CONCEPT EMBEDDING FOR DEEP NEURAL FUNCTIONAL ANALYSIS OF GENES AND DEEP NEURAL WORD SENSE DISAMBIGUATION OF BIOMEDICAL TEXT

by

Ahmad Pesaranghader

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
June 2019

© Copyright by Ahmad Pesaranghader, 2019

To the memory of my mother, Nasrin Hakimian, who inspired me to have faith in God, taught me to care and respect, encouraged me to be curious and believe in my dreams, and showed me so much could be done with (little) love

Table of Contents

List of Tables

List of Figures

Abstract

As far as Gene Ontology (GO) is concerned, most of the existing gene functional similarity measures combine information content-based semantic similarity scores of single GO-term pairs to estimate gene functional similarity, whereas a few models base their approach on Jaccard similarity to compare GO terms in groups for this measurement. However, almost all of these measures are dependent on the ever-changing structure of GO, they are slow and task-dependent, and do not consider the valuable natural language definition of GO terms. The first part of this thesis introduces the simDEF model which avoids these drawbacks by considering the advantage of distributed representation of GO terms using their text definitions. Manual feature engineering, large dimensions of distributed GO-term vectors, the use of traditional metrics to aggregate GO-term similarity scores prior to computation of gene functional similarity, and, resorting to separate evaluation of each sub-ontology in GO (biological process, cellular component, or molecular function) in a biological task, are challenges that can be addressed by Deep Learning. Therefore, we introduce deepSimDEF that avoids the majority of the above-mentioned issues. For this purpose, deepSimDEF network(s) learn low-dimensional vectors of GO terms and gene products, and then learn how to calculate the functional similarity of protein pairs using these vectors (a.k.a. embeddings). By considering all GO sub-ontologies, deepSimDEF increases yeast PPI predictability by $\sim$4%, shows a Pearson's correlation improvement >6% with yeast gene expression and >4% with human gene expression, and improves correlation with yeast sequence homology by up to 11%. The beneficial method for distributed representations of GO terms can be utilized in other domains of Machine Learning for low-dimensional embedding of concepts. In the second part of this thesis, this concept embedding method is evaluated in the task of Word Sense Disambiguation of natural text. Hence, deepBioWSD, a one-size-fits-all model is devised which consists of a single Bidirectional Long Short-Term Memory network classifier. We use the MSH-WSD dataset to compare WSD algorithms while macro and micro accuracies are employed as evaluation metrics. We show deepBioWSD outperforms the existing supervised models in (biomedical) text WSD by achieving the state-of-the-art performance of 96.82% for macro accuracy.

List of Abbreviations Used

AUC Area under (the ROC) Curve

BLAST Basic Local Alignment Search Tool

BLSTM Bidirectional Long Short-Term Memory

BMA best-match average

BP Biological Process

CC Cellular Component

CUI Concept Unique Identifiers

FS functional similarity

GloVe Global Vectors for Word Representation

GO Gene Ontology

IC Information Content

IEA Inferred from Electronic Annotation

LRBS log-reciprocal BLAST score

LSA Latent semantic analysis

LSTM Long Short-Term Memory

MF Molecular Function

MSH Medical Subject Headings (MeSH)

NLP Natural Language Processing

PMI Pointwise Mutual Information

PPI Protein-protein interactions

RRBS relative reciprocal BLAST score

SOC second-order co-occurrence

SS semantic similarity

UMLS Unified Medical Language System

WSD Word Sense Disambiguation

ZSL Zero-shot Learning

Acknowledgements

I praise to the Lord Almighty for his blessings and mercy.

I am heartily grateful to my beloved parents, Dr. Majid Pesaranghader and Nasrin Hakimian, and to my siblings, Narges and Ali, for all sacrifices they made for me throughout my life.

I am deeply thankful to my supervisors, Dr. Stan Matwin and Dr. Marina Sokolova for the invaluable support they provided me throughout the preparation of this thesis, and for the incredible autonomy they gave me in the conduct of my research. Their friendly and encouraging guidance has been crucial at all stages of my doctoral studies. Likewise, I am thankful to the committee members of my thesis defence, Dr. Jimmy Huang, Dr. Robert Beiko, Dr. Evangelos E. Milios and Dr. Hong Gu for their comments and the time they devoted to reading this thesis and improving the overall quality of the final submission.

I would also like to show my sincere gratitude to Dr. Robert Beiko, Dr. Ali Pesaranghader (my brother), and Xiang Jiang for sharing their pearls of wisdom with me that enabled me to develop a deeper understanding of the subjects during the course of this research.

Ahmad Pesaranghader

Chapter 1

Part I: Prologue

1 Introduction

1 .1 Gene Ontology and Gene Function Analysis

The Gene Ontology project (GO) [9] is a bioinformatics initiative to characterize all the important features of genes and gene products within a cell using a structured and controlled vocabulary. UniProt [36], SwissProt [17], and many other biomedical databases are annotated by the GO terms in order to communicate semantic roles of biomedical entities. In addition, countless biomedical and biological studies have been receiving benefits from GO and GO annotations directly or indirectly in their experiments. These studies run the gamut from generation and employment of more focused biological networks of interest (e.g. protein-protein interaction networks [114], co-expression networks [43, 149, 222], and gene co-functional networks [198, 88]) to investigation of novel techniques of drug discovery [54, 218], disease-discovery [251, 120], and cancer treatment [125, 236]. These studies largely employ GO term semantic similarity measures which subsequently leads to functional similarity prediction of genes. This prediction will help a study to come up with their final inference, production, or tool - however, the results of semantic/functional similarity measures usually will be integrated with other biological metrics or statistical measures for a more reliable proposition. Likewise, since the *in vitro* biomolecular experiments designed for validating gene functions are expensive, in the recent years, the automatic computation of gene function prediction of the biomedical entities using their GO annotations has been under data mining investigations extensively [216, 183]. To this end, every year, the GO offers a repertoire of functional terms for CAFA competition[1] [180, 53, 80] aiming for Critical Assessment of protein Function Annotation algorithms. In these regards, these developed methods and software tools get evaluated against a wide

[1] `http://biofunctionprediction.org/`

range of biological problems such as prediction of protein-protein interaction (PPI) [214, 131], analysis of gene expressions [236, 73], comparison of homologous genes [31, 176, 38], and evaluation of functional annotations of enzymes [68, 174].

In several recent studies, distinguished from those that propose various invariants of semantic similarity and functionality similarity measures, the effect of different aspects of these measures such as the effect of annotation size [94] and the effect of shared information on semantic calculations are investigated [15, 245]. For a comprehensive review of these studies refer to [41, 121]. As to the significant impact of the former, certain number of studies have been concerned with the enrichment of GO annotations by means of biological data hand in hand with computational tools [195, 27], while some of them even employ deep learning tools such as autoencoders in order to facilitate reaching this goal [28]. Moreover, due to the importance and wide applicability of these measures in the domain, speeding up these functional similarity measures has been the subject of study in several research works as well [212, 98]. The problem of computational efficiency for pair-wise approaches (e.g. information content-based measures [184]) is even more prominent because they are dependent on the combination of semantic similarity.

Part 2 of this thesis is concerned with these biological studies as well as biological concepts and tools available. Furthermore, it proposes two models, namely simDEF and deepSimDEF, to facilitate and speed up the process of gene function analysis.

1 .2 Word Sense Disambiguation of Natural Text

In the Natural Language Processing (NLP) community, Word Sense Disambiguation (WSD) has been described as the task which selects the appropriate meaning (sense) to a given word in a text or discourse where this meaning is distinguishable from other senses potentially attributable to that word. These senses could be seen as the target labels of a classification problem. That is, machine learning seems to be a possible way to tackle this problem. WSD task is a potential intermediate task [228] for many other NLP systems, including mono and multilingual Information Retrieval, Information Extraction, Machine Translation or Natural Language Understanding.

One of the important use-cases of a WSD model is in the biomedical domain in

which lots of polysemous terms[2], acronyms, and abbreviations exist. For example, the simple word *cold* has several senses and may refer to a *disease*, a *temperature sensation*, or an *environmental condition*, (or even it is an acronym for a harsh lung-related disease called *chronic obstructive lung disease*). The intended specific sense is determined by the textual context in which an instance of the ambiguous word appears. In "I am taking aspirin for my cold" the *disease* sense is intended, in "Let's go inside, I'm cold" the *temperature sensation* sense is meant, while "It's cold today, only 2 degrees", implies the *environmental condition* sense. Therefore, automatically identifying the intended sense of ambiguous words improves the proper inference of biomedical text data for the clinical and biomedical applications.

Furthermore, the need of enhanced WSD capabilities appears in many applications whose aim is not language understanding or its usage in a specific domain. Among others, we could mention: Machine Translation [21, 137], Information Retrieval [204, 250, 47, 156], Semantic Parsing [18], Speech Synthesis and Recognition [146], Acquisition of Lexical Knowledge [119], Lexicography [209], and etc..

The Part 3 of this thesis is concerned about WSD models and the advantages they provide to the NLP applications. Importantly, in the second half of that part, in Chapter 5, we propose a model named deepBioWSD that is mainly developed to deal with the problem of needing to have multiple WSD classifiers in a supervised setting of sense prediction.

1 .3 A Quick Excursion to the Deep Learning Land

With the revival of deep feedforward neural networks around 2006 [13, 66], deep learning methods have become prevalent in the research community. These are representation learning methods that compose multiple non-linear modules to obtain multiple levels of representation [99]. These non-linear modules can transform the representation of the raw input at one level into a representation at a higher, more abstract level. The key advantage of deep learning is that human engineers do not design these layers of features and, therefore, the least feature engineering is needed [58]. Over the last few years, deep learning methods have brought about breakthroughs in image recognition and speech recognition [65, 93] and proved their beneficial usage

[2]A polysemous term has many meanings

in many natural language processing (NLP) tasks [34, 217, 247, 206]. Since deep learning methods demonstrate their excellent performance in the general domain, recently they have gained attention from biomedical and bioinformatics communities as well. Therefore, their effectiveness has been evaluated in text mining of biomedical data [30, 106] and segmentation of medical images [48, 199]. These studies also benefits from autoencoders (AEs), variational autoencoders (VAEs [90]), and generative adversarial networks (GANs [59]) for feature construction, information extraction, and synthetic generation of biomedical images or other biological/biomedical entities [210, 28, 84, 190].

deepSimDEF presented in Chapter 3 (in Biology Part 2), and deepBioWSD proposed in Chapter 5 (in NLP Part 3) are designed and benefit from deep learning concepts and methods.

2 Research Problems and Motivations

This thesis addresses two research problems related to A) functional analysis of genes, and B) word sense disambiguation of natural text. Regarding the former, we introduce the deepSimDEF model to improve the similarity estimations of the gene products and their functionalities. Regarding the latter, we introduce deepBioWSD which helps to improve the disambiguation accuracy of a natural language or specific domain (e.g., biomedical or financial documents). We state and discuss each research problem and objective as follows:

2 .1 Problem/Objective I: Biological Attribute Embedding for Gene Function Analysis

Problem. As far as GO is concerned, most of the existing gene functional similarity measures combine semantic similarity scores of single GO term pairs to estimate gene functional similarity (pair-wise measures), whereas others compare GO terms in groups for this measurement (group-wise measures) [211, 170] (for further details refer to Chapter 2 and Chapter 3). However, almost all of these measures are strictly dependent on the ever-changing topological structure of GO; they are very slow and extremely task-dependent leaving no room for their generalization, and none of them takes the valuable textual definition of GO terms into consideration. Our first model,

simDEF (presented in Chapter 2), avoids these drawbacks by taking into account the significant advantage of distributed (vector-based) representation of GO terms using their textual definitions. In contrast to information content (IC) based semantic similarity measures such as Resnik [184] and Lin [110] which depend solely on the structure of GO, this distributed representation helps for better semantic similarity measurement of GO terms leading to more accurate gene functional similarity estimation. However, simDEF suffers from some unaddressed yet important shortcomings, many of which are shared with the existing models. Manual feature engineering, large dimensions of distributed GO-term vectors, the use of traditional metrics to aggregate GO-term semantic similarity scores prior to computation of genes functional similarity, and, resorting to separate evaluation of each sub-ontology in GO in a biological task, are some of these inadequacies. These limitations and problems present the challenges of measuring genes functional similarity reliably.

Objective. As an objective to deal with the above-mentioned problems, by relying on the expressive power of deep neural networks, we lay out and develop the deepSimDEF model (presented in Chapter 3). Briefly, deepSimDEF is an efficient model for measuring functional similarity of proteins and other gene products (e.g., microRNA and mRNA) using natural language definitions of GO terms annotating those genes. For this purpose, deepSimDEF neural network(s) (single-channel and multi-channel) learn low-dimensional vectors of GO terms and gene products and then learn how to calculate the functional similarity of protein pairs using these learned vectors which are known as embeddings.

2 .2 Problem/Objective II: Natural Language Concept Embedding for Word Sense Disambiguation

Problem. The reason for the importance of WSD lies in the ambiguity of human language [135], which is so pervasive that huge numbers of words can be interpreted in multiple ways depending on the context in which they occur. Therefore, resolving the sense ambiguity of words is obviously essential for many Natural Language Understanding applications [75]. However, current WSD models are mostly designed based on the notion of "*one classifier per (one ambiguous) word*" [194]. Meaning a large amount of labelled data are needed for training of a supervised WSD model

(which consists of a large number of disambiguation classifiers); these supervised models typically outperform unsupervised, semi-supervised, and knowledge-based models. However, sense annotation of a large amount of data is an unattainable and arduous task to achieve as the labelling process is labor-intensive and time-consuming (mainly when it is done in specific domains in which expert knowledge is needed). This is a challenging problem as it poses a formidable hurdle on the way of real-life implementation of supervised WSD algorithms.

Objective. The objective of Part 3 of this thesis is to tackle this inadequacy in addressing the task of WSD, especially when supervised training of a model is concerned. For this purpose, inspired by the approach to generate embeddings of biological entities (e.g., genes and gene products) introduced in Part 2, we introduce deepBioWSD that can largely benefit from sense embedding of natural language semantic units (i.e., senses or concepts). To do so, deepBioWSD offers a one-size-fits-all model that consists of a Bidirectional Long Short-Term Memory network devised to be trained on all available training data of all ambiguous instances, meaning we do not dedicate one (labelled) ambiguous instance only to its associated classifier (and vice versa). These considerations alleviate the need for large numbers of training samples as the network can be trained in unsupervised fashion as well. This is because as a single classifier for all instances it shares statistical strength across all words and their contexts by scaling well when the vocabulary size increases.

2 .3 Research Methodology

We followed an incremental methodology for this thesis. It comprised seven general tasks, as depicted in Figure 1.1. We began with the literature review, and the data collection and/or generation task. We then designed our simDEF and deepSimDEF models for gene function analysis, and our single-network and deepBioWSD models for word sense disambiguation of natural text. The adjustment of parameters, experiments, and discussion were the oncoming tasks. The central hexagon, i.e., the analysis task, had an influential role, because we discussed pros and cons of existing methods and established our developments based on that. Finally, the process was like a lattice where all tasks were interactively fulfilled (rather than being in a queue).

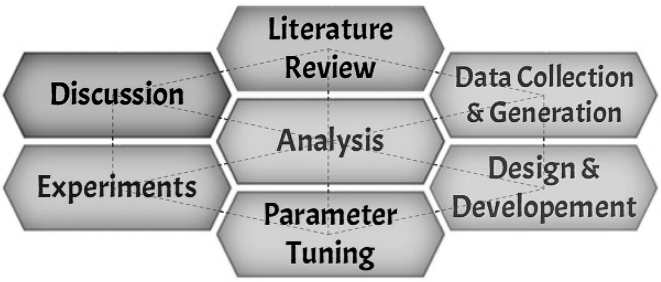

Figure 1.1: Research Methodology Lattice

Additionally, in order to ensure the machine learning reproducibility of the conducted experiments and the achieved results would be easily possible for every one, throughout the studies we made sure the guideline/checklist[3] from Dr. Joelle Pineau for reproducibility was met. For that reason, the source codes of the algorithms are shared and are publicly available through online repositories.

2 .4 Main Contributions of the Thesis

Despite several popular and well-performing word embedding methods (e.g., Word2Vec [128], GloVe [151], ...), still there is shortage of promising concept embeddings methods with which several critical and essential "research problem" would become more feasible to address in practical sense. Regarding this, the main contributions of the thesis are as follows:

1. We build a promising concept embedding method based on natural language definition of concepts coming from:
   - Gene Ontology (GO)
   - Unified Medical Language System (UMLS)
2. In Bioinformatics, we propose models that can deal with Functional Analysis of Genes. For such models we have:
   - simDEF Model (refer to Chapter 2)
   - deepSimDEF Model (refer to Chapter 3)

[3]https://www.cs.mcgill.ca/ jpineau/ReproducibilityChecklist.pdf

3. In NLP, we introduce and develop models which are capable of dealing with Word Sense Disambiguation of natural language data. These models include:

    - single-classifier BLSTM (refer to Chapter 4)
    - deepBioWSD Model (refer to Chapter 5)

The description of the concept embedding method is almost repeated in every chapter depending on the bioinformatics or NLP setup - and that is what glues Part 2 and Part 3.

2 .5 Research Accomplishments and Deliverables

We list our research accomplishments and deliverables, each of which made this thesis possible, as below:

1. **Journal Paper:** "*simDEF: definition-based semantic similarity measure of gene ontology terms for functional similarity analysis of genes*" (Bioinformatics, Oxford, England) authorship: A. Pesaranghader et al., [153]

2. **Journal Paper:** "*deepSimDEF: deep neural embeddings of gene products and Gene Ontology terms for functional analysis of genes*" (Journal of Genome Biology - to be submitted) authorship: A. Pesaranghader et al.

3. **Conference Proceeding:** "*One single deep bidirectional LSTM network for word sense disambiguation of text data*" (In Advances in Artificial Intelligence: 31st Canadian Conference on Artificial Intelligence, Canadian AI 2018, Toronto, ON, Canada) authorship: A. Pesaranghader et al., [157]

4. **Journal Paper:** "*deepBioWSD: effective deep neural word sense disambiguation of biomedical text data*" (JAMIA: Journal of the American Medical Informatics Association) authorship: A. Pesaranghader et al., [154]

5. The simDEF and deepSimDEF source code, embeddings, and prepared data:

    - https://github.com/ahmadpgh/simDEF
    - https://github.com/ahmadpgh/deepSimDEF

6. The deepBioWSD source code, embeddings, and prepared data:

    - https://github.com/iwera-git/deepBioWSD

We state throughout the studies of this thesis we made sure that the guideline/checklist[4] from Dr. Joelle Pineau for machine learning reproducibility was met. This list is used by DL/ML community (e.g., NeuroIPS, ICML).

2 .6 Thesis Organization

The remainder of this thesis is organized as follows. After the current chapter, we provide Part 2 which is concerned with biological aspects of the study. Chapter 2 in that part introduces simDEF that partially addresses some of the existing issues regarding gene function analysis as long as the employment of Gene Ontology is concerned. In Chapter 3, deepSimDEF offers low-dimensional embeddings of biological entities (e.g., genes and gene products) and improves simDEF even further by employing deep learning methods. After dealing with biological aspects of the study, in Part 3 we address the Natural Language Processing task of WSD. In the first chapter of that part, Chapter 4, we propose a model that consists of a single WSD classifier. This model, however, is designed based on the consideration of the word embeddings of the context words. This consideration of word embeddings instead of sense embeddings, however, occurs at the cost of accuracy and true sense prediction (the proposed model is evaluated in the general language context). Benefiting from our approach for gene embedding generation, in Chapter 5, we propose deepBioWSD which outperforms our preceding word embedding-based model as well as the other existing supervised WSD models; the evaluation is done in the biomedical domain. Finally, in Part 4, we conclude the thesis and discuss future work in Chapter 6. Each of the Chapters 2 to 5 are provided in such a way that they would be as self-contained as possible, meaning some minor overlap in the introduction and the description of the tools employed might be noticed. Moreover, in the beginning of every chapter, a short summary regarding the content of that chapter and what is delivers and accomplishes is provided.

[4]https://www.cs.mcgill.ca/ jpineau/ReproducibilityChecklist.pdf

In order to help a reader quickly and conveniently navigate among the results provided in the thesis, Table 1.1 for Part 2 and Table 1.2 for Part 3 are provided respectively. For biological experiments, following what is common in the literature, we compared the results of the proposed models against protein-protein interaction, gene expression and sequence homology datasets in order to validate the superiority of the proposed algorithms. For word sense disambiguation experiments, we considered two cases: comparison between the existing supervised WSD models (using either SenseEval-3 or MSH-WSD benchmark datasets), and, evaluation of the proposed model(s) with different hyperparameter settings (ablation study). Since the sample ambiguous terms in SenseEval-3 come from different dictionaries/thesauri, it caused difficulty for sense embedding construction. Therefore, we limited deepBioWSD model to MSH-WSD dataset which is only dependent on the Unified Medical Language System (UMLS) meta-thesaurus.

Table 1.1: Navigation guideline for functional similarity models and the correlation experiments in Part 2, Biology and Bioinformatics

	PPIs		**Gene Expression**		**Sequence Homology**	
	F1-score	AUC	Spearman	Pearson	Spearman	Pearson
simDEF Chapter 2	✓Table 2.5	✓Table 2.4	-	✓Table 2.3	-	✓Table 2.1, 2.2
deepSimDEF Chapter 3	✓Table 3.4	-	✓Table 3.6	✓Table 3.5	✓Table 3.8	✓Table 3.9

Table 1.2: Navigation guideline for WSD models and conducted experiments in Part 3, Natural Language Processing

	SensEval-3		**MSH-WSD**	
	between-all-models	within-our-model	between-all-models	within-our-model
single BLSTM Chapter 4	✓Table 4.3	✓Table 4.4	-	-
deepBioWSD Chapter 5	-	-	✓Table 5.7	✓Table 5.8

Chapter 2

Part II: Biological Attribute Embedding for Function Analysis of Genes
simDEF for Gene Function Analysis

1 Summary

Motivation – Measures of protein functional similarity are essential tools for function prediction, evaluation of protein-protein interactions (PPIs) and other applications. Several existing methods perform comparisons between proteins based on the semantic similarity of their Gene Ontology (GO) terms; however, these measures are highly sensitive to modifications in the topological structure of GO, tend to be focused on specific analytical tasks and concentrate on the GO terms themselves rather than considering their textual definitions.
Results – We introduce simDEF, an efficient method for measuring semantic similarity of GO terms using their GO definitions, which is based on the Gloss Vector measure commonly used in natural language processing. The simDEF approach builds optimized definition vectors for all relevant GO terms, and expresses the similarity of a pair of proteins as the cosine of the angle between their definition vectors. Relative to existing similarity measures, when validated on a yeast reference database, simDEF improves correlation with sequence homology by up to 50%, shows a correlation improvement >4% with gene expression in the biological process hierarchy of GO and increases PPI predictability by >2.5% in F1-score for molecular function hierarchy.
Publication – Original paper authored by Pesaranghader et al. [153] is available in: https://doi.org/10.1093/bioinformatics/btv755 (Journal of Oxford Bioinformatics)

2 Introduction

Gene Ontology (GO) [9] describes the attributes of genes and gene products using a structured vocabulary. Many biomedical databases, such as UniProt [36] and

SwissProt [17], are annotated by GO terms to communicate semantic meanings of biomedical entities. Computing functional similarity of biomedical entities has been applied to problems such as prediction of protein-protein interaction (PPI) [179], gene expression studies [141] and homology analysis [176]. Also, in the context of text mining various studies have aimed to enhance the literature-based GO annotation of gene products [83, 82].

There are two main computational models available to measure similarity of terms. *Ontology-based models* take advantage of lexical structures in their estimation of term similarity. Edge-based ontology measures like Wu [231] and RSS [230] consider the number of edges along the paths that link two GO terms. Node-based measures (which we designate as information-content-based), such as Resnik [184], Jiang [78], Lin [110], Schlicker et al. [192], TCSS [77], GraSM [37] and AIC [200] compare the properties of the terms augmented with the properties of their ancestors or descendants. IC vectors [155] represent IC values in distributed forms in the computation of semantic similarity. Hybrid measures such as those of Wang et al. [221], Liu et al. [113] and HRSS [229] combine node-based and edge-based measures. While these measures first compute semantic similarity of two gene products and then aggregate the results as a single functional similarity value, group-wise measures such as simUI [49], simGIC [171] and SORA [211] calculate similarity by measuring two sets of GO terms annotating these genes. Huang et al. [71] also proposed a similarity measure where gene functional similarity is based on vector representations of their GO terms. Ontology-based measures suffer from three important limitations: first, they depend on the constantly changing topological structure of GO; second, they use incomplete GO annotations to compute statistical information; and third, they offer no guarantee of generalization to multiple biological tasks.

Distributional-based approaches derive from John Rupert Firth's idea [51] that a term is characterized by the company it keeps in its context. Measures following this notion calculate terms specifications from relevant text data and represent them in a vector space for subsequent computation of their similarities. The Gloss Vector semantic relatedness measure [147] is a distributional-based approach with a wide application in natural language processing. This measure constructs definitions (glosses) of terms from a predefined thesaurus, and estimates semantic relatedness of

two terms as the cosine of the angle between those terms' gloss-vectors. Interpolation of content words of a text corpus into the terms definition was shown to outperform the direct definition comparison. Gloss vectors offer a new opportunity to exploit the information of GO term definitions and to infer gene functional similarity. Liu et al. [115] successfully applied the Gloss Vector measure to the biomedical domain using MEDLINE as the text corpus and the unified medical language system and WordNet for the construction of extended definitions of medical concepts. The Gloss Vector approach requires a frequency cut-off in selecting the best features describing one term [156, 158]. We have developed simDEF, an optimized version of the Gloss Vector targeted to analysis of gene functions. Here, by using MEDLINE as the text corpus, we compare the performance of simDEF with other leading approaches, and demonstrate its effectiveness using comparisons based on sequence homology, gene expression and PPI data.

2 .1 Measures of Semantic Similarity Already Applied in GO Context

Most early semantic similarity measures were developed for linguistic studies in natural language processing. Recently, semantic similarity measurement methods have been applied to and further developed and tailored for biological uses as listed below. Considering GO and gene product annotations as information resources, the semantic similarity measures investigated in this chapter employing these resources are as follows:

Resnik Measure. Resnik (1995) [184] uses the concept of "information content" (IC) to define a semantic similarity measure. The IC for a term located in an ontology is based on the probability or *p(t)* of occurrence of that term in a corpus.

$$p(t) = \frac{\text{freq}(t)}{\text{freq (root)}} \qquad (2\ .1)$$

freq(t) is the frequency of t and all its descendants in the ontology summed together. Generally, IC of a term in an ontology indicates how informative that term is in that

ontology. As a rule of thumb, the closer to the root, the less informative that term will be. IC of the term t is given by:

$$IC(t) = -\log(p(t)) \tag{2 .2}$$

The more information two terms share, the higher their similarity. The shared information is captured by the set of common ancestors in the graph. The amount of shared information and thus the similarity between the two terms is quantified by the IC of their least common ancestors (LCA). This leads us to the following formula for similarity measurement of two terms in an ontology:

$$\mathrm{sim}_{Resnik}(t_1, t_2) = \max\left(IC\left(LCA\left(t_1, t_2\right)\right)\right) \tag{2 .3}$$

Jiang and Conrath Measure. Since the Resnik measure considers only the IC of ancestors and ignores input terms level of specificity, Jiang and Conrath (1997) [78] deal with this issue by taking the IC of the input term into account:

$$\mathrm{sim}_{Jiang}(t_1, t_2) = 1 + IC\left(LCA\left(t_1, t_2\right)\right) - \frac{IC\left(t_1\right) + IC\left(t_2\right)}{2} \tag{2 .4}$$

Lin Measure. Since Jiang was originally an unnormalized distance measure, Lin (1998) [110] proposed a new similarity measure to resolve that issue:

$$\mathrm{sim}_{Lin}(t_1, t_2) = \frac{2 \times IC\left(LCA\left(t_1, t_2\right)\right)}{IC\left(t_1\right) + IC\left(t_2\right)} \tag{2 .5}$$

GraSM Measure. Resnik uses the most informative common ancestor (LCA), but GraSM [37] takes into account the average ICs for all disjoint common ancestors instead of choosing only the maximum IC among all the disjoint common ancestors.

GraSM assumes that two common ancestors are disjunctive if there are independent paths from both ancestors to the GO term:

$$\mathrm{sim}_{GrasM}\left(t_1, t_2\right) = \mathrm{avg}\left(IC\left(LCA\left(t_1, t_2\right)\right)\right) \tag{2 .6}$$

Wang Measure. Wang et al. [221] attempts to improve existing measures by aggregating the semantic contributions of ancestor terms in the GO graph. Formally, a GO term c can be represented as $DAG_c = (c;\, T_c;\, E_c)$ where Tc is the set including term c and all of its ancestor terms in the GO graph, and E_c is the set of edges connecting the GO terms in DAG_c (edges which connect Tc terms). By defining the semantic value (SV) of term c as the aggregate contribution of all terms in DAG_c to the semantics of term c, Wang proposes terms closer to term c inDAG_c contribute more to its semantics. The semantic value (SV) of a GO term c is:

$$SV(c) = \sum_{t \in T_c} S_c(t) \tag{2 .7}$$

where S_c is semantic contribution of term c or its ancestors into c's meaning. The semantic contribution of term t to term c is calculable by:

$$S_c(t) = \begin{cases} 1 & \text{if } t = c \\ \max\left\{\omega_{\mathrm{e}} \times S_c(t') | t' \in \text{ children of } t\right\} & \text{if } t \neq c \end{cases} \tag{2 .8}$$

where ω_{e} is the "semantic contribution factor" for edge e $\in E_c$ linking term t with its child term t'. Finally, the semantic similarity between two GO terms t_I and t_2 is:

$$\mathrm{sim}_{Wang}\left(t_1, t_2\right) = \frac{\sum_{t \in T_1 \cap T_{t_2}} \left(S_{t_1}(t) + S_{t_2}(t)\right)}{SV\left(t_1\right) + SV\left(t_2\right)} \tag{2 .9}$$

AIC Measure. AIC or Aggregated Information Content [200] is the latest variation of IC-based semantic similarity measures which considers the aggregate contribution of the ancestors of a GO term to the semantics of that GO term. In their study, they first propose the semantic weight of GO term t as:

$$SW(t) = \frac{1}{1 + e^{-(IC(t))^{-1}}} \tag{2 .10}$$

and then, by considering A_x as the ancestor set of term x to the root (including x itself), the semantic value SV(x) of the GO term x is computed by adding the semantic weights of its ancestors:

$$SV(x) = \sum_{t \in A_x} SW(t) \tag{2 .11}$$

Having the above values, the semantic similarity between GO terms t_1 and t_2 based on their aggregate IC is as follows:

$$\mathrm{sim}_{AIC}\left(t_1, t_2\right) = \frac{\sum_{t \in A_1 \cap A_{t_2}} 2 \times SW(t)}{SV\left(t_1\right) + SV\left(t_2\right)} \tag{2 .12}$$

simGIC Measure. simGIC or Graph Information Content similarity [171] is a functional similarity of gene products. It directly employs the IC of GO terms associated with two gene products. For two gene products A and B with annotation sets of T_A and T_B, simGIC is given by:

$$\mathrm{simGIC}(A, B) = \frac{\sum_{t \in T_A \cap T_B} IC(t)}{\sum_{t \in T_A \cup T_B} IC(t)} \tag{2 .13}$$

simUI Measure. Like SimGIC, simUI or Union-Intersection similarity [49] is a functional similarity of gene products. simUI is given by the number of terms in the intersection of T_A and T_B divided by the number of terms in their union.

$$\text{simUI}(A, B) = \frac{\text{COUNT}_{t \in T_A \cap T_B}}{\text{COUNT}_{t \in T_A \cup T_B}} \quad (2.14)$$

2 .2 Gloss Vector Semantic Relatedness Measure

Generally, this measure constructs definitions (glosses) of the terms from a predefined thesaurus and estimates the semantic relatedness of two terms using the cosine of the angle between those terms' gloss-vectors. Pedersen et al. [147] proposed this measure as a combination of term definitions from a thesaurus and cooccurrence data from a text corpus. In their approach, every word in the definition of one term from WordNet gets replaced by its context vector from the co-occurrence data from the corpus, and then all of these context vectors summed together build that term's definition-vector (gloss-vector). The Gloss Vector measure is highly valuable as it employs both terms definitions and empirical knowledge implicit in a text corpus. The Gloss Vector comprises five steps:

1. Construction of first order co-occurrence matrix by scanning and counting bigram frequencies (i.e. words that cooccur) in the corpus

2. Removing insignificant words using low and high-frequency cut-off points (done by elimination of very low/high frequent bigrams),

3. Using a taxonomy (or a linked thesaurus), developing an extended definition for a term by adding definitions of the directly linked terms to a target term in the taxonomy to the definition of that term,

4. Constructing a definition matrix (all definition vectors) by employing the thresholded first-order matrix from step 2 (cut-off first-order matrix) and the extended definitions from step 3, and finally

5. Estimation of semantic relatedness for a concept-pair (pair of input terms).

3 Experimental Data

3 .1 Gene Ontology and GO Annotations

GO comprises three GOs which express different biological attributes: *biological process* (BP) for processes such as metabolism or cell proliferation; *cellular component* (CC) such as the nucleus or cell membrane; and *molecular function* (MF) such as catalytic or binding activities. GO is maintained and constantly updated by a group of curators[1].

