bioRxiv preprint doi: https://doi.org/10.1101/103648; this version posted July 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

A novel Word2vec based tool to estimate semantic similarity of genes by using Gene Ontology terms

Dat Duong[1], Eleazar Eskin[1,2], Jingyi Jessica Li[2,3]

Department of Computer Science[1], Human Genetics[2], Statistics[3], University of California, Los Angeles, United States of America.

Abstract

The Gene Ontology (GO) database contains GO terms that describe biological functions of genes and proteins in the cell. A GO term contains one or two sentences describing a biological aspect. GO database is used in many applications. One application is the comparison of two genes or two proteins by first comparing semantic similarity of the GO terms that annotate them. Previous methods for this task have relied on the fact that GO terms are organized into a tree structure. In this old paradigm, the locations of two GO terms in the tree dictate their similarity score. In this paper, we introduce a new solution to the problem of comparing two GO terms. Our method uses natural language processing (NLP) and does not rely on the GO tree. In our approach, we use the Word2vec model to compare two words. Using this model as the key building-block, we compare two sentences, and definitions of two GO terms. Because a gene or protein is annotated by a set of GO terms, we can apply our method to compare two genes or two proteins. We validate our method in two ways. In the first experiment, we measure the similarity of genes in the same regulatory pathways. In the second experiment, we test the model's ability to differentiate a true protein-protein network from a randomly generated network. Our results are equivalent to those of previous methods which depend on the GO tree. This gives promise to the development of NLP methods in comparing GO terms.

1 Introduction

The Gene Ontology (GO) project founded in 1998 is a collaborative effort that provides consistent descriptions of genes and proteins across different databases and species. The GO database contains vocabularies referred to as GO terms which describe the functions of genes and proteins. This GO database is divided into three categories: cellular components (CC), molecular functions (MF) and biological processes (BP). The CC category contains terms describing the components of the cell and can be used to locate a protein. The MF category contains terms describing chemical reactions such as *catalytic activity* or *receptor binding*. These terms do not specify the genes or proteins involved in the reactions or the locations of the events. The BP category contains terms describing a series of events. The general rule to assist in distinguishing between a biological process and a molecular function is that a biological process must have more than one distinct step[3].

To make these definitions more concrete, consider the GO terms GO:0005622, GO:0016787, and GO:0001822 in the CC, MF, and BP category respectively. GO:0005622 has the name

bioRxiv preprint doi: https://doi.org/10.1101/103648; this version posted July 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

"intracellular" and definition "The living contents of a cell; the matter contained within (but not including) the plasma membrane, usually taken to exclude large vacuoles and masses of secretory or ingested material". GO:0016787 has the name "hydrolase activity" and definition "Catalysis of the hydrolysis of various bonds". GO:0001822 has the name "kidney development" and definition "The process whose specific outcome is the progression of the kidney over time, from its formation to the mature structure".

One can use the GO database in various applications[3]. Here we focus on the task of measuring the semantic similarity of two genes or two proteins. Because a gene or protein is described by a set of GO terms, we must first address the problem of measuring the semantic similarity of two GO terms.

The problem of comparing two GO terms very much relates to how the GO terms are organized in the gene ontology. The GO terms within each category are organized into an directed acyclic graph (DAG) where there is only one $root$ node[3]. In this tree of GO terms (or GO tree), a more generic term (i.e. cell cycle phase) is nearer to the root, whereas a very specific term (i.e. mitotic cell cycle phase) is nearer to a leaf. Because of this design, GO terms with a direct ancestor (i.e. sibling nodes) are deemed to be more related than GO terms with a distal ancestor. Moreover, because of this design, existing methods to measure the semantic similarity of two GO terms rely heavily on the GO tree. Broadly speaking, these methods are either node-based or edge-based[10]. The key focus in all node-based methods is the evaluation of the information content of the common ancestors of two GO terms. In brief, the information content of a GO term measures the usefulness of the term by evaluating how often the term is used to annotate a gene or a protein. Terms that are used sparingly have very high information content because they are very specific and can be used to distinguish genes. Node-based methods have been shown to be quite successful and are very popular[10].

