Context-Aware Mapping of Gene Names using Trigrams

ThaiBinh Luong [1,2]
thaibinh.luong@yale.edu

Nam Tran [1]
nam.tran@yale.edu

Michael Krauthammer [1,2]
michael.krauthammer@yale.edu

[1] Department of Pathology, Yale University, New Haven, CT, USA
[2] Program for Computational Biology and Bioinformatics, Yale University, New Haven, CT, USA

Abstract

We present a method for the mapping of gene names to Entrez Gene identifiers. We first resolve lexical variation by transforming domain terms into their unique trigrams, and use this representation for a preliminary term mapping. We then perform fine-mapping via contextual analysis of the abstract that contains the domain term. We have formalized our method as a sequence of matrix manipulations, allowing for a fast and coherent implementation of the algorithm. We pair our method with existing approaches for entity recognition, and achieve an F-score of 0.761 in the BioCreative 2 Gene Normalization Task.

Keywords:

1 Introduction

Our paper addresses the Gene Normalization Task of the BioCreative 2 Challenge. We approach this task as a term identification problem, which can be subdivided into three modular stages: term recognition, term classification, and term mapping [1]. Our method presented here focuses on the third step, the mapping of biomedical terms to some controlled vocabulary, which we think is most relevant with respect to the Gene Normalization Task. The advantage of our approach is that our mapping strategy is independent from the underlying term recognition and classification process, and can therefore be paired with a multitude of previously published methods for recognizing and classifying terms.

We believe that two fundamental processes are at play when mapping biomedical terms: A first *approximate* mapping of a term to known biomedical concepts, and a subsequent fine-mapping using contextual analysis. The first step analyzes lexical term variation, and results in a prioritized list of possible biomedical concepts. It is solely based on the local features (aka the letters/words) of the unmapped biomedical term. The second step is a contextual analysis of the term mentioning. Only the latter enables the definite placement of the term with respect to a unique biomedical concept. We believe that this approach may be similar to the way we humans approach the term mapping problem. After encountering a novel gene name, which looks similar (but not identical) to known gene names, we can infer the correct gene by comparing the context (such as a scientific abstract) with the previously encountered literature. The contextual analysis may result in the identification of similar, already known abstracts, that discuss known genes. If the novel gene name is similar to the names of those known genes, we can easily make the final term assignment.

There are several noteworthy features of our approach: First, we are clearly separating the local and contextual mapping, enabling the experimental examination of both processes individually. Second, our local analysis is fast and efficient, avoiding the traditional string matching techniques. Similarly, we perform a fast contextual analysis with respect to thousands of previously published abstracts.

2 Methods and Results

As discussed above, we approach the task as a term mapping problem. The idea is to use existing programs for entity recognition, and then use the methods described below to map recognized and classified strings to external gene identifiers (in our case: Entrez GeneIDs). For entity recognition, we use Abner [2] (both Biocreative and NLPBA settings) and LingPipe[1] (GeneTag model), two programs with excellent recall and precision. We process PubMed abstracts three times, for each program and setting. Each run gives rise to a separate list of recognized entities, which are then separately mapped to Entrez GeneIDs. A majority vote is then cast to determine the list of abstract-specific GeneIDs.

We use a combination of two methods to map recognized entities to their appropriate gene identifiers: the *Trigram Method*, and the *Network Method*. Both methods require preprocessing, using resources from Entrez Gene, to construct a set of method-specific matrices.

2.1 Trigram Method

The first method, as mentioned earlier, is designed to quickly retrieve a list of possible gene identifiers, which are good mapping candidates for each entity recognized by Abner/LingPipe. The method should be fast, but does not need to resolve uncertainties, such as homonymy. In short, our method utilizes an approximate representation of a gene names, by transforming a name into the set of its unique trigrams. The similarity between 2 gene names is the number of their common trigrams (i.e. the intersection of their sets of trigrams). This approach allows for the fast mapping of a gene name to a dictionary of gene names, such as the Entrez Gene resource, with its associated gene identifiers.

To accomplish this, we first need a preprocessing step, in which all the unique gene names/synonyms ("gene strings") from the Entrez Gene resource are identified, and split into a set trigrams, a succession of three alphanumeric characters. For example, the gene string "lypla1" (the official symbol of GeneID 10434) would be split into 4 trigrams: "lyp", "ypl", "pla", and "la1".