A GO annotation consists of a GO term associated with a specific reference and an evidence code to indicate how a given annotation is supported. Out of all the evidence codes available, Inferred from Electronic Annotation (IEA) is not assigned by a curator and is thus the least reliable so we treat them separately. GO and the required GO annotations were downloaded from the GO website (http://geneontology.org November 2, 2015).

3 .2 MEDLINE Abstracts

MEDLINE (https://mbr.nlm.nih.gov/Download/) contains over 20 million citations of biomedical articles from 1966 to the present. The database includes journal articles from medicine, pharmacy, dentistry, nursing, healthcare and covers the literature in biology and biochemistry. For this study, we used MEDLINE 2013 as the corpus to build a first-order word-word co-occurrence matrix for the later computation of second-order co-occurrence (SOC) matrices which are used by simDEF.

3 .3 Validation Datasets

Sequence homology

We used bitscores from the Basic Local Alignment Search Tool (BLAST) [4] to create our sequence homology dataset. In the first step, we performed an all-versus-all comparison of proteins in the yeast *Saccharomyces cerevisiae* database [26] with an expectation-value threshold of 0.1. The *e*-value is the number of expected hits of

[1]The Gene Ontology Consortium (GOC) integrates resources from a variety of research groups, from model organisms to protein databases to the biological research communities actively involved in the development and implementation of the Gene Ontology. For more information refer to http://geneontology.org/docs/go-consortium/

similar quality (score) that could be found just by chance. *e*-value of 10 means that up to 10 hits can be expected to be found just by chance, given the same size of a random database. *e*-value can be used as a first quality filter for the BLAST search result, to obtain only results equal to or better than the number given by the *e*-value option. Although this threshold is liberal, the corresponding bitscores associated with *e*-values near this threshold will be very low and have a minimal effect on our analysis.

The bitscore is the required size of a sequence database in which the current match could be found just by chance. The bitscore is a log_2 scaled and normalized raw-score (it is non-symmetric). Each increase by *1* doubles the required database size ($2^{bit-score}$). Unlike *e*-value, bitscore does not depend on database size; and since the bitscore gives the same value for hits in databases of different sizes it can be used for searching in an constantly increasing database. As a general rule of thumb, the higher the bitscore, the better the sequence similarity. Since a bitscore for query and subject proteins is not symmetrical, we calculate log-reciprocal BLAST score (LRBS) and relative reciprocal BLAST score (RRBS) to express the general sequence similarity of protein pairs. For proteins A and B, the LRBS and RRBS are:

$$LRBS(A,B) = log\Big(\frac{Bitscore(A,B) + Bitscore(B,A)}{2}\Big) \qquad (3.1)$$

$$RRBS(A,B) = \frac{Bitscore(A,B) + Bitscore(B,A)}{Bitscore(A,A) + Bitscore(B,B)} \qquad (3.2)$$

Finally, after LRBS and RRBS computations, we have a dataset of 20,167 protein pairs from the yeast *S.cerevisiae* database along with their LRBS and RRBS sequence similarity scores. All proteins in the dataset have their own GO annotations from the CC, BP and MF ontologies without considering IEAs.

Gene expression

The gene expression dataset comes from the study by Jain and Bader [77]. In their study, the gene-expression dataset for *S.cerevisiae* was downloaded from GeneMANIA [226] and other microarray experiments. The authors prepared test datasets of 5000 *S.cerevisiae* gene pairs randomly selected from a list of all possible pairs of proteins

in their gene expression dataset. This was done independently for CC, BP and MF annotations of gene products. Since in our experiments we mainly consider genes with non-electronic annotations (without IEAs), we used 4800 fitting gene pairs from their study.

Protein-protein interaction

For the PPI experiment, we employed subsets of the yeast PPI dataset from Wu et al. [229]. In that study, for each GO, independent gold-standard positive datasets for yeast were built from a core subset of the Database of Interacting Proteins (DIP) [189]. Negative datasets were independently generated by randomly choosing annotated protein pairs in BP, CC and MF, which are absent from a combined dataset of all possible PPIs. Since for different GOs the numbers of generated PPI pairs are different and more importantly many of them do not have GO annotations after excluding IEA, we selected subsets of 3000 positive and 3000 negative PPIs for each ontology from that study to evaluate our measure against other similarity measures in a PPI prediction task.

4 Methods

4 .1 Method Definition

Pointwise Mutual Information (PMI) is a measure of association used in information theory. In computational linguistics, the PMI for two given words indicates the likelihood of finding one word in a text document that includes the other word. PMI is formulated as:

$$\mathrm{PMI}\left(w_1, w_2\right) = \log \frac{p\left(w_1, w_2\right)}{p\left(w_1\right) \times p\left(w_2\right)} \tag{4 .1}$$

where $p(w_1, w_2)$ is the probability that words w_1 and w_2 co-occur in a document, and $p(w_1)$ and $p(w_2)$ for w_1 and w_2, respectively, are the marginal probabilities of their occurrence in a document. It is expected that rare words are highly associated with and descriptive of each other, yet due to their sparse nature their bigram frequency

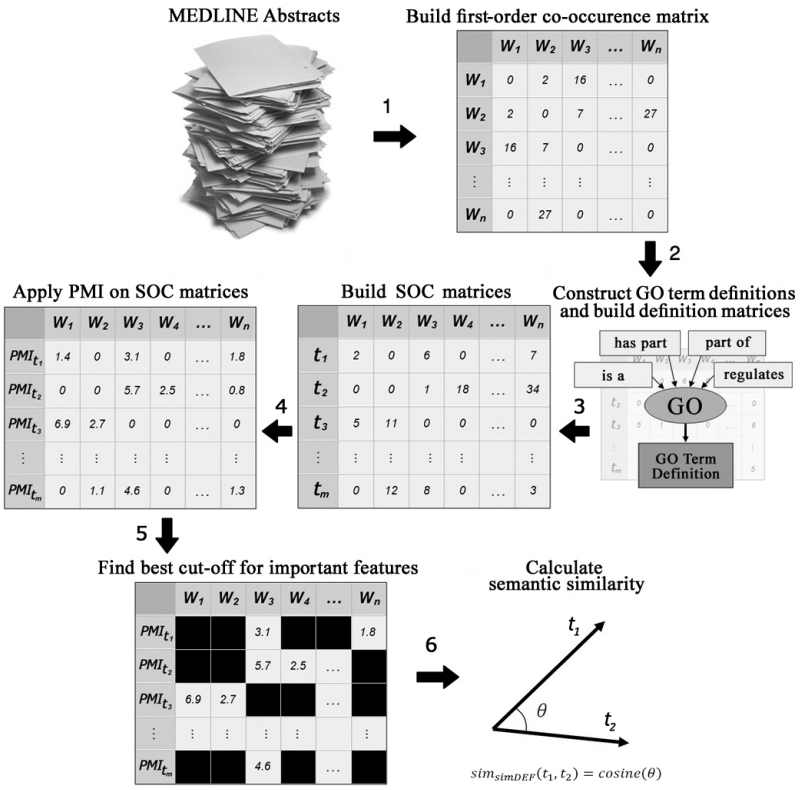

Figure 2.1: Computation of the simDEF semantic similarity measure

(i.e., number of times they have been seen next to each other) is small in the corpus. This is the main drawback of the Gloss Vector measure in selection of the best descriptive features.

We use PMI in our proposed measure, simDEF, for statistical elimination of insignificant features (words). simDEF requires procedures for building the co-occurrence matrix from a proper text corpus, constructing extended definitions for GO terms using GO term definitions, and finding words that are appropriate descriptors of that GO term. simDEF comprises six steps (Figure 2.1).

Step 1 – counting bigrams and building the first-order co-occurrence matrix. After discarding punctuation, changing all characters to lowercase, and removing stop words (a pre-defined list of 204 non-informative words like *a* and *the*) from the MEDLINE corpus, a list of bigrams and their frequencies for all the content words is constructed. A window size of 2 is used for extraction of bigrams. This window size controls how close two words can appear in bigrams. Stemming was found

to reduce accuracy and was not adopted in simDEF. Then, by ignoring the order of occurrence in a bigram, we transform it from a bigram list to a co-occurrence list. Finally, we construct the first-order co-occurrence matrix, which is symmetric and sparse and represents the contextual information of MEDLINE words. Cell values in the first-order matrix represent how many times the word associated with its row is seen in this corpus alongside the word associated with its column.

Step 2 – definition construction of GO terms and then building BP, CC, and MF definition matrices. In this step, we construct an extended definition for every term in GO. From the theoretical perspective, definition extension of parent GO terms (i.e., broader concepts) with their children's definitions (i.e., narrower and more specific definitions) adds more specific information. Although child GO terms may contain contradictory information, this information may nonetheless provide essential context when calculating functional similarity with other genes (which may in turn be augmented with conflicting information). From the practical perspective, we examined all the combinations of definition extension considering GO relationships such as *is_a*, *has_part*, *part_of*, *regulates*, *siblings* and *synonyms*. What is represented in Figure 2.1 yielded the best results in our experiments conducted in this study. Improvement in the results using relationships such as *part_of* and *regulates* indicates that besides the similarity, simDEF accounts for relatedness as well. See Subsection 4 .2 for more in-depth explanation of why definition extension can be beneficial.

Each GO term has an identifier, a representative name, a GO definition, a namespace defining the sub-ontology of the GO term and other information such as its relationship to the other GO terms. For example, GO:0001104 has the representative name '*RNA polymerase II transcription cofactor activity*' and belongs to the MF hierarchy. This GO term has the definition '*Interacting selectively and non-covalently with an RNA polymerase II (RNAP II) regulatory transcription factor and also with the RNAP II basal transcription machinery in order to modulate transcription. Cofactors generally do not bind DNA, but rather mediate PPIs between regulatory transcription factors and the basal RNAP II transcription machinery.*' In order to make this definition even richer we concatenate definitions of its direct parents (i.e. GO:0003712 or '*transcription cofactor activity*' and GO:0001076 or '*RNA polymerase II transcription*

factor binding transcription factor activity') and direct children (i.e., GO:0001105 or '*RNA polymerase II transcription co-activator activity*' and GO:0001106 or '*RNA polymerase II transcription co-repressor activity*') to its definition. We also add this GO term's representative name to this extended definition considering this name as part of its own definition. This process is done for all GO terms in BP, CC and MF. Now we see that for each word in the definition of a GO term we have an associated first-order co-occurrence vector calculated in Step 1. After changing all characters to lowercase and removing punctuation and stop words from these extended definitions we store them in different matrices for three different sub-ontologies. In these matrices, the value of a cell represents how many times the word associated with its column appears in the definition of the GO term associated with its row.

Step 3 – building Second-Order Co-occurrence (SOC) matrices. To build the SOC vector for a GO term we sum the first-order co-occurrence vectors from the words in the constructed definition of that GO term (i.e., compute the centroid), and then normalize the result vector by the number of words in the definition. We do this process separately for each GO. The results are three different matrices for BP, CC and MF; each row again represents a GO term and features are the words. We have three SOC matrices for BP, CC and MF at the end of this step.

Step 4 – PMI on SOC matrices (PMI-on-SOC matrices). In our similarity measure, PMI-on-SOC matrices replace a conventional approach of low- and high-frequency cut-offs for detection of insignificant features or words in the Gloss Vector measure. We statistically measure the level of association between GO terms and their describing features in the SOC matrices and then apply a cut-off threshold on this level in the next step. Following the equation 4 .1, here, PMI(t_i, w_j) measures the level of association between GO term i and feature j to discover how descriptive the word j of that GO term is. PMI is biased toward low-frequency words (due to *logarithm* utilized in the formula for the measurement of dependency) and consequently tends to favour them by assigning them a higher degree of importance [155]; in order to resolve this weakness, we employ the add-one technique. Before applying PMI on a matrix, all the elements of the matrix are incremented by 1 unit.

Step 5 – removing insignificant features from the PMI-on-SOC vectors. Defining a PMI threshold allows us to skip those words which provide low information for GO terms in their constructed PMI-on-SOC vectors. By using the available dataset in an iterative way, we gradually increase the threshold of PMI cut-off from zero and then evaluate the results generated by simDEF. Depending on the biomedical task, for a chosen cut-off threshold, criteria such as Pearson's correlation (see Subsection 5 .1 and Subsection 5 .2) or AUC (see Subsection 5 .3) can be used for the performance evaluation of the estimated similarity results. In general, as cut-off thresholds increase we tend to get better results until a point where performance starts to drop rapidly. Therefore, by recording this curve for different performance results and cut-off points we try to find the optimal cut-off point in order to keep only those informative features describing one GO term. Also, to avoid this interval being sensitive to the choice of dataset, we use 5-fold cross validation to predict the extent to which the threshold will generalize to an independent dataset. This cut-off selection is done separately for the three constructed PMI-on-SOC matrices of the BP, CC and MF ontologies.

Step 6 – calculating semantic similarity. In this final step, the semantic similarity among GO terms is estimated. The cosine of the angle between optimized PMI-on-SOC vectors of two GO terms will indicate the degree of similarity for those terms. For the final usage of the measure, the last produced matrix is loaded into memory and used for measuring similarity between GO terms. In these matrices, each row stores the calculated optimized definition vector of its associated GO term.

As, in most cases, gene products are annotated with more than one GO term in the same ontology hierarchy (BP, CC or MF), there are several methods to measure the functional similarity of gene products based on the semantic similarity of these GO terms. MAX and AVE define functional similarity between gene products as the maximum or average semantic similarity values, respectively, over the GO terms annotating the genes. MAX has been shown to be more useful for a PPI task [230]. If T_A and T_B are the sets of GO terms which annotate proteins A and B, respectively,

the MAX for their functional similarity measurement is achieved by:

$$\mathrm{sim}_{\mathrm{MAX}}(A,B) = \mathrm{MAX}_{t_1 \in T_A, t_2 \in T_B}\left(\mathrm{sim}\left(t_1,t_2\right)\right) \tag{4 .2}$$

Azuaje et al. [10] developed the best-match average (BMA) method, in which each term of the first protein is paired only with the most similar term of the second one and vice versa. The best-match average (BMA) method is found to be the best for evaluation of semantic similarity measures and the correlation of its results with sequence homology and gene expression data [172]. BMA for two gene products A and B with n and m GO annotations is given by:

$$\mathrm{sim}_{\mathrm{BMA}}(A,B) = \frac{1}{2}\left(\frac{1}{n}\sum_{t_1 \in T_A}\mathrm{MAX}_{t_1,t_2 \in T_B}\left(\mathrm{sim}\left(t_1,t_2\right)\right) + \frac{1}{m}\sum_{t_2 \in T_B}\mathrm{MAX}_{t_1 \in T_A,t_2}\left(\mathrm{sim}\left(t_1,t_2\right)\right)\right) \tag{4 .3}$$

Consider that in these formulae T_A of different ontologies would be different (likewise for T_B). Therefore, we will achieve three different protein functional similarity values for three different gene ontologies.

MAX and BMA measure similarity between two gene products by combining semantic similarities between their terms. Semantic similarity estimation was used to evaluate the Resnik [184], Lin [110], Jiang [78], GraSM [37], Wang [221], AIC [200] and simDEF measures (see Subsection 2 .1 for their definitions and formulas). In contrast, groupwise measures like simGIC [171] and simUI [49] are functional similarity measures by nature and do not rely on combining similarities between individual terms to assess gene product similarity, but calculate it directly by their annotation sets. By employing GO annotations for the previous measures and MEDLINE for the simDEF as the needed corpora, we implemented these measures as appropriate, and reported results alongside the best cut-off point for feature removal in each task.

4 .2 On the Importance of Definition Extension

Here, we provide an illustrative example in order to demonstrate the valuable benefits that will come through extending definition of one GO term with the definitions of its directly related GO terms.

Consider the provided information in Figure 2.2 (a). This information is directly extracted from the GO for the two parent GO terms (broader concepts) GO:0051917 and GO:0001910 and their children (more specific concepts). Take into account that for each of these parent GO terms, their children provide contradictory definitions.

We added the "Keywords" field among the provided information since they tried to capture the short yet concise message of their definitions just to illustrate the impact of definition extension. Considering the presented keywords, our vector space consists of 6 features: *modulate*, *stop*, *activate*, *frequency*, *fibrinolysis*, *LMC*. Therefore, without definition extension, the vector representation of GO terms will be what is represented in 2.2 (b) (to keep everything simple we work only with binary values here).

Since the feature *frequency* is 1 everywhere and provides no unique information we can skip it. The result matrix would be what is shown in Figure 2.2 (c).

Now, in different scenarios, let's compute some cosine similarities from this result matrix and compare it with our intuition of similarity.

1. Comparing similarity of the parent GO:0051917 with its children and also siblings together:

   `sim(GO:0051917, GO:0051918) = 0.50000`

   `sim(GO:0051917, GO:0051919) = 0.50000`

   `sim(GO:0051918, GO:0051919) = 0.50000`

   Problem: Intuitively, we expect to see less similarity between the child concepts (siblings) because there are more specific terms compared to their parent and therefore should share less information. But it is not captured here.

2. Comparing two different branches:

   `sim(GO:0051917, GO:0001911) = 0`

   `sim(GO:0051917, GO:0001912) = 0`

   Problem: From the natural language perspective we are aware that if one GO term can modulate something, it is capable to stop or activate it. Therefore, we need to capture this similarity information between two GO terms which are characterized by these features in order to distinguish them from the other

a)

id: GO:0051917 (***the first parent GO term***)
name: regulation of fibrinolysis
namespace: biological process
definition: "Any process that modulates the frequency, rate or extent of fibrinolysis, an ongoing process that solubilizes fibrin, resulting in the removal of small blood clots."
Keywords: modulate, frequency, fibrinolysis

---

id: GO:0051918
name: negative regulation of fibrinolysis
namespace: biological process
definition: "Any process that stops, prevents, or reduces the frequency, rate or extent of fibrinolysis, an ongoing process that solubilizes fibrin, resulting in the removal of small blood clots."
is_a: GO: 0051917
Keywords: stop, frequency, fibrinolysis

---

id: GO:0051919
name: positive regulation of fibrinolysis
namespace: biological process
definition: "Any process that activates, maintains or increases the frequency, rate or extent of fibrinolysis, an ongoing process that solubilizes fibrin, resulting in the removal of small blood clots."
is_a: GO: 0051917
Keywords: activate, frequency, fibrinolysis

---

id: GO:0001910 (***the second parent GO term***)
name: regulation of leukocyte mediated cytotoxicity
namespace: biological process
definition: "Any process that modulates the frequency, rate, or extent of leukocyte mediated cytotoxicity."
Keywords: activate, frequency, LMC

---

id: GO:0001911
name: negative regulation of leukocyte mediated cytotoxicity
namespace: biological process
definition: "Any process that stops, prevents, or reduces the rate of leukocyte mediated cytotoxicity."
is_a: GO:0001910
Keywords: stop, frequency, LMC

---

id: GO:0001912
name: positive regulation of leukocyte mediated cytotoxicity
namespace: biological process
definition: "Any process that activates or increases the frequency, rate or extent of leukocyte mediated cytotoxicity."
is_a: GO:0001910
Keywords: stop, frequency, LMC

b)

	modulate	stop	activate	frequency	fibrinolysis	LMC
GO:0051917 (PAR)	*1*	*0*	*0*	*1*	*1*	*0*
GO:0051918 (CHD)	*0*	*1*	*0*	*1*	*1*	*0*
GO:0051919 (CHD)	*0*	*0*	*1*	*1*	*1*	*0*
GO:0001910 (PAR)	*1*	*0*	*0*	*1*	*0*	*1*
GO:0001911 (CHD)	*0*	*1*	*0*	*1*	*0*	*1*
GO:0001912 (CHD)	*0*	*0*	*1*	*1*	*0*	*1*

c)

	modulate	stop	activate	fibrinolysis	LMC
GO:0051917 (PAR)	*1*	*0*	*0*	*1*	*0*
GO:0051918 (CHD)	*0*	*1*	*0*	*1*	*0*
GO:0051919 (CHD)	*0*	*0*	*1*	*1*	*0*
GO:0001910 (PAR)	*1*	*0*	*0*	*0*	*1*
GO:0001911 (CHD)	*0*	*1*	*0*	*0*	*1*
GO:0001912 (CHD)	*0*	*0*	*1*	*0*	*1*

d)

	modulate	stop	activate	fibrinolysis	LMC
GO:0051917 (PAR)	*1*	*1*	*1*	*1*	*0*
GO:0051918 (CHD)	*1*	*1*	*0*	*1*	*0*
GO:0051919 (CHD)	*1*	*0*	*1*	*1*	*0*
GO:0001910 (PAR)	*1*	*1*	*1*	*0*	*1*
GO:0001911 (CHD)	*1*	*1*	*0*	*0*	*1*
GO:0001912 (CHD)	*1*	*0*	*1*	*0*	*1*

Figure 2.2: A Piece of Information from the GO and the Definition Matrices for it

completely irrelevant GO terms. But the result achieved here cannot find this semantic connection between the terms modulate, stop and activate. This issue is not taken into account in the previous problem either (i.e. fibrinolysis is the only feature that relates those GO terms)

3. Discovering relatedness of two GO terms (not necessarily similarity measurement)[2]:

   `sim(GO:0051918, GO:0001912) = 0`

   Problem: One of the drawbacks of existing semantic similarity measures is that they just account for similarity and not relatedness. We believe one of the advantages of simDEF over those measures is its ability to discover these sorts of relationships and treat them differently. For example here, we expect to see some, even though small, degree of relatedness between two GO terms characterized by stop and activate. Due to the poor definitions of GO terms in this example we could not address this goal.

Now, let's extend the GO term definitions by adding their direct children/parents definitions to them. Following this rule, keywords for different GO term will be (again, for simplicity we do not consider frequency of the words in the definitions here):

Keywords (GO:0051917): *modulate*, *stop*, *activate*, *frequency*, *fibrinolysis*
Keywords (GO:0051918): *modulate*, *stop*, *frequency*, *fibrinolysis*
Keywords (GO:0051919): *modulate*, *activate*, *frequency*, *fibrinolysis*
Keywords (GO:0001910): *modulate*, *stop*, *activate*, *frequency*, *LMC*
Keywords (GO:0001911): *modulate*, *stop*, *frequency*, *LMC*
Keywords (GO:0001912): *modulate*, *activate*, *frequency*, *LMC*

Now, the matrix presented in Figure 2.2 (b) will change to what we have in Figure 2.2 (c). By revisiting the problematic scenarios discussed above we will examine if the consideration of definition extension addresses those problems.

[2]The term semantic similarity is often confused with semantic relatedness. Semantic relatedness includes any relation between two terms, while semantic similarity only includes "is a" relations.

1. Comparing similarity of the parent GO:0051917 with its children and also siblings together:

   `sim(GO:0051917, GO:0051918) = 0.86603`

   `sim(GO:0051917, GO:0051919) = 0.86603`

   `sim(GO:0051918, GO:0051919) = 0.66667`

   Solution: We observe that definition extension addressed our first intuition as we expect to achieve less similarity between two siblings than their similarities with their parent.

2. Comparing two different branches:

   `sim(GO:0051917, GO:0001911) = 0.57735`

   `sim(GO:0051917, GO:0001912) = 0.57735`

   Solution: We observe that definition extension addressed our understanding of human language

3. Discovering relatedness of two GO terms (not necessarily similarity measurement):

   `sim(GO:0051918, GO:0001912) = 0.33333`

   Solution: Definition extension can help us better to measure the degree of relatedness between two GO terms as well.

5 Results

5 .1 Correlation with Sequence Similarity

Several authors have compared the performance of different semantic similarity measures by testing how well these measures correlate with sequence similarity. Various studies [117] showed that the more similar two sequences are the more similar their ontological annotations will be.

To evaluate the semantic similarity measures, we used two distinct sequence similarity measures: LRBS and RRBS with the formulae of (1) and (2). LRBS is similar

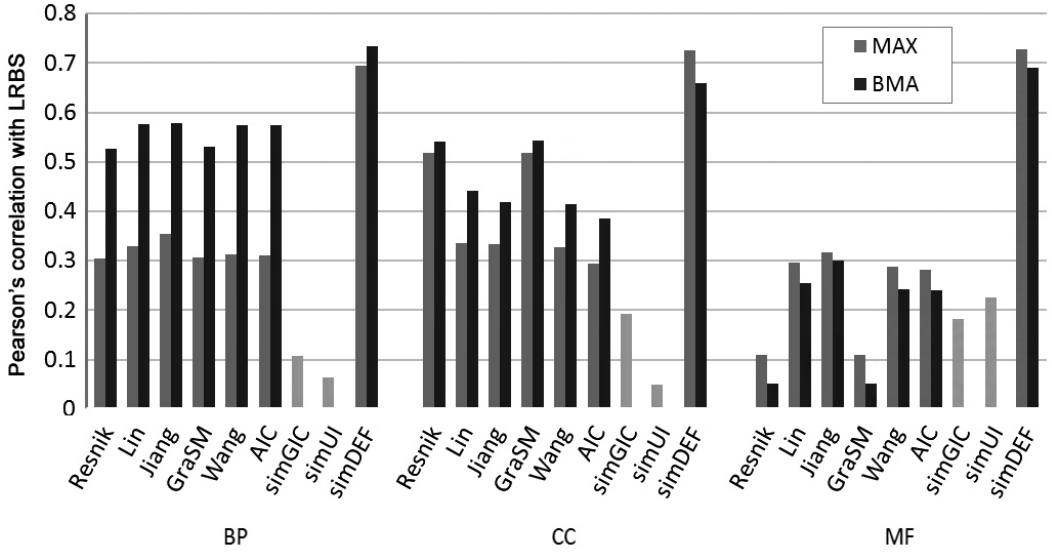

Figure 2.3: Pearsons correlation between semantic measures and LRBS (IEA–)

to the sequence similarity measure used previously by Lord, but compensates for the fact that BLAST scores are not symmetric. RRBS, suggested by Joshi and Xu [85], is another indicator of functional similarity acting like the sequence identity percentage by taking amino acid substitutions into account. Figure 2.3 shows the degree of correlation between LRBS and the functional similarity estimations calculated by semantic measures of 20,167 protein pairs (without IEAs included).

In all cases, whether we use MAX or BMA, simDEF correlates with sequence similarity better than the other IC-based measures. The high correlation between simDEF and LRBS in the MF ontology is notable as it is more than the twice of the second best measure's result (Jiang). Table 2.1 shows the exact numerical results of this experiment (with and without IEAs).

The other metric used for sequence similarity measurement is RRBS which is not directly affected by sequence length (unlike LRBS). We assessed whether the dependency on sequence length affects the outcome of the evaluation. Figure 2.4 shows the degree of correlation between the similarity estimations calculated by semantic measures and RRBS. RRBS, like LRBS, shows the highest degree of correlation with simDEF among the similarity measures.

In general, measures of functional similarity correlate better with LRBS sequence similarity than RRBS. We also observe among IC-based measures tested here that

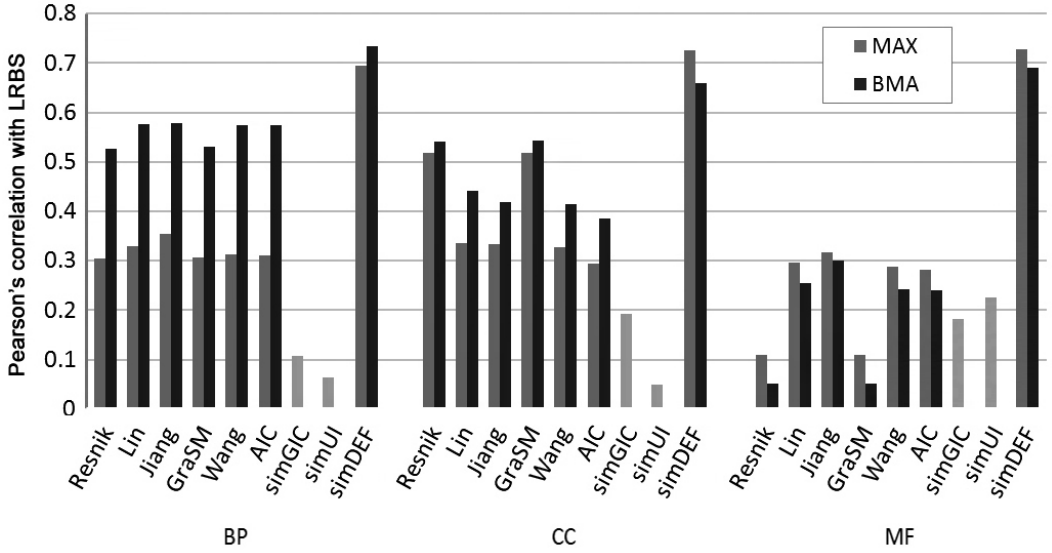

Figure 2.4: Pearson's correlation between semantic measures and RRBS (IEA–)

no single measure is superior to all others in the BP, CC and MF ontologies, which suggests task-dependency of these measures. AIC, the latest variant of IC-based measures, does not offer any improvement over the earlier measures. The Wang topological measure of similarity works only slightly better than the IC-based measures in the RRBS sequence similarity comparison of BP. The correlation results for LRBS and RRBS also demonstrate that BMA is the appropriate metric for functional similarity measurement of proteins from BP and CC points of view when we use IC-based measures while for simDEF in CC it is reverse. The difference between results generated by BMA and MAX for simDEF is typically small, whereas other pairwise semantic similarity measures tend to show larger discrepancies. Table 2.2 shows the exact results of this experiment.