Unlike node-based methods, edge-based methods measures the distance (or average distance when more than one path exists) between two GO terms. Edge-based methods have one serious problem. Despite being organized into a tree, the GO terms at the same level do not always have the same specificity because different gene properties require different levels of detailed explanation. Thus, edge-based methods suffer from the problem of *shallow annotation*: terms separating by the same distance are assigned the same similarity score regardless of their positions in the GO tree[16]. Different schemes to weigh the edges according to their positions in the GO tree were suggested but failed to fully resolve the problem[9]. Thus, we do not further consider edge-based methods in our paper.

In this paper, we approach the problem of comparing two GO terms from a different angle. We apply a method that is independent of the GO tree and is from the natural language processing (NLP) domain. We follow a building-block approach in which we solve easier problems before the harder ones. We first define a metric to compare two words. In this task, we train the Word2vec model using open access articles on Pubmed, so that we can use this model to represent a word as a N-dimensional vector[11]. Cosine similarity is used to compare two words. Using this metric as the key block, we devise a model to compare two GO terms. A GO term has a definition which is usually one or two sentences describing a biological feature. To assess the similarity of two GO terms, we consider a more simplistic view by treating the definition of each GO term as a set

bioRxiv preprint doi: https://doi.org/10.1101/103648; this version posted July 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

of words, and we use the modified Hausdorff distance to measure the distance between two sets. Finally, because every gene is annotated with a set of GO terms, we also use the modified Hausdorff distance to measure the similarity of two genes.

We name the metric in our model w2vGO. We create a simple user interface to compare two GO terms using this metric. Our software, input data, and results are available on github [1]. We conduct two experiments to compare w2vGO against two popular node-based methods Resnik and Aggregate Information Content (AIC)[14,16]. In the first experiment, we use the three metrics to measure the similarity scores of genes in the same regulatory pathways. A good method would give high similarity scores for these pairs of genes. In the second experiment, we test the three metrics at differentiating a real protein-protein interaction network from a network where the interactions are randomly assigned. Our results show that w2vGO is as good as Resnik and AIC. This gives many promises to the use of NLP methods in comparing GO terms and genes.

2 Method

2.1 Node-based methods to measure similarity between two GO terms

There are many node-based methods, but we choose the Resnik and AIC methods as the baseline for the following reasons. Resnik method is one of the very first methods to quantify the similarity between two GO terms[14]. Interestingly, despite being very simple, Resnik method has been shown to outdo some of its extensions in several test datasets[10,12]. AIC method is recently new, and previous works that compare various approaches to measure similarity scores have not yet extensively experimented with this method[16].

2.1.1 Resnik method

The most basic node-based method introduced by Resnik in 1999 relies on the information content (IC) of a GO term[14]. IC of a GO term t is computed as $\text{IC}(t) = -\log(p(t))$ where $p(t)$ is the probability of observing a term t in the ontology. $p(t)$ is computed as $p(t) = \frac{\text{freq}(t)}{\text{freq}(root)}$. The term $\text{freq}(t)$ counts the frequency of a term t, where $\text{freq}(t) = \text{count}(t) + \sum_{c \in \text{child}(t)} \text{freq}(c)$.

Function $\text{count}(t)$ is the number of genes annotated with the term t and $\text{child}(t)$ are the children of t. Based on this definition, $\text{IC}(root) = 0$, and a node near the leaves has higher IC than nodes at upper levels. To compute a similarity score of the GO terms a, b, one finds the most informative common ancestor of these two terms.

$$\text{Resnik}(a, b) = \max_{p \in \{\text{par}(a) \cap \text{par}(b)\}} \text{IC}(p) \tag{1}$$

where $\text{par}(t)$ denotes every ancestor of term t. We use the R library GOSim to compute Resnik similarity score. $\text{Resnik}(a, b)$ ranges from 0 to infinity because the probability $p(t)$

