Unsupervised Information Extraction for Finding Gene Functions

Ehsan Emadzadeh¹, Azadeh Nikfarjam¹, Rachel E. Ginn¹ and Dr. Graciela Gonzalez¹

¹Department of Biomedical Informatics - Arizona State University

Abstract
Finding gene functions discussed in a literature is imperative to information extraction from biomedical documents. Automated, computational methodologies can reduce the need for manual curation significantly and improve quality of other related Information Extraction (IE) systems. We propose an open information extraction method for BioCreative IV GO shared task (Subtask b)—a workshop designed to find gene function terms (GO terms) for different genes in an article. The proposed open IE approach is based on distributional semantic similarity over the gene ontology terms. The method does not require the annotated data for training, which makes it highly generalizable. We achieve the f-measure of 0.26 for test-set in the official submission for BioCreative-GO shared task.

Introduction
Text mining biomedical literature aims to reduce manual labor and provide more enriched information to empower research and medical treatments. Lu et al. (1) demonstrated that there is an increasing interest to use text mining techniques for curation workflows. Currently, literature curation struggles with a lack of automated annotation techniques—particularly for gene ontology annotations (1). As medical technology advances and more curation sources become available, this need magnifies. In medical informatics alone, the number of indexed articles has increased by an average of 12% each year between 1987 and 2006 (2). With an increasing number of publications detailing even more complex information, the need to have reliable and generalizable computational techniques increases rapidly.

Finding gene functions discussed in literature is crucial to genomic information extraction. Currently, tagging the gene functions in published literature is a mainly manual process. The curators find gene function evidence by reviewing each sentence in the article and mapping the results to gene ontologies. Gene Ontology (GO) (3) is a set of controlled vocabulary that defines gene product functions. BioCreative IV is a National Institutes of Health (NIH) workshop which aims to automate gene functional curation though computational methods. With a focus on gene functions, it includes two sub tasks: a) Retrieving GO evidence sentences for relevant genes, b) Predicting GO terms for relevant genes. We focus on sub task b, which finds the related gene functions (GO terms) in a set of genes discussed in an article. More details about the shared task and the corpus can be found in Auken et al. (4). This task is very similar to BioCreative I subtask
2.2 which was held in 2004 (5). Blaschke et al. (5) summarized the results for BioCreative I. For subtask 2.2 the highest precision was reported to be 34.62% (6). BioCreative IV GO subtask 2 includes an annotated corpus which enables to measure recall and f-measures. Couto et al. (7) used the IR technique to find related sentences and GO terms. Chiang et al. (6) combined sentence classification with pattern mining. Ray et al. (8) proposed a solution based on probabilistic model and Naïve Bayes classifier. Most of the participants in the previous related task focused on information content and statistical models combined with machine learning. Here, we propose an unsupervised method based on distributional semantic similarity that can be easily applied for different types of texts and ontologies.

Material and Methods
Our method is based on distributional semantic similarity of sentences to GO terms. We use semantic vectors package (9) implementation of LSA (10) with random indexing (11) to calculate semantic similarities. GO terms' semantic vectors are created based on GO names defined in GO; one semantic vector is created for each term in the ontology. Stop-words are removed from GO name and they are generalized by Porter stemming (12).

Figure 1 shows the overall flow of our proposed method. After creating GO semantic vectors, the question is to find whether or not a sentence is related to a gene. We do this by using lexical patterns and generalizing the sentence and gene symbol (e.g. removing the numbers and non alphabetic characters). If “Sentence Gene Matcher” predicts that a sentence is related to a gene, then we calculate semantic similarity of the sentence to all GO terms using already generated semantic vectors. The “Go Finder” module finds all related GO terms to the sentence and generates the triplet of sentence, gene and GO term. Finally the output in the shared task expected format is generated by “BioC output generator”. In next section we explain about the “GO Finder” module in more details.

“GO Finder” Module
GO Finder finds related GO terms for each sentence. We define G as a set of top m GO terms with highest semantic similarity to the sentence. D is the set of top n GO terms with high similarity to the abstract of the related article. The following function returns top k similar GO terms for a given query:

\[ \text{TopSimilarGO}(\text{query}, k) = \{x | x \in \text{GTerms} \land |\{y \in \text{GTerms} | \text{Sim}(x, \text{query}) < \text{Sim}(y, \text{query})\}| < k\} \]

And G and D sets are:
\[ G(\text{sentence}) = \text{TopSimilarGO}(\text{sentence}, m) \]
\[ D(\text{abstract}) = \text{TopSimilarGO}(\text{abstract}, n) \]
If a sentence is predicted to have the gene mention, the predicted GO terms for the sentence and gene are the conjunction of top similar GO terms to the sentence (set G) and top similar GO terms to the related abstract (set D):

$$GeneGO(gene, sentence, abstract) = \{G(sentence) \cap D(abstract)\} \text{ if HasGene(sentence, gene) else } \{\}$$

A GO term with the highest semantic similarity to the sentence in GeneGO set will be chosen as the final GO annotation for each gene in the sentence. For example if a sentence top \(m=2\) similar GO terms are \{g5, g10\} and the abstract top \(n=5\) GO terms are \{g4, g8, g5, g2, g9\}, then the final predicted GO terms for the sentence related to the gene will be \{g5\}. \(m\) and \(n\) are tuning parameters that control precision and recall.

