspect models for scoring retrieved sentences.

4.2.2 Generalizable Aspect Model

In the above models, we only take into consideration the word usage for each aspect. However, as the training sentences are prepared from a sampled set of genes, they also contain gene-specific information which we want to avoid when summarizing genes that are different from this training set. For example, sentences about a gene which encodes transcription factor usually contain many terms about transcription, like "DNA binding, regulation, transcriptional". The desirable language model extracted to represent each aspect should be general and gene-independent, thus can be used to summarize any new gene.

However, in previous methods, there is no mechanism to filter out this gene-specific information from the aspect language model. Recall that each training sentence has two features: it is about a gene and it is associated with an aspect. In previous models, we only considered the association between each training sentence and its relevant aspect, but ignored the information that each training sentence is originally from a specific gene. The specific semantic aspect and the specific gene can both be regarded as indicating a "context" to which the sentence belongs. In this sense, the models presented in the previous subsection can consider only one context information (*i.e.*, aspect context), but are not applicable to model multiple types of context information (*i.e.*, aspects and genes). We thus adopt the contextual probabilistic model (CPLSA) proposed in (Mei and Zhai, 2006) to address both the aspect context and the gene context when estimating the aspect models. In effect, this is to further filter out the gene-specific word distributions from the aspect models. We thus expect to see it achieves better performance than the methods presented in the previous subsection.

Specifically, by modeling a training sentence in the context of all sentences about the same aspect, we can extract aspect language models. On the other hand, by modeling sentences in the context of all sentences about the same gene, we can

model the gene-specific information and distinguish it from the aspect language models. This model intends to make the aspect models more general and applicable to arbitrary genes.

In our task, each document is associated with two types of contexts: the gene g which it describes and the aspect k which it is associated with. From the view of each different context (*i.e.*, aspects and genes), the corresponding language model represents the word distribution of this context. Our goal is to extract the language models associated with the aspect contexts but not the gene contexts. The details of contextual theme analysis and the original CPLSA model can be found in (Mei and Zhai, 2006).

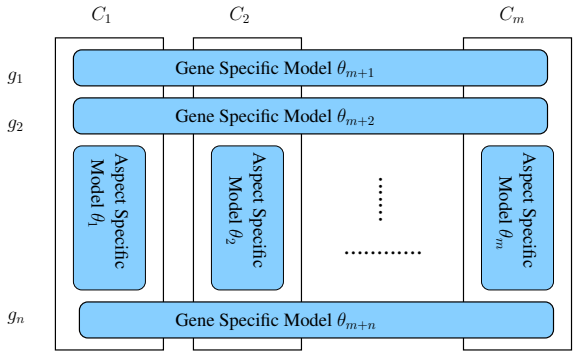

Fig. 4. Generalizable Aspect Model Extraction by Contextual Analysis

Figure. 4 illustrates the idea for modeling two types of context information, *i.e.*, aspects and genes. In this model, we assume that document d (with context features C_d) is generated by generating each word in it as follows: (1) Choose a view v_i from a context according to the view distribution $p(v_i|d, C_d)$; (2) Generate a word from the language model θ_i of the view v_i. That is, $p_d(w) = \sum_{i=1}^{m+n} p(v_i|d, C_d)p(w|\theta_i)$, where θ_i $(i = 1, 2, \ldots, m)$ represents the m aspect models we are interested to extract, and θ_i $(i = m+1, m+2, \ldots, m+n)$ represents the n gene-specific language models we want to filter out. The log-likelihood of the whole collection is

$$\log p(\mathcal{C}) = \sum_{d \in \mathcal{C}} \sum_{w \in V} c(w, d) \log \sum_{i=1}^{m+n} p(v_i|d, C_d)p(w|\theta_i)$$

The parameters are the view selection probability $p(v_i|d, C_d)$, the theme distribution $p(w|\theta_i)$.