[1] github.com/datduong/word2vec2compareGenes

bioRxiv preprint doi: https://doi.org/10.1101/103648; this version posted July 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

ranges from 0 to 1. In this model, the similarity score of a GO term t to itself is not 1. Second, when a, b have only $root$ as a common ancestor, then Resnik$(a, b) = 0$. This is problematic because leaf nodes are more informative than other types of nodes. Consider an example where the $root$ is the only common ancestor of the pair a, b and the pair c, d. Next, suppose that a, b are leaf nodes. c is the parent of a, and d is the parent of b. $root$ is the parent of both c, d. One would then expect that Resnik$(a, b) <$ Resnik(c, d); however, one would obtain Resnik$(a, b) =$ Resnik(c, d).

2.1.2 Aggregate Information Content (AIC) Method

The AIC method by Song et al.[16] amends the two problems in Resnik method. To encode the fact that leaf nodes are more informative, AIC defines a knowledge function of term t as $k(t) = 1/\text{IC}(t)$ which is used to measure its semantic weight $sw(t) = 1/(1+\exp(-k(t)))$. Here $sw(root) = 1$. Semantic value $sv(t)$ of t is then $sv(t) = \sum_{p \in \text{path}(t)} sw(p)$.

The function path(t) contains every ancestor of t and the term t itself. Because of the function sv, the method is named aggregate information content. Usually, $sv(a) < sv(b)$ when term a is nearer to the root than b. The similarity score of two GO terms a, b is defined as

$$\text{AIC}(a, b) = \frac{2 \sum_{p \in \{\text{path}(a) \cap \text{path}(b)\}} sw(p)}{sv(a) + sv(b)} \tag{2}$$

AIC(a, b) ranges from 0 to 1. In this model, AIC$(a, a) = 1$. When a, b have only $root$ as the common ancestor, then AIC$(a, b) = 2/(sv(a) + sv(b))$ which depends on where a, b are on the GO tree. We use the R library GOSim to get the IC values, and code the AIC metric by ourselves in R.

2.2 Word2vec model

The Word2vec model converts a word into a N-dimensional vector where the user can choose N. Word2vec transforms similar words into similar vectors, thus enabling one to numerically quantify the similarity between two words by using Euclidean distance or cosine similarity. At the heart of the Word2vec is the neural network model with one input layer, one hidden layer, and one output layer[11]. Like all neural network models, Word2vec requires a training data. Loosely speaking, one can view the mechanic of the Word2vec model as a 2-step mapping[15]. In the first step, a word w from the input layer is mapped into a N-dimensional vector at the hidden layer. In the second step, this vector is mapped back into the word $\hat{w}$. Word2vec chooses the values in the N-dimensional vector at the hidden layer so that $w = \hat{w}$ for every w in the training data. The purpose of this paper is not to rigorously discuss Word2vec model; we are interested in adopting this model to measure the similarity of GO terms and compare it with other methods. Interested readers are encouraged to read the original manuscript by Mikolov et al.[11], and the introduction to Word2vec by Rong[15].

bioRxiv preprint doi: https://doi.org/10.1101/103648; this version posted July 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

2.2.1 Measure similarity of two words using Word2vec

One must train the Word2vec model before using it, and the types of training data can influence the application of the model. We set $N = 250$ and train the Word2vec to recognize biological words. We use 15 GB of data from open access articles on Pubmed. The raw count of unique and repeated words is 7,762,873,870. We remove words which appear less than 100 times in the whole training data, thus reducing the final number of unique words to be 380,594. We use the Python library gensim to train the Word2vec model[13]. A simple Python user interface is available at github [2].

There are two important details here. First, the training data does not contain definitions of GO terms found in the GO database. This helps us avoid data reusing. Second, theoretically speaking, Word2vec model can train on the GO terms in the Pubmed data, so that one can convert a GO term into a vector. Unfortunately, the Ids of the GO terms are not used too often in published papers, and detecting definitions of GO terms in papers is a different type of research problem[17]. For these reasons, we use the Word2vec model as a metric to compare two biological words.