Regarding the results for correlation of semantic similarity measures with LRBS and RRBS scores we have the exact results shown in Table 2.1 (in each column, the boldface numbers are the highest and the underscored numbers are the second best results):

5 .2 Correlation with Gene Expression

Correlation with gene expression is another desirable criterion [161]. Since genes involved in the same process tend to exhibit similar expression patterns, we could

Table 2.1: Pearson's correlation of semantic similarity measures for three GO ontologies against sequence similarity (LRBS and RRBS) without IEA (IEA−)

Semantic Measure		**LRBS** BP	**LRBS** CC	**LRBS** MF	**RRBS** BP	**RRBS** CC	**RRBS** MF
Resnik [184]	MAX	0.2523	0.4906	0.0441	0.3043	0.5165	0.1117
	BMA	0.3997	0.4554	0.0099	0.5272	0.5399	0.0508
Lin [110]	MAX	0.1851	0.2379	0.1595	0.3301	0.3363	0.2958
	BMA	0.4223	0.4112	0.1486	0.5749	0.4413	0.2558
Jiang [78]	MAX	0.2394	0.2495	0.1852	0.3539	0.3327	0.3181
	BMA	0.4852	0.4423	0.1997	0.5781	0.4165	0.2998
GraSM [37]	MAX	0.2573	0.4938	0.0443	0.3069	0.5188	0.1107
	BMA	0.4035	0.4611	0.0098	0.5301	0.5438	0.0518
Wang [221]	MAX	0.1874	0.2476	0.1545	0.3255	0.3309	0.2822
	BMA	0.4292	0.4398	0.1462	0.5743	0.4165	0.2426
AIC [200]	MAX	0.1654	0.1961	0.1631	0.3107	0.2961	0.2802
	BMA	0.4181	0.3436	0.1478	0.5725	0.3856	0.2413
simGIC [171]		0.415	0.2108	0.1447	0.1064	0.1938	0.1833
simUI [49]		0.1793	0.3883	0.2874	0.0612	0.0502	0.2266
simDEF [153]	MAX	0.5971	**0.7272**	0.2374	0.6943	**0.7263**	**0.7272**
	BMA	**0.6454**	0.6912	**0.5366**	**0.7341**	0.6585	0.6892

Table 2.2: Pearson's correlation of semantic similarity measures for three GO ontologies against sequence similarity (RRBS and RRBS) with IEA (IEA+)

Semantic Measure		**LRBS** BP	**LRBS** CC	**LRBS** MF	**RRBS** BP	**RRBS** CC	**RRBS** MF
Resnik [184]	MAX	0.1868	0.3929	-0.019	0.1965	0.4055	0.0403
	BMA	0.4278	0.3939	-0.004	0.4129	0.4709	0.0215
Lin [110]	MAX	0.1449	0.1579	0.0505	0.2102	0.2404	0.1724
	BMA	0.4842	0.3486	0.1213	0.4531	0.3803	0.2102
Jiang [78]	MAX	0.2137	0.2112	0.1578	0.3167	0.2886	0.2997
	BMA	0.5691	0.4259	0.2431	0.5585	0.4252	0.3508
GraSM [37]	MAX	0.1887	0.3938	-0.019	0.1976	0.4103	0.0403
	BMA	0.4314	0.3966	-0.004	0.4166	0.4741	0.0216
Wang [221]	MAX	0.2187	0.1871	0.0904	0.3217	0.2412	0.1609
	BMA	0.5741	0.3623	0.1201	0.5688	0.3912	0.1998
AIC [200]	MAX	0.1232	0.1207	0.0459	0.1888	0.1935	0.1531
	BMA	0.4757	0.2752	0.1133	0.4501	0.3123	0.1907
simGIC [171]		0.1893	0.1909	0.1138	0.0803	0.1612	0.1603
simUI [49]		0.1795	0.3213	0.3105	0.1553	0.0398	0.3301
simDEF [153]	MAX	0.5292	**0.5924**	0.1001	0.5801	**0.5907**	0.6273
	BMA	**0.6517**	0.5821	**0.5103**	**0.6107**	0.5685	**0.6402**

expect good semantic similarity estimations calculated on the BP ontology to be correlated with the expression similarity (Yang et al., [234]). For our experiments, the evaluation is done against the available standard reference of 4800 gene expression values. Here, we report Pearson's correlation between gene expression data and the results from simGIC, simUI and BMA of pairwise measures. We focus on the BMA criterion as it always gave higher correlations. Pearson's correlation between gene expression and semantic measures for CC, BP and MF ontologies with and without IEAs considered are shown in Table 2.3[3].

Table 2.3: Pearson's correlation of semantic measures for three GOs using BMA against gene expression data (IEA+ and IEA–)

Semantic measure	**Including IEA**			**Excluding IEA**		
	BP	**CC**	**MF**	**BP**	**CC**	**MF**
Resnik	0.2659	<u>0.4562</u>	<u>0.2514</u>	0.2593	<u>0.4426</u>	<u>0.2231</u>
Lin	0.2541	0.3864	0.2155	0.2567	0.3842	0.2075
Jiang	0.2022	0.3217	0.1566	0.1757	0.2845	0.1708
GraSM	<u>0.2677</u>	0.4542	**0.2516**	<u>0.2624</u>	0.4395	**0.2252**
Wang	0.1911	0.3013	0.1306	0.1638	0.2805	0.1672
AIC	0.2466	0.3735	0.2149	0.2439	0.3593	0.2078
simGIC	0.0812	0.1542	0.1204	0.0667	0.1328	0.1422
simUI	0.1272	0.2418	0.0654	0.0628	0.0773	0.0455
simDEF	**0.3098**	**0.4649**	0.2325	**0.3071**	**0.4559**	0.2166

The highest correlations in all cases are seen with the CC ontology, followed by BP and MF. Although the difference in correlation coefficients is not as striking as in the homology example, simDEF outperforms the next best method, GraSM, by 4% on the BP ontology and 12% on the CC ontology. GraSM has the best correlation for MF, 12% better than simDEF, which was also outperformed by the Resnik. Correlation coefficients were generally higher for datasets with IEAs, suggesting that electronic annotations have some value when investigating gene-expression profiles.

Wang et al. [220] and Sevilla et al. [196] showed that the correlation between gene expression and semantic similarity was negligible when semantic similarity values were low, but the two measures were highly related when semantic similarity was high. Xu et al. [233] further showed a linear relationship for gene pairs with high levels of

[3]In each column, the boldface numbers are the highest and the underscored numbers are the second best results

expression correlation. We examined this trend by comparing Resnik against simDEF for variable numbers of the highest correlated genes. For this purpose, after sorting gene expression data from the highest to the lowest values, we measured correlation of these data with Resnik and simDEF as we go from the top correlated expressions to the bottom. Figure 2.5 demonstrates the trend of change for this test.

Considering other studies' findings and our result demonstrated in Figure 2.5, we see that by being more focused on highly correlated gene expression pairs the overall correlation between functional similarity and gene expression increase only when we take the BP ontology into account. For CC and MF the reverse is true. The other important point learned for BP is that by employing simDEF as semantic measure, when we ignore electronic annotation we get better correlation with highly-correlated gene expression data while this is not true for Resnik. Moreover, we observe that for BP and CC simDEF works better than Resnik no matter which subset we consider. Nevertheless, this issue does not hold for MF and we only get better results from simDEF when we focus on higher correlated genes in terms of their expression.

5 .3 Comparison with PPIs

Semantic similarity can also be used as an indicator for the plausibility of putative PPIs, as proteins that interact in the cell in vivo are expected to participate in similar cellular locations and BPs. Like other studies (Jain and Bader [77]; Wu et al. [229]), we formulated this as a classification problem and checked how well the different semantic similarity measures perform for predicting true PPIs. For this purpose, the MAX and BMA results are directly interpreted as the classification probability of 'Interaction' and 'Not Interaction'. The higher this value is, the higher the probability of interaction will be. We applied this approach to a dataset of 6000 PPI pairs for each GO while half of the data have positive labels (due to experimentally confirmed PPIs) and the other half have negative labels.

In our evaluation, the results of prediction were investigated by receiver operating characteristic (ROC) curves, with area under the curve (AUC) as the main accuracy criterion. Here, we report only MAX since, as we expected from the previous studies, for all the cases MAX predicted better results compared with BMA. Table 2.4 shows the values of AUC for different semantic measures, including a hybrid measure that

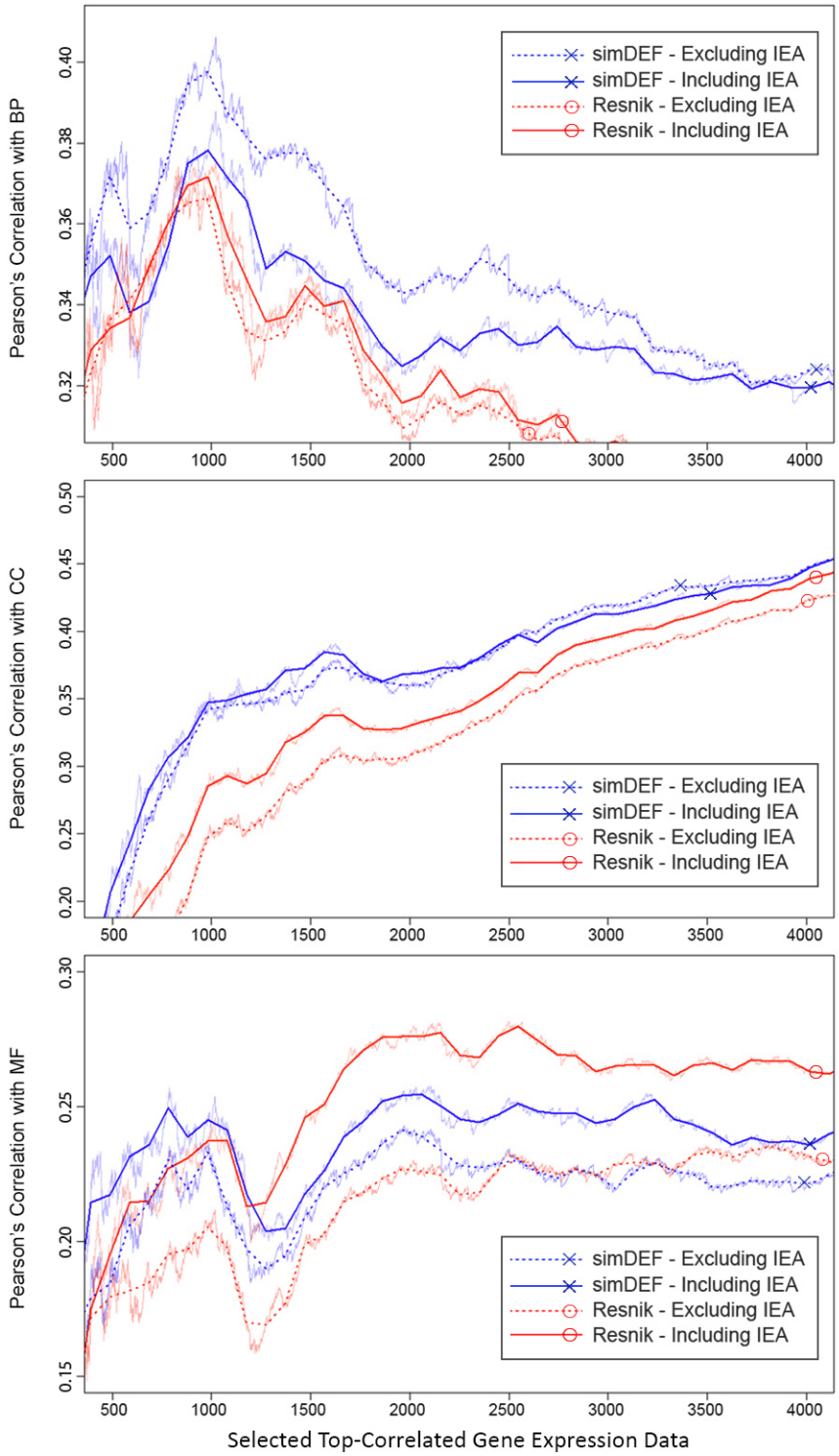

Figure 2.5: Relationship of gene expression correlation and semantic similarity in three GO ontologies.
X = 500 means that only the 500 most highly correlated gene pairs were considered when generating the correlation scores

uses the average of the simDEF and Resnik values as the probability of interaction.

Table 2.4: AUC of the semantic similarity measures for three GOs using MAX in the PPI task on the yeast dataset (IEA+ and IEA–)

Semantic measure	**Including IEA**			**Excluding IEA**		
	BP	**CC**	**MF**	**BP**	**CC**	**MF**
Resnik	0.8961	**0.8658**	0.7969	0.8685	**0.8525**	0.7429
Lin	0.8856	0.7588	0.7814	0.8629	0.7805	0.7419
Jiang	0.8719	0.7555	0.7613	0.8541	0.7467	0.7621
GraSM	0.8965	**0.8658**	0.7969	0.8691	0.8488	0.7413
Wang	0.8687	0.7835	0.7612	0.8483	0.7507	0.7496
AIC	0.8812	0.7623	0.7802	0.8613	0.7727	0.7427
simGIC	0.8014	0.8003	0.7025	0.7415	0.7673	0.6634
simUI	0.7999	0.7364	0.6921	0.7413	0.7098	0.6705
simDEF	**0.9086**	0.7742	**0.8202**	**0.9059**	0.8001	**0.8115**
simDEF + Res	**0.9264**	**0.8809**	**0.8306**	0.9039	**0.8564**	0.8073

As in the gene expression case, we found that including IEA records from GO improved the accuracy (in this case, the AUC). simDEF gave the highest accuracy when the BP and MF ontologies were used, while Resnik and GraSM performed best for CC. The hybrid classifiers AUC results are represented in the last row of Table 2.4. This result shows that simDEF is useful on all three ontologies, whether alone or as a complement to the Resnik measure. We believe different approaches of simDEF and IC-based semantic measures in similarity estimation is the main reason for this improvement. With consideration of IEA, the ROC for CC ontology shown in Figure 2.6 represents that the combination of Resnik and simDEF benefits from the results of simDEF and Resnik both.

ROC is not always the only best approach to evaluate a classifier's performance in a PPI task (Jain and Bader [77]; Wu et al. [229]). Therefore, in our second experiment, by keeping the feature cut-off point of simDEF as before, considering Resnik as the baseline measure, and including IEA in the evaluation, we calculated different F1-scores for different classification cut-off points in the simDEF, Resnik and hybrid measures. Then, we compared the calculated mean and maximum F1-score values. While the mean and maximum F1-scores can be indicators of one classifiers performance in the detection of positive interactions (similar to AUC), maximum F1-score also helps in selection of the best classification cut-off point of a classifier

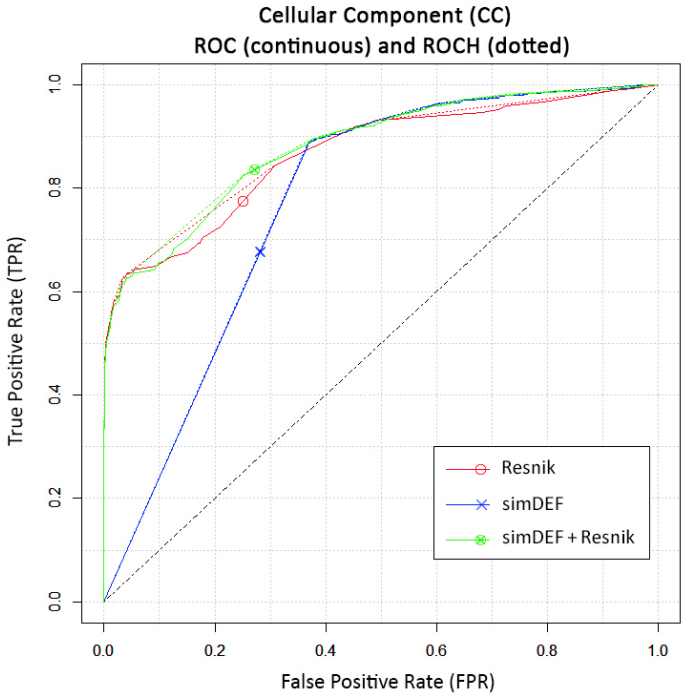

Figure 2.6: ROC evaluation of the simDEF, Resnik and the hybrid measure of them by MAX for the PPI task at different classification cut-offs based on the yeast dataset using CC ontology (IEA+)

having its ROC curve. The mean and maximum F1-score results are shown in Table 2.5.

Table 2.5: F1-score of the simDEF, Resnik and the hybrid measure by MAX for the PPI task (IEA+)

Semantic measure	**Mean of F1-score**			**Max of F1-score**		
	BP	**CC**	**MF**	**BP**	**CC**	**MF**
Resnik	0.5973	0.5719	0.4699	0.8416	0.7815	0.7264
simDEF	**0.8154**	**0.7591**	**0.7084**	**0.8483**	**0.7889**	**0.7521**
simDEF + Res	0.6318	0.6686	0.5921	**0.8519**	**0.7962**	**0.7546**

The simDEF prediction of PPIs based on the F1-score is always better than the results achieved by Resnik. Even though Resnik gave the best AUC for the CC ontology, the simDEF mean F1-score is considerably higher than that of Resnik, while the maximum scores differ by <1%. For the other two ontologies the improved mean of the F1-scores in the simDEF measure against Resnik is notable. For MF the difference between max F1-score in the hybrid measure is >2.5% compared with

Resniks F1-score itself. We also see this improvement in the result is due more to simDEF than to the Resnik measure.

6 Discussion

Our approach to similarity estimation based on shared context makes intuitive sense, as concepts which share closely related attributes in their representation should exhibit high levels of similarity. We have shown that implementing these ideas via the Gloss Vector representation yields improved effectiveness across the majority of ontologies and problem types. For the yeast database, simDEF increases the correlation of semantic similarity with sequence homology by 50%, yields an increase of >4% in correlation with gene expression on the BP ontology, and improves the PPI prediction F1-score by >2.5% on the MF ontology.

A key advantage of simDEF in comparison with IC-based measures is its reduced dependency on annotation data, and the GO structure. New GO terms typically do not have rich annotation information, which can influence the IC calculation of all GO terms as they depend on the root frequency which itself depends on all GO term frequencies. In contrast, simDEF needs to access only the direct parents and children of one GO term to expand that GO terms definition.

7 Conclusion

This chapter introduced simDEF, an efficient method for measuring semantic similarity of GO terms using their GO definitions, which was based on the Gloss Vector measure commonly used in natural language processing. We showed that thus seemantic similarity measure can be helful In future work, simDEF can be evaluated against Enzyme Commission (EC) and protein family (Pfam) similarities. Gene clustering and orthologous protein distinguishing tasks present yet another opportunity for simDEF performance evaluation. Moreover, further investigation of miss-classified PPIs will help to improve that aspect of study. simDEF also needs to be tested on the other species than *S.cerevisiae* as well. Moreover, other statistical measures of association, such as Chi-square and log-likelihood, can be examined in replacement of PMI for further improvement of simDEF. More in-depth studies can also find out

if using larger window sizes of bigrams or even tri-grams in the word extraction of MEDLINE abstracts would improve the achieved results. Also, current advancement in deep neural networks for the low-dimensional yet more accurate representation of GO terms leaves room for further investigation of semantic similarity measures in the distributional model. Many of these aspects will be investigated in the next chapter in depth.

Chapter 3

deepSimDEF for Deep Neural Embedding of Biological Attributes and Deep Neural Gene Function Analysis

1 Summary

Background – There exists a plethora of measures to evaluate functional similarity (FS) of genes; measures which are widely used in many bioinformatics applications including identifying co-expressed genes, predicting protein-protein interactions, and prioritization of disease genes. These FS measures are mostly derived from Information Contents (IC) of Gene Ontology (GO) terms annotating genes. However, existing measures employing IC of terms based their results on different hand-engineered and application-specific metrics in order to quantify the degree of shared information between two genes given their GO annotations.

Results – deepSimDEF, however, by relying on the power of deep neural networks, is an efficient model that learns this FS aggregation metric automatically given a set of genes and their paired annotation data. Once trained, deepSimDEF is able to measure FS of genes, that were absent at the time of training, when provided with their new paired GO annotations. To this end, deepSimDEF learns low-dimensional vectors of GO terms and gene products, and then calculates FS using these learned vectors (i.e., embeddings). Relative to best-performing similarity measures, by considering all GO sub-ontologies, when validated on a yeast reference database, deepSimDEF increases PPI predictability by $\sim$4%, shows a correlation improvement >6% with gene expression, and improves correlation with sequence homology by up to 11%.

Conclusions – As far as similarity of genes with respect to their GO annotations is concerned, next to providing GO term and gene product embeddings, deepSimDEF offers a powerful, flexible, easily transferable and adaptable deep neural architecture, as well as a software tool, that a wide range of problems in proteomics and genomics can benefit from.

2 Background

With the revival of *deep neural networks* around 2006 [13, 66], deep learning methods have become prevalent in the research community. These methods are basically representation learning techniques that combine multiple non-linear modules to obtain multiple levels of representation [99]. These modules can transform the representation of the raw input at one level into a representation at a higher, more abstract level. The key advantage of deep learning is that human engineers do not design these layers of features and, therefore, the least feature engineering is needed as features are learned 'dynamically' and 'automatically'. As a result, over the last decade, deep learning methods have brought about breakthroughs in image and speech recognition [65, 93, 59], two challenging tasks that traditionally took years of experts' efforts to design handcrafted features which were not close to perfect. Considering the excellent performance of the deep learning methods in the general domain, in recent years, their effectiveness has been evaluated in the biological domain as well. For example, BioVec [8], inspired by the Word2Vec [128] widely used in *natural language processing* (NLP), is an initiative in the biomedical domain to offer a solution for an unsupervised data-driven distributed representation of biological sequences. The learned vectors (also known as *embeddings*) can be used later on in other machine learning models addressing biological tasks. For a comprehensive review of deep learning applications in biology, medicine, and medical imaging an avid reader can refer to [29, 111].

The *Gene Ontology* project (GO) [9] is a bioinformatics initiative to characterize important features of genes and gene products using a controlled vocabulary. UniProt [6], SwissProt [17], and many other biomedical databases are annotated with *GO terms* to describe the semantic role of biomedical entities. Since *in vitro* biomolecular experiments to validate gene functions are expensive, recently, *functional similarity* (FS) measurement of genes from their GO annotations has become the focus of several challenges such as the Critical Assessment of protein Function Annotation algorithms (CAFA) [180, 80], whereas the ongoing developed methods have been compared against vast biological problems such as prediction of protein-protein interaction (PPI) [22, 240, 22], analysis of gene expressions [197, 222], protein function prediction [241, 92, 242, 180], protein subcellular localization prediction [20, 218, 235], and study of homologous genes [109]. Dessimoz et al. in [42] provide

a thorough overview of GO, and the molecular biology analyses and applications it corrects or facilitates.

As far as GO is concerned, there exist two main computational classes of FS measurements. *Ontology-based methods* take advantage of GO structure in their model in which typically GO term *semantic similarity* (SS) values are computed *pair-wise* prior to drawing on them for the gene functional estimation. The proposed SS measures revolve around the idea of shared *Information Contents* (IC) [184] of GO terms annotating genes. The IC-based FS measures of Resnik [184], Lin [110], Jiang [78], GraSM [37] and AIC [200] depend on these engineered SS measures. Recently, Dutta et al. [45] presented a new approach (which we call clusteredGO in our evaluation) that utilized IC of the GO terms and the GO graph to do GO term clustering. In contrast to this approach, while pair-wise FS measures first compute SS of two gene products and then aggregate the results as a single FS value using another engineered metric, *group-wise* FS measures such as simUI [49], simGIC [171] and SORA [211] directly calculate FS by measuring the distance between two sets of GO term annotations. Motivated by *Jaccard distance* [103], the group-wise measures are less computationally intensive, however, this occurs at the cost of accuracy. This process of FS estimation is executed and then reported for every GO sub-ontology separately (refer to Gene Ontology and GO annotations regarding GO sub-ontologies). Apart from the above-mentioned engineered metrics, more than a decade ago, Schlicker et al. also proposed another handcrafted metric to combine FS scores from every GO sub-ontology into a single FS score through computing the root mean square of the sub-ontologies results [193]. Examined on this metric, recently, Weichenberger et al. in one part of their study showed that the consideration of combined information from all three GO sub-ontologies can reduce the error rate in a task to discriminate between orthologues and random protein pairs [227].

Distributional-based methods of FS measurement are based upon Firth [51], which characterized one natural language term by the company the term keeps in its context. Our previous work, a text-mining approach called simDEF [153] to compare the text definitions of two GO terms, was inspired by this notion to address several drawbacks of the ontology-based methods. Recently Duong et al. [44] by introducing their distributional definition-based model called AicInferSentGO attempted to improve

simDEF even further by proposing a new approach for (distributed) vector representation of GO terms. Even though simDEF and AicInferSentGO demonstrated the significant advantage of distributed vector representation of GO terms, they suffered from important shortcomings, some of which are still shared with even the most recent methods: manual metric and feature engineering for aggregating GO-term SS scores prior to the computation of gene FS; large dimensions of the 'static' GO-term vectors; and, typically separate consideration of each sub-ontology of GO for a biological task at hand due to the lack of certainty on how the downstream biological attributes from those sub-ontologies should be combined. The paired *multi-channel deepSimDEF* neural network presented in this chapter attempts to address all of these shortcomings simultaneously.

deepSimDEF relies on GO annotation data while BioVec takes into account only the sequence information of the biological entities in which the functional characteristics of those biological entities are not entirely encoded. *Supervised* deepSimDEF neural networks are also designed to address the biological tasks by themselves; i.e., the main output of the deepSimDEF is a prediction model, where GO-term and gene-product embeddings will be the by-products of the training process. However, in contrast to previous FS measures in which the estimation results hinged upon the choice of hand-engineered metrics such as *Minimum* (MIN), *Maximum* (MAX), *Average* (AVG), *Best-Match Average* (BMA), *Average Best-Match* (ABM), and *modified Hausdorff distance* (MHD) to aggregate the SS scores and the underlying information of two annotation sets [173, 122, 44], a deepSimDEF network automatically learns this quantification regarding an application of interest, and later, measures FS of genes which are absent at the time of training. Prior to training, deepSimDEF networks are ideally initialized with our pretrained GO-term embeddings that we compute in advance. We tested the performance of deepSimDEF against the FS measures of Resnik [184], Lin [110], Jiang [78], GraSM [37], AIC [200], clusteredGO [45], simGIC [171], simDEF [153] and AicInferSentGO [44] introduced above (see Chapter 2 and Subsection 2 .1 for their details). However, there existed more FS measures in the literature several of which are surveyed in [121]. With the consideration of the biological experiments in the original works of the FS measures and the experiments designed in this study, we aimed at choosing the most well-known and the most recent

measures which were the best representative of their types and the most challenging landmarks to overcome regarding the conducted tasks.

3 Experimental Data

Prior to explaining the results of the study, we discuss the resources that we employed as well as the experimental data that were created during the course of study on which the experiments were based.

3 .1 Gene Ontology and GO annotations

GO terms annotating genes and gene products are structured in three mutually-exclusive sub-ontologies, namely, *biological process* (BP), *cellular component* (CC) and *molecular function* (MF). Generally, a BP term describes a change or complex of changes on the level of granularity of the cell or organism that is mediated by one or more gene products; metabolism and cell proliferation are examples of such BP terms. A CC term, such as the nucleus or cell membrane, defines a part of a cell or its extra-cellular environment where a gene product may be located. An MF term is the enduring potential of a gene product instance to perform actions, such as catalysis or binding activities, on the molecular level of granularity.

Each GO annotation consists of an association between a gene and a GO term with a specific reference and an evidence code that shows how a given annotation is supported. Out of all the evidence codes, inferred from electronic annotation (IEA) and no biological data available (ND) are the least reliable. For experiments of this study, the latest Gene Ontology and the GO annotations of yeast *Saccharomyces cerevisiae* were downloaded from the Gene Ontology website[1].

3 .2 MEDLINE Abstracts

MEDLINE[2] includes over 20 million citations of life sciences and biomedical articles from 1966 to the present. Combined with the GO term definitions, we employed the MEDLINE 2013 bigram list[3] to build our pretrained GO-term embeddings.

[1] http://www.geneontology.org/page/download-ontology (as of Nov. 2018)
[2] https://www.nlm.nih.gov/databases/download/pubmed_medline.html
[3] https://mbr.nlm.nih.gov/Download/

3 .3 Evaluation and Validation Datasets

Protein-protein interaction

We built a PPI dataset from a list of manually curated positive *physical interactions* (PI) provided in yeast *Saccharomyces cerevisiae* database[4]. As to negative interactions, following what is common in the literature, a list of negative interactions was independently generated by randomly choosing annotated protein pairs which were absent from the provided list of PPIs (including genetic and high-throughput interactions). After removing those proteins that had no GO term annotations from all three sub-ontologies of BP, CC and MF (without considering IEA and ND annotations), each pair of interacting proteins was labeled with 1 indicating a positive interaction, or 0 which offered no interaction. The final balanced PPI dataset contained 28,996 interactions in total.

Gene expression

Yeas (Saccharomyces cerevisiae). Having a microarray gene expression data from a study by Eisen et al. [46], our gene expression dataset was built by integrating gene expression data constructed for 2465 yeast genes under 79 biological conditions (4 experiments on cell cycle, sporulation, temperature shock and diauxic shift processes). We computed the absolute Pearson correlation of all possible gene-gene pairs based on the expression values calculated on these 79 biological conditions - regardless of whether their correlation was positive or negative as we focused on the strength of expressions, and then applied Fisher's z transformation to these results to convert them into normally distributed variables suitable for parametric statistical testing. After removing those genes that had no GO annotations, all the genes in the final dataset had their own GO annotations from all three sub-ontologies (without considering IEAs and NDs). The final dataset contained 2,149,701 gene-gene pairs along with the transformed Pearson's correlation of their expressions.

Human (Homo sapiens). The data comes from a study by [15] (due to a large size a random subset of the original data is use). In their study, probes from the human U133A array were mapped to their Refseq identifiers which were then mapped to

[4]https://downloads.yeastgenome.org/curation/literature/interaction_data.tab

Uniprot identifiers. Genes without any GO annotations in GOA were removed. After filtering, the all-pairs correlation of 5688 genes was calculated resulting in 16,173,828 unique correlation pairs (we only used 300,000 pairs which covered 8228 genes that had full GO annotations). The absolute value of the correlation was calculated between expression pairs to attempt to detect a relationship either negative or positive.

Sequence homology

We employed sequence homology[5] data from our previous study [153]. To create this dataset, we used bitscores from the Basic Local Alignment Search Tool (BLAST) algorithm [3]. Since a bitscore for query and subject proteins is not symmetrical, we computed log-reciprocal BLAST score (LRBS) and relative reciprocal BLAST score (RRBS) to express the general sequence similarity of yeast protein pairs. After computation of LRBS and RRBS, we had a dataset of 16,570 protein pairs along with their LRBS and RRBS sequence similarity scores. We removed a few protein pairs from the original dataset due to new changes in Gene Ontology and GO annotation data. All proteins in the final dataset had GO annotations from the BP, CC and MF sub-ontologies (non-IEA and non-ND annotations).

4 Method

4 .1 Pretraining of GO-term Embeddings

Initialization of a neural network with pretrained embeddings has proven to be effective in a variety of applications [23, 232]. Inspired by studies for (high-dimensional) distributed representation of biomedical concepts [115, 161] and the low-dimensional vector representation of words [104, 11] we pretrained GO-term embeddings in six steps depicted in Figure 3.1. The pretraining of GO-term embeddings closely followed our approach in the work explained in Part 2 in which we pretrained sense embeddings for every concept in the Unified Medical Language System (UMLS) to

[5] A homologous gene (or homolog) is a gene inherited in two species by a common ancestor; hence, it is a binary concept. While homologous genes can be similar in sequence, similar sequences are not necessarily homologous. Orthologous are homologous genes where a gene diverges after a speciation event, but the gene and its main function are conserved. If a gene is duplicated in a species, the resulting duplicated genes are paralogs of each other, even though over time they might become different in sequence composition and function.

address the word sense disambiguation (WSD) of biomedical text data. For pretraining of GO-term embeddings, however, we dealt with three GO sub-ontologies of BP, CC and MF, in which GO terms had biologically-concerned text definitions that were represented as low-dimensional vector embeddings. As we show in Experimental Results, these pretrained vectors facilitate and accelerate the exploration and exploitation of training data in order to gain more accurate knowledge regarding a biological task in hand.