The mixture model can be fit to a contextualized collection $\mathcal{C}$ using a maximum likelihood estimator. The EM algorithm can be used to estimate the parameters by the following updating formulas, where $\{z_{w,i,d}\}$ is a hidden variable and $p(z_{w,i,d} = 1)$ indicates that the word w in document d is generated by model θ_i.

$$p(z_{w,i,d}=1) = \frac{p^{(n)}(v_i|d,C_d)p^{(n)}(w|\theta_i)}{\sum_{i'=1}^{m+n} p^{(n)}(v_{i'}|d,C_d)p^{(n)}(w|\theta_{i'})}$$

$$p^{(n+1)}(v_i|d,C_d) = \frac{\sum_{w\in V} c(w,d)p(z_{w,i,d}=1)}{\sum_{i'=1}^{m+n}\sum_{w\in V} c(w,d)p(z_{w,i',d}=1)}$$

$$p^{(n+1)}(w|\theta_i) = \frac{\sum_{d\in\mathcal{C}} c(w,d)p(z_{w,i,d}=1)}{\sum_{w'\in V}\sum_{d\in\mathcal{C}} c(w',d)p(z_{w',i,d}=1)}$$

The estimated θ_i ($i = 1, 2, \ldots, m$) can then be assumed to be our semantic aspect models for scoring retrieved sentences.

5 Evaluation Experiments

5.1 Experiment Setup

We retrieve 22092 MedLine abstracts as our document collection using the keyword "Drosophila". We use the Lemur Toolkit [3] to implement the summarizers we proposed. As explained in Section 3.3.1, we use the subcollection which consists of 20% genes in FlyBase for model training purpose. It contains 7391 sentences in total.

The first stage of keyword retrieval is very straight forward, and can be implemented using standard information retrieval techniques. To handle gene name variations and improve the accuracy of retrieval, we apply a heuristic method on top of regular keyword matching. Methods developed for this stage are intuitive and standard as a general retrieval task on genomic literature, thus we would not present a formal evaluation here. Instead, we focus the evaluation on the second stage, *i.e.*, sentence extraction, which is more challenging. In our experiments, we compare the performance of different summarizers, by means of extracting sentences from the same set of sentences retrieved for each target gene.

5.1.1 Gold Standard

Since the semi-structured literature summarization is a novel problem, there is no existing gold standard. It is very difficult to create a large judgment set manually. In our experiments, we randomly select 20 genes from FlyBase for evaluation. For each gene, we ask two experts to assign each candidate sentence to at most two most relevant aspects separately. Then the sentences that are decided as relevant to

[3] http://www.lemurproject.org/, a C++ toolkit supporting a variety of information retrieval functions.

a certain aspect are collected as the judgment for this gene in this corresponding aspect. There is no constraint on the length of the gold standard summary. These two sets of 20 multiple aspect based summaries are used as the gold standard for our experiments, based on which all summaries generated by the different approaches are evaluated.

5.1.2 Evaluation Metrics

ROUGE is an evaluation package suggested by DUC [4] and commonly used (Sun et al., 2005) to automatically evaluate both single-document summarization and multi-document summarization systems (Lin and Hovy, 2003; Lin, 2004). It provides a suite of evaluation metrics to measure the similarity between system generated summaries and the judgments in several ways, such as n-gram overlapping, longest common subsequence and skip-bigram co-occurrence, instead of simple matching/non-matching. This is especially desirable for gene summarization because the sentences retrieved for a gene are from multiple papers, thus there are usually multiple sentences which are very similar to each other in terms of covering some aspects of the gene. All these sentences are reasonable to be selected for a summary. Among all the evaluation metrics in ROUGE, ROUGE-N(models n-gram co-occurrence, N = 1, 2, 3) and ROUGE-W-1.2 generally perform well in evaluating single-document summarization according to (Lin and Hovy, 2003; Lin, 2004). We evaluate our system with all the metrics provided by ROUGE, and report ROUGE-1, ROUGE-2, ROUGE-3 and ROUGE-W-1.2.

5.1.3 The Baselines

Since there is no existing system for semi-structured summarization, we select several general purpose summarization systems as the baselines to be compared with our methods. MEAD [5] is a well known, publicly available summarization toolkit which accommodates both extractive multi-document summarization and single-document summarization (Radev et al., 2003). Three baseline summarizers are provided by MEAD: *random* extraction - the summarizer extracting sentences randomly, *lead-based* extraction - the summarizer extracting sentences from the front of each document, and *Mead-Single* - a featured single document summarization system which integrates text features such as keywords, sentence length, sentence position, and cluster centroids. We use the default setting of the MEAD toolkit. In all these three systems, we pool all the sentences retrieved by the Retrieval Module for each gene as a single document. These general purpose summarization methods are reasonable baselines to evaluate our system, and denoted as ***RAND***, ***LEAD***, ***MEAD*** respectively in our experiments.