To compare two vector representations of two words, we use the cosine similarity. In NLP, cosine similarity is preferred over Euclidean distance because it is bounded whereas, Euclidean distance is not[8]. We define the function $\text{w2v}(z, v)$ as the similarity score of two words z, v.

2.2.2 Measuring similarity of two GO terms using Word2vec

A GO term comes with a definition which is usually one or two sentences describing a biological feature. For example, the GO term id GO:0003700 has the definition: "Interacting selectively and non-covalently with a specific DNA sequence in order to modulate transcription. The transcription factor may or may not also interact selectively with a protein or macromolecular complex".

When a GO term definition has more than one sentence, we concatenate these sentences into the same sentence by ignoring the period symbol. For example, the two sentences for the term GO:0003700 is considered as one long sentence.

Thus, the task to compare two GO terms reduces to the problem of comparing their definitions which are two sentences. Suppose GO term a, b have sentences Z, V as their definitions respectively. We treat two sentences $Z =$ "$z_1\ z_2\ z_3 \ldots z_N$" and $V =$ "$v_1\ v_2\ v_3 \ldots v_M$" as two unordered sets of words $Z = \{z_1, z_2 \ldots z_N\}$ and $V = \{v_1, v_2 \ldots v_M\}$. We use the modified Hausdorff distance (MHD) to measure the similarity of sentences (or sets) Z and V[2]. Here, we add weights to the MHD and name the metric weighted MHD (WMHD)

$$\text{WMHD}(Z, V) = \min \left\{ \sum_{i=1\ldots N} \text{content}(z_i) \max_{j=1\ldots M} \text{w2v}(z_i, v_j), \quad \sum_{j=1\ldots M} \text{content}(v_j) \max_{i=1\ldots N} \text{w2v}(z_i, v_j) \right\} \tag{3}$$

$\text{content}(w)$ is the weight of the word w and is used very frequently to distinguish common words from rare ones. In using WMHD with Word2vec model, the weights of

[2] github.com/datduong/word2vec2compareGenes

bioRxiv preprint doi: https://doi.org/10.1101/103648; this version posted July 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

words can help avoid the influence of hub-words (i.e. words such as $cell$, dna, $activity$) which are ubiquitously associated with many other words[7]. content(w) is very similar to the IC function[8].

$$\text{content}(w) = -\log\left(\frac{\text{number of times word } w \text{ appears in training data}}{\text{training data size}}\right) \tag{4}$$

There are more sophisticated models that consider the word-ordering in the sentences, but they are not ideal in this paper for two reasons. First, suppose one has a metric that compares two words. When applying this metric to compare two sentences by using Hausdorff distance or any other metrics to compare two sets, considering word ordering does not always greatly improve performance[1]. Second, other methods that directly handle word-ordering when comparing two sentences often take a sentence as a whole input[6,18]. This idea does not fit well into the building-block approach. The GO database is continuously evolving, with new definitions being added and old definitions being modified[5]. Thus, it is better to have a system where we can measure the similarity of two words, so that we can then use this metric as the key block to build a model that compares two sentences.

In any case, we have defined a metric to measure the two GO terms a, b with definitions Z, V under the Word2vec paradigm. Because the a GO term a and its definition Z are two equivalent entities, for simplicity, we define w2vGO(a, b) = WMHD(Z, V) to be the similarity score of a, b. Theoretically, w2vGO(a, b) ranges from -1 to 1 because the function w2v(z, v) ranges from -1 to 1. However, in practice, we have yet to observe a negative similarity score.

2.3 Measuring similarity of two genes

A gene is annotated with several GO terms within each of the three GO categories. For example, the gene HOXD4 which is important for morphogenesis is annotated by these GO terms GO:0003677, GO:0003700, and GO:0006355. Thus, we can view any gene A as a set of GO terms. A GO term a is in the set A (i.e. $a \in A$) if a is used to annotate A.