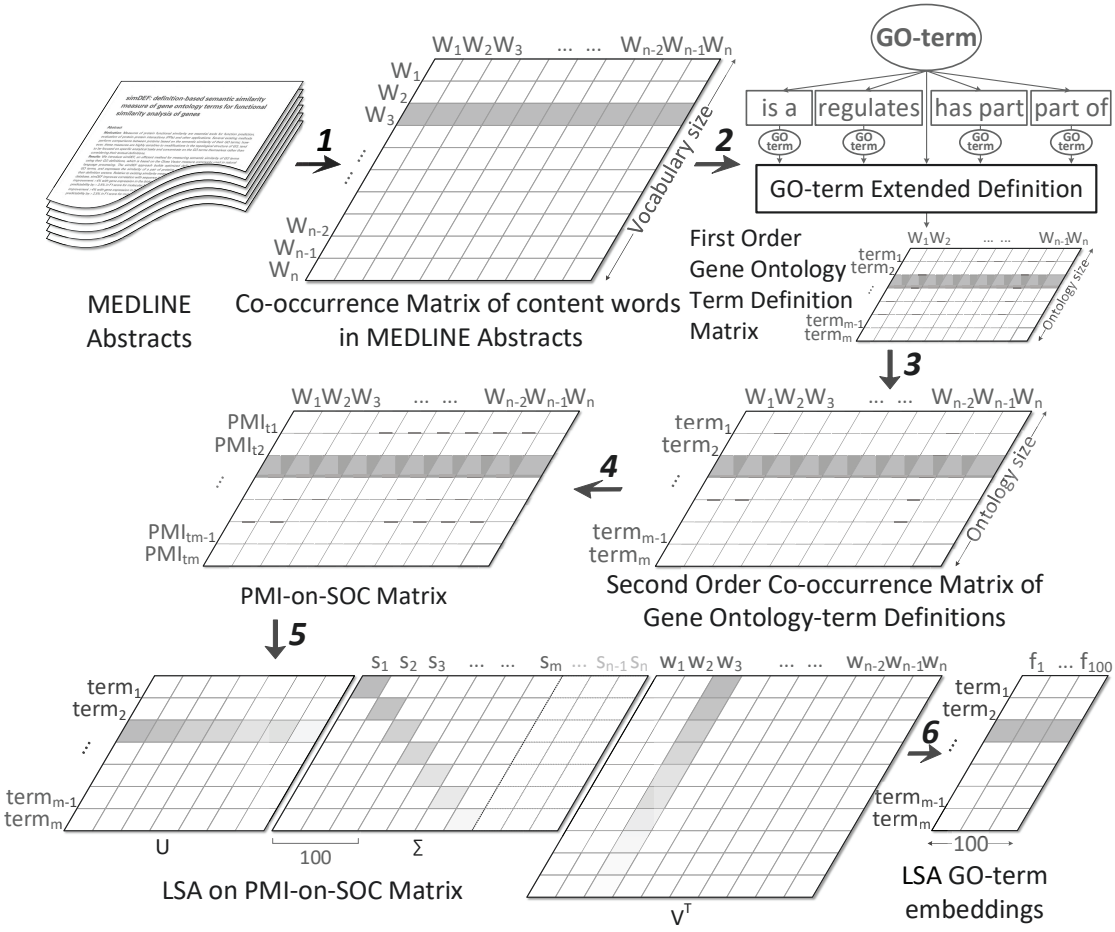

Figure 3.1: Definition-based embedding model of the Gene Ontology terms. The pretraining of GO-term embeddings consists of 6 steps. Breifly, the Second-order vector representation of GO terms prevents sparsity (of word features) in the First-order representation of their text definitions; Pointwise Mutual Information statistically defines the degree of association between GO terms and these second-order word features; and Latent Semantic Analysis reduces the result high-dimensional vectors to a size proper for initialization of a deepSimDEF network. Steps 2–6 are executed for each sub-ontology of BP, CC and MF separately.

In essence, in this pretraining approach, the *Second-order* computation of vector

representation of GO terms (their text definitions) prevents the issue of sparsity of word features in the *First-order* vector representation of their text definitions; *Pointwise Mutual Information* statistically defines the degree of association between each GO term and its second-order word features; and *Latent Semantic Analysis* aims at condensing the final high-dimensional vectors to a size proper for a deep neural network. These steps explained below are executed in advance in order to compute GO-term embeddings before training our deepSimDEF networks which these embeddings initialize. Steps 2–6 were executed for each sub-ontology separately.

Step 1 – MEDLINE Word Co-occurrence Matrix. After discarding punctuation, changing all characters to lowercase, and removing stop-words from the MEDLINE bigram list, a list of bigrams and their frequencies for all the content words in the GO term definitions were constructed. We built a Co-occurrence Matrix from this bigram list of MEDLINE abstracts; a symmetric and sparse matrix that stored contextual information of the MEDLINE words in which we were interested.

Step 2 – Definition Extension and Definition Matrix. In this step, following the simDEF guideline [153], we constructed an extended definition for every GO term. The definition extension of a concept by the definitions of its neighbour concepts in a taxonomy, such as their parents and children, enriches that concept's semantics [115] and to some extent avoids sparsity of the first-order word features in its original and typically brief definition. For this reason, we extended the original definition of every GO term by adding the definitions of the other GO terms which were directly related to that term in the GO structure. The Definition Matrix stored the frequency of the words in every GO-term extended definition. If one word (i.e., W_i word feature) did not appear in a GO term definition the frequency was 0 – which still could indicate sparsity in these vectors despite the extension.

Step 3 – Normalized Second-order Co-occurrence (SOC) Matrix. Each of the W_i word features from the previous step has an associated co-occurrence vector that we computed in Step 1. Following [115, 153, 154], these rich co-occurrence vectors helped to resolve the issue of sparsity of the first-order definition vectors of the GO terms further through the construction of the second-order vector presentations of

the definitions.

To build a normalized SOC vector of a GO term, we first summed the MEDLINE co-occurrence vectors of the content words in that GO term's extended definition, and then divided the resulting vector by the number of words in that definition (these frequency statistics were stored in the Definition Matrix built in the previous step). In other words, we took the centroid of the co-occurrence vectors associated with the words in one definition, and then normalized the result by the number of constituent vectors in the summation in order to deal with variable lengths of the GO term definitions.

Step 4 – Pointwise Mutual Information (PMI) on SOC Matrix. Not all word features associated with a GO term are equally important [156]. PMI, as in Eq. (8 .1), statistically measures the level of association between one GO term, i.e., every associated word in its normalized SOC vector denoted by $word_i$, and the word features W_j. This statistical approach is a replacement for the naive consideration of word feature frequency cut-off threshold for the removal of low-frequency occurrences [115]. As a principal rule in NLP, the total frequency of one occurrence indicates how informative that occurrence is, stating the less frequent the occurrence is in a series of events, the more informative that occurrence will be in general [182] – an important consideration ignored in the low-frequency cut-off threshold [156]. PMI on the other hand took these total frequencies into consideration through $p(word_i)$ and $p(W_j)$ probabilities denoted in Eq. (8 .1). Once PMI values were calculated for all the GO terms and word features, our validation sets helped to set a low cut-off threshold for the removal of (statistically) irrelevant features. As a common practice in the computation of PMI values, we also applied the Laplace (add-one) smoothing technique to the Normalized SOC Matrix in advance to avoid bias towards infrequent occurrences [40].

$$PMI(word_i, W_j) = log \frac{p(word_i, W_j)}{p(word_i) \times p(W_j)} \tag{4 .1}$$

Step 5 – Latent Semantic Analysis (LSA) on PMI-on-SOC Matrix. LSA is a statistical approach of acquisition and representation of semantics that allows similarities among the elements of a language – such as words or sentences – to be

computed based on their co-occurrence patterns in a large corpus [140]; a computational model of meaning that closely mimics human understanding of the contextual use of language widely used for information retrieval and machine understanding of text [96]. Hence, unlike standard keyword-based methods, LSA can detect subtle aspects of semantic content. Employing this statistical approach, formulated by Eq. (8 .2), LSA used *Singular Value Decomposition* (SVD) algorithm that resulted in two square and unitary matrices $\boldsymbol{U}$ an $\boldsymbol{V}^T$, and a non-negative diagonal matrix $\boldsymbol{\Sigma}$ that held singular values on its diagonal in a non-increasing order [57].

$$PMI\_on\_SOC = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{4 .2}$$

Step 6 – Reducing the Rank of Singular Values. The reduced dimension semantic representation from LSA allows comparison by computing the semantic similarity between individual terms or groups of terms in a more efficient manner. We use this dimensionality reduction technique to prepare our well-sized GO-trem embeddings for an effective deepSimDEF network initialization. Having Eq. (8 .3), we truncated the SVD to 100 for low-dimensional representation of GO terms. The resulting matrix (its columns) contained 100 *principal components* of the original matrix. Basically, these principal components are calculated from a covariance matrix which is encoded in $\boldsymbol{\Sigma}$ in the form of the square root of its eigenvalues (i.e., singular values) [57]. That is, principal components with larger associated variances represent interesting structure, while those with lower variances indicate noise. Determined by our validation sets in the conducted experiments, embedding sizes smaller than 100 yielded worse results whereas higher dimensions did not improve the accuracy and just increased the training time of the networks.

$$GO\_terms\_LSA\_embeddings = \boldsymbol{U}\boldsymbol{\Sigma}_{100} \tag{4 .3}$$

4 .2 deepSimDEF Network Definition

deepSimDEF offers *single-channel* and *multi-channel* network architectures which learn and represent the shared information of two proteins based on their GO annotations, and then measure FS of genes for an application of interest. While a

single-channel network only considers annotations of one sub-ontology, as depicted in Figure 3.2 for the BP sub-ontology, the multi-channel architecture, with more layers shown in Figure 3.3, takes into account all the three GO sub-ontologies together. The 6 layers fundamental to both deepSimDEF architectures are described as follows.

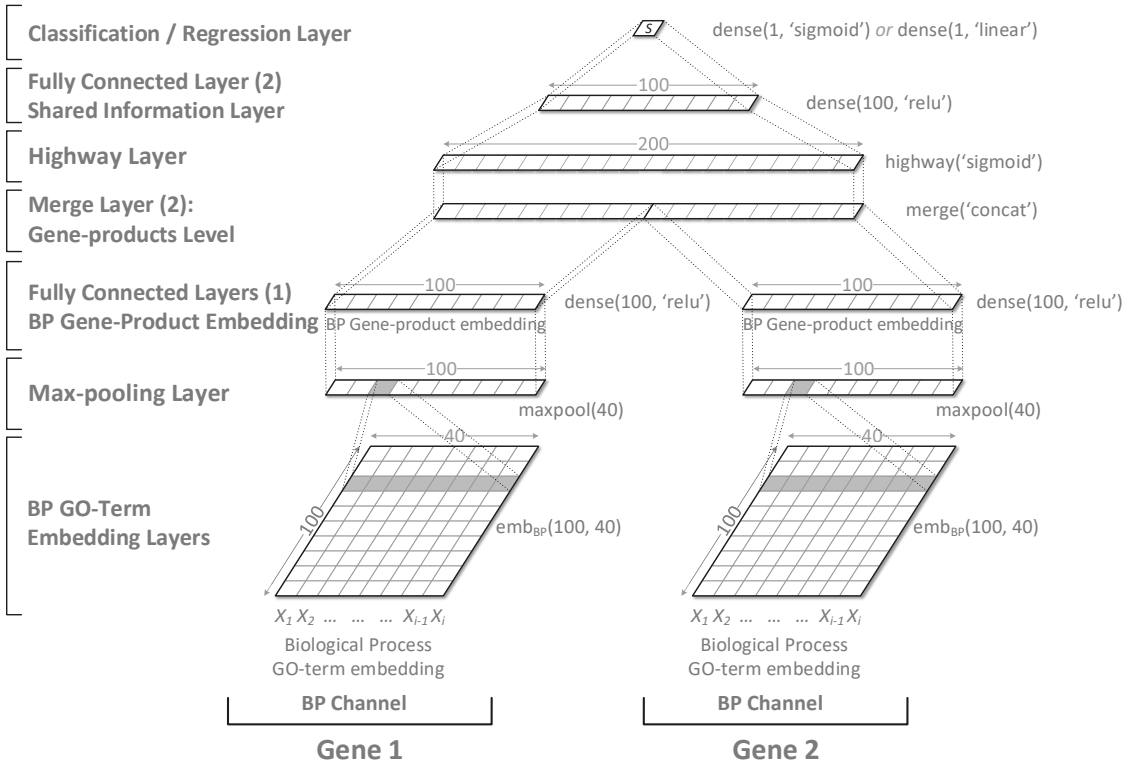

Figure 3.2: Paired single-channel deepSimDEF network architecture for BP.
The paired single-channel deepSimDEF architecture consists of 7 layers for functional similarity measurement of genes and gene products using their Gene Ontology term annotations from one of the GO sub-ontologies. For two input genes, their annotations are fed to the network in the first layer in which they will be represented as two lists of 100-dimensional embedding vectors. Max-pooling layer condenses each of these two lists into a 100-dimensional row-vector. Merge and highway layers together encode the degree of shared information of these two pooled vectors. In several locations of the architecture, fully-connected layers are considered for better representation of their underlying layers. Finally, based on the biological application in hand, the result of the prediction network is computed which can be either a scalar or a classification probability. The weights of the two pairs are shared during the network training and testing. This architecture is shown for BP, however, for CC and MF it stays the same with only having different length of the input annotations in the first layer that is defined by the maximum number of annotations given to a gene from that sub-ontology.

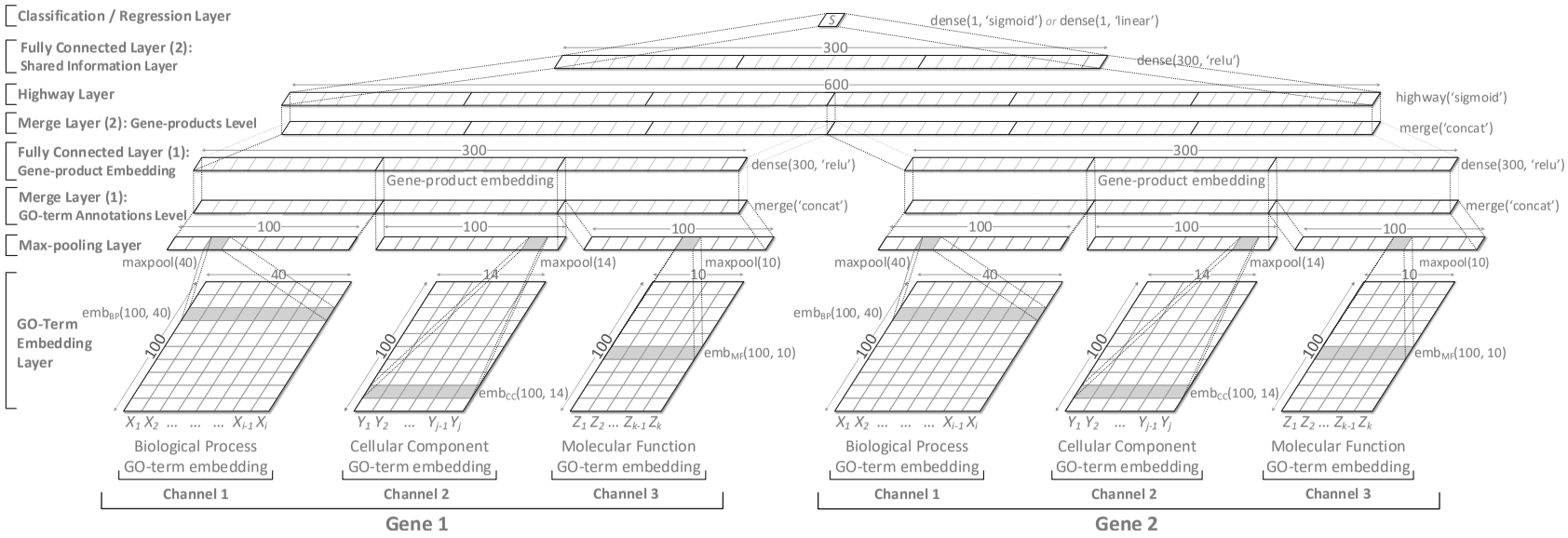

Figure 3.3: Paired multi-channel deepSimDEF network architecture.

The paired multi-channel deepSimDEF architecture consists of 8 layers for functional similarity measurement of genes and gene products using their Gene Ontology term annotations from all three GO sub-ontologies. For two input genes, their full GO annotations are fed to the network in the first layer in which they will be represented as six lists of 100-dimensional embedding vectors (three lists for each gene). Max-pooling layer condenses each of these six lists into a 100-dimensional row-vector, and subsequently the first merge layer concatenates them into two rich row-vectors regarding the gene and the sub-ontologies they come from. The second merge layer as well as highway layer together encode the degree of shared information of these two pooled and then merged vectors. In different locations of the architecture, fully-connected layers are considered for better representation of their underlying layers. Finally, based on the biological application in hand, the result of the prediction network is computed which can be either a scalar or a classification probability. The weights of the two pairs are shared during the network training and testing. In an attempt to increase accuracy of the predictions, in contrast to single-channel architecture, this architecture considers all GO annotations from all three sub-ontologies of BP, CC and MF.

GO-term Embedding Layer. The GO term annotations of two proteins are fed to the model as indexes taken from three fixed sets of $\mathrm{GO_{BP}}$, $\mathrm{GO_{CC}}$, or/and $\mathrm{GO_{MF}}$. These sets contain the indexed GO terms of a particular database from the sub-ontologies of BP, CC, and MF (yeast database in our case). Each set is also associated with a *look-up table* of 100-dimension in row size; e.g., $\mathrm{W}_{LT\_BP} \in \mathrm{R}^{100\times|\mathrm{GO_{BP}}|}$ is the look-up table for BP GO terms of the yeast database. These tables, ideally initialized with our pretrained LSA GO-term embeddings, are parameters of the model. First, for every protein, its GO term indexes transform into vectors by looking up their GO-term embeddings. Then, within the embedding layer, for each sub-ontology, the two input proteins are represented as two lists of fixed length t_0, each list containing the 100-dimensional GO embeddings of those two genes' annotations looked up already (Eq. (4 .4)). In the architectures, for consistency across GO annotations of all genes, whenever the annotation sets of a gene had the length of less than t_0, we padded the annotation list with a generic vector of a large negative value (padding was repeated whenever needed); subsequent Max-pooling Layer later suppressed the effect of this generic vector and the final estimations were calculated only based on the actual annotations. Without the consideration of IEAs (i,e., IEA−), the fixed length for BP, CC, and MF were 40, 14, and 10, respectively (i.e., the longest number of the GO term annotations of a gene in the yeast database from that sub-ontology); for IEA+ these numbers were 44, 17, and 33.

$$\boldsymbol{X}_{ebm} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{t_0}] \in R^{100\times t_0} \tag{4 .4}$$

where $\boldsymbol{x}_i$ denotes the GO-term embedding of the ith BP GO annotation of a protein. An embedding layer is denoted by *emb(100, t_0)* in the figures.

Max-pooling Layer. Max-pooling operations are commonly used to extract global features from convolution [93]. In our method, since we deal with the sets of annotations instead of sequences of words or adjacent pixels, we do not need any convolution layer, so max-pooling is applied directly to the embeddings. Generally, a pooling layer aggregates the input vectors by taking the maximum over a set of intervals. Here,

for the output of an embedding layer, the max operation is applied over all column-features, which is denoted by *maxpool(t_0)*. We also considered *flattening* of the resulting pooled column-vector into a row feature-vector representation as an integrated part of the max-pooling layer. This flattening needs to be done prior to passing the results of a lower layer to a higher fully-connected layer. After max-pooling, proteins with different lengths of GO annotations are represented with 100-dimensional global feature vectors each for one sub-ontology (e.g., $\boldsymbol{b}_{pool} \in R^{1\times 100}$ is the pooling layer result for BP).

Merge Layer. Depending on whether we use the paired single-channel or the paired multi-channel architecture, we have one or two merge layers. For a paired single-channel architecture, we have only one merge layer, which happens at the gene-product (similarity) level due to the paired nature of the input data. That means prior to the extraction and representation of the shared information between two gene products, their individual feature-vectors need to be merged through the *concatenation* technique. For the multi-channel architecture however, besides having a merge layer at the gene-product level, we have one more merge layer that occurs at the GO term annotations level. For a given gene product of an input protein pair, this extra merge layer is used to concatenate the three 100-dimensional feature-vectors of the BP, CC, and MF annotations from the max-pooling layer. In the multi-channel architecture, at the GO term annotations level, $\boldsymbol{m}_{go\_multi} \in R^{1\times 300}$ is the result of the merge layer. At the gene-product level, $\boldsymbol{m}_{gp\_single} \in R^{1\times 200}$ and $\boldsymbol{m}_{gp\_multi} \in R^{1\times 600}$ are the results of the merge layers for the paired single-channel and paired multi-channel architectures, respectively. Merge layers are denoted by *merge('concat')*.

Fully-connected Layer. The fully-connected layer takes a d_0-dimensional input row-vector $\boldsymbol{x}_{fch} \in R^{d_0}$ to learn higher level feature representations of the underneath layers[6]:

$$\boldsymbol{h} = ReLU(\boldsymbol{W}_h \cdot \boldsymbol{x}_{fch} + \boldsymbol{b}_h) \tag{4 .5}$$

where $\boldsymbol{W}_h \in R^{n_{hid}\times d_0}$, n_{hid} is the size of the fully-connected hidden layer, $\boldsymbol{b}_h \in$

[6]In the equations, $\cdot$ denotes matrix multiplication.

$R^{n_{hid}}$ is the bias vector, and $ReLU$ is rectified linear activation function [133]. The output of the first fully-connected layer can be seen as the embeddings of the input gene products. Depending on whether the single-channel or multi-channel network is employed, this embedding size can be 100-dimensional or 300-dimensional. The fully-connected hidden layers are denoted by *dense(n_{hid}, 'relu')*. At the similarity level, the output of the fully-connected layer improves for better representation of the shared information between two given gene products.

Highway Layer. In the previous measures including simDEF, for FS estimation of two input gene products, human-engineered aggregation metrics were used – while the SS scores of their pair-wise GO annotations made the inputs of these metrics. However, there is no consensus in the literature on what metric is the best choice for the aggregation of the shared information, as from one biological experiment to another the results vary, and even sometimes, the conclusions contradict each other [62]. In the deepSimDEF model, the highway layer [203] is devised in such a way that the model itself properly learns an adaptive representation of the provided information of the two input gene products encoded in the lower layer for the comparison of their as well as other gene products functionality. This representation uses a *gating mechanism* that controls the flow of information from the two gene products into an aggregated high-level representation. This adaptive representation of the shared information strengthens an affine transformation - similar to what is presented in Eq. (4 .5) - with a non-linear transform function $\boldsymbol{T}$. We refer to the vector $\boldsymbol{T}$ as the transform gate since it expresses how the output is produced through *transforming* or *carrying* the input. If we consider the size of the concatenated feature vectors of two input genes to be d_1-dimensional, $\boldsymbol{T}$ can be formulated as:

$$\boldsymbol{T} = \sigma(\boldsymbol{W}_T \cdot \boldsymbol{x}_{fch} + \boldsymbol{b}_T) \tag{4 .6}$$

where $\boldsymbol{W}_T \in R^{n_{hid} \times d_1}$ is the weight matrix, n_{hid} is the size of the fully-connected hidden layer and here is equal to d_1 since we do not want to expand or shrink the representation result at this stage, $\boldsymbol{b}_T \in R^{n_{hid}}$ is the bias vector, and σ is a *sigmoid function* employed in the original paper as the transform function [203]. If we want to represent two extreme cases which apply either transform state or block (or carry)

state on the input data, Eq. (4 .7) formulates that for us:

$$\boldsymbol{x}' = \begin{cases} \boldsymbol{x}_{fch}, & if\, \boldsymbol{T} = 0 \\ \sigma(\boldsymbol{W}_h \cdot \boldsymbol{x}_{fch} + \boldsymbol{b}_h), & if\, \boldsymbol{T} = 1 \end{cases} \tag{4 .7}$$

Therefore, depending on the output of the transform gates, a highway layer should smoothly vary its behavior between that of a plain layer with a non-linear activation of interest (if $\boldsymbol{T} = 1$; in deepSimDEF we achieved better results with sigmoid function) and that of a layer which simply passes its inputs through (if $\boldsymbol{T} = 0$). Eq. (4 .8) formulates this favorable behavior of a highway layer[7]:

$$\boldsymbol{x}' = \sigma(\boldsymbol{W}_h \cdot \boldsymbol{x}_{fch} + \boldsymbol{b}_h) \odot \boldsymbol{T} + \boldsymbol{x}_{fch} \odot (1 - \boldsymbol{T}) \tag{4 .8}$$

The transform gate - which is the principal component in the deepSimDEF network(s) for a high-level representation of the shared information of two input genes, and all the weights in the highway layer, will be learned during the training phase. The highway layer is denoted by *highway('sigmoid')*.

Classification / Regression Layer. Depending on whether an experiment conducted in a study is formulated as a classification problem or as a regression problem, the output of the last dense layer is fully connected to either a *softmax classification* layer (e.g., for our PPI experiment) or a *linear regression* layer (for the gene expression and sequence homology experiments). After the lower layer processing, a fixed dimensional feature vector $\boldsymbol{x}_{cl}$ or $\boldsymbol{x}_{rg} \in R^{d_2}$ is the input to the classification/regression layer, with a sigmoid or linear activation, whose output is the FS estimation of the genes. For a classification task we have:

[7]In the equation, $\odot$ implies element-wise multiplication.

$$p(y = i|\boldsymbol{x}_{cl}) = \frac{exp(\boldsymbol{W}_{out_i} \cdot \boldsymbol{x}_{cl} + \boldsymbol{b}_{out_i})}{\sum_{j=1}^{n_{out}} exp(\boldsymbol{W}_{out_j} \cdot \boldsymbol{x}_{cl} + \boldsymbol{b}_{out_j})} \quad (4.9)$$

where $p(y = i|\boldsymbol{x}_{cl})$ outputs probability distribution over labels, $\boldsymbol{W}_{out} \in R^{n_{out} \times d_2}$, n_{out} is the size of the classification layer (for the PPI prediction it is equal to two types), $\boldsymbol{b}_{out} \in R$ is the bias vector, and d_2 is either 100-dimensional (for single-channel) or 300-dimensional (for multi-channel architecture). The classification layer is denoted by *dense(1, 'sigmoid')*. For a regression task:

$$\hat{y} = \boldsymbol{W}_{out} \cdot \boldsymbol{x}_{rg} + \boldsymbol{b}_{out} \quad (4.10)$$

where $\hat{y}$ outputs a scalar value, $\boldsymbol{W}_{out} \in R^{1 \times d_2}$, d_2 is either 100- or 300-dimensional feature vectors depending on the chosen architecture, and $\boldsymbol{b}_{out} \in R$ is the bias vector. The regression layer is denoted by *dense(1, 'linear')*.

Since a deepSimDEF network needs to be symmetric and produce the same result for the two input pairs of [g_1, g_2] and [g_2, g_1], all equivalent layers of the paired networks must share the same weights; this applies to the embedding layers as well (similar to Siamese network[8]). Meaning, for each sub-ontology, we only have one look-up table (initialized randomly or with the pretrained LSA GO-term embeddings). In the training phase and during *back-propagation*, this table(s) will be updated simultaneously for every gene product in a training gene product pair. We also used *dropout* [202] of 0.3 on the fully-connected and highway layers to allow a more accurate generalization. The parameters of the networks are optimized to maximize the correlation between the estimated FS of gene products predicted by the models and the target scores in the training datasets. This selection was done in a 10-fold cross-validation manner where validation splits chose the best parameters using an *early stopping*

[8]Siamese network is an artificial neural network that use the same weights while working in tandem on two different input vectors to compute comparable output vectors.

strategy [177]. Additionally, since the weight matrices of the highway layer for the concatenated feature-vectors of the paired networks are not symmetric and do not update symmetrically, we not only trained the networks on ($[g_1, g_2]$, *score*) instances, we also trained them on ($[g_2, g_1]$, *score*) instances.

5 Experimental Results

The experiments were designed in a 10-fold cross-validation fashion. Meaning, in every experiment (i.e., dealing with PPI, gene expression, or sequence homology), we randomly divided the total number of proteins (in that experiment) into 10 non-overlapping sets. In a 10-time experiment, each time we reserved one of those sets, and all the protein pairs in which they occurred, for testing; the rest of the protein pairs were employed for network training (10% of them were set aside for validation in advance). After the hyper-parameters were selected, the final networks were trained on the whole training set that contained the validation set and then were evaluated on the test set protein pairs in which at least one protein was unseen during training, therefore, no proteins were presented in both the training and testing sets. Through this design, we attempted to break the inter-connection between the pairs of the gene products in the training data (i.e., we avoided direct transitive inference between the protein pairs). Additionally, as to the motivation for conducting this design, we assumed the reserved proteins were new proteins which were discovered recently, so there existed no concrete prior knowledge (i.e., their GO annotations) about their functionality. Therefore, following what is common in the literature regarding SS and FS, we wanted to see how deepSimDEF networks worked with a set of entirely new proteins. Despite the negligible variance in the results, for a more solid conclusion, we repeated the shuffling of proteins in every experiment 10 times and the average of all 100 results was considered as the final result of that experiment.

5 .1 Semantic Similarity of Pretrained GO-term Embeddings

In the following subsections, we will show that the initialization of a deepSimDEF network with pretrained GO-term embeddings improved the results of the experiments (refer to Pretraining of GO-term Embeddings to see how these embeddings

are constructed; for how they initialize the deepSimDEF networks see GO-term Embedding Layer). In essence, our pretraining method organizes embeddings of the GO terms within a Euclidean space based on those GO terms' semantics (arranging books in a physical library is an appropriate analogy for this attempt). Once introduced to a network, these embeddings put that network in a proper state prior to training leading to faster convergence and more accurate results. For three randomly selected GO terms from a pool of >16,000 biological process (BP) terms, Table 3.1 shows the 5 top-most similar GO terms to those terms drawn from our pretrained GO-term embeddings using *cosine* similarity (in the library analogy they are similar books arranged next to the given book title). We can see for a given GO-term query, the returned GO terms are very close conceptually. For cellular component (CC) and molecular function (MF), we observed the same sense-similarity organizations in their embedding spaces as well.

Table 3.1: Sense similarity results for three BP terms over pretrained embeddings

Query	GO term ID	GO term Name
Q #1	**GO:0072521**	**purine-containing compound metabolic process**
1	GO:0072523	purine-containing compound catabolic process
2	GO:0072527	pyrimidine-containing compound metabolic process
3	GO:0072529	pyrimidine-containing compound catabolic process
4	GO:0052803	imidazole-containing compound metabolic process
5	GO:0046453	dipyrrin metabolic process
Q #2	**GO:0045292**	**mRNA cis splicing, via spliceosome**
1	GO:0000398	mRNA splicing, via spliceosome
2	GO:0048024	regulation of mRNA splicing, via spliceosome
3	GO:0000380	alternative mRNA splicing, via spliceosome
4	GO:0090615	mitochondrial mRNA processing
5	GO:0000395	mRNA 5'-splice site recognition
Q #3	**GO:0001116**	**protein-DNA-RNA complex assembly**
1	GO:0001115	protein-DNA-RNA complex subunit organization
2	GO:0001117	protein-DNA-RNA complex disassembly
3	GO:0071165	GINS complex assembly
4	GO:0071824	protein-DNA complex subunit organization
5	GO:0032986	protein-DNA complex disassembly

Semantic similarity of pretrained GO-term embeddings of Cellular Component and Molecular Function subontologies are demonstrated in Table 3.2 and Table 3.3 respectively.