[4] http://duc.nist.gov/

[5] http://www.summarization.com/mead/

In addition to the three runs for the above baselines, we run four experiments to evaluate our proposed methods in sentence extraction as follows:

- Run 1(***VSM***): use the vector space model proposed in Section 4.1.
- Run 2(***baseLM***): use the simple language modeling approach presented in Section 4.
- Run 3(***DAM***): use the discriminative aspect model proposed in Section 4.2.1.
- Run 4(***GAM***): use the generalizable aspect model proposed in Section 4.2.2.

5.2 Comparison of Sentence Extraction Methods

Using ROUGE, We evaluate the result summaries on six aspects and two gold standards separately, then take the average score over the six aspects and two standards as the final evaluation. Table 3 summarizes the averaged *Average_R* score for all evaluated methods on sentence extraction. We vary the length of summaries for each aspect among 1, 5, 10 and 15 sentences. Note that as we did not finely tune the parameters for the four proposed methods, the results for them are not necessarily their optimal results. The default selection of parameters are presented in Table 2 unless otherwise specified.

Table 2
Default Selection of Parameter Values

Method	baseLM	DAM	GAM
Parameter Value	$\lambda_B = 0.1$	$\lambda_B = 0.8$	$\lambda_B = 0.3$

From Table 3, we make a number of observations as the following:

- *Comparison among baseline methods*:
  Among the three baseline methods, ***MEAD*** performs significantly better than ***RAND*** and ***LEAD***. This is not surprising, since ***MEAD*** is a featured document summarization system which integrates many text features, whereas the other two simply extract sentences randomly or from the front of each document. The advantages of ***MEAD*** over the other two methods decreases when the summary grows longer. This is because the evaluation on very long summaries will be affected by the unavoidably redundancy within the summary, thus the evaluation might not reliably reflect the effectiveness of the methods.
- *Baseline methods vs. proposed methods*:
  By most metrics, our proposed methods perform better than the baseline methods. We also observe that, the evaluation result of metric ROUGE-1 is inconsistent with the others especially when the summary length increases (*i.e.*, $>= 10$ sentences). Only in this case, the best baseline approach ***MEAD*** achieves better score than our proposed methods. In this case, the other two baseline methods (***RAND***, ***LEAD***) also show exceptionally high scores. This is because ROUGE-1 only measures the coverage of unigrams. In the standard summary, the common

background words, like "gene, protein, Drosophila", occur very frequently, most of which appear alone (i.e., not associated in an n-gram phrase). When the generated summary length increases, the matching of these common background words will affect the evaluation scores significantly. However, in terms of other metrics like ROUGE-2, ROUGE-3, the common background words are less effective because they measure bigram/trigram co-occurrences. Under these metrics, ***MEAD*** could not achieve higher score than our proposed methods. We also observe consistent result by ROUGE-4. The outperforming of language models and the vector space model over baseline approaches indicate that our proposed methods are useful for automatically generating semi-structured gene summaries from biomedical literature, while the general purpose summarizers do not serve this purpose very well.

Generally speaking, a reasonably short summary (e.g., 5 sentences for each aspect of a gene) is enough to capture the key information of a gene and is preferred. Long summaries usually carry redundant information. By investigating the scores over different summary length, we also noticed that the advantage of our proposed methods over baseline methods are most significant when the summary is short. The main reason is that, unlike MEAD, we do not consider any redundancy removal in sentence selection. The longer the summary is, the more the result will be affected by the redundancy between the sentences picked by our methods. It also suggests a direction for our future work to develop techniques on removing redundancy in generating a semi-structured summary.