Table 2 summarizes the number of sentences in the training set which was detected by “Sentence Gene Matcher” as relevant to a gene and also annotated to have a gene function. The table shows that “abstract”, “front” and “title2” of each document are the most important sections that might include gene function. We found that the first sentences of paragraphs have information about GO terms, but including all sentences in a paragraph will significantly reduce the precision. Therefore, we limit searching for the gene functions to the mentioned sections of the article. We choose one set of values for \(m\) and \(n\), for “Front”, “Abstract” and “Title2” (\(m\)-FAT, \(n\)-FAT), and choose a different set for the first sentence of the paragraphs (\(m\)-Paragraph, \(n\)-Paragraph). Next section shows detail analysis of the impact of the tuning parameter on precision and recall.
### Results and Discussion

To achieve the highest f-measure, tuning parameters \((m\) and \(n\)) needs to be adjusted accordingly. We use two sets of values for \(m\) and \(n\); one set for the first sentence of each paragraph and another for other passage types (“abstract”, “front” and “title2”). Figure 2-a depicts precision, recall and f-measure change in respect to \(m\)-FAT changes. As \(m\)-FAT increases, precision declines and recall increases. The maximum f-measure is achieved for \(m\)-FAT=3. Therefore we assign \(m\)-FAT to 3, and try to find the best value for \(m\)-Paragraph. Figure 2-b shows the change of performance based on change of \(m\)-Paragraph. The best f-measure of 0.304 is achieved for \(m\)-FAT=3 and \(m\)-Paragraph=4.

When \(m\)-Paragraph varies, the change in f-measure is not as significant as when \(m\)-FAT varies. In addition, recall is almost constant for \(m\)-FAT >3. This shows that considering more than 4 GO terms for each sentence in FAT sections does not help us much and can only decrease the precision. On the other hand, considering only one top GO term for the first sentence of each paragraph gives the maximum boost to the recall.

We have improved parameter tuning after official submission and Figure 2 shows slightly better results than the official submission. In the first run, we tried to get the high f-measure; for run 2 and 3 we tried to get high precision and high recall respectively. Table 2 shows values of tuning parameters for each run.

### Table 2

<table>
<thead>
<tr>
<th>Passage</th>
<th>With Gene Function</th>
<th>Total</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>front</td>
<td>26</td>
<td>67</td>
<td>39%</td>
</tr>
<tr>
<td>title_2</td>
<td>149</td>
<td>797</td>
<td>19%</td>
</tr>
<tr>
<td>abstract</td>
<td>225</td>
<td>1253</td>
<td>18%</td>
</tr>
<tr>
<td>paragraph</td>
<td>1700</td>
<td>20703</td>
<td>8%</td>
</tr>
<tr>
<td>fig_title_caption</td>
<td>17</td>
<td>412</td>
<td>4%</td>
</tr>
<tr>
<td>fig_caption</td>
<td>99</td>
<td>6009</td>
<td>2%</td>
</tr>
<tr>
<td>table_title_caption</td>
<td>0</td>
<td>47</td>
<td>0%</td>
</tr>
<tr>
<td>title_1, title_3, title_4</td>
<td>0</td>
<td>26</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 2. Summarizes the number of sentences in the training set which was detected by “Sentence Gene Matcher” as relevant to a gene and also annotated to have a gene function.
In this work we proposed an unsupervised approach for gene function extraction from documents. Here we only use GO terms’ names for creating semantic vectors. We tried using GO terms description but it does not help. Using more fine tuned vocabulary set for each GO term can result in more accurate vectors and probably increases the performance of this method. In addition, using term-term semantic similarity for expanding sentence terms can be evaluated. In this work we used annotations for finding the important passage types, evaluating the method and finding the best settings for the parameters. The main advantage of using unsupervised open IE technique is that it can easily be generalized and applied to similar relation extraction problems. The results from this method can be used as a baseline for supervised systems. In the future, we plan to combine this approach with supervised techniques.
Funding
This work was partially supported by National Library of Medicine under [grant number R01LM011176]. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Library of Medicine.

References