Let m be the number of all the possible trigrams (that occur across all strings in the Entrez Gene resource), then a string s is represented by an m-vector v of 0 and 1, such that $v_i = (i\text{th trigram} \in s)$ for all $1 \leq i \leq m$.

The similarity between two strings u and v is defined as the dot product $u \bullet v$.

Let n be the number of all the unique Entrez Gene strings. Let A_S be the n x m matrix whose rows are the vector transposes of the strings' representations. We can then easily determine the similarities of a query string u (i.e. the trigram representation of the string recognized by Abner/LingPipe) to all the Entrez Gene strings by computing the product

$$r_S = A_S u$$

The results vector r_S is of dimension n, the number of unique gene strings. The similarity scores need to be normalized, in order to penalize improper string matches. For example, suppose our query string is "abl". Gene strings that contain words such as "transpos*abl*e", "dis*abl*e", or "vari*abl*e" will receive the same similarity scores as a simple gene string "abl". For this reason, we take into consideration how well a query string is contained within an Entrez Gene string, ie whether the number of trigrams in the query sting matches the number of trigrams in the gene string. Vice versa, we also calculate how well an Entrez Gene string is contained within the query string. We thus weight the results vector r_S accordingly, assigning the highest weights to gene strings that match the query string exactly (are perfectly contained within each other).

We denote the normalized results vector r_{Sn}. The latter vector contains similarity scores for each gene string. However, we are interested in finding the maximum similarity score on the gene level, i.e. looking at each synonym of a gene (a set of gene strings) and selecting the synonym (gene string) with the highest score. This is done by probing results vector r_{Sn} in a gene-by-gene fashion. To accomplish this, we construct an n x l matrix, A_{GS}, where l is the number of unique GeneIDs, and n is the number of unique Entrez Gene strings as described above.

A value of "1" in $A_{GS(i,j)}$ implies that GeneID j is associated with gene string i. We then update A_{GS} by r_{Sn}.

$$A_{GSu} = diag(r_{Sn})A_{GS}$$

From A_{GSu}, we construct a vector g_S, which is of size l, the number of unique (human) GeneIDs.

[1] http://www.alias-i.com/lingpipe

$$g_S = [\left|A_{GSu}^{(1)}\right|_\infty, \left|A_{GSu}^{(2)}\right|_\infty, ..., \left|A_{GSu}^{(l)}\right|_\infty]$$

Here, $A_{GSu}^{(i)}$ is the *i*th column vector of A_{GSu} and $|\ \ |_\infty$ is the maximum norm. Thus $g_{S(j)}$ represents the highest scoring gene string per GeneID j.

2.2 Network Method

The first method calculates g_S, a vector of size *l*, the number of human genes, with $g_{S(i)}$ representing the trigram-similarity score of gene *i* (with respect to a recognized entity *E*). It is possible that several genes have the same similarity score, and we need another method for pinpointing the correct gene identifiers. To accomplish this, the Network Method examines the words (context) of the abstract, where the entity has been recognized. The idea is as follows: Assume that the Trigram Method determines that a recognized entity *E* may be linked to two different gene identifiers (gene *A* and *B*) with equal similarity scores. The network method compares the abstract *a*, where the entity has been recognized, to a collection of abstracts where gene A and B have been positively identified. If the content of abstract *a* is closer to the set of abstracts linked to gene A, we label entity *E* with gene identifier A. We devised a method to rapidly perform the above procedure across all human genes. As in the Trigram Method, there is a need to preprocess external resources to create method-specific matrices. We use the Entrez gene2pubmed resource to identify *p* abstracts that are positively linked to human genes (often, several abstracts are linked to a single human GeneID). We preprocess those abstracts to extract a list of unique and stemmed words, and weigh those words according to a normalized TF*IDF measure. We then construct a *p* x *q* matrix A_N, where *p* is the number of abstracts and *q* is the number of unique stemmed words that appear across all *p* abstracts. Furthermore, we construct a *p* x *l* matrix A_{GN} associating abstracts with their GeneIDs (similar to the matrix A_{GS} in the Trigram Method above). We follow a similar procedure as outlined in the Trigram Method above. Given an input abstract containing the recognized entity *E*, we transform the abstract into a *q*-vector *u* and calculate

$$r_N = A_N u$$

r_N is of size *p*, the number of abstracts, and contains the resulting similarity scores of the input abstract *a* to the abstracts in A_N. We then can easily[2] group the abstracts that are mapped to the same GeneID by calculating

$$g_N = A_{GN} r_N$$

Vector g_N is of size *l*, the number of unique (human) GeneIDs, and contains the similarity scores of the abstract *a* to each GeneID[3].