- *Probabilistic language model vs. vector space model*:
  Vector space model performs slightly better than language models only when the generated summary is very short (*i.e.*, one-sentence summary). In fact, this is not surprising because we used a simple implementation of vector space model without document length normalization. It favors longer sentences. Indeed, we observed that the sentences picked by this model are longer than those picked by the others. Thus when the number of sentence is small, it has a higher chance to cover more words in the gold standard summary, especially unigrams, which makes it achieve slightly higher ROUGE-1 score since the reported Average-R score favors high recall. However, when investigating N-grams instead of single terms, long sentences do not necessarily have high recall. That is why when ROUGE-2, 3 are used, ***VSM*** is not the best. Also when the result summary length increases, this outperforming disappears and vector space model is beaten by language models with larger margin.
- *Baseline language model vs. discriminative aspect model*:
  The discriminative aspect model (***DAM***) only performs better than the simple baseline language model (***baseLM***) when the length of the summary is 5 sentences. These two methods seem quite comparable. From this evaluation, it is unclear to see which one applies better in our task. Further studies in comparing more elaborate vector space models and optimized discriminative aspect model on this summarization task may provide more insight.
- *Generalizable aspect model vs. other probabilistic language models*:
  The generalizable aspect model (***GAM***) is the most sophisticated method among

the three probabilistic language model approaches and the only one which takes into consideration the gene labels of training sentences. With all evaluation metrics, it performs the best among the proposed language models as expected. ***GAM*** also outperforms all the other summarization methods with most evaluation metrics. The only exceptions are that it has lower ROUGE-1 score (1) than the vector space model when generating single-sentence summaries; (2) than the baseline ***MEAD*** when generating 10- and 15-sentence summaries. The reasons are already discussed above. According to the evaluation, the generalizable aspect model based on contextual probabilistic analysis appears to be a very promising approach for our semi-structured gene summarization task.

Table 3
Evaluation of Sentence Extraction Methods

Len	Metric	RAND	LEAD	MEAD	VSM	baseLM	DAM	GAM
1	ROUGE-1	0.0746	0.0604	0.1052	**0.1398**	0.1163	0.1082	0.1282
1	ROUGE-2	0.0235	0.0199	0.0393	0.0967	0.0911	0.0766	**0.1017**
1	ROUGE-3	0.0144	0.0114	0.0257	0.0880	0.0844	0.0687	**0.0947**
1	ROUGE-W-1.2	0.0232	0.0210	0.0339	0.0512	0.0467	0.0411	**0.0513**
5	ROUGE-1	0.2824	0.2426	0.3628	0.3668	0.3471	0.3588	**0.3676**
5	ROUGE-2	0.1222	0.1014	0.1729	0.2745	0.2584	0.2698	**0.2817**
5	ROUGE-3	0.0864	0.0737	0.1288	0.2498	0.2335	0.2456	**0.2581**
5	ROUGE-W-1.2	0.0859	0.0788	0.1101	0.1266	0.1281	0.1298	**0.1344**
10	ROUGE-1	0.4570	0.4173	**0.5212**	0.4731	0.4831	0.4905	0.4969
10	ROUGE-2	0.2464	0.2151	0.2950	0.3584	0.3671	0.3660	**0.3788**
10	ROUGE-3	0.1917	0.1706	0.2363	0.3239	0.3315	0.3288	**0.3438**
10	ROUGE-W-1.2	0.1414	0.1368	0.1633	0.1588	0.1721	0.1709	**0.1746**
15	ROUGE-1	0.5510	0.5228	**0.6227**	0.5292	0.5696	0.5711	0.5746
15	ROUGE-2	0.3122	0.2962	0.3865	0.4055	0.4378	0.4299	**0.4424**
15	ROUGE-3	0.2411	0.2409	0.3178	0.3671	0.3952	0.3861	**0.4009**
15	ROUGE-W-1.2	0.1705	0.1714	0.1973	0.1746	0.1983	0.1953	**0.2005**

In Table 4, we show a sample structured summary generated for the well-studied gene *Abl* by collecting the best sentence ranked by the vector space model for each aspect. In this summary, all the extracted sentences are quite informative as judged by biologists. For comparison, the human-generated FlyBase summary of the same gene is in Figure 1.