Table 3.2: Sense similarity results for three CC terms over pretrained embeddings

Query	GO term ID	GO term Name
Q #1	**GO:0000109**	**nucleotide-excision repair complex**
1	GO:0033061	DNA recombinase mediator complex
2	GO:0009380	excinuclease repair complex
3	GO:0019812	type I site-specific deoxyribonuclease complex
4	GO:1990391	DNA repair complex
5	GO:1990249	nucleotide-excision repair, DNA damage recognition complex
Q #2	**GO:0000306**	**extrinsic component of vacuolar membrane**
1	GO:0032419	extrinsic component of lysosome membrane
2	GO:0019898	extrinsic component of membrane
3	GO:0031312	extrinsic component of organelle membrane
4	GO:0035452	extrinsic component of plastid membrane
5	GO:0031313	extrinsic component of endosome membrane
Q #3	**GO:0044611**	**nuclear pore inner ring**
1	GO:0070762	nuclear pore transmembrane ring
2	GO:0044614	nuclear pore cytoplasmic filaments
3	GO:0031080	nuclear pore outer ring
4	GO:0044612	nuclear pore linkers
5	GO:0044615	nuclear pore nuclear basket

5 .2 Comparison with PPIs

Protein-protein interactions play a key role in various aspects of the structural and functional organization of the cell; studies demonstrate knowledge of PPIs unveils the molecular mechanisms of biological processes that lead to rational drug design [132]. To this end, it has been shown that the (aggregate of) SS values can be employed as an indicator for the plausibility of putative PPIs [246]. This is because proteins that interact in the cell *in vivo* are expected to participate in similar cellular locations and to be involved in close or/and related biological processes [153], attributes that are expressed by Gene Ontology annotations.

Like other studies, we formulated this as a classification problem and checked how well a deepSimDEF network, as a tool of FS measurement, performed to predict true PPIs. For this purpose, the result of an FS measure (ours or others') was directly interpreted as the classification probability of "Interaction" and "Not Interaction". Generally, the higher this value is, the higher the probability of interaction will be. In our evaluation, represented in Table 3.4, the results of predictions from deepSimDEF and other similarity measures are compared with respect to F1-scores computed for

Table 3.3: Sense similarity results for three MF terms over pretrained embeddings

Query	GO term ID	GO term Name
Q #1	**GO:0044653**	**dextrin alpha-glucosidase activity**
1	GO:0044654	starch alpha-glucosidase activity
2	GO:0032450	maltose alpha-glucosidase activity
3	GO:0090600	alpha-1,3-glucosidase activity
4	GO:0004558	alpha-1,4-glucosidase activity
5	GO:0033919	glucan 1,3-alpha-glucosidase activity
Q #2	**GO:0071667**	**DNA/RNA hybrid binding**
1	GO:0097098	DNA/RNA hybrid annealing activity
2	GO:0001069	regulatory region RNA binding
3	GO:0003697	single-stranded DNA binding
4	GO:0001067	regulatory region nucleic acid binding
5	GO:1990471	piRNA uni-strand cluster binding
Q #3	**GO:0000034**	**adenine deaminase activity**
1	GO:0008892	guanine deaminase activity
2	GO:0004126	cytidine deaminase activity
3	GO:0004131	cytosine deaminase activity
4	GO:0047974	guanosine deaminase activity
5	GO:0035888	isoguanine deaminase activity

each classifier (i.e., each gene FS measure). Among the aggregation metrics used in the previous studies, MAX yielded the highest PPI prediction results, so, we considered it in our evaluation and presented it in the table.

Table 3.4: F1-score of deepSimDEF in the PPI prediction task of the yeast dataset for three sub-ontologies compared to other FS measures aggregated by MAX

Semantic Measure	Excluding IEA (%)			Including IEA (%)		
	BP	CC	MF	BP	CC	MF
Resnik [184]	82.93	77.77	72.12	83.09	77.93	72.64
Lin [110]	82.43	76.61	71.63	82.79	76.73	71.84
Jiang and Conrath [78]	82.49	76.75	71.78	82.85	76.86	71.97
GraSM [37]	82.93	77.86	72.01	83.12	77.96	72.58
AIC [200]	82.49	76.59	71.41	82.79	76.73	71.67
clusteredGO [45]	82.98	77.78	72.23	83.18	77.93	72.76
simGIC [171]	78.89	73.72	69.93	79.46	74.43	70.23
simDEF [153]	83.39	78.41	75.21	83.79	78.63	75.63
AicInferSentGO [44]	83.21	78.48	75.09	83.73	78.71	75.61
$\text{deepSimDEF}_{\text{single-channel-rand-emb}}$	81.48 ± 0.45	80.52 ± 0.42	77.11 ± 0.45	81.91 ± 0.49	80.64 ± 0.51	78.08 ± 0.39
$\text{deepSimDEF}_{\text{multi-channel-rand-emb}}$		82.82 ± 0.36			83.43 ± 0.33	
$\text{deepSimDEF}_{\text{single-channel-LSA-emb}}$	**85.42 ± 0.23**	**82.41 ± 0.24**	**80.78 ± 0.18**	**85.84 ± 0.31**	**82.77 ± 0.23**	**81.49 ± 0.19**
$\text{deepSimDEF}_{\text{multi-channel-LSA-emb}}$		**87.37 ± 0.29**			**87.69 ± 0.26**	

deepSimDEF, when pretrained with LSA GO-term embeddings, in a single-channel

network achieved F1-score improvement of >2% compared to the second best method's results, simDEF in BP (∼4% on average for all three sub-ontologies). Also, in the multi-channels we observed a further increase of ∼2% compared to BP of deepSimDEF which yielded the best result among the evaluated FS measures and among the three sub-ontologies (∼4% improvement compared to simDEF in BP); this indicates the consideration of all three sub-ontologies together provides us with a better PPI prediction model. clusteredGO slightly improved the results of Resnik, whereas the group-wise simGIC represented the worst performance among the evaluated FS measures. Comparing simDEF with AicInferSentGO, we observed their results were very close – which, due to their definition-based nature, was expected. What separates simDEF and AicInferSentGO is their way of GO term vector representation; simDEF relies on extended definitions of GO terms; AicInferSentGO, however, computes them using an approach already proposed in [35] for supervised learning of sentence representations using a Bidirectional Long Short-Term memory (BiLSTM) encoder. Also, we observed that the consideration of IEAs for PPI prediction slightly improved all results.

By comparing the results of the deepSimDEF networks when started with random embeddings versus when initialized with pretrained LSA GO-term embeddings, we understood the influential impact of the latter as we achieved >4% increase for the multi-channel and >3% for the single-channel networks in the F1-scores. This means that the initialization of the networks with LSA embeddings avoids some critical local minima that a deepSimDEF network can fall into during training if it is initialized with random weights.

5 .3 Correlation with Gene Expression

Highly correlated genes are often functionally related and participate in the same biological processes. Previous studies have evaluated the performance of their FS measures by calculating the correlation between their estimations and gene-expression correlation data [236, 15].

Wu et al. in [229], by achieving a poor correlation between their GO-based FS measure and gene expression from microarray data of yeast and human argued that the inconsistent results experienced in the previous studies indicate the correlations

between GO-based FS measures and gene co-expression data are sensitive to the source of data and the method of evaluation. Similar to Wang et al. [221], we believe this inconsistency stems from the inherent complexity of the gene-expression data, and the fact that there exists no direct correlation between GO annotations and gene-expression data that one ideal GO-based FS measure can completely discover and portray. We believe, however, deep neural networks, such as ours, have the potential of exploiting this (non-linear) complexity and discovering the underlying inner dependency to the greatest degree possible.

In our evaluation, represented in Table 3.5, the Pearson's correlation coefficients between the FS measures and the gene-expression data were studied (see Table 3.6 for Spearman's correlation results). In contrast to the PPI experiment, however, BMA metric showed better correlation with the gene-expression data, hence, we considered it in the evaluation.

Table 3.5: Pearson's correlation of deepSimDEF and yeast gene expression data for three sub-ontologies compared with other FS measures aggregated by BMA

Semantic	**Excluding IEA**			**Including IEA**		
Measure	BP	CC	MF	BP	CC	MF
Resnik [184]	0.241	0.393	0.213	0.225	0.396	0.221
Lin [110]	0.217	0.362	0.201	0.201	0.374	0.203
Jiang and Conrath [78]	0.185	0.335	0.178	0.178	0.345	0.181
GraSM [37]	0.245	0.393	0.215	0.228	0.389	0.226
AIC [200]	0.206	0.353	0.201	0.196	0.371	0.205
clusteredGO [45]	0.209	0.325	0.201	0.193	0.345	0.205
simGIC [171]	0.066	0.152	0.142	0.081	0.156	0.121
simDEF [153]	0.323	0.401	0.191	0.329	0.403	0.218
AicInferSentGO [44]	0.323	0.399	0.189	0.329	0.405	0.212
deepSimDEF$_{\text{single-channel-rand-emb}}$	0.448 ± 0.025	0.414 ± 0.026	0.333 ± 0.021	0.452 ± 0.031	0.418 ± 0.029	0.343 ± 0.026
deepSimDEF$_{\text{multi-channel-rand-emb}}$		0.462 ± 0.018			0.466 ± 0.021	
deepSimDEF$_{\text{single-channel-LSA-emb}}$	**0.451 ± 0.019**	**0.414 ± 0.017**	**0.335 ± 0.016**	**0.456 ± 0.015**	**0.422 ± 0.017**	**0.343 ± 0.017**
deepSimDEF$_{\text{multi-channel-LSA-emb}}$		**0.464 ± 0.015**			**0.469 ± 0.012**	

The pretrained deepSimDEF, in the single-channel networks, improved Pearson's correlation with the expression data by >8% (on average); >5% for deepSimDEF in BP compared to the second best results achieved by simDEF and AicInferSentGO in CC. In contrast to the previous FS measures which in general expressed better correlation results in CC, deepSimDEF represented higher correlation with the expression data in BP. Moreover, in the previous FS measure, GraSM showed better correlation with the expression data for MF whereas simDEF and AicInferSentGO acted better in the CC and BP departments; nevertheless, single-channel deepSimDEF networks outperformed them all in the three sub-ontologies. Additionally, as expected, in the multi-channel deepSimDEF architecture we observed an increase of >1% in

the Pearson's correlation result compared to the single-channel model architecture.

Similar to the PPI experiment, after taking IEA annotations into account, we gained minor improvements in the correlation results with the expression data. In contrast to the PPI experiment, however, initialization of a network with pretrained GO-term embeddings did not increase the correlation results with the expression data significantly; considering the large data size in this experiment this observation made sense since the GO-term embeddings could be trained almost entirely from scratch during the network training; nevertheless, initialization of the networks with the pretrained GO-term embeddings accelerated the training process.

The Spearman's correlation results between the FS measures and the gene expression data are shown in Table 3.6.

Table 3.6: Spearman's correlation of deepSimDEF and yeast gene expression data for three sub-ontologies compared with other FS measures aggregated by BMA

Semantic	**Excluding IEA**			**Including IEA**		
Measure	BP	CC	MF	BP	CC	MF
Resnik [184]	0.052	0.193	0.102	0.038	0.189	0.106
Lin [110]	0.029	0.163	0.091	-0.005	0.169	0.088
Jiang and Conrath [78]	-0.027	0.136	0.067	-0.028	0.139	0.066
GraSM [37]	0.056	0.193	0.104	0.042	0.183	0.101
AIC [200]	-0.003	0.154	0.091	-0.009	0.165	0.091
clusteredGO [45]	0.021	0.126	0.091	-0.013	0.139	0.091
simGIC [171]	0.032	0.042	0.031	0.033	0.056	0.036
simDEF [153]	0.134	0.201	0.101	0.142	0.196	0.103
AicInferSentGO [44]	0.134	0.199	0.078	0.142	0.198	0.097
deepSimDEF$_{\text{single-channel-rand-emb}}$	0.239 ± 0.025	**0.214 ± 0.024**	**0.202 ± 0.022**	**0.245 ± 0.021**	**0.212 ± 0.028**	**0.208 ± 0.023**
deepSimDEF$_{\text{multi-channel-rand-emb}}$		0.249 ± 0.021			**0.262 ± 0.017**	
deepSimDEF$_{\text{single-channel-LSA-emb}}$	**0.242 ± 0.019**	0.206 ± 0.015	0.197 ± 0.019	0.241 ± 0.016	0.211 ± 0.015	0.195 ± 0.015
deepSimDEF$_{\text{multi-channel-LSA-emb}}$		**0.251 ± 0.013**			0.254 ± 0.012	

We also conducted the same experiment working with human gene expression data. As presented in Table 3.7, we observe deepSimDEF outperforms all other gene function similarity measures when compared against human gene expression data in terms of Pearson correlation.

5 .4 Correlation with Sequence Similarity

Proteins with similar sequence are usually homologous and thus have a similar function [176, 38]. For that reason, proteins in a newly sequenced genome are routinely annotated using the sequences of similar proteins in other genomes.

Every gene pair in our sequence homology data is accompanied by the LRBS and RRBS scores indicating the level of sequence similarity of their component genes.

Table 3.7: Pearson's correlation of deepSimDEF and human gene expression data for three sub-ontologies compared with other FS measures aggregated by BMA

Semantic Measure	Excluding IEA			Including IEA		
	BP	CC	MF	BP	CC	MF
Resnik [184]	0.225	0.373	0.225	0.242	0.368	0.242
Lin [110]	0.217	0.319	0.231	0.212	0.355	0.216
Jiang and Conrath [78]	0.195	0.345	0.186	0.185	0.375	0.201
GraSM [37]	0.255	0.402	0.265	0.237	0.396	0.237
AIC [200]	0.226	0.343	0.212	0.207	0.351	0.215
clusteredGO [45]	0.215	0.346	0.212	0.205	0.336	0.225
simGIC [171]	0.106	0.131	0.122	0.102	0.168	0.134
simDEF [153]	0.307	0.411	0.202	0.336	0.418	0.221
AicInferSentGO [44]	0.312	0.376	0.209	0.346	0.411	0.228
$\text{deepSimDEF}_{\text{single-channel-rand-emb}}$	0.448 ± 0.028	0.414 ± 0.029	0.333 ± 0.031	0.452 ± 0.025	0.418 ± 0.027	0.343 ± 0.024
$\text{deepSimDEF}_{\text{multi-channel-rand-emb}}$		0.438 ± 0.024			0.434 ± 0.021	
$\text{deepSimDEF}_{\text{single-channel-LSA-emb}}$	**0.425 ± 0.021**	**0.401 ± 0.022**	**0.319 ± 0.024**	**0.434 ± 0.027**	**0.411 ± 0.017**	**0.321 ± 0.022**
$\text{deepSimDEF}_{\text{multi-channel-LSA-emb}}$		**0.434 ± 0.017**			**0.457 ± 0.014**	

Pesquita et al. [172] noted the relationship between semantically-derived shared information from Gene Ontology and RRBS is non-linear. Therefore, in our experiment with sequence data, the results of non-linear Spearman's correlations were primarily considered for the evaluation of the FS measures (see Table 3.9 for Pearson's correlation results).

Table 3.8 shows deepSimDEF outperformed all the existing FS measures in the correlation task with the yeast sequence homology data (the pretrained single-channel deepSimDEF improved the correlation results by >8% for RRBS, and >7% for LRBS). In contrast to deepSimDEF, among the previous FS measures, there existed no single measure that was superior to all others with respect to all the three sub-ontologies of BP, CC and MF; additionally, compered to IC-based measures, distributional definition-based measures consistently worked better. The multi-channel deepSimDEF improved these FS results even more by at least 3% (>10% and ∼11% compared to AicInferSentGO for RRBS and LRBS scores, respectively).

In the previous FS measures, MAX and BMA metrics also showed inconsistency in their correlation results with sequence homology data as depending on the measure and the sub-ontology of choice one metric could work better than the other. deepSimDEF architecture design, however, fundamentally alleviates this dependency on the manually engineered aggregation metrics of such. Additionally, initialization of the deepSimDEF networks with the pretrained embeddings improved the correlation results with the sequence homology data in RRBS for ∼1%; this increase for LRBS was ∼2%. Regarding RRBS and LRBS comparison, we noticed that the FS measures correlated better with RRBS scores in all cases with respect to Spearman's

Table 3.8: Spearman's correlation of deepSimDEF and other FS measures for three sub-ontologies against yeast sequence homology (RRBS and LRBS) (IEA+)

Semantic Measure		**LRBS** BP	**LRBS** CC	**LRBS** MF	**RRBS** BP	**RRBS** CC	**RRBS** MF
Resnik [184]	MAX	0.389	0.401	0.342	0.563	0.438	0.432
	BMA	0.443	0.409	0.312	0.546	0.415	0.491
Lin [110]	MAX	0.391	0.392	0.384	0.483	0.486	0.477
	BMA	0.452	0.401	0.362	0.564	0.526	0.531
Jiang and Conrath [78]	MAX	0.398	0.383	0.397	0.483	0.483	0.468
	BMA	0.452	0.398	0.386	0.557	0.528	0.491
GraSM [37]	MAX	0.398	0.392	0.397	0.486	0.489	0.477
	BMA	0.458	0.401	0.386	0.579	0.522	0.531
AIC [200]	MAX	0.391	0.392	0.384	0.492	0.484	0.473
	BMA	0.452	0.409	0.386	0.564	0.526	0.491
clusteredGO [45]	MAX	0.392	0.386	0.397	0.483	0.489	0.477
	BMA	0.458	0.409	0.362	0.579	0.516	0.529
simGIC [171]		0.415	0.381	0.396	0.554	0.516	0.552
simDEF [153]	MAX	0.661	0.628	0.673	0.683	0.641	0.689
	BMA	0.682	0.602	0.646	0.693	0.628	0.703
AicInferSentGO [44]	MAX	0.663	0.633	0.673	0.676	0.644	0.689
	BMA	0.687	0.602	0.651	0.698	0.635	0.701
deepSimDEF	single-channel-rand-emb	0.757 ± 0.047	0.701 ± 0.037	0.751 ± 0.049	0.791 ± 0.035	0.713 ± 0.041	0.802 ± 0.041
deepSimDEF	multi-channel-rand-emb		0.788 ± 0.034			0.809 ± 0.032	
deepSimDEF	single-channel-LSA-emb	**0.769 ± 0.029**	**0.719 ± 0.031**	**0.765 ± 0.036**	**0.799 ± 0.042**	**0.724 ± 0.034**	**0.814 ± 0.035**
deepSimDEF	multi-channel-LSA-emb		**0.806 ± 0.028**			**0.819 ± 0.026**	

correlation, and deepSimDEF was not an exception to that.

The Pearson's correlation results between the FS measures and the sequence homology data are shown in the Table 3.9.

Table 3.9: Pearson's correlation of deepSimDEF and other FS measures for three sub-ontologies against yeast sequence homology (RRBS and LRBS) (IEA+)

Semantic Measure		**LRBS** BP	**LRBS** CC	**LRBS** MF	**RRBS** BP	**RRBS** CC	**RRBS** MF
Resnik [184]	MAX	0.504	0.547	0.469	0.629	0.568	0.483
	BMA	0.558	0.549	0.437	0.615	0.548	0.547
Lin [110]	MAX	0.511	0.392	0.506	0.544	0.613	0.527
	BMA	0.567	0.545	0.489	0.631	0.655	0.578
Jiang and Conrath [78]	MAX	0.517	0.525	0.525	0.548	0.618	0.517
	BMA	0.571	0.544	0.507	0.623	0.661	0.544
GraSM [37]	MAX	0.518	0.531	0.519	0.546	0.621	0.525
	BMA	0.581	0.546	0.515	0.645	0.649	0.587
AIC [200]	MAX	0.511	0.534	0.513	0.552	0.613	0.521
	BMA	0.573	0.551	0.508	0.631	0.662	0.546
clusteredGO [45]	MAX	0.506	0.533	0.523	0.547	0.615	0.525
	BMA	0.577	0.556	0.489	0.643	0.646	0.576
simGIC [171]		0.533	0.521	0.516	0.622	0.644	0.606
simDEF [153]	MAX	0.778	0.768	0.799	0.751	0.769	0.736
	BMA	0.799	0.744	0.772	0.759	0.757	0.749
AicInferSentGO [44]	MAX	0.781	0.777	0.799	0.741	0.775	0.736
	BMA	0.806	0.742	0.779	0.761	0.768	0.751
deepSimDEF	single-channel-rand-emb	0.848 ± 0.043	0.838 ± 0.039	0.871 ± 0.038	0.851 ± 0.037	0.839 ± 0.042	0.848 ± 0.038
deepSimDEF	multi-channel-rand-emb		0.883 ± 0.031			0.877 ± 0.029	
deepSimDEF	single-channel-LSA-emb	**0.869 ± 0.026**	**0.845 ± 0.026**	**0.877 ± 0.028**	**0.853 ± 0.031**	**0.842 ± 0.029**	**0.851 ± 0.026**
deepSimDEF	multi-channel-LSA-emb		**0.892 ± 0.019**			**0.879 ± 0.017**	

5 .5 On the Importance of 'Highway Layer'

Prior to working with 'unseen' genes, a deepSimDEF network itself learns how the shared information of two 'known' genes should be transferred to a higher level representation. This learned representation will ideally dictate the degree of FS association

for all genes with respect to a particular biological task or data for which an assumption of gene FS is made already (e.g., the indication of function similarity/relatedness of genes that are closely linked to each other in a PPI network, or the association of homologous genes and their functionality). Even though this aggregation of the shared information can be learned by a fully-connected layer, we suspected a Highway network (described in Section Method, Subsection Highway Layer) could do this task more effectively due to its gating mechanism. The experiments on the validation and test split genes supported this belief. As we demonstrate in Table 3.10, regarding the PPI experiment, we achieved an increase of >1% when a Highway network was designed in the deepSimDEF architecture; we experienced the same range of improvement when we worked with the gene-expression and sequence homology data.

Table 3.10: F1-scores for deepSimDEF with a highway network compared to the deepSimDEF with a fully-connected layer in the task of PPI prediction

Network Architecture + Aggregation Layer		**Excluding IEA (%)**			**Including IEA (%)**		
		BP	CC	MF	BP	CC	MF
deepSimDEF sgl-ch	Fully connected	84.72	81.23	79.78	84.93	81.42	80.26
deepSimDEF mlt-ch	Fully connected		85.94			86.41	
deepSimDEF sgl-ch	Highway network	**85.42**	**82.41**	**80.78**	**85.84**	**82.77**	**81.49**
deepSimDEF mlt-ch	Highway network		**87.37**			**87.69**	

5 .6 'Negative Control' Experiments

To make sure correct annotations of gene products played an important role in the understanding of their functionality, and also a valid deepSimDEF model training, we conducted several negative control experiments by randomly assigning GO term annotations to the genes. For our random annotations, we considered several scenarios to be completely certain of the importance of correct GO annotations in the whole process. As presented in Figure 3.4, first, we fully stripped genes of their original GO term annotations and assigned fully random annotations to them and conducted experiments listed above, and then, slowly injected true annotations to the data to observe how the results changed by having more true GO term annotations assigned. We noticed as we added more true annotations to the training/test data the results improved. For example, the results of PPI experiments were ∼50% for the F1-score when fully random annotations were given to the genes; once we started to inject true

annotations to the data, those results began to increase. This observation was true for the correlation results of the gene expression and sequence homology experiments as well.

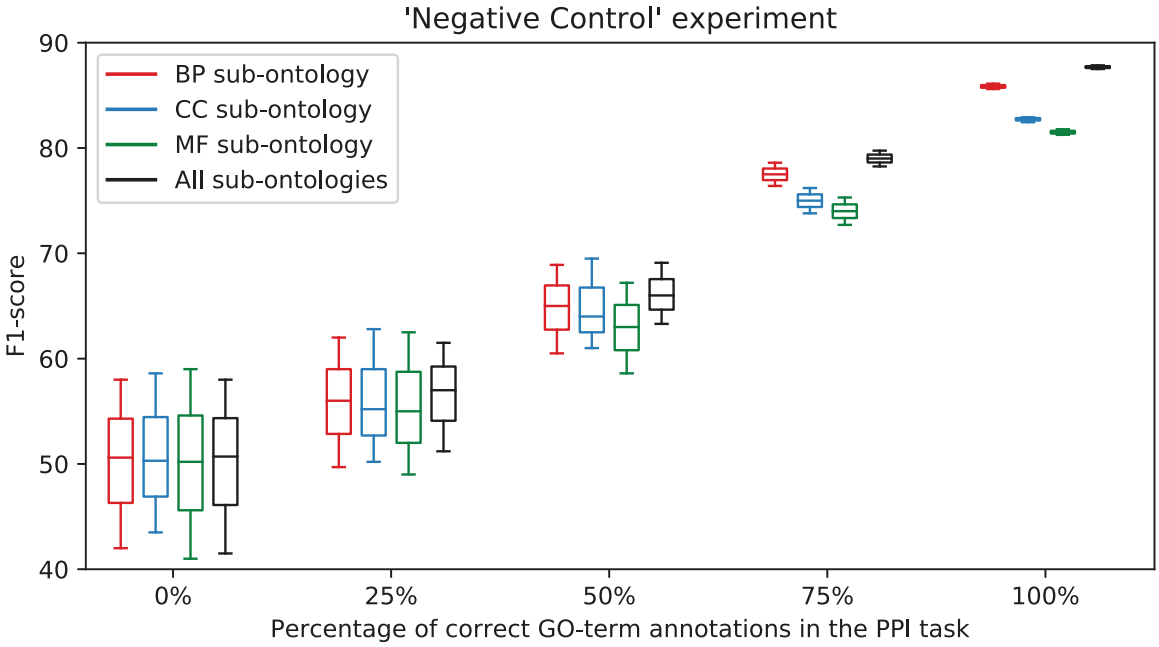

Figure 3.4: Negative control experiment to verify the importance of correct GO term annotations for a reliable model training (IEA+).
In the experiment we first stripped the examined gene from any correct GO term annotations and assigned completely random GO term annotations to them and then trained and tested the model based on the those GO term annotations. Then, we gradually removed random annotations and replaced them with the original and the correct GO term annotations to see the effect of correct annotations for model training and the prediction of PPIs. Having the embedding layers of the networks initialized with pretrained GO-term embeddings and the rest of weights randomly assigned, we repeated this experiment 10 times to find the mean and the variance of the F1-scores in each consideration.

6 Discussion

Besides automatically learning how to aggregate the shared information of the two compared genes (given their GO term annotations) through the means of a Highway Layer, another innovative aspect regarding deepSimDEF architecture is the systematic idea of using GO annotations of all three sub-ontologies at the same time. We

believe the absence of this notion in the other FS measures mainly stems from their dependence on the topological structure of Gene Ontology and the lack of certainty on how this basically distinguished information should be combined – which largely forced the existing FS measure to consider each sub-ontology separately and then offer the best one at the end. However, the multi-channel deepSimDEF network makes use of all the annotations concurrently. To this end, the multi-channel architecture introduces parallel biological information into a deepSimDEF network with multiple types of GO-term embeddings aiming to combine this distinguished yet entwined information, and subsequently, to further improve the performance of a biological application.

One important aspect regarding the hyper-parameter setting of the deepSimDEF networks was that for all the experiments, one set of hyper-parameters always helped to get the optimal results for the networks (multi- or single-channel). For example, if we changed the embeddings size in one experiment and observed a decline or an improvement in the results, for other experiments, we observed the same trend in the results applying the same changes to their networks. This helped the structure of the final model stay the same and not change from one experiment to another, which can be very beneficial to the future biological applications meant to be studied and investigated later.

7 Conclusion

deepSimDEF, a novel deep neural-based model for gene function prediction, results in a valuable low-dimensional distributed representation of GO terms and gene products (e.g., RNAs and proteins), also known as embeddings, and introduces a powerful, flexible, easily transferable and adaptable deep neural architectures that a wide range of problems in proteomics and genomics can benefit from. When applied to yeast database, our single-channel and multi-channel models outperformed the best-performing FS measures in the tasks of PPI prediction, correlation with gene expression as well as correlation with sequence homology data by gaining a large margin of improvement.

One important future work direction regarding deepSimDEF is extrinsic consideration of what it offers to address biological tasks to which FS and SS measures

have been applied already. These applications run the gamut from microRNA function analysis [149, 95, 236] and coexpression network construction [222, 70, 69, 39] to drug-discovery [201, 107, 112] and cancer treatment studies [191, 139, 108]. Some studies also intrinsically evaluated their FS measure against Enzyme Commission (EC) [150, 248] and protein family (Pfam) similarities [15]. Moreover, further investigation of miss-classified PPIs will help to improve deepSimDEf's prediction of positive interactions. deepSimDEF needs to be tested on the other species other than yeast as well. Also, more recent deep neural techniques may help to improve the quality of the pretrained GO-term embeddings. In spite of what we explained above, in an in-depth study, the prediction result of a deepSimDEF FS network can be seen secondary to the production of the GO-term and gene-product embeddings as, similar to what was suggested in [8, 128], they can help to acquire further insights into GO terms as well as gene products from the hidden (arithmetic) relationships among their embeddings. Also, adding an *attention mechanism* [237, 55] to the network architecture might further improve the FS results. Last but not least, in the context of *transfer learning*, more studies are needed to be done to discover how the learned information from a biological task for an organism can be transferred to another organism in an attempt to put their networks in a more proper state prior to training.

Chapter 4

Part III: Natural Language Concept Embedding for Word Sense Disambiguation
A single Bidirectional Long Short-Term Memory Network for Word Sense Disambiguation of Natural Text

1 Summary

Due to recent technical and scientific advances, we have a wealth of information hidden in unstructured text data such as offline/online narratives, research articles, and clinical reports. To mine these data properly, attributable to their innate ambiguity, a Word Sense Disambiguation (WSD) algorithm can avoid numbers of difficulties in Natural Language Processing (NLP) pipeline. However, considering a large number of ambiguous words in one language or technical domain, we may encounter limiting constraints for proper deployment of existing WSD models. This chapter attempts to address the problem of one-classifier-per-one-word WSD algorithms by proposing a single Bidirectional Long Short-Term Memory (BLSTM) network which through the consideration of senses and context sequences works on all ambiguous words collectively. Evaluated on SensEval-3 benchmark, we show the result of our model is comparable with top-performing WSD algorithms. We also discuss how applying additional modifications alleviates the model fault (i.e., consideration of two embedding spaces) and the need for more training data.
Publication – Original paper authored by Pesaranghader et al. [157] is available in: https://doi.org/10.1007/978-3-319-89656-4˙8 (In Proceedings of Canadian AI conference, Toronto, 2018)

2 Introduction

Word Sense Disambiguation (WSD) is an important problem in Natural Language Processing (NLP), both in its own right and as a stepping stone to other advanced tasks in the NLP pipeline, applications such as machine translation [215] and question answering [72]. WSD specifically deals with identifying the correct sense of a word, among a set of given candidate senses for that word, when it is presented in a brief narrative (surrounding text) which is generally referred to as *context*. Consider the ambiguous word '*cold*'. In the sentence "*He started to give me a cold shoulder after that experiment*", the possible senses for cold can be *cold temperature* (S1), *a cold sensation* (S2), *common cold* (S3), or *a negative emotional reaction* (S4). Therefore, the ambiguous word cold is specified along with the sense set {S1, S2, S3, S4} and our goal is to identify the correct sense S4 (as the closest meaning) for this specific occurrence of cold after considering - the semantic and the syntactic information of - its context.

In this effort, we develop our supervised WSD model that leverages a Bidirectional Long Short-Term Memory (BLSTM) network. This network works with neural sense vectors (i.e., *sense embeddings*), which are learned during model training, and employs neural word vectors (i.e. *word embeddings*), which are learned through an unsupervised deep learning approach called GloVe (Global Vectors for word representation)[151] for the context words. By evaluating our one-model-fits-all WSD network over the public gold standard dataset of SensEval-3 [126], we demonstrate that the accuracy of our model in terms of F-measure is comparable with the state-of-the-art WSD algorithms'.