To asses the similarity between two genes A and B (within a specific GO category), we must compare two sets of GO terms. There are many metrics for this task[10,12]. Here, we use the modified Hausdorff distance[2]. The original MHD works with distances where, given two objects a, b, a small distance $d(a, b)$ implies that a, b are near[2]. Similarity score $s(a, b)$ used to measure two GO terms a, b is the opposite of the distance function. A high $s(a, b)$ implies that a,b and be are near. Therefore, we redefine the MHD to measure the similarity score of two genes A, B to be

$$\text{MHD}(A, B) = \min\left\{ \frac{1}{|A|}\sum_{a\in A}\max_{b\in B} s(a, b), \quad \frac{1}{|B|}\sum_{b\in B}\max_{a\in A} s(a, b) \right\} \tag{5}$$

In the above, the function $s(a, b)$ is a generic placeholder for measuring GO terms a, b. For example, if one uses Resnik, AIC, or w2vGO similarity score then $s(a, b)$ = Resnik(a, b), AIC(a, b) or w2vGO(a, b) respectively.

bioRxiv preprint doi: https://doi.org/10.1101/103648; this version posted July 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

MHD(A, B) ranges from 0 to 1 for AIC. Theoretically, it is from 0 to infinity for Resnik and -1 to 1 for w2vGO. In practice, we have seen that MHD(A, B) for Resnik is bounded above by 1, and w2vGO is bounded from below by 0 (Figure 1).

2.4 P-value of similarity score between two genes

When the similarity scores between two genes is 0 or 1, it is easy to interpret the outcome. But for any other values of similarity score, it is not always easy to determine how truly similar the two genes are. To this end, we generate the null distribution of a similarity score by computing the scores for many randomly chosen pairs of genes. Here, as an example, we use only the GO terms in the BP ontology to compare genes. Figure 1 shows that w2vGO, AIC, and Resnik have different null distributions. The 95% quantile for w2vGO, AIC, and Resnik are 0.61921, 0.51886, and 0.33588, respectively. By using these distributions, one can compute the the p-value of a similarity score by finding its rank with respect to these distributions. These distributions are used later in the result section.

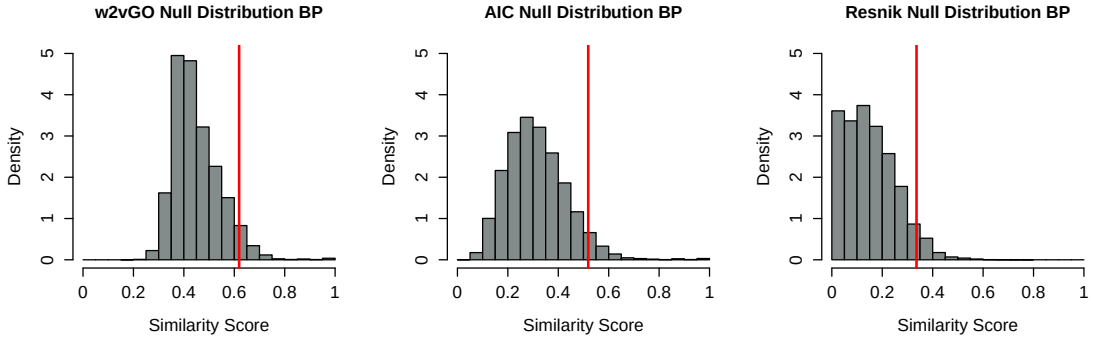

Figure 1: The distribution of similarity scores for many randomly chosen pairs of genes in the BP ontology.

3 Result

We compare the performance of the w2vGO metric against the traditional Resnik and the recently introduced AIC metric. We download the latest GO term definitions (dated 11/30/2016) and GO annotation data for human genes (dated 11/30/2016) at the geneontology.org. We divide the GO annotation data for human into the BP, MF, and CC categories. The data and results in this section are available at github.