Table 4
Text summary of gene *Abl* by our approach

GP	The Drosophila melanogaster abl and the murine v-abl genes encode tyrosine protein kinases (TPKs) whose amino acid sequences are highly conserved.
EL	In later larval and pupal stages, abl protein levels are also highest in differentiating muscle and neural tissue including the photoreceptor cells of the eye. abl protein is localized subcellularly to the axons of the central nervous system, the embryonic somatic muscle attachment sites and the apical cell junctions of the imaginal disk epithelium.
SI	The DNA sequence encodes a protein of 1520 amino acids with sequence homology to the human c-abl proto-oncogene product, beginning at the amino terminus and extending 656 amino acids through the region essential for tyrosine kinase activity.
MP	The mutations are recessive embryonic lethal mutations but act as dominant mutations to compensate for the neural defects of abl mutants.
GI	Mutations in the Abelson tyrosine kinase gene show dominant interactions with fasII mutations, suggesting that Abl and Fas II function in a signaling pathway that controls proneural gene expression.
WFPI	We have examined the expression of the abl protein throughout embryonic and pupal development and analyzed mutant phenotypes in some of the tissues expressing abl. abl protein, present in all cells of the early embryo as the product of maternally contributed mRNA, transiently localizes to the region below the plasma membrane cleavage furrows as cellularization initiates.

6 Discussion

Although our study is focused on a specific semi-structured text summarization problem (i.e., summarizing gene information), the problem setup and the proposed methods are general and applicable to other instances of the problem.

Gene summarization is one of the many cases where an ideal summary should have some structure and the summary sentences should be grouped according to this structure. For example, a very common information need is to find opinions about products from the Web. A summary with positive opinions separated from negative opinions would be much more useful than one with the opinions mixed. Also, the products in the same family (e.g., all cameras) tend to share some common subtopic dimensions (e.g., battery, lens, resolution), and an informative summary about products should ideally separate sentences into these different dimensions.

In general, the search results for any broad topic will likely contain results that can be grouped into different subtopics; correspondingly, a semi-structured summary

with summary sentences selected for each subtopic would often be desirable.

As a general setup, our problem definition involves the following elements: (1) A set of documents to be summarized. (2) A set of aspects to define the structure of the summary. (3) Training sentences for each aspect. Clearly, this setup can be applied to other instances of the semi-structured summarization problem provided that we can collect training sentences for each aspect. Indeed, such a problem setup is most useful for domains where the definition of aspects is natural (e.g., in product summarization) and we may easily obtain training data. In case we do not have training data, it is also possible to create some training data as long as meaningful aspects can be predefined in the domain.

Given that a problem fits to this general setup, all our proposed methods, including the general two-stage strategy and specific sentence extraction methods for the second stage, can be applied. Although the problem can also be cast as a sentence classification problem on top of a "flat summarization" problem, this strategy would not allow us to focus on the desired aspects and the summary can easily be biased toward any dominating aspect. In our approach, we ensure the coverage of each aspect through decomposing the summarization task into aspect-specific summarization tasks.

Naturally, for any specific problem, various kinds of heuristics can be exploited to further improve performance for both stages. For example, we have exploited gene synonym resources to improve the retrieval performance in the first stage for gene summarization.

7 Conclusion and Future Work

In this paper, we studied a novel problem in biomedical text mining: automatic generation of semi-structured gene summaries. We developed a system which employed information retrieval and information extraction techniques to automatically summarize information about genes from PubMed abstracts. We proposed several representative methods for solving the problem and used our system to investigate which computational strategies would be most suitable for our problem. The methods were tested on 20 randomly selected genes, and evaluated by ROUGE.

The results show that the two generic methods proposed, *i.e.*, vector space model and probabilistic models, are both very effective for sentence extraction. The fact that our proposed methods outperform the baseline general-purpose summarizers indicates that the general-purpose summarizers are not very effective for gene summarization, and considering the special characteristics of the gene summarization problem (i.e., different semantic aspects) is important for improving summarization performance. Specifically, the generalizable aspect model (***GAM***) based on contex-

tual probabilistic analysis performs the best in most cases, and appears as a very promising approach for this task.

We realized that one obvious limitation of our approach was its dependence on the high-quality data in FlyBase. To address this issue, we will in the future incorporate more training data from databases of other model organisms and resources such as GeneRIF in Entrez Gene. We believe the mixture of data from different resources will reduce the domain bias and help build a general tool for gene summarization. Also, the six structured aspects defined in this work are not the only possible ones; the proposed methods can easily generate new aspects given corresponding training sets.

We have not considered the removal of redundant information in generated summaries. To further improve our system, we will develop techniques to integrate other text features for redundancy removal.