We outline the organization of the rest of the chapter as follows. In Section 3 , we briefly explore earlier efforts in WSD and discuss recent approaches that incorporate deep neural networks and word embeddings. Our main model that employs BLSTM with the sense and word embeddings is detailed in Section 4 . We then present our experiments and results in Section 5 supported by a discussion on how to avoid some drawbacks of the current model in order to achieve higher accuracies and demand less number of training data which is desirable. Finally, in Section 7 , we conclude with some future research directions for the construction of sense embeddings as well as applications of such model in other domains such as biomedicine.

3 Background and Related Work

Generally, there are three categories of WSD algorithms: supervised, knowledge-based, and unsupervised. Supervised algorithms consist of automatically inducing classification models or rules from labeled examples [249]. Knowledge-based WSD approaches are dependent on manually created lexical resources such as WordNet [129] and the Unified Medical Language System[1] (UMLS) [158]. Unsupervised algorithms may employ topic modeling-based methods to disambiguate when the senses are known ahead of time [89]. For a thorough survey of WSD algorithms refer to Navigli [134].

3 .1 Neural Embeddings for WSD

In the past few years, there has been an increasing interest in training neural word embeddings from large unlabeled corpora using neural networks [33][127]. Word embeddings are typically represented as a dense real-valued low dimensional matrix $\boldsymbol{W}$ (i.e. a *lookup table*) of size $d \times v$, where d is the predefined embedding dimension and v is the vocabulary size. Each column of the matrix is an embedding vector associated with a word in the vocabulary and each row of the matrix represents a latent feature. These vectors can subsequently be used to initialize the input layer of a neural network or some other NLP model. GloVe [151] is one of the existing unsupervised learning algorithms for obtaining these vector representations of the words in which training is performed on aggregated global word-word co-occurrence statistics from a corpus.

Besides word embeddings, recently, computation of sense embeddings has gained the attention of numerous studies as well. For example, Chen et al. [24] adapted neural word embeddings to compute different sense embeddings (of the same word) and showed competitive performance on the SemEval-2007 data [136].

3 .2 Bidirectional LSTM

Long Short-Term Memory (LSTM), introduced by Hochreiter and Schmidhuber (1997) [67], is a gated recurrent neural network (RNN) architecture that has been designed to

[1] https://www.nlm.nih.gov/research/umls/

address the vanishing and exploding gradient problems of conventional RNNs. Unlike feedforward neural networks, RNNs have cyclic connections making them powerful for modeling sequences. A Bidirectional LSTM is made up of two reversed unidirectional LSTMs [60, 79]. For WSD this means we are able to encode information of both preceding and succeeding words within context of an ambiguous word, which is necessary to correctly classify its sense.

4 One Single BLSTM Network for WSD

Given a document and the position of a target word, our model computes a probability distribution over possible senses related to that word. The architecture of our model, depicted in Figure 4.1, consist of 6 layers which are a sigmoid layer (at the top), a fully-connected layer, a concatenation layer, a BLSTM layer, a cosine layer, and a sense and word embeddings layer (on the bottom).

In contrast to other supervised neural WSD networks in which generally a softmax layer - with a cross entropy or hinge loss - is parameterized by the context words and selects the corresponding weight matrix and bias vector for each ambiguous word's senses [86][208], our network shares parameters over all words' senses. While remaining computationally efficient, this structure aims to encode statistical information across different words enabling the network to select the true sense (or even a proper word) in a blank space within a context.

Due to the replacement of their softmax layers with a sigmoid layer in our network, we need to impose a modification in the input of the model. For this purpose, not only the contextual features are going to make the input of the network, but also, the sense for which we are interested to find out whether that given context makes sense or not (no pun intended) would be provided to the network. Next, the context words would be transferred to a sequence of word embeddings while the sense would be represented as a sense embedding (the shaded embeddings in Figure 4.1). For a set of candidate senses (i.e., $\{s_1, ..., s_n\}$) for an ambiguous term, after computing *cosine similarities* of each sense embedding with the word embeddings of the context words, we expect the sequence result of similarities between the true sense and the surrounding context communicate a pattern-like information that can be encoded through our BLSTM network; for the incorrect senses this premise does not hold. Several WSD studies

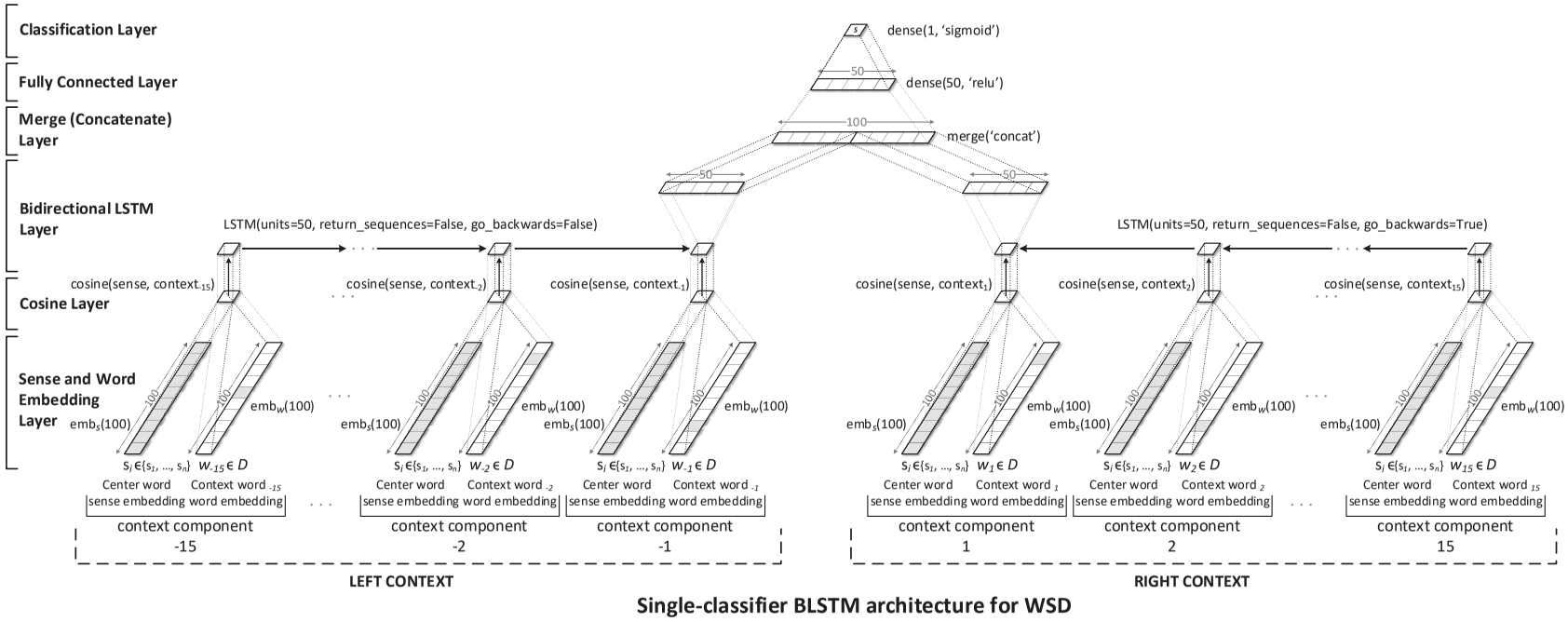

Figure 4.1: The single model of deep Bidirectional LSTM for Word Sense Disambiguation of text data.
A series of (left and right) context components are centered around the ambiguous word. The cosine similarities between the context words and the examined sense as the outputs of the first two layers are fed to two LSTM networks with different directions. Then, the concatenated outputs of LSTMs are fed to a (binary) neural network sense classifier consisting of one fully-connected layer and a sigmoid unit. Finally, an argmax over the outputs of all the sigmoids for the set of candidate senses selects the true sense, confirming this sequence of cosine similarities is the best match for the correct sense based on the learned cosine similarities patterns during training of the network.

already incorporated the idea of sense-context cosine similarities in their models [123].

4 .1 Model Definition

For one instance (or one document), the input of the network consists of a sense and a list of context words (left and right) which paired together form a list of context components. For the context D which encompasses the ambiguous term T, that takes the set of predefined candidate senses $\{s_1, ..., s_n\}$, the input for the sense s_i for which we are interested in to find out whether the context is a proper match will be determined by Equation (4 .1). Then, this input is copied (next) to $|D|$ positions of the context to form the first pair of the context components.

$$\boldsymbol{l}_i = \boldsymbol{W}_s^l \cdot \boldsymbol{v}_s(s_i),\ i \in \{1, ..., n\}. \tag{4 .1}$$

Here, $\boldsymbol{v}_s(s_i)$ is the one-hot representation of the sense corresponding to $s_i \in \{s_1, ..., s_n\}$. A one-hot representation is a vector with dimension V_s consisting of $|V_s|-1$ zeros and a single one which index indicates the sense. The V_s size is equal to the number of all senses in the language (or the domain of interest). Equation (4 .1) will have the effect of picking the column (i.e. sense embeddings) from $\boldsymbol{W}_s^l$ corresponding to that sense. The $\boldsymbol{W}_s^l$ (stored in the sense embeddings lookup table) is initialized randomly since no sense embedding is computed a priori.

Regarding the context word inputs that form the second pairs of context components, at position m in the same context D the input is determined by:

$$\boldsymbol{x}_m = \boldsymbol{W}_w^x \cdot \boldsymbol{v}_w(w_m),\ m \in \{-|D|/2, ..., -2, -1, 1, 2, ..., |D|/2\}. \tag{4 .2}$$

Here, $\boldsymbol{v}_w(w_m)$ is the one-hot representation of the word corresponding to $w_m \in D$. Similar to a sense one-hot representation (V_s), this one-hot representation is a vector with dimension V_w consisting of $|V_w|-1$ zeros and a single one which index indicates the word in the context. The V_w size is equal to the number of words in the language (or the domain of interest). Equation (4 .2) will choose the column (i.e. word embeddings) from $\boldsymbol{W}_w^x$ corresponding to that word. The $\boldsymbol{W}_w^x$ (stored in the word embeddings lookup table) can be initialized using pre-trained word embeddings; in

this work, GloVe vectors are used.

On the other hand, the output of the network that is examining sense s_i is

$$\hat{y}_{s_i} = \sigma(\boldsymbol{W}_{out} \cdot \boldsymbol{h}_{cl} + \boldsymbol{b}_{out}),\ s_i \in \{s_1, ..., s_n\} \tag{4 .3}$$

where $\boldsymbol{W}_{out} \in R^{1\times 50}$ and $\boldsymbol{b}_{out} \in R$ are the weights and the bias of the classification layer (sigmoid), and $\boldsymbol{h}_{cl}$ is the result of the merge layer (concatenation).

When we train the network, for an instance with the correct sense and the given context as inputs, $\hat{y}_{s_i}$ is set to be **1.0**, and for incorrect senses they are set to be **0.0**. During testing, however, among all the senses, the output of the network for a sense that gives the highest value of $\hat{y}_{s_i}$ will be considered as the true sense of the ambiguous term, in other words, the correct sense would be:

$$\arg\max_{s_i}\{\hat{y}_{s_1}, ..., \hat{y}_{s_n}\},\ s_i \in \{s_1, ..., s_n\}\ . \tag{4 .4}$$

By applying softmax to the result of estimated classification values, $\{\hat{y}_{s_1}, ..., \hat{y}_{s_n}\}$, we can show them as probabilities; this facilitates interpretation of the results.

Further, the hidden layer h_{cl} is computed as

$$\boldsymbol{h}_{cl} = ReLU(\boldsymbol{W}_h \cdot [\boldsymbol{h}^L_{C_{-1}}; \boldsymbol{h}^R_{C_{+1}}] + \boldsymbol{b}_h) \tag{4 .5}$$

where $ReLU$ means rectified linear unit; $[\boldsymbol{h}^L_{C_{-1}}; \boldsymbol{h}^R_{C_{+1}}]$ is the concatenated outputs of the right and left traversing LSTMs of the BLSTM when the last context components are met. $\boldsymbol{W}_h$ and $\boldsymbol{b}_h$ are the weights and bias for the hidden layer.

4 .2 Validation for Selection of Hyper-parameters

SensEval-3 data [126] on which the network is evaluated, consist of separate training and test samples. In order to find hyper-parameters of the network 5% of the training samples were used for the validation in advance. Once the hyper-parameters are selected, the whole network is trained on all training samples prior to testing. As to the loss function employed for the network, even though is it common to use *(binary) cross entropy* loss function when the last unit is a sigmoidal classification, we

observed that *mean square error* led to better results for the final *argmax* classification (Equation (4 .4)) that we used. Regarding parameter optimization, RMSprop [64] is employed. Also, all weights including embeddings are updated during training.

4 .3 Dropout and Dropword

Dropout [202] is a regularization technique for neural network models where randomly selected neurons are ignored during training. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass, and any weight updates are not applied to the neuron on the backward pass. The effect is that the network becomes less sensitive to the specific weights of neurons, resulting in better generalization, and a network that is less likely to overfit the training data. In our network, dropout is applied to the embeddings as well as the outputs of the merge and fully-connected layers.

Following the dropout logic, *dropword* [76] is the word level generalizations of it, but in *word dropout* the word is set to zero while in dropword it is replaced with a specific tag. The tag is subsequently treated just like one word in the vocabulary. The motivation for doing dropword and word dropout is to decrease the dependency on individual words in the training context. Since by replacing word dropout with dropword we observed no change in the results, only word dropout was applied to the sequence of context words during training.

5 Experiments

In SensEval-3 data (*lexical sample task*[2]), the sense inventory used for nouns and adjectives is WordNet 1.7.1 [129] whereas verbs are annotated with senses from Wordsmyth[3]. Table 4.1 presents the number of words under each part of speech, and the average number of senses for each class.

As stated, training and test data are supplied as the instances of this task; and the task consist of disambiguating one indicated word within a context.

[2]http://www.senseval.org/senseval3
[3]http://www.wordsmyth.net/

Table 4.1: Summary of senses in SensEval-3

Class	Number of words	Average senses
Nouns	20	5.8
Verbs	32	6.31
Adjectives	5	10.2
Total	57	6.47

Table 4.2: Hyper-parameter used for the experiments and the ranges that were searched during tuning. '-' indicates no tuning was performed on that parameter.

Hyper-prameter	Range searched	Values used
Context size	[10, 100] [Left, Right]	[15 Left, 15 Right]
Embedding size	{50, 100, 200, 300}	100
BLSTM hidden layer size	[50, 300]	2*50
Dropout on sense/word embeddings	[0, 50%]	20%
Dropout on LSTM outputs	[0, 70%]	50%
Dropout on fully-connected layer	[0, 70%]	50%
Word dropout	[0, 50%]	20%
Sense embedding initialization	-	Random $\in$ unif(-0.1, 0.1)
Word embedding initialization	-	GloVe[4] (uncased)

5 .1 Experimental Settings

The hyper-parameters that were determined during the validation is presented in Table 5.6. The preprocessing of the data was conducted by lower-casing all the words in the documents and removing numbers. This results in a vocabulary size of $|V| = 29044$. Words not present in the training set are considered unknown during testing. Also, in order to have fixed-size contexts around the ambiguous words, the padding and truncating are applied to them whenever needed.

5 .2 Results

Between-all-models comparisons - When SensEval-3 task was launched 47 submissions (supervised and unsupervised algorithms) were received addressing this task. Afterwards, some other papers tried to work on this data and reported their results in separate articles as well. We compare the result of our model with the top-performing

[4]Wikipedia and Gigaword (400K vocab): https://nlp.stanford.edu/projects/glove/

Table 4.3: F-measure results for SensEval-3 (English lexical samples)

Rank	Method	F-measure(%)
1	Multi-classifier BLSTM [86]	73.4
1	IMS+adapted CW [208]	73.4
2	htsa3 [61]	72.9
3	IRST-Kernels [205]	72.6
4	**Our Single-classifier BLSTM**$_{least-square-error-loss}$	**72.5**
5	**Our Single-classifier BLSTM**$_{cross-entropy-loss}$	**72.4**
6	nusels [102]	72.4
35	IRST-Ties	58.9
37	R2D2	57.2
39	NRC-Coarse	48.5
40	NRC-Coarse2	48.4
42	DLSI-UA-LS-SU	44.4

and low-performing algorithms [5] (supervised). We show our single model sits among the 5 top-performing algorithms, considering that in other algorithms for each ambiguous word one separate classifier is trained (i.e. in the same number of ambiguous words in a language there have to be classifiers; which means 57 classifiers for this specific task). Table 5.7 shows the results of the top-performing and low-performing supervised algorithms.

The first two algorithms represent the state-of-the-art models of supervised WSD when evaluated on SensEval-3. Multi-classifier BLSTM [86] consists of deep neural networks which make use of pre-trained word embeddings. While the lower layers of these networks are shared, upper layers of each network are responsible to individually classify the ambiguous that word the network is associated with. IMS+adapted CW [208] is another WSD model that considers deep neural networks and also uses pre-trained word embeddings as inputs. In contrast to Multi-classifier BLSTM, this model relies on features such as POS tags, collocations, and surrounding words to achieve their result. For these two models, softmax constitutes the output layers of all networks. htsa3 [61] was the winner of the SensEval-3 lexical sample. It is a Naive Bayes system applied mainly to raw words, lemmas, and POS tags with correction of the a-priori frequencies. IRST-Kernels [205] utilizes kernel methods for pattern

[5] low-performing algorithms are listed for a better comparison among the supervised WSD models

Table 4.4: WSD single-classifier BLSTM with other pieces or hyper-parameters

Network (Our Single-classifier)	**F-measure(%)**
Full network in Figure 4.1	**72.5**
BLSTM with reverse directions in Figure 4.1	68.9
BLSTM with a shuffled context	67.3
Fully-connected layers instead of BLSTM layer	70.2
BLSTM without GloVe for the context (all weights are random)	65.6
BLSTM without word dropout	71.1
BLSTM with a larger context size [25 left, 25 right]	71.4

abstraction, paradigmatic and syntagmatic information and unsupervised term proximity on British National Corpus (BNC), in SVM classifiers. Likewise, nusels [102] makes use of SVM classifiers with a combination of knowledge sources (part-of-speech of neighboring words, words in context, local collocations, syntactic relations. The second part of the table lists the low-performing supervised algorithms [126]. Considering their ranking scores we see that there are unsupervised methods that outperform these supervised algorithms.

Within-our-model comparisons - Besides several internal experiments to examine the importance of some hyper-parameters to our network, we investigated if the sequential follow of cosine similarities computed between a true sense and its preceding and succeeding context words carries a pattern-like information that can be encoded with BLSTM. Table 4.4 presents the results of these experiments.

The first row shows the best result of the network that we described above (and depicted in Figure 4.1). Each of the other rows shows one change that we applied to the network to see the behavior of the network in terms of F-measure. In the middle part, we are specifically concerned about the importance of the presence of a BLSTM layer in our network. So, we introduced some fundamental changes in the input or in the structure of the network. Generally, it is expected that the cosine similarities of closer words (in the context) to the true sense be larger than the incorrect senses' [123]; however, if a series of cosine similarities can be encoded through an LSTM (or BLSTM) network should be experimented. We observe if reverse the sequential follow of information into our Bidirectional LSTM, we shuffle the order of the context

words, or even replace our Bidirectional LSTMs with two different fully-connected networks of the same size 50 (the size of the LSTMs outputs), the achieved results were notably less than 72.5%.

In the third section of the table, we report our changes to the hyper-parameters. Specifically, we see the importance of using GloVe as pre-trained word embeddings, how word dropout improves generalization, and how context size plays an important role in the final classification result (showing one of our experiments).

6 Discussion

From the results of Table 4.3, we notice our single WSD network, despite eliminating the problem of having a large number of WSD classifiers, still falls short when is compared with the state-of-the-art WSD algorithms. Based on our intuition and supported by some of our preliminary experiments, this deficiency stems from an important factor in our BLSTM network. Since no sense embedding is made publicly available for use, the sense embeddings are initialized randomly; yet, word embeddings are initialized by pre-trained GloVe vectors in order to benefit from the semantic and syntactic properties of the context words conveyed by these embeddings. That is to say, the separate spaces that the sense embeddings and the (context) word embeddings come from enforces some delay for the alignment of these spaces which in turn demands more training data. Furthermore, this early misalignment does not allow the BLSTM to fully take advantage of larger context sizes which can be helpful. Our first attempt to deal with such problem was to pre-train the sense embeddings by some techniques - such as taking the average of the GloVe embeddings of the (informative) definition content words of senses, or taking the average of the GloVe embeddings of the (informative) context words in their training samples - did not give us a better result than our random initialization. Our preliminary experiments though in which we replaced all GloVe embeddings in the network with sense embeddings (using a method proposed by Chen et al. [24]), showed considerable improvements in the results of some ambiguous words. That means both senses and context words (while they can be ambiguous by themselves) come from one vector space. In other words, the context would also be represented by the possible senses that its words can take. This idea not only can help to improve the results of the current model, it

can also avoid the need for a large amount of training data since senses can be seen in both places, center and context, to be trained.

7 Conclusion

In contrast to common one-classifier-per-each-word supervised WSD algorithms, we developed our single network of BLSTM that is able to effectively exploit word orders and achieve comparable results with the best-performing supervised algorithms. This single WSD BLSTM network is language and domain independent and can be applied to resource-poor languages (or domains) as well. As an ongoing project, we also provided a direction which can lead us to the improvement of the results of the current network using pre-trained sense embeddings.

For future work, besides following the discussed direction in order to resolve the inadequacy of the network regarding having two non-overlapping vector spaces of the embeddings; this in turn would lead to less number of labelled data needed for trainingn. We plan to examine the network on technical domains such as biomedicine as well. In this case, our model will be evaluated on MSH WSD dataset[6] prepared by National Library of Medicine[7] (NLM). Also, construction of sense embeddings using (extended) definitions of senses [153][163] can be tested. Moreover, considering that for many senses we have at least one (lexically) unambiguous word representing that sense, we also aim to experiment with unsupervised (pre-)training of our network which benefits form quarry management by which more training data will be automatically collected from the web. For future work, the current model can be evaluated against extrinsic tasks such as text symmetrization or more specific applications serving as multi-term topics focused crawling [166, 165] in online/streaming environments [168, 167, 169].

[6] https://wsd.nlm.nih.gov/collaboration.shtml
[7] https://www.nlm.nih.gov/

Chapter 5

deepBioWSD: a one-size-fits-all network for an effective deep neural Word Sense Disambiguation of biomedical text data

1 Summary

Objective – In biomedicine, there is a wealth of information hidden in unstructured narratives such as research articles and clinical reports. To exploit these data properly, a word sense disambiguation (WSD) algorithm prevents downstream difficulties in the natural language processing applications pipeline. Supervised WSD algorithms largely outperform un- or semisupervised and knowledge-based methods; however, they train 1 separate classifier for each ambiguous term, necessitating a large number of expert-labeled training data, an unattainable goal in medical informatics. To alleviate this need, a single model that shares statistical strength across all instances and scales well with the vocabulary size is desirable.

Materials and Methods – Built on recent advances in deep learning, our deepBioWSD model leverages 1 single bidirectional long short-term memory network that makes sense prediction for any ambiguous term. In the model, first, the Unified Medical Language System sense embeddings will be computed using their text definitions; and then, after initializing the network with these embeddings, it will be trained on all (available) training data collectively. This method also considers a novel technique for automatic collection of training data from PubMed to (pre)train the network in an unsupervised manner.

Results – We use the MSH WSD dataset to compare WSD algorithms, with macro and micro accuracies employed as evaluation metrics. deepBioWSD outperforms existing models in biomedical text WSD by achieving the state-of-the-art performance of 96.82% for macro accuracy.

Conclusions – Apart from the disambiguation improvement and unsupervised training, deepBioWSD depends on considerably smaller amount of expert-labelled data as

it learns the target and the context terms jointly. These merit deepBioWSD to be conveniently deployable in real-time biomedical applications.

Publication – Original paper authored by Pesaranghader et al. [154] is available in: https://doi.org/10.1093/jamia/ocy189 (Journal of Oxford JAMIA)

2 Introduction

With recent advances in biomedicine, we see a massive amount of biomedical text data being generated every day. To gain knowledge from these data, developing natural language processing (NLP) tools that mine them accurately within a reasonable time is crucially important. NLP components that include named entity recognition programs[63] syntactic parsers [56], and relation extractors [100, 118] build the foundation of many high-level biomedical information extraction and knowledge discovery applications [97, 142, 101]. Also, it is shown that the biomedical text data such as scientific articles [19], clinical narratives [87], and health-related social media posts [188], abound with ambiguous terms (hereafter, instead of saying *ambiguous word* we use *ambiguous term* because a [biomedical] conceptual unit that we try to disambiguate can be represented by a series of words; as in *malignant B-cell lymphoma* or *benign B-cell lymphoma* for the target ambiguous term *B-cell lymphoma*). In the lowest level, surrounded by this innate ambiguity, all other components and the full biomedical application will suffer if it is not resolved properly.

A word sense disambiguation (WSD) algorithm attempts to predict the correct sense of a term within a context given a set of candidates. For example, in the sentence "Ca intakes in the United States and Canada appear satisfactory among young adults," the sense set for *Ca* consists of *Canada* (s_1), *California* (s_2), *calcium* (s_3), and *cornu ammonis* (s_4) and the goal is to predict the correct sense s_3 for this specific occurrence of *Ca*. It is shown that this automatically identifying the intended sense of ambiguous words improves the performance of clinical and biomedical applications such as medical coding and indexing,[178, 130] detection of adverse drug event,[219] automatic medical reporting,[32, 223] and other secondary uses of data such as information retrieval and extraction,[138] and question-answering systems.[185] These capabilities are becoming essential tasks due to the growing

amount of information available to researchers, the transition of healthcare documentation and patient-practitioner interaction toward electronic health records and automatic expert systems, and the push for quality and efficiency in health care.

Supervised machine learning WSD algorithms typically build one separate classifier for each ambiguous term, which will be trained solely on the instances of that term. That is, to train an accurate WSD model, a large amount of annotated instances are needed, the curation of which will be expensive and labor-intensive particularly in health informatics [175, 243]. Recent studies in the biomedical domain incorporate expert-involved active learning techniques to accelerate the labeling process of this training data [225, 224]. Nevertheless, considering the multiclassifier design of the traditional supervised WSD models, the real-world implementation of them in the domain is still impracticable.

We introduce a one-size-fits-all deepBioWSD architecture for disambiguation of biomedical text data, a deep learning-based model that unifies all disambiguation classifiers into 1 single network. In a supervised manner, this network will be trained on all existing instances of the ambiguous terms as 1 group of training data in which sense-context pair and $s_i \in \{1.0, 0.0\}$ constitute the input and the output, respectively. While the network encodes the shared information among all instances, for a given training-instance, it learns the senses of the unlabeled terms in the context and the sense of the labeled center term at the same time. To this end, our architecture employs a bidirectional long short-term memory network (BLSTM), and works with neural sense embeddings, which can be pretrained.

3 Supervised WSD in Biomedicine

Jimeno-Yepes et al. [81] prepared the National Library of Medicine's MSH WSD dataset in 2011 with naive Bayes accuracy of 93.84% (NB [these abbreviations are used during evaluation of the WSD algorithms]). Later, traditional discriminative models with rigorous linguistic and biomedical specific features were used for WSD evaluation [124, 14]. To avoid an intense feature engineering, recently, the state-of-the-art accuracy of 95.97% was achieved by Jimeno-Yepes [239] using unigrams and word embeddings with support vector machines ($\text{SVM}_{\text{Yepes}}$); they also reported the accuracy of 94.87% for their long short-term memory networks (LSTMs). In

another supervised model, Antunes and Matos [5] used bag-of-words as local features and word embeddings as global features and reported accuracy of 95.6% when SVM classifiers were employed ($SVM_{Ant\text{-}Mat}$). To eliminate an extreme need for extensive amount of annotated data to train classifier of each term, Sabbir et al. [187] recently developed a knowledge-based model at the cost of accuracy (92.24%, KN). In another recent knowledge-based study, Duque et al. [50] reported accuracy of 71.52% on MSH WSD for their system called Bio-Graph that employs a PageRank algorithm to work with occurrence graphs built from MEDLINE abstract to address WSD (Bio-Graph). Moreover, MetaMap [7] is a highly configurable program developed at the National Library of Medicine (NLM) to map biomedical text to the UMLS Metathesaurus or, equivalently, to discover Metathesaurus concepts referred to in text. MetaMap for WSD uses a knowledge-intensive approach based on symbolic, natural-language processing (NLP) and computational-linguistic techniques.

4 Neural Embeddings for WSD

With recent interests in training neural word embeddings from large raw corpora [33, 127, 151], several studies included pretrained word embeddings in their WSD models, some of which were concerned with biomedical text [74, 144, 86, 157]. Lately, computation of sense embeddings has gained the attention of researchers as well where their importance in the WSD tasks has been investigated [12, 148, 24]; however, the mapping of these hardly interpretable inducted sense embeddings to a sense inventory (e.g., the Unified Medical Language System [UMLS]) has been the main bottleneck for their wider employment in WSD systems [16]. In the deepBioWSD model, first, we build our sense embeddings using the UMLS text definition of senses; then, these embeddings initialize our BLSTM network before training.

5 Bidirectional LSTM

LSTMs address the vanishing gradient problem in RNNs by incorporating gating functions into their state dynamics [67] (see Appendix A). Standard Recurrent Neural Networks (RNNs) and LSTMs, however, have restrictions as the future input information cannot be reached from the current state, so, a Bidirectional LSTM fuses

1 forward and 1 backward LSTM [60]. In WSD, this means we are able to encode the information of both preceding and succeeding words with respect to a pivotal ambiguous term. Kgebck and Salomonsson [86] proposed a partially shared multiclassifier WSD model with BLSTMs that employed word embeddings ($\text{BLSTMs}_{\text{Kg-Sal}}$). In our previous work which we described in Chapter 4, we developed a single-classifier WSD model with just 1 BLSTM network ($\text{BLSTM}_{\text{Pes-etal}}$) [157]; this model, however, uses 2 separate word and sense spaces for the context and center words, which caused inconsistency and worse performance. As we will see, the deepBioWSD network is only dependent on sense space for both center and context terms, an architectural improvement over $\text{BLSTM}_{\text{Pes-etal}}$ network for better sense prediction, faster training, and less dependency on expert-labeled data. Other existing BLSTM-based WSD algorithms are Seq2Seq-inspired models, which typically underperform conventional supervised WSD models [207, 181, 1].

6 Zero-shot Learning

Zero-shot learning (ZSL) aims at predicting labels for instances that belong to classes that were not directly seen during training [2, 186]. The underlying secret ensuring the success of ZSL is to find an intermediate semantic representation to transfer the knowledge learned from seen classes to unseen ones [244]. The scalability of the model is of utmost importance since a large amount of unlabeled data is generally present and can be received by interaction with the environment [91], which is the case in medical informatics. We show deepBioWSD with a unitary and uniform network architecture that it offers benefits from ZSL as ambiguous terms in the context would be trained indirectly at the time of direct training on another ambiguous term's labelled instances (leading to need for less amount of training instances); that also, in turn, prevents the "cold start" problem that exists in other supervised WSD algorithms. Regarding cold start problem, when a model cannot draw any inferences as it has not yet gathered sufficient information related to a subject matter or application; hence, training of the model from scratch with sufficient amount of labelled data seems inevitable.