3.1 Measuring similarity scores for genes in the same regulatory pathway

Previous work has shown that Resnik produces good results when used to compare genes in the same regulatory pathways[4]. The developers of AIC did not compare AIC against Resnik using genes in regulatory pathways[16]. In this experiment, we show that w2vGO

bioRxiv preprint doi: https://doi.org/10.1101/103648; this version posted July 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

and AIC are comparable to Resnik when used for said task. We extract the genes from 100 regulatory pathways in human from the Kyoto Encyclopedia of Genes and Genomes website (www.genome.jp/kegg).

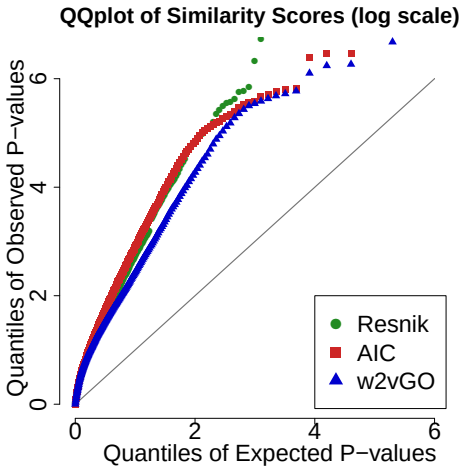

Figure 2: QQ plot of p-values for similarity scores of genes in the same regulatory pathways. BP ontology is used.

In this experiment, we use only the BP ontology because the genes in the same regulatory pathway are expected to involve in the same biological process, but they may not have the same molecular function or reside in the same part of the cell. We treat the genes within a pathway as a completely connected graph and compute the similarity scores for all the pairs of genes. For the 100 regulatory pathways, there are a total of 72086 pairs.

Because these similarity scores are computed for genes in the same pathways, we expect their p-values to be very inflated as compared to the quantiles of a uniform density. The p-value of a similarity score is computed using the null distribution that is described in section 2.4. Figure 2 confirms this expectation and shows that all three metrics are very comparable.

3.2 Measuring similarity scores for protein-protein interaction network

We use the protein-protein interaction (PPI) data prepared by Mazandu and Mulder[10]. This PPI data contains 6031 interactions; 5366 of which have both interacting partners annotated by the BP gene ontology, and 5580 pairs have both interacting partners annotated by the CC gene ontology. We trim this PPI data further, keeping only human proteins that can be mapped to some gene names. This mapping is done by using uniprot.org. The final data has 2898 and 2866 pairs for BP and CC category, respectively.

In this experiment, it is only best to use the BP and CC category, because two interacting proteins can be in the same biological process or located in the same part of the cell. It is not guaranteed that they share the same molecular function[9]. We treat the BP and the CC category separately.

Like in Mazandu and Mulder[10], our goal is to compare how well each metric differentiates a true PPI network from a randomly made PPI network. For each PPI network, we use Resnik, AIC, and w2vGO to compute the similarity scores of the edges (i.e. pairs of proteins). Then we prepare a random PPI network having the same number of edges, by randomly selecting protein pairs that do not occur in the real PPI network. The real and random PPI network have the same proteins; we only require that they have different interacting partners.

To compare the performance of w2vGO against Resnik and AIC, we find the area under the curve (AUC) of the Receiver Operator Curve (ROC) curve. The real and random PPI network provide the true positive and false negative rate, respectively. The AUC is computed by plotting the true positive versus false negative rate at different thresholds

bioRxiv preprint doi: https://doi.org/10.1101/103648; this version posted July 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

and computing the area under this curve. AUC value goes from 0 to 1, with 1 being the best prediction power.

For the BP category, the AUC for w2vGO, AIC, and Resnik are 0.83796, 0.85128, 0.82308 respectively. For the CC category, the AUC for w2vGO, AIC, and Resnik are 0.78596, 0.77704, 0.76587 respectively. The prediction power of using BP ontology is better than that of using CC. This agrees with Mazandu and Mulder[10]. Intuitively, proteins that share similar biological processes are likely to interact, whereas proteins in the same part of the cell do not necessarily interact. Again, the three methods are very comparable.