2.3 Combining the Methods

The vectors of trigram scores and network scores for each Entrez Gene, g_S and g_N, are now combined to assign the final GeneID for each recognized entity *E*. We first look at g_S, and read the set of those GeneIDs with a perfect score of 1. If the set consists of a single GeneID, we assign that ID to the entity *E*. If the set is >1, we sort the set by the network score g_N, and assign the highest ranked GeneID to the entity *E*. By default, we do not assign a GeneID if there is no entry in g_S with a perfect score of 1 (this measure aims at eliminating incorrectly recognized entities).

2.4 Results

We evaluated our two methods on the Biocreative 2 GN testing set, which consisted of 262 abstracts discussing human genes. The task was to identify all the gene identifiers of those genes. We submitted a single run of our program to the BioCreative Challenge. Our combination of the Trigram and Network methods yielded a recall of 0.740, a precision of 0.784, and an f-score of 0.761. Subsequent analysis of the Trigram method on the same set produced results, which were slightly lower than the Trigram/Network method, as expected. The recall and precision were 0.684 and 0.707, respectively, and the f-score was 0.695.

[2] Not shown is a normalization step, where we normalize g_N with respect to the number of abstracts that link to a particular GeneID.

[3] The BioCreAtIvE 2 GN training and testing sets contained abstracts that were part of Entrez's gene2pubmed file. We checked whether a training and testing abstract *a* was part of gene2pubmed, and removed the mapping information of abstract *a* from A_{GN} (the identification of the correct gene identifiers for abstract *a* would otherwise be trivial).

We also analyzed our method's ability for gene name mapping in presence of a perfectly marked-up corpus (ie perfect entity recognition), where we assign GeneIDs to all entities. Our preliminary data suggest that we can achieve an accuracy of 0.78 for mapping to the correct GeneID (unpublished data).

The main reason for mis-mapping stems from the issue of "containment". Our computation of trigram scores favors genes that more closely contain the entity *and* do not contain extra trigrams. Another source of incorrect mapping can be attributed to the lack of a close variation in the gene_info file (our "dictionary"). The last major category of incorrect mapping are those entities in which we cannot correctly disambiguate between two genes that have the same trigram score, but very close network scores. In many of the cases, the group of lexically equivalent genes belong to the same family of genes.

3 Discussion

We describe a coherent, matrix-based method for approximate and contextual term mapping in the biomedical domain. We believe that our approach is unique in that it provides a coherent framework in solving both the problem of lexical variation and term ambiguity of gene names.

A trigram-representation of phrases has been previously described as being useful in finding synonyms in the biomedical domain [3]. Here, we show that the use of trigrams is similarly effective for mapping of gene names. Also, there exist earlier studies that discuss the inclusion of contextual information for term mapping (see for example [4, 5]). We think that our method adds an elegant solution to this problem, by providing a fast vector-space method for resolving gene name ambiguity in a large biomedical dictionary (Entrez Gene), without the need for machine learning. There is ample room for expansion of our method. We are investigating different ways of combining the results vectors g_S and g_N of the Trigram and Network method, respectively. We also need to address the problem of gene names that consist of fewer than 3 characters. An obvious solution is the use of a bigram representation.

References

[1] Krauthammer, M. and G. Nenadic, *Term identification in the biomedical literature.* J Biomed Inform, 2004. 37(6): p. 512-26.

[2] Settles, B., *ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text.* Bioinformatics, 2005. 21(14): p. 3191-2.

[3] Aronson, A.R., et al., *The NLM Indexing Initiative.* Proc AMIA Symp, 2000: p. 17-21.

[4] Liu, H., S.B. Johnson, and C. Friedman, *Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS.* J Am Med Inform Assoc, 2002. 9(6): p. 621-36.

[5] Schuemie, M.J., J.A. Kors, and B. Mons, *Word sense disambiguation in the biomedical domain: an overview.* J Comput Biol, 2005. 12(5): p. 554-65.