7 Experimental Data

7 .1 Unified Medical Language System

The UMLS (https://www.nlm.nih.gov/research/umls/) is a terminology integration system that contains Metathesaurus and SPECIALIST Lexicon. The Metathesaurus holds ∼3.4 million biomedical and clinical concepts (hereafter, we use concept and sense [of a term] interchangeably) by maintaining their hierarchical relationships. Each concept has a unique identifier called CUI (Concept Unique Identifier), a set of representative terms, and a text definition. The Metathesaurus provided us with the sense sets of the ambiguous terms. The SPECIALIST Lexicon resource contains information about common English vocabulary and biomedical terms by offering tools for language processing. We used its programs to demarcate terms in the contexts; in our early example, *the United States* is an unambiguous term (CUI: C0041703) consisting of 3 words, and *satisfactory* is a single-word ambiguous term (C0205410, C1547307). The latest UMLS release 2018AA was used in the study. This release covers >83 000 ambiguous representative terms.

7 .2 MEDLINE Abstracts

MEDLINE includes over 20 million citations of life sciences and biomedical articles from 1966 to the present. Combined with the UMLS concept definitions, we employed MEDLINE 2013 bigram-list (https://mbr.nlm.nih.gov/Download/) to create our sense embeddings.

7 .3 Validation Datasets

We employed the MSH WSD dataset (https://wsd.nlm.nih.gov/collaboration.shtml) for the evaluation of WSD algorithms [81]. This dataset provides 37,888 instances for 203 ambiguous terms (including abbreviations) that take 25 senses (∼100 instances per each sense are provided). Prepared from MEDLINE, every instance of a target ambiguous term is manually annotated with a CUI within the sense set of that term. For example, an instance of *Ca* is labeled with either C0006823 (Canada), C0006675 (California), C0006754 (calcium), or C3887642 (cornu ammonis); while every instance of the target term *lymphogranulomatosis* takes the sense C0036202

(benign lymphogranulomatosis) or C0019829 (malignant lymphogranulomatosis).

8 Materials and Methods

8 .1 Pretraining of Sense Embeddings

Inspired by studies for (high-dimensional) distributed representation of biomedical concepts [115, 153, 162], and low-dimensional vector representation of words [105, 11], we pretrained UMLS sense embeddings in 6 steps as depicted in Figure 5.1. In essence, the second-order computation of vector representation of concepts prevents the issue of sparsity (of word features) in the first-order vector representation of their definitions, pointwise mutual information statistically defines the degree of relevance between each biomedical concept and its (second-order) word features, and latent semantic analysis aims at condensing the final high-dimensional vectors to a size proper for a deep neural network. These steps briefly explained below are executed in advance to compute sense embeddings of the UMLS concepts before training our deepBioWSD network which they initialize (see below and Subsection 8 .2 for further details of each part).

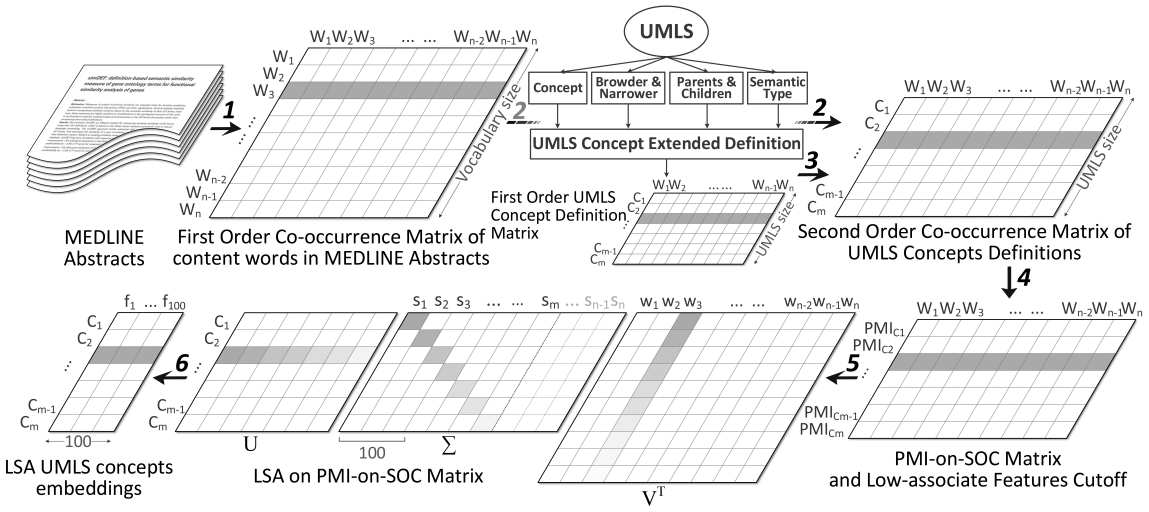

Figure 5.1: Definition-based sense embedding model for the UMLS concepts. The figure represents different steps in our unsupervised method to generate low-dimensional sense embeddings for the Unified Medical Language System (UMLS) concepts. These embeddings initialize of disambiguation deep neural network. C: concept; f: new feature; LSA: latent semantic analysis; PMI: pointwise mutual information; S: salient feature; SOC: second-order co-occurrence; W: word feature.

Step 1 – Bigrams and MEDLINE word co-occurrence matrix. We built a co-occurrence matrix from the bigram-list of MEDLINE abstract. This matrix is symmetric and sparse, and represents the contextual information of the MEDLINE words.

Step 2 – UMLS concept definition extension and definition matrix. The definition extension of concepts by their neighbour concepts' in an ontology/thesaurus enriches their semantic [143, 116]. When applied to the UMLS concepts, words in the extended definitions have associated co-occurrence vectors from MEDLINE computed in step 1. For every (extended) definition, the definition matrix stores the frequency of these word features.

Step 3 – Second-order co-occurrence (SOC) matrix. To build a SOC vector of a concept, we first summed the MEDLINE co-occurrence vectors of the content words in that concept's extended definition, and then normalized the result vector by the number of words in the definition. In other words, we took the centroid of the vectors associated with each word in the definition, and then normalized the result to uniformly treat the different size definitions.

Step 4 – Pointwise mutual information (PMI) on SOC matrix. Not all word features associated with a concept are equally important. PMI, as in Equation (8 .1), statistically measures the level of association between the concepts (their associated words; i.e., $word_i$) and the word features (i.e., $word_j$), instead of naive consideration of word feature frequency cutoff threshold [152, 164, 160]. Once PMI values are calculated - with respect to the (frequency) probabilities of (co-)occurrences of these words, our validation set helps to set a low cutoff threshold for the removal of irrelevant features. We applied the add-1 smoothing technique to the SOC matrix in advance to avoid bias toward infrequent occurrences [153, 159].

$$PMI(sense_i, word_j) = log \frac{p(sense_i, word_j)}{p(sense_i) \times p(word_j)} \qquad (8\ .1)$$

Step 5 – Latent semantic analysis (LSA) on PMI-on-SOC matrix. LSA, given by

Equation (8 .2), uses a singular value decomposition algorithm that resulted 2 square and unitary matrices $\boldsymbol{U}$ and $\boldsymbol{V}^T$, and a non-negative diagonal matrix $\boldsymbol{\Sigma}$ that holds singular values on its diagonal in a non-increasing order.

$$PMI\_on\_SOC = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{8 .2}$$

Step 6 – Reducing the rank of singular values. Having Equation (8 .3), we truncated the singular value decomposition to 100 for low-dimensional representation of UMLS concepts. Determined by our validation set, smaller embedding sizes yielded worse WSD results, and higher dimensions did not improve the accuracy and just increased the training time.

$$sense\_embeddings = \boldsymbol{U}\boldsymbol{\Sigma}_{100} \tag{8 .3}$$

8 .2 The Rationale Behind Different Considerations in Pre-trained Sense-embeddings Method

Bigrams

The bigrams of the MEDLINE abstracts are used to compute unsupervised sense-embeddings prior to supervised training of the network which works with senses of any size of words (this step is executed separately to compute sense-embeddings before starting to train the network). These pre-computed sense-embeddings initialize the weight of the first layer of the network in order to put the network in proper state. The benefits are 2-fold: faster training and more-importantly avoiding to fall into local minima in the backpropagation process which could have happened with a greater probability if the network's embedding weights were initialized randomly (i.e., better classification result with pre-trained sense-embeddings as presented in Table 5.7).

To calculate sense-embeddings, our method is inspired by a work done by Lie et al. [115] work in which they use bigrams. In their study, the bigrams, combined with the text definition of UMLS concepts, produce distributed representations of the UMLS

concepts on which we build our sense-embeddings after adding PMI and LSA to the approach to get more accurate final presentations that are low-dimensional.

The idea of using "bigrams+definitions" instead of only definition is to avoid sparsity in vector representation and to avoid the"hard semantic overlap" of the definitions. Just as a toy example assume we have two different concepts with definitions provided as below:

- $\boldsymbol{Sense_1}$: *Chemotherapy is a regimen typically taken for cancer treatment.*
- $\boldsymbol{Sense_2}$: *A combination of immunotherapy and radiation therapy to treat metastasis cells.*

If we only consider the definition overlap between these two definitions (their exact content words), there is no overlap or relatedness between them whatsoever, except for the uninformative word "a" (i.e., the first-order vectors fails similarity estimation here). However, by considering bigram vectors created from MEDLINE for each content word in the definitions we can overcome this hard overlap. This is because in MEDLINE chemotherapy usually occurs in the context in which *immunotherapy* and *radiation* most probably occur, i.e. have more similar bigram vectors. The same reasoning for *therapy* and *treatment*, as well as *cancer* and *metastasis*. If we computed bigram vectors of all content words in definitions in advance using MEDLINE abstracts, the second-order vector representation of a definition would be the summation of the bigram-vectors of the content words in that definition (the centroid of all bigram-vectors associated with that definition). Once computed for all definitions, we observe the results convey concept similarity/relatedness more accurately.

Definition Extension

In the above-mentioned work by Lie et al. [115] (Figure 6 in their paper), they presented by having richer text definitions for concepts, the method results in more accurate vector representations, hence, they suggested the idea of definition extension by their immediate parents and children in the UMLS. In our experiments, we observed the same trend, however not that dramatic as if any lacking in the generated sense-embeddings, the supervised training of the network later on attempts to cover for that. Moreover, definition extension is the safest option since it (almost) causes

no extra computational overhead as the size of the matrices remain the same with and without them (i.e., the number of word features and concepts are the same). For further detailed justification on the importance of definition extension you can refer to the subsection 4 .2.

Pointwise Mutual Information (PMI)

Not all word-features associated with a concept are equally important. We employ PMI to deal with Gloss Vectors (above paper) low/high frequency cut-off phase when addressing the degree of importance. This naive approach of cutting is indecisive and causes loss of valuable information. This is because, first, we believe when a concept is specific (or concrete) in its meaning it needs more specific (or informative) words in its definition describing that concept. Second, it is that when one term is more specific in the meaning has lower frequency in the corpus compared to those terms that are general (or abstract). As an example of these considerations, we can juxtapose these two definitions:

- ***Cancer***: *A term for diseases in which abnormal cells divide without control and can invade nearby tissue.*
- ***Gliosarcomas***: *Rare mixed tumors of the brain and the spinal cord which contain malignant neuroectodermal (glial) and mesenchymal components including spindleshaped fibrosarcoma cells.*

We can see the term *gliosarcomas* is described with more specific and detailed terms which are naturally less frequent in the corpus (consequently their bigrams are of lower frequency as well). Therefore, the above-mentioned issues call into question the idea of fixed frequency cut-off points since by removing the low frequency terms/bigrams as we face a trouble defining pointed concepts like *gliosarcomas* since we waste some beneficial and expressive amount of information. Therefore, before considering any strict low and high cut-off points to remove features, we need to measure the relative level of association between concepts and their word-features; the help of PMI measure (Equation (8 .1)) achieves this.

It is highly possible that $word_i$ (as in row for a concept) and $word_j$ (as in column for a feature-word) co-occur in the MEDLINE with rather low frequency but they can

be associated with and descriptive of each other (e.g., very few yet important concepts are dependent on $word_i$). The problem of the conventional low and high frequency cut-off in Gloss Vector measure is that it takes into account only the frequency of $word_i$ and $word_j$ co-occurred in the MEDLINE without being concerned with frequencies of $word_i$ and $word_j$ individually describing concepts. Once computed, for the removal of uninformative word-features, 0.7 was found to be the appropriate cut-off point with the help of our validation sets (i.e., low association values were changed to 0.0).

8 .3 deepBioWSD Network Definition

In contrast to other supervised WSD networks, in which a *softmax* layer with a *cross-entropy* or *hinge loss* is often parametrized to select the corresponding weight matrix and bias vector for every sense of an ambiguous term, our network shares parameters over all senses. Given an instance and the position of a target term, the deepBioWSD network computes a probability distribution over candidate senses of that term.

The architecture of our network consists of 7 layers (Figure 5.2). Due to the replacement of the conventional softmax layer with a linear or sigmoid layer, we imposed a modification to the input. That is, apart from the contextual features, the sense for which we want to discover whether the given context is meaningful will be provided as input. For an ambiguous term with the sense set $\{s_1, , s_n\}$, the network runs n times (for every sense) and the highest-confidence sense would be selected. In lower layers, to determine proximity of the senses and the given context, after computing cosine similarities of each candidate sense (embedding) with the senses of the context terms, the sequential result of the cosine similarities between the correct sense and the surrounding context communicate a pattern-like information that our BLSTM layer encodes which consequently yields higher confidence in the upper regression layer; however, for the incorrect senses, this premise of homogeneity and proximity does not hold (i.e., match and mismatch classification of a sense and a given context). Several studies already incorporated the idea of sense-context cosine similarities in their WSD models.[157, 52, 123] Nevertheless, the context terms, which are determined by the SPECIALIST Lexicon during the disambiguation process, can be ambiguous themselves. To deal with their ambiguity, just before the cosine layer,

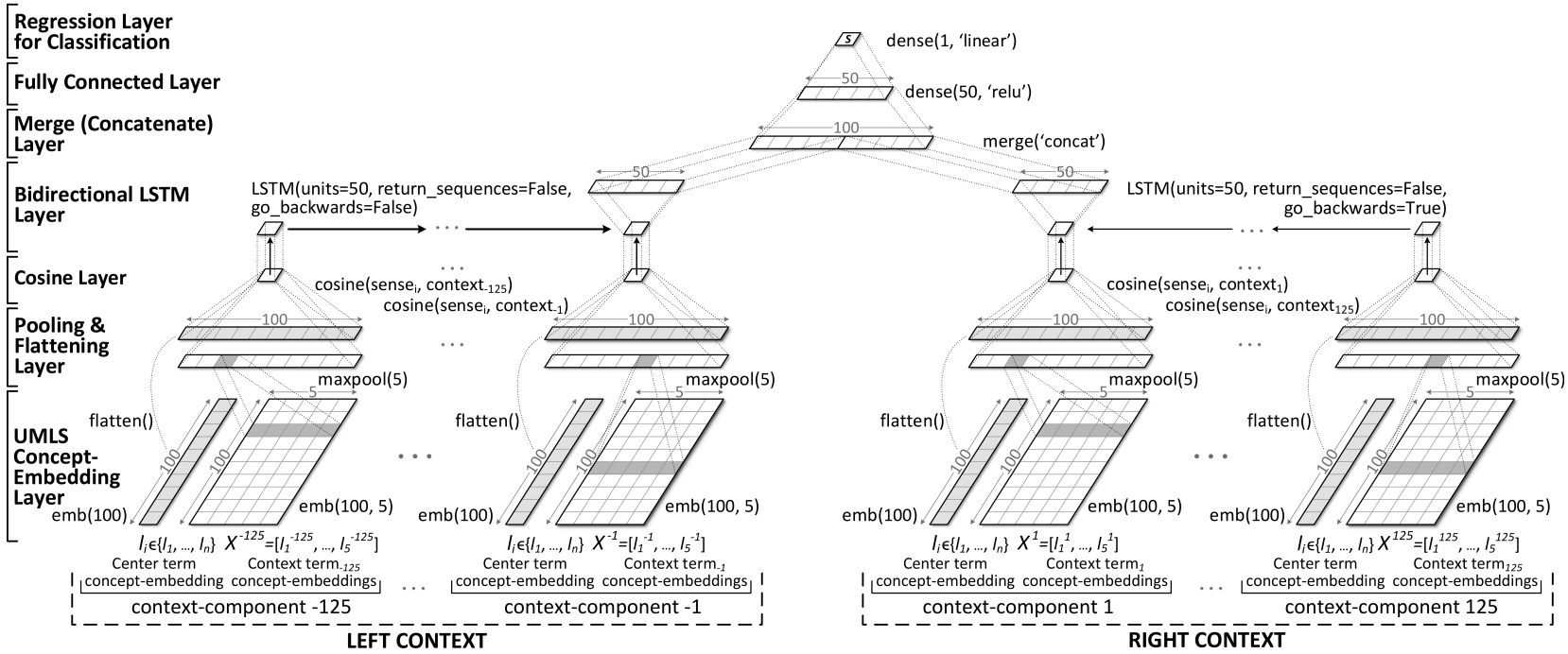

Figure 5.2: deepBioWSD network architecture.

The figure illustrates our 1-size-fits-all deepBioWSD network which treats all center ambiguous terms (and their instances) uniformly. The *emb* represents embedding size, l_i is the current candidate sense (or label) under investigation, and $\boldsymbol{X}_j$ is the jth term in the context (left or right). Besides training on the center terms, the embeddings of the context terms would be updated and learned (i.e., disambiguated) during training. LSTM: long short-term memory network; UMLS: Unified Medical Language System.

a pooling layer is devised, the result of which learns the senses of the ambiguous terms appeared in the context. This means the network takes gradients with respect to (shared) sense embeddings of both the target term and the context terms at the same time.

UMLS Concept Embedding Layer. For one instance, the input of the network consists of a sense and a list of (left and right) context terms, which paired together form a list of context components. For context D, which encompasses an ambiguous term with the sense set of $\{s_1, , s_n\}$, the embedding layer weights for the examined input sense s_i, is determined by Equation (8 .4). Then, this input is copied to $|D|$ positions of the context to form the first pair of the context-components and set the same embedding weights in the layer.

$$\boldsymbol{l}_i = \boldsymbol{W}_s^l \cdot \boldsymbol{v}_s(s_i), \quad i \in \{1, ..., n\} \tag{8 .4}$$

where, $l_i \in \mathbb{R}^{100}$ and $v_s(s_i)$ is the 1-hot representation of the sense. A 1-hot representation is a vector with the dimension V_s consisting of $|Vs|-1$ zeros and a single one that indicates the index of a sense in a look-up table; the V_s size is equal to the number of CUIs in the UMLS. The $\boldsymbol{W}_s^l$ is the shared look-up table for the center terms and context terms; it is initialized with the sense embeddings that we computed in advance. Equation (8 .4) have the effect of picking the column (i.e., a sense embedding) from $\boldsymbol{W}_s^l$ corresponding to that sense.

Regarding a context term input, which forms the second pair of a context component, at position k in the same context D the embedding weights are determined by:

$$\boldsymbol{x}^k = [\boldsymbol{l}_1, ..., \boldsymbol{l}_m] \in \mathbb{R}^{100 \times m}, \quad k \in \{-|D|/2, ..., -2, -1, 1, 2, ..., |D|/2\} \tag{8 .5}$$

where, l_i is set by Equation (8 .5), and k is the position of the term in the context (left

or right) while $|D|/2$=125 is a hyperparameter of the network (padding or truncating was applied wherever needed). m is another hyperparameter that typically should be equal to the size of the largest sense set; however, in the experiments of the study we observed an inverse relationship between the sense set size and the occurrence frequency of the terms, therefore we limited m to be 5. This means only those terms in the context were inputted to the network that had the sense set of size 5 or less (i.e., some infrequent terms were ignored). This resulted in a faster convergence with no accuracy loss. For those terms with the sense set size of <5, a generic embedding vector of very large negative numbers was employed to fill in the void senses; this helped *maxpooling* consider only the sense embeddings of a context term.

Pooling and Flattening Layer. Here, max operation is applied over all rows per each context terms sense embeddings, denoted as maxpool(5) in Figure 5.2. After maxpooling, each context term is represented with a 100-dimensional global feature vector. We also flattened the result column vector into a row vector as an integrated part of the maxpooling layer; that is, at position k in the context, the pooling and flattening layer gives $\bar{l}^k \in \mathbb{R}^{1\times 100}$ for a target term sense and $\overline{x}^k \in \mathbb{R}^{1\times 100}$ for the predicted context term sense. Despite the intuitive use case of maxpooling to deduce the proper sense of a context term, experimentally it worked better than *averagepooling*.

Cosine Layer. In $|D|$ positions of context components, the cosine similarities between the embedding vector of the examined sense and the maxpooled of the context terms are calculated. Computed by Equation (8 .6), the results are 2 row-vectors of size $|D|/2$ each containing the cosine similarities of the context components of their side:

$$\begin{aligned} \boldsymbol{c}_{left} &= [c_l^1, ..., c_l^k] \in \mathbb{R}^{1\times |D|/2}, \quad k \in \{-|D|/2, ..., -2, -1\} \\ \boldsymbol{c}_{right} &= [c_r^1, ..., c_r^k] \in \mathbb{R}^{1\times |D|/2}, \quad k \in \{1, 2, ..., |D|/2\} \\ c^i &= cosine(\bar{\boldsymbol{l}}^i, \overline{\boldsymbol{x}}^i) = \frac{\sum \bar{\boldsymbol{l}}^i \odot \overline{\boldsymbol{x}}^i}{\|\bar{\boldsymbol{l}}^i\| \times \|\overline{\boldsymbol{x}}^i\|}, \quad i \in \{1, ..., k\} \end{aligned} \tag{8 .6}$$

Bidirectional LSTM Layer. With 1 forward and 1 backward LSTM networks, we have a left context-dedicated LSTM network that receives the cosine similarities from left to right, and right context-dedicated LSTM network that receives the cosine similarities from right to left. $\boldsymbol{c}_{left}$ and $\boldsymbol{c}_{right}$ are the inputs of these networks; their outputs are the vectors $\boldsymbol{h}_{left} \in \mathbb{R}^{1\times 50}$ and $\boldsymbol{h}_{right} \in \mathbb{R}^{1\times 50}$, each encoding the received information from one side of the target ambiguous term (50 is another hyperparameter).

Concatenation Layer. This layer concatenates the output row vectors of the BLSTM layer:

$$\boldsymbol{h}_{merge} = [\boldsymbol{h}_{left}, \boldsymbol{h}_{right}] \in \mathbb{R}^{1\times 100} \tag{8 .7}$$

Fully Connected Layer. Further, for a better representation, a hidden fully connected layer $\boldsymbol{h}_{fc}$ is devised which is:

$$\boldsymbol{h}_{fc} = ReLU(\boldsymbol{h}_{merge} \cdot \boldsymbol{W}_h + \boldsymbol{b}_h) \in \mathbb{R}^{1\times 50} \tag{8 .8}$$

where, *ReLU* is rectified linear unit function; $\boldsymbol{W}_h \in \mathbb{R}^{100\times 50}$ and $\boldsymbol{b}_h \in \mathbb{R}^{1\times 50}$ are the weights and bias for the hidden layer. The result of this layer embeds the input sequence into a vector of size 50.

Regression for Classification Layer - This layer outputs a single value that is computed by:

$$\hat{y}_{s_i} = \boldsymbol{h}_{fc} \cdot \boldsymbol{W}_{out} + \boldsymbol{b}_{out}, \quad s_i \in \{s_1, ..., s_n\} \tag{8 .9}$$

where, $\boldsymbol{W}_{out}$ and $\boldsymbol{b}_{out}$ are the weights and the bias of the regression for classification layer (linear), and $\boldsymbol{h}_{fc}$ is the result of the previous hidden fully-connected layer.

During network training, for an instance with its given context and the correct sense as inputs, $\hat{y}_{s_i}$ is set to be **1.0**, whereas for the same context with incorrect senses it is set to be **0.0**. During testing, however, among all the senses, the output of the network for a sense that gives the highest value of $\hat{y}_{s_i}$ is considered as the true sense of the ambiguous term. In other words, the correct sense is:

$$\underset{s_i}{\text{argmax}}\{\hat{y}_{s_1}, ..., \hat{y}_{s_n}\}, \quad s_i \in \{s_1, ..., s_n\} \tag{8 .10}$$

By applying softmax to the results of the estimated values $\{\hat{y}_{s_1}, ..., \hat{y}_{s_n}\}$, we can represent them as probabilities. This will facilitate interpretation of them especially when deepBioWSD is benefiting from an *active learning* setting where intricacy and importance of one instance can be measured.

The final recommended hyperparameters of the network which were determined during validation are provided in Table 5.6.

8 .4 Unsupervised Collection of Training Data

Considering the uniform structure of deepBioWSD, we also aimed at collecting more training data on which deepBioWSD could be pretrained. For this purpose, we employed Entrez Direct (https://www.ncbi.nlm.nih.gov/books/NBK179288/) to automatically gather data from PubMed. So, we devised a *query management* scheme that benefited from the notion of polyonymy of a concept (polyonomy is the employment of multiple names for the same concept): besides ambiguous representative terms, usually, one concept has other representative terms that are unambiguous (e.g., *lymphogranulomatosis* vs *malignant lymphogranulomatosis*). By sending queries to PubMed for these unambiguous terms, we obtain abstracts for which we already know the true sense. It allowed us to create samples of unsupervised instances in a

large quantity (see the following subsection). For each (unambiguous) sense query, we only considered the first 500 instances retrieved from PubMed (excluding the MSH WSD instances). A total of 180,175 instances were automatically prepared as PubMed returned <500 abstracts for some sense queries.

Entrez Direct

Since this study aimed at developing a single network that is convenient to pre-train and maintain - a model that can benefit from continuous learning - we made use of Entrez Direct to automatically collect training data from PubMed. Entrez Direct (https://www.ncbi.nlm.nih.gov/books/NBK179288/) is a set of executables that provides a UNIX command line interface to the E-utilities which makes access to the NCBI's suite of interconnected databases (PubMed, Gene, Structure, and etc.) possible.[12] We explore how to pre-train the network with these automatically collected data prior to the supervised training. We also investigate how the network trained only on these data compares to an existing non-supervised WSD algorithm.

Here we represent how Entrez Direct helps to send *unique sense-queries* to PubMed and retrieve abstracts. The example that we provide here is for the ambiguous term *moles* with the sense-set: C0324740, C0027960.
The process of data preparation is executed in three steps (fully-automated):

1. Selection of unique representative terms of a concept/sense (in the UMLS)

2. Sending unique sense-queries to PubMed through Entrez Direct interface, collecting the data, and labeling them with the sense of the sense-query they are associated with (notice that the Entrez Direct query returns the abstracts in bulk)

3. Finding unique representative terms in the collected abstracts and replacing them with the ambiguous representative term

Step 1:

- List of representative terms for C0324740: talpidae, moles, talpidae family mole, family talpidae, mole Unique representative terms for C0324740: talpidae, family talpidae, talpidae family mole

- List of representative terms for C0027960: skin mole, naevus, nevi, nevus, moles, skin moles, mole of skin, mole Unique representative terms for C0027960: nevi, skin moles

Step 2:

- ≫ `esearch db pubmed query ‘((‘‘talpidae’’[TIAB] OR ‘‘family talpidae’’[TIAB] OR ‘‘talpidae family mole’’[TIAB] OR ‘‘moles’’[TIAB])) AND ‘‘moles’’[MESH])’ | efetch format xml > C0324740.xml`

- ≫ `esearch db pubmed query ‘((‘‘nevi’’[TIAB] OR ‘‘skin moles’’[TIAB])) AND ‘‘nevus’’[MESH])’ | efetch format xml > C0027960.xml`

Step 3 (sample excerpts retrieved and processed):

- PMID: 29309911 (one of the abstracts retrieved for the first query)

  - *Results unequivocally demonstrate that the presence of β4 Ser and β5 Gly, together with a low DPG sensitivity Hb phenotype, predates the radiation of the* ***~~family Talpidae~~ moles****, and did not evolve as an adaptation to fossorial life.* (True label: C0324740)

- PMID: 29718885 (one of the abstracts retrieved for the second query)

  - *All patients (100%) had multiple pigmented* ***~~nevi~~ moles*** *on the face and a lack or thinning of subcutaneous tissue around the neck and face.* (True label: C0027960)

Table 5.1: Sense similarity for candidate senses of the ambiguous term, *CP*

Cerebral Palsy	**Propionibacterium acnes**	**Cleft Palate**
Convulsion	Staphylococcus	Glossoptosis
Spastic syndrome	Propionibacterium	Cleft Lip
Muscle Dystonia	Stomatococcus	Omodysplasia
Dysdiadochokinesis	Micrococcus	Congenital Megacolon
Choreoathetosis	Flavobacterium	Ectromelia
Quadriplegia	Neisseriaceae	Polydactylism
Trismus	Acidovorax	Teething
Hemiplegia	Abiotrophia	Congenital Aniridia
Muscle Hypertonia	Paenibacillaceae	Omphalocele
Muscle Spasticity	Helicobacter	Syndactyly

9 Results

9 .1 Sense similarity of Pretrained Embeddings

We will see our method for pre-training of the sense embedding plays an important role in sense predictions. Table 5.1 represents an example of a sense similarity for the ambiguous term *Ca* that takes 4 different senses. In the table, instead of representing the uninterpretable identifiers of the concepts in the UMLS, their selected representative terms are shown. Providing just one example here, we observed that other senses followed the same sense similarity pattern in the sense embedding space.

Table 5.1 represents a (cosine) sense similarity example for the ambiguous term *CP* (computed over the pretrained sense embeddings; i.e., books in the library). Each column header represents one sense of *CP*, and the listed terms below are the closest UMLS concepts to that meaning of *CP*. In the table, instead of unfamiliar sense CUIs, the selected representative terms of the concepts are shown. Providing just 1 example here, we observed other senses followed the same sense similarity organization in the sense embedding space as well.

The following tables (Table 5.2, 5.3, 5.4, 5.5) represent the sense similarities for the ambiguous term *Iris*, *Sterilization*, *OCD*, *Ca*, respectively:

Table 5.2: Sense similarity for candidate senses of the ambiguous term, *Iris*

Eye iris	**Iris Plant**
Uvea	Capparis
Esodeviation	Bryonia
Entire orbital region	Trichosanthe
Exodeviation	Petiveria
Phoria	Daphne
Ophthalmoparesis	Pseudotsuga
Exophoria	Hydrangeaceae
Eye lens	Luffa
Epicanthal fold	Clusia

Table 5.3: Sense similarity for candidate senses of the ambiguous term, *Sterilization*

Reproductive Sterilization	**Sterilization**
Hysterotomy	Incineration
Hysterectomy	Freeze Drying
Ovariectomy	Blast
Cystectomy	Fluoropolymer
Ovariohysterectomy	Lightnings
Male Circumcision	Synchrotron
Orchidectomy	Low Density Lipoproteins
Vaginotomy	Polytetrafluoroethylene
Penectomy	Mercury

Table 5.4: Sense similarity for candidate senses of the ambiguous term, *OCD*

Obsessive Compulsive Disorder	**Osteochondritis Dissecans**
Alexithymia	Osteochondrosis
Murder	Brachymetatarsia
Hypomania	Entire bony skeleton
Forgetfulness	Melorheostosis
Apprehension	Epiphysis
Excitability	Osteoradionecrosis
Sluggishness	Exostosis
Hysteria	Bone Fracture
Paranoia	Condyle

Table 5.5: Sense similarity for candidate senses of the ambiguous term, *Ca*

Canada	**California**	**Calcium**	**Cornu ammonis**
Geographic Area	Ohio	Calcium Carbonate	Cerebellar Cortex
Racial Group	Alabama	Calcium Sulfate	Dentate Gyrus
Nova Scotia	Maryland	Mineral	Tissue of Brainstem
Country	Idaho	Silicate	Cerebellopontine Angle
United States	Montana	Apatite	Hindbrain
North America	Wyoming	Alkali Metals	Olfactory Cortex
America	West Virginia	Potassium Compound	Limbic System
New Brunswick	South Carolina	Lithium	Diencephalon
France	North Carolina	Ions	Olivary Nucleus

Table 5.6: Hyper-parameter settings in deepBioWSD network

Hyper-prameter	**Range searched**	**Values used**
Context size	[10, 150] [Left, Right]	[125 Left, 125 Right]
Embedding size	{50, 100, 200, 300}	100
BLSTM hidden layer size	[50, 300]	2*50
Dropout on sense embeddings	[0, 50%]	25%
Context term dropout	[0, 50%]	25%
Dropout on merged & fully-connected layers	[0, 70%]	50%
Sense embedding initialization	-	PMI-LSA pre-training Method

9 .2 Experimental Settings of the deepBioWSD Network

The network hyper-parameters that were determined during validation are presented in Table 5.6. In other words, this final architecture was discovered and was confirmed by our experiments on validation sets prior to reporting the results on the held-out test data. To have a fixed-size context, padding or truncating was applied wherever needed. Regarding optimization, RMSprop was employed.[213] Also, all weights including embeddings were updated during training. Moreover, all the context-sense inputs were shuffled during training.