The general problem of generating semi-structured summaries represents a new research direction in text summarization. Although our approaches are proposed for gene summarization, they are general and can also be applied to other semi-structured summarization problems, such as automated summarization of product reviews or blog articles in multiple aspects. Further improving our approaches and developing new general methods for semi-structured summarization are all very interesting future research directions.

8 Acknowledgments

The National Science Foundation (NSF) supported this research through Award 0425852 in the Frontiers in Integrative Biological Research (FIBR) program, for BeeSpace - An Interactive Environment for Analyzing Nature and Nurture in Societal Roles (http://www.beespace.uiuc.edu). The work is also supported in part by an NSF ITR Grant 0428472. We thank Todd Littell for his help to integrate this work into the BeeSpace system and many helpful discussions.

References

Goldstein, J., Kantrowitz, M., Mittal, V., Carbonell, J., 1999. Summarizing text documents: sentence selection and evaluation metrics. In: SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 121–128.

Hearst, M. A., Pedersen, J. O., 1996. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In: Proceedings of SIGIR 1996. pp. 76–84.

Hersh, W., Bhupatiraju, R. T., 2003. Trec genomics track overview. In: Proceeding of Text Retrieval Conference (TREC). pp. 14–23.

Hirschman, L., Colosimo, M., Morgan, A., Yeh, A., 2005. Overview of biocreative task 1b: Normailized gene lists. BMC Bioinformatics 6, S11.

Hirschman, L., Park, J. C., Tsujii, J., Wong, L., Wu, C. H., 2002. Accomplishments and challenges in literature data mining for biology. Bioinformatics 18, 1553–1561.

Iliopoulos, I., Enright, A., Ouzounis, C., 2001. Textquest: Document clustering of medline abstracts for concept discovery in molecular biology. In: Proceedings of PSB 2001. pp. 384–395.

Kraaij, W., Spitters, M., van der Heijden, M., 2001. Combining a mixture language model and naive bayes for multi-document summarisation. In: Proceedings of the DUC2001 workshop, New Orleans.

Kummamuru, K., Lotlikar, R., Roy, S., Singal, K., Krishnapuram, R., 2004. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In: Proceedings of WWW 2004. pp. 658–665.

Kupiec, J., Pedersen, J., Chen, F., 1995. A trainable document summarizer. In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 68–73.

Lin, C., 2004. Rouge: a package for automatic evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004). pp. 74–81.

Lin, C.-Y., Hovy, E., 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. pp. 71–78.

Ling, X., Jiang, J., He, X., Mei, Q., Zhai, C., Schatz, B., 2006. Automatically generating gene summaries from biomedical literature. In: Proceedings of Pacific Symposium of Biocomputing (PSB) 2006. pp. 41–50.

Luhn, H. P., 1958. The automatic creation of literature abstracts. IBM Journal of Research and Development 2 (2), 159–165.

Mei, Q., Zhai, C., 2006. A mixture model for contextual text mining. In: Proceedings of the 2006 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (KDD'06). pp. 649–655.

R. A. Drysdale, M. A. C., Consortium, T. F., 2005. Flybase: genes and gene models. Nucleic Acids Res. 33, 390–395.

Radev, D. R., Teufel, S., Saggion, H., Lam, W., Blitzer, J., Qi, H., Celebi, A., Liu, D., Drabek, E., 2003. Evaluation challenges in large-scale document summarization: the mead project. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. pp. 375–382.

Shatkay, H., Feldman, R., 2003. Mining the biomedical literature in the genomic era: An overview. Journal of Cell Biology 10, 821–856.

Sun, J., Shen, D., Zeng, H., Yang, Q., Lu, Y., Cheni, Z., 2005. Web-page summarization using clickthrough data. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information

Retrieval, (SIGIR'05). pp. 194–201.
Zamir, O., Etzioni, O., 1999. Grouper: A dynamic clustering interface to web search results. In: Proceeding of WWW 1999.
Zeng, H., He, Q., Chen, Z., Ma, W., Ma, J., 2004. Learning to cluster web search results. In: Proceeding of SIGIR 2004.
Zhai, C., Velivelli, A., Yuk, B., 2004. A cross-collection mixture model for comparative text mining. In: Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (KDD'04). pp. 743–748.