4 Discussion

In this paper, we explore an entirely new approach to measure the similarity score between two GO terms and between two genes. Our method depends on the Word2vec model to compare two words. Using this as the key building-block, we compare two sentences and then two definitions of two GO terms. Next we compare two sets of GO terms; this task is equivalent to comparing two genes because genes are annotated by sets of GO terms. Unlike previous methods which claim to measure the semantic similarity yet rely very much on the GO tree, our model is entirely from the natural language processing domain and is independent of the GO tree. In measuring similarity of genes in the same regulatory pathways and protein-protein interaction, our model is at least comparable to the Resnik and AIC model.

Contributions

DD and JJL came up with the problem. DD downloaded and cleaned the data, came up with and coded the solution, and wrote the paper. JJL gave advice on database to be used. DD and JJL discussed the results. DD, JJL, and EE read the paper.

References

[1] Achananuparp, P., Hu, X., and Shen, X. (2008). The evaluation of sentence similarity measures. In International Conference on Data Warehousing and Knowledge Discovery, pages 305–316. Springer.

[2] Dubuisson, M.-P. and Jain, A. K. (1994). A modified hausdorff distance for object matching. In Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on, volume 1, pages 566–568. IEEE.

[3] Gene Ontology Consortium (2017). Expansion of the gene ontology knowledgebase and resources. Nucleic acids research, **45**(D1), D331–D338.

[4] Guo, X., Liu, R., Shriver, C. D., et al. (2006). Assessing semantic similarity measures for the characterization of human regulatory pathways. Bioinformatics, **22**(8), 967–973.

bioRxiv preprint doi: https://doi.org/10.1101/103648; this version posted July 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

[5] Guzzi, P. H., Agapito, G., Milano, M., et al. (2016). The impact of gene ontology evolution on go-term information content. arXiv preprint arXiv:1612.09499.

[6] He, H., Gimpel, K., and Lin, J. (2015). Multi-perspective sentence similarity modeling with convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1576–1586.

[7] Levy, O., Goldberg, Y., and Ramat-Gan, I. (2014). Linguistic regularities in sparse and explicit word representations. In CoNLL, pages 171–180.

[8] Li, Y., McLean, D., Bandar, Z., et al. (2006). Sentence similarity based on semantic nets and corpus statistics. IEEE Transactions on Knowledge and Data Engineering, **18**(8), 1138–1150.

[9] Mazandu, G. K. and Mulder, N. J. (2012). A topology-based metric for measuring term similarity in the gene ontology. Advances in bioinformatics, **2012**.

[10] Mazandu, G. K. and Mulder, N. J. (2014). Information content-based gene ontology functional similarity measures: Which one to use for a given biological data type? PLoS ONE, **9**(12), e113859.

[11] Mikolov, T., Sutskever, I., Chen, K., et al. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.

[12] Pesquita, C., Faria, D., Falcão, A. O., et al. (2009). Semantic similarity in biomedical ontologies. PLoS Computational Biology, **5**(7), e1000443.

[13] Rehurek, R. and Sojka, P. (2011). Gensim–python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic.

[14] Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. J. Artif. Intell. Res.(JAIR), **11**, 95–130.

[15] Rong, X. (2014). word2vec parameter learning explained. arXiv preprint arXiv:1411.2738.

[16] Song, X., Li, L., Srimani, P. K., et al. (2014). Measure the semantic similarity of GO terms using aggregate information content. IEEE/ACM Transactions on Computational Biology and Bioinformatics, **11**(3), 468–476.

[17] Tuan, L. A., Kim, J.-j., and Ng, S.-K. (2013). Gene ontology concept recognition using cross-products and statistical methods. In BioCreative Challenge Evaluation Workshop vol., page 174.

[18] Tymoshenko, K., Bonadiman, D., and Moschitti, A. (2016). Convolutional neural networks vs. convolution kernels: Feature engineering for answer sentence reranking. In Proceedings of NAACL-HLT, pages 1268–1278.