9 .3 First WSD Experiment: Direct Learning From Center Terms

Between-all-models comparisons: Table 5.7 compares the deepBioWSD with the other WSD algorithms. Despite those for which we already had the accuracy results on MSH WSD dataset, $\text{BLSTMs}_{\text{Kg-Sal}}$ and $\text{BLSTM}_{\text{Pes-etal}}$ were reimplemented with their best hyperparameters chosen, a few of which were slightly different from

Table 5.7: Accuracy results for MSH-WSD dataset

Method	Algorithm	Macro Acc(%)	Micro Acc(%)
Unsupervised	Bio-Graph	71.52	-
	KB	92.24	-
	deepBioWSD$_{\textit{with random embeddings}}$	92.16	91.93
	deepBioWSD$_{\textit{with pre-trained embeddings}}$	**92.67**	**92.51**
Supervised	MetaMap with WSD	81.77	-
	NBs	93.84	-
	SVM$_{\text{Ant-Mat}}$	65.60	-
	LSTMs	94.87	94.78
	SVMs	95.97	95.81
	BLSTMs$_{\text{Kg-Sal}}$	95.64	95.47
	BLSTM$_{\text{Pes-etal}}$	95.53	95.39
Supervised (sigmoid)	deepBioWSD$_{\textit{with random embeddings}}$	93.53	93.40
	deepBioWSD$_{\textit{with pre-trained embeddings}}$	95.79	95.63
	deepBioWSD$_{\textit{pre-trained unsupervised w/o sense embdgs}}$	96.44	96.25
	deepBioWSD$_{\textit{pre-trained unsupervised w/ sense embdgs}}$	96.71	96.52
Supervised (linear)	deepBioWSD$_{\textit{with random embeddings}}$	93.88	93.71
	deepBioWSD$_{\textit{with pre-trained embeddings}}$	96.14	95.96
	deepBioWSD$_{\textit{pre-trained unsupervised w/o sense embdgs}}$	96.64	96.47
	deepBioWSD$_{\textit{pre-trained unsupervised w/ sense embdgs}}$	**96.82**	**96.64**

their original papers (e.g., different context size). What we report here for deepBioWSD is based on 10-fold validation experiments we conducted after considering training, validation, and test splits; other models might not necessarily follow this strategy.

Supervised. Instances of every term in 203 terms included in MSH WSD data were divided into 10 non-overlapping folds in which one fold was put aside for a final testing in a 10-time validation. Training on the rest of the 9 folds, we first randomly selected 5% as a validation set to tune hyperparameters and to find the proper number of epoch the network needed to train. After hyperparameters were chosen, the final model was trained on the whole training set (including the validation set), and then was evaluated on the 203 test data folds taken out already. In the experiments, macro and micro accuracies were considered for hyperparameter tuning as well as for the final evaluation of the test data (refer to Table 5.6 for the hyperparameters). After computing the test results of the all 10 times of validation, their average was considered as the results of the models. For the description of macro and micro accuracies refer to Appendix B.

Unsupervised. After finding the proper structure of the network, we experimented with 2 scenarios. First, the network was trained on the automatically collected data where the MSH WSD instances made the test data. Second, the network was pre-trained on these unsupervised data and then it was retrained and evaluated according to the supervised layout described previously.

These results indicate the importance of pretrained sense embeddings initializing the network. Their influence, however, is minimal when the network is pretrained on the unsupervised training data. In that case, the network produces sense embeddings from scratch, and the final updated embeddings are the byproduct of the network. Overall, deepBioWSDs single network architecture outperforms unsupervised KB and (multiclassifier) supervised WSD algorithms in the biomedical WSD task. Regarding training time, deepBioWSD also showed better efficiency.

In order to test time efficiency, we trained deepBioWSD network on a single GPU (NVIDIA GeForce GTX 980M) in a dedicated time-frame of 4 hours. For per-term WSD model of $\text{BLSTMs}_{\text{Kg-Sal}}$ [86] we divided this time by the number of terms 203 and assigned that portion of time to train classifier of each term in the model. The average result was far worse than what we gained for deepBioWSD when the network was only initialized by sense embeddings (86.87% vs. 93.67% for Macro accuracy). For $\text{BLSTM}_{\text{Pes-etal}}$, [157] (presented in Chapter 4) since we have two separate and disjoint spaces for the target term and context terms, the communication of information occurs very slowly as the alignment of these sense and word spaces is the first and utmost requirement for accurate predictions, which in turn demands more training data (86.17% vs. 93.67% for Macro accuracy, for the same time-frame of training). Also, by the same token for these models, we realized that in per-term models the full-training time of a classifier varies from one term to another as understandably some terms are more challenging to be trained (since they are treated in isolation). This case was less intense for deepBioWSD though as it attempted to learn about all senses jointly and more fairly; that is because at training time the network aims at making sense of the contexts (in the same space) at the same time rather than focusing on a single term. It means deepBioWSD can be trained for just a few hours (still not fully-trained) and then put in an application; in parallel the model can constantly continue towards full-training and updating itself while it is in use (with

Table 5.8: deepBioWSD with other architectural settings

Network (Our Single-classifier)	**Macro Accuracy(%)**
Full network in Figure 5.2	**96.82**
BLSTM with reverse directions in Figure 5.2	93.86
BLSTM with a shuffled context	91.98
Fully-connected layers instead of BLSTM layer	95.23
BLSTM on the left & BLSTM on the right	95.81

some resemblance to life-long machine learning) [25].

Regarding the storage to keep the training data in, we encountered no difficulty; however, it may cause problems when the training data grows larger. However, this can be easily dealt with using distributed storage systems. Moreover, due to batch training of the network, similar to deep per-term models, deepBioWSD can benefit from multi-GPU training and data parallelism for faster.

Within-our-model comparisons: We also studied if the flow of cosine similarities between a true sense and its preceding and succeeding terms (their senses) carried a sequential information that one BLSTM could encode and learn from. Therefore, according to what Table 5.8 represents, we introduced some changes in the input or in the structure of the network to verify that. We observed if we reverse the sequential flow of information into our BLSTM, we shuffle the order of the context terms, or replace our LSTMs with 2 fully connected networks of the same size 50, the achieved results were notably less than our original structure. Expectedly, due to a variable size of the original contexts (which forced padding/truncation), replacement of LSTMs with BLSTMs had negative effects.

9 .4 Second WSD Experiment: Indirect Learning from Context Terms

Considering zero-shot learning, we also experimented if training on one target term's instances led to indirect insights into other terms. As an example, assume we are training the ventricles instance, "Coronal measurements of both ventricles were similar when obtained by US and MRI images"; having *ventricles* (meaning *cerebral ventricles* here) as the target ambiguous term, we gain knowledge about the context terms as well, including *US* and *MRI*. In a new *US* instance, this insight helps the

Table 5.9: Accuracy results for indirect learning from the context terms

Stage	Supervised Setting	Macro Acc(%)	Micro Acc(%)
Before Training	deepBioWSD$_{\textit{with random embeddings}}$	49.37	49.53
	deepBioWSD$_{\textit{with pre-trained embeddings}}$	**65.46**	**65.73**
After Training	deepBioWSD$_{\textit{with random embeddings}}$	67.32	66.92
	deepBioWSD$_{\textit{with pre-trained embeddings}}$	**82.08**	**81.85**

network to predict if *US* means *United States* or *ultrasonography*.

To investigate indirect learning, we randomly divided 203 numbers of MSH WSD terms into 10 non-overlapping folds, and then held (instances of) one of the folds for testing (as unseen data) and the rest for training (10-time 10-fold validation). We selected 5% of the training set as a validation set to tune hyperparameters. The final network was trained on the whole training set and then was evaluated on the test set (averaged the individual test results on the unseen target terms).

Table 5.9 represents the average of the 10 times of validation. These results clearly represent the influence of pretrained sense embeddings on the predictions. More importantly, we observe, when deepBioWSD is not directly trained on one term's instances, the preserved statistical information learned from the context (and its maxpooled embeddings) guides the network for more accurate sense prediction of that term when located at the center. Furthermore, with the current state of the network, the model will not suffer from the cold start problem because the model has been gaining the momentum, and with smaller amount of training data needed, it will be fully trained on unseen terms in short order as well. Except for BLSTM$_{\text{Pes-etal}}$, for which the results of this experiment were completely random (in all cases), we could not envision and conduct the experiment for the other supervised algorithms due to their multiclassifier design.

10 Discussion

The deepBioWSD introduces an unorthodox WSD network in which all conceptual pieces of the biomedical domain (i.e., pivotal and contextual terms) are designed to be interconnected-pieces that constantly communicate information to solve the jigsaw puzzle of WSD. The network, however, found 2 types of instances challenging. First,

when the syntactic structure with similar semantic theme surrounding the candidate senses were very similar (e.g., *veterinary assistant* and *veterinary medicine* for the ambiguous term *veterinary*). Second, when the senses are semantically so close that they share the same immediate parent in the UMLS, or 1 term directly subsumes the other sense (as in senses for *borrelia*, *heregulin*, and *HGF* in the MSH-WSD dataset) (see Appendix C).

We let MeSH and SNOMED CT demarcate the context terms (following the previous studies) [52, 158]. We found however by adding more vocabularies from the UMLS, fewer context terms will be ignored during prediction as the model will be inclusive of more biomedical terms or senses. For example, the term *12-step program* appeared frequently in the context of *AA* when it meant *Alcoholics Anonymous* (another meaning is *amino acid*); however, *12-step program* belongs to neither MeSH nor SNOMED CT, whereas the National Cancer Institute ontology (NCI) covers it. This consideration of more vocabularies was helpful, as it slightly improved the results with a smaller context size needed. Nonetheless, with more vocabularies, the possible number of senses one term can take grows, which to some extent offsets the advantage of a smaller context size.

11 Conclusions

One future work direction can be consideration of other unsupervised biomedical sense embedding methods in the model. Adding an attention mechanism to the network architecture might further improve the disambiguation results as well [238]. Also, more comprehensive and systematic study for the collection of unsupervised training data is needed. The model can also be evaluated on an extrinsic task with real-world applications (e.g., Clinical Information Extraction) [138]. Moreover, adding more text data from external resources such as Wikipedia to the definitions of concepts can enrich their meaning and improve their vector representation, hence, an interesting study to investigate [160].

Chapter 6

Part IV: Epilogue
Conclusions and Future Work

1 Conclusions

We restate that as a priority, throughout the studies of this thesis we made sure that the guideline/checklist[1] from Dr. Joelle Pineau for machine learning reproducibility was met. The list is used by DL/ML community (e.g., NeuroIPS, ICML). For that reason, the source codes of the algorithms are shared and are publicly available on online repositories - refer to Section 2 .5.

The conclusions of the thesis are concerned with two individual components of the study discussed in Part 2 and Part 3.

1 .1 Biological Attribute Embedding for Function Analysis of Genes

Our approach for functional similarity estimation based on the shared context makes intuitive sense, as concepts which share closely related attributes in their representation should exhibit higher levels of similarity. We showed that implementing these ideas via the deep learning tools, which helped for low-dimensional distributed representations of GO terms and gene products, improved effectiveness of the correlation of functional similarity with sequence homology data, namely, LRBS and RRBS scores, gene expression, as well as protein interaction (PPI) data. For the yeast *Saccharomyces cerevisiae* database, relative to best-performing similarity measures, by considering all GO sub-ontologies (i.e., with a multi-channel deepSimDEF network), deepSimDEF increased PPI predictability by $\sim$4%, showed a correlation improvement $>$6% with gene expression data, and improved correlation with sequence homology by up to 11%. However, these increases using paired single-channel deepSimDEF networks was a little less. More importantly, these improvements compared to the

[1] https://www.cs.mcgill.ca/ jpineau/ReproducibilityChecklist.pdf

previous functional similarity measures, mainly those that use semantic similarity measures at the backend, was much larger indicating superiority of the deepSimDEF model over the traditional IC-based functional similarity measures.

Compared to IC-based measures, deepSimDEF has less dependency on ever-changing structure of GO. A key advantage of deepSimDEF in comparison with simDEF is its automatic and supervised training of the GO-term embeddings by means of one particular available biological data (i.e., the training dataset) and GO annotations. Later, these embeddings and the trained networks will be evaluated on a separate piece of data which is meant for testing. Once the testing passes all evaluation stages, the model will be trained on the whole data and will be available for use in a multitude of biological applications. One important phenomenon that deepSimDEF networks address is the method for aggregation of the GO annotations of two compared gene products. While all the previous similarity measures employ or introduce a human engineered metric such as MAX or BMA, deepSimDEF networks attempt to devise and propose the best possible way of aggregation of the shared information by means of a highway layer. Furthermore, in contrast with a single-channel network which deals only with one sub-ontology in GO, concurrent flow of annotations of the genes from all three sub-ontologies into the multi-channel deepSimDEF network provides the measure with richer information which consequently results in more accurate functional similarity estimation.

Succinctly, accomplishments of this part can be listed as follows:

1. We proposed a method for GO term vector embedding using their text definition

2. We built a deep neural network named deepSimDEF used for gene function analysis which offered:

   (a) single-channel and multi-channel networks for gene similarity estimation

   (b) PPI predictability increase by $\sim$4% (for yeast database, multi-channel architecture)

   (c) correlation improvement >6% with gene expression (for yeast database, multi-channel architecture)

(d) correlation improvement >4% with gene expression (for human database, multi-channel architecture)

(e) correlation improvement with sequence homology by up to 11% (for yeast database, multi-channel architecture)

(f) a method to embed genes and genes products in low-dimensional vector space based on their given GO annotations

1 .2 Natural Language Concept Embedding for Word Sense Disambiguation

This part of study addressed the critical problem of WSD in NLP by introducing and developing a novel deep Bidirectional Long Short-Term Memory (BLSTM) network named deepBioWSD. For the training of deepBioWSD, first, we initialized the BLSTM network using pretrained sense vector embeddings. Then, we trained the network on the biomedical textual data that was already manually annotated/labelled. As to the calculation of the pretrained sense embeddings, we made use of Unified Medical Language System (UMLS) and MEDLINE abstracts and also employed Pointwise Mutual Information (PMI) and Latent Semantic Analysis/Indexing (LSA/LSI). Finally, for evaluation, we tested the converged model on a holdout set that was absent during training. The experimental result on the MSH-WSD dataset (MeSH WSD dataset from National Library of Medicine, NLM) represented that the introduced deep learning model outperforms the state-of-the-art supervised methods in terms of accuracy results. Specifically, deepBioWSD achieves 96.82% for macro accuracy when was trained and evaluated based on the supervised labelled instances. In another scenario, when deepBioWSD was trained on training instances automatically collected from the web and annotated (in completely unsupervised fashion) it achieved the accuracy of 92.67%, outperforming the state-of-the-art unsupervised WSD models. We also showed that deepBioWSD was capable of predicting the sense of the terms for which we do not have any direct training data instances. This is done through a semi zero-shot-learning training of deepBioWSD, meaning the information regarding those (unlabelled) terms were learned when they occurred in the context of the directly annotated ambiguous terms.

Succinctly, accomplishments of this part can be listed as follows:

1. We introduced a method for sense embedding of semantic natural language units (i.e., concepts) using their text definition
2. We built a deep neural network named deepBioWSD used for word sense disambiguation of biomedical text which offered:
   (a) a single BLSTM WSD network that considerably needs less number of training instances
   (b) this network can be trained in an unsupervised fashion through automatic collection of instances from the web
   (c) deepBioWSD achieves the state-of-the-are 96.82% for macro accuracy when evaluated on MSH-WSD (supervised training)
   (d) deepBioWSD achieves the state-of-the-are 92.67% for macro accuracy when evaluated on MSH-WSD (unsupervised training)

2 Future Work

Similar to conclusion, future work can be separated based on what we discussed and presented in Part 2 and Part 3.

2 .1 Future Work for deepSimDEF

For the future work, Enzyme Commission (EC) similarity and orthologous protein distinguishing tasks present another opportunity for deepSimDEF performance evaluation. deepSimDEF networks as gene functional similarity measures need to be tested on the other species other than *Saccharomyces cerevisiae* as well. deepSimDEF models can also be utilized in the context of transfer learning for more improvement and faster training[145]. This transfer learning can occur by training on one species and then testing or fine-tuning on another species, or from one biological application to another application. Moreover, what deepSimDEF offers to the community is more than a functional similarity measure. deepSimDEF opens door to thinking that any biological entity can be presented in the form of embeddings. To give a direction,

we know that a protein domain is a conserved part of a given protein sequence that can evolve, function, and exist independently of the rest of the protein chain. Many proteins consist of several structural domains, and one domain may appear in a variety of different proteins. Molecular evolution also uses domains as building blocks as these may be recombined in different arrangements to create proteins with different functions. Having these kinds of knowledge in biology, and our understanding of deepSimDEF model, one will have a motivation to see if by the means of any relevant biological data it would be possible to represent these units in the form of embeddings, and then through statistical metrics or visualization tools infer some implicit knowledge hidden among the protein domains. deepSimDEF idea can also be applied to any (biomedical and biological) ontology of interest as well. For example, Online Mendelian Inheritance in Man[2] (OMIM) is a continuously updated ontology of human genes and genetic disorders and traits, with a particular focus on the gene-phenotype relationships. This ontology has proved its significant usefulness for discovery of diseases and drugs in many studies that take full advantage of the similarity between the biological entities covered in this ontology. Through a well-designed study, which follows the steps of deepSimDEF model in order to accurately quantify the relationships among genes and their phenotypic expressions, we might reach a breakthrough for a more advance approach of disease and drug discovery. And last but not least, a creative modification of deepSimDEF model from a functional similarity measure to a function assignment algorithm can be an interesting adventure to explore as well.

2 .2 Future Work for deepBioWSD

The outcome of deepBioWSD is directly applicable to a wide range of NLP applications. These applications run the gamut from machine translation and automatic text summarization to information extraction and query answering in any given language or domain; these applications can also cover specific tasks such as detection of adverse drug reactions from social media data and association discovery of diagnosis codes from electronic medical records (EMR). Apart from the intrinsic evaluation of deepSimDEF conducted in this study, any of these applications can provide a desirable testing field for the extrinsic evaluation of this WSD model. Moreover, by

[2] `https://www.omim.org/`

growing more interests in unsupervised pretraining of sense embeddings in the natural language domain, these new approaches can replace our sense embedding method in order to be evaluated and further improve the result of deepBioWSD. Additionally, better strategies for the collection of unsupervised training data from the web should be devised. Furthermore, consideration of an attention mechanism in the network design of deepBioWSD can improve its sense predictability.

Bibliography

[1] Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E. Mercer. A novel neural sequence model with multiple attentions for word sense disambiguation. In *17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, Orlando, FL, USA, December 17-20, 2018*, pages 687–694, 2018.

[2] Zeynep Akata, Scott E. Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 2927–2936, 2015.

[3] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[4] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

[5] Rui Antunes and Sérgio Matos. Supervised learning and knowledge-based approaches applied to biomedical word sense disambiguation. *J. Integrative Bioinformatics*, 14(4), 2017.

[6] Rolf Apweiler, Amos Bairoch, Cathy H Wu, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, et al. Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 32(suppl_1):D115–D119, 2004.

[7] Alan R Aronson and François-Michel Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.

[8] Ehsaneddin Asgari and Mohammad RK Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one*, 10(11):e0141287, 2015.

[9] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25, 2000.

[10] Francisco Azuaje, Haiying Wang, and Olivier Bodenreider. Ontology-driven similarity approaches to supporting gene functional assessment. In *Proceedings of the ISMB'2005 SIG meeting on Bio-ontologies*, pages 9–10, 2005.

[11] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247, 2014.

[12] Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. Breaking sticks and ambiguities with adaptive skip-gram. In *Artificial Intelligence and Statistics*, pages 130–138, 2016.

[13] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

[14] Bjoern-Toby Berster, J Caleb Goodwin, and Trevor Cohen. Hyperdimensional computing approach to word sense disambiguation. In *AMIA Annual Symposium Proceedings*, volume 2012, page 1129. American Medical Informatics Association, 2012.

[15] Paul W Bible, Hong-Wei Sun, Maria I Morasso, Rasiah Loganantharaj, and Lai Wei. The effects of shared information on semantic calculations in the gene ontology. *Computational and structural biotechnology journal*, 15:195–211, 2017.

[16] Chris Biemann, Simone Paolo Ponzetto, Stefano Faralli, Alexander Panchenko, and Eugen Ruppert. Unsupervised does not mean uninterpretable: The case for word sense induction and disambiguation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 86–98, 2017.

[17] Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J Martin, Karine Michoud, Claire O'donovan, Isabelle Phan, et al. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic acids research*, 31(1):365–370, 2003.

[18] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *Artificial Intelligence and Statistics*, pages 127–135, 2012.

[19] Delroy Cameron, Ramakanth Kavuluru, Thomas C. Rindflesch, Amit P. Sheth, Krishnaprasad Thirunarayan, and Olivier Bodenreider. Context-driven automatic subgraph creation for literature-based discovery. *Journal of Biomedical Informatics*, 54:141–157, 2015.

[20] Zhen Cao, Xiaoyong Pan, Yang Yang, Yan Huang, and Hong-Bin Shen. The lnclocator: a subcellular localization predictor for long non-coding rnas based on a stacked ensemble classifier. *Bioinformatics*, 1:10, 2018.

[21] Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[22] Charles E Chapple, Benoit Robisson, Lionel Spinelli, Céline Guien, Emmanuelle Becker, and Christine Brun. Extreme multifunctional proteins identified from a human protein interaction network. *Nature communications*, 6:7412, 2015.

[23] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750, 2014.

[24] Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. A unified model for word sense representation and disambiguation. In *EMNLP*, pages 1025–1035, 2014.

[25] Zhiyuan Chen and Bing Liu. *Lifelong Machine Learning, Second Edition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2018.

[26] J Michael Cherry, Caroline Adler, Catherine Ball, Stephen A Chervitz, Selina S Dwight, Erich T Hester, Yankai Jia, Gail Juvik, TaiYun Roe, Mark Schroeder, et al. Sgd: Saccharomyces genome database. *Nucleic acids research*, 26(1):73–79, 1998.

[27] Davide Chicco and Marco Masseroli. Ontology-based prediction and prioritization of gene functional annotations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(2):248–260, 2016.

[28] Davide Chicco, Peter Sadowski, and Pierre Baldi. Deep autoencoder neural networks for gene ontology annotation predictions. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 533–540. ACM, 2014.

[29] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.

[30] Sung-Pil Choi. Extraction of protein–protein interactions (ppis) from the literature by deep convolutional neural networks with various feature embeddings. *Journal of Information Science*, page 0165551516673485, 2016.

[31] Wyatt T Clark and Predrag Radivojac. Analysis of protein function and its prediction from amino acid sequence. *Proteins: Structure, Function, and Bioinformatics*, 79(7):2086–2096, 2011.

[32] Kevin Bretonnel Cohen, Dina Demner-Fushman, Sophia Ananiadou, and Jun'ichi Tsujii, editors. *Proceedings of the Workshop on Biomedical Natural Language Processing, BioNLP@IJCNLP 2015, Beijing, China, July 30, 2015*. Association for Computational Linguistics, 2015.

[33] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[34] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[35] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[36] UniProt Consortium. The universal protein resource (uniprot) in 2010. *Nucleic acids research*, 38(suppl_1):D142–D148, 2009.

[37] Francisco M Couto and Mário J Silva. Disjunctive shared information between ontology concepts: application to gene ontology. *Journal of biomedical semantics*, 2(1):5, 2011.

[38] D Cozzetto and DT Jones. Computational methods for annotation transfers from sequence. *Methods in molecular biology (Clifton, NJ)*, 1446:55, 2017.

[39] Megan Crow, Anirban Paul, Sara Ballouz, Z Josh Huang, and Jesse Gillis. Exploiting single-cell expression to characterize co-expression replicability. *Genome biology*, 17(1):101, 2016.

[40] Kushal Dave, Steve Lawrence, and David M Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

[41] Christophe Dessimoz and Nives Škunca. The gene ontology handbook. *Methods in molecular biology*, 2016.

[42] Christophe Dessimoz and Nives Škunca. *The Gene Ontology Handbook*. Springer, 2017.

[43] Juan J Díaz-Montaña, Norberto Díaz-Díaz, and Francisco Gómez-Vela. Gfd-net: A novel semantic similarity methodology for the analysis of gene networks. *Journal of biomedical informatics*, 68:71–82, 2017.

[44] Dat Duong, Wasi Uddin Ahmad, Eleazar Eskin, Kai-Wei Chang, and Jingyi Jessica Li. Word and sentence embedding tools to measure semantic similarity of gene ontology terms by their definitions. *Journal of Computational Biology*, 2018.

[45] Pritha Dutta, Subhadip Basu, and Mahantapas Kundu. Assessment of semantic similarity between proteins using information content and topological properties of the gene ontology graph. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(3):839–849, 2018.

[46] Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

[47] Bilel Elayeb, Wiem Ben Romdhane, and Narjès Bellamine Ben Saoud. Towards a new possibilistic query translation tool for cross-language information retrieval. *Multimedia Tools and Applications*, 77(2):2423–2465, 2018.

[48] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

[49] Seth Falcon and Robert Gentleman. Using gostats to test gene lists for go term association. *Bioinformatics*, 23(2):257–258, 2006.

[50] Andres Duque Fernandez, Mark Stevenson, Juan Martinez-Romo, and Lourdes Araujo. Co-occurrence graphs for word sense disambiguation in the biomedical domain. *Artificial Intelligence in Medicine*, 87:9–19, 2018.

[51] John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

[52] Lucie Flekova and Iryna Gurevych. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[53] Iddo Friedberg and Predrag Radivojac. Community-wide evaluation of computational function prediction. *The Gene Ontology Handbook*, pages 133–146, 2017.

[54] Yuanyuan Fu, Yanzhi Guo, Yuelong Wang, Jiesi Luo, Xuemei Pu, Menglong Li, and Zhihang Zhang. Exploring the relationship between hub proteins and drug targets based on go and intrinsic disorder. *Computational biology and chemistry*, 56:41–48, 2015.

[55] Shang Gao, Michael T Young, John X Qiu, Hong-Jun Yoon, James B Christian, Paul A Fearn, Georgia D Tourassi, and Arvind Ramanthan. Hierarchical attention networks for information extraction from cancer pathology reports. *Journal of the American Medical Informatics Association*, 25(3):321–330, 2017.

[56] Sahil Garg, Aram Galstyan, Ulf Hermjakob, and Daniel Marcu. Extracting biomolecular interactions using semantic parsing of biomedical text. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2718–2726, 2016.

[57] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.

[58] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[59] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[60] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[61] Cristian Grozea. Finding optimal parameter settings for high performance word sense disambiguation. In *Proceedings of Senseval-3 Workshop*, 2004.

[62] Pietro H Guzzi, Marco Mina, Concettina Guerra, and Mario Cannataro. Semantic similarity analysis of protein data: assessment with biological features and issues. *Briefings in bioinformatics*, 13(5):569–585, 2012.

[63] Maryam Habibi, Leon Weber, Mariana L. Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 2017.

[64] G Hinton, N Srivastava, and K Swersky. Rmsprop: Divide the gradient by a running average of its recent magnitude. *Neural networks for machine learning, Coursera lecture 6e*, 2012.

[65] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[66] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[67] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[68] Gemma L Holliday, Rebecca Davidson, Eyal Akiva, and Patricia C Babbitt. Evaluating functional annotations of enzymes using the gene ontology. *The Gene Ontology Handbook*, pages 111–132, 2017.

[69] Frantisek Honti, Stephen Meader, and Caleb Webber. Unbiased functional clustering of gene variants with a phenotypic-linkage network. *PLoS computational biology*, 10(8):e1003815, 2014.

[70] Chia-Lang Hsu, Hsueh-Fen Juan, and Hsuan-Cheng Huang. Functional analysis and characterization of differential coexpression networks. *Scientific reports*, 5:13295, 2015.

[71] Da Wei Huang, Brad T Sherman, Qina Tan, Jack R Collins, W Gregory Alvord, Jean Roayaei, Robert Stephens, Michael W Baseler, H Clifford Lane, and Richard A Lempicki. The david gene functional classification tool: a novel biological module-centric algorithm to functionally analyze large gene lists. *Genome biology*, 8(9):R183, 2007.

[72] Jason C Hung, Ching-Sheng Wang, Che-Yu Yang, Mao-Shuen Chiu, and George Yee. Applying word sense disambiguation to question answering system for e-learning. In *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*, volume 1, pages 157–162. IEEE, 2005.

[73] Curtis Huttenhower, Matt Hibbs, Chad Myers, and Olga G Troyanskaya. A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics*, 22(23):2890–2897, 2006.

[74] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 897–907, 2016.

[75] Nancy Ide and Jean Véronis. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics*, 24(1):2–40, 1998.

[76] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691, 2015.

[77] Shobhit Jain and Gary D Bader. An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology. *BMC bioinformatics*, 11(1):562, 2010.

[78] Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

[79] Xiang Jiang, Erico N de Souza, Ahmad Pesaranghader, Baifan Hu, Daniel L Silver, and Stan Matwin. Trajectorynet: An embedded gps trajectory representation for point-based classification using recurrent neural networks. In *Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering*, pages 192–200. IBM Corp., 2017.

[80] Yuxiang Jiang, Tal Ronnen Oron, Wyatt T Clark, Asma R Bankapur, Daniel DAndrea, Rosalba Lepore, Christopher S Funk, Indika Kahanda, Karin M Verspoor, Asa Ben-Hur, et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome biology*, 17(1):184, 2016.

[81] Antonio J Jimeno-Yepes, Bridget T McInnes, and Alan R Aronson. Exploiting mesh indexing in medline to generate a data set for word sense disambiguation. *BMC bioinformatics*, 12(1):223, 2011.

[82] Bo Jin, Vicky Chen, Lujia Chen, and Xinghua Lu. Mapping annotations with textual evidence using an sclda model. In *AMIA Annual Symposium Proceedings*, volume 2011, page 834. American Medical Informatics Association, 2011.

[83] Bo Jin and Xinghua Lu. Identifying informative subsets of the gene ontology with information bottleneck methods. *Bioinformatics*, 26(19):2445–2451, 2010.

[84] Dakai Jin, Ziyue Xu, Youbao Tang, Adam P Harrison, and Daniel J Mollura. Ct-realistic lung nodule simulation from 3d conditional generative adversarial networks for robust lung segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 732–740. Springer, 2018.

[85] Trupti Joshi and Dong Xu. Quantitative assessment of relationship between sequence similarity and function similarity. *BMC genomics*, 8(1):222, 2007.

[86] Mikael Kågebäck and Hans Salomonsson. Word sense disambiguation using a bidirectional lstm. *arXiv preprint arXiv:1606.03568*, 2016.

[87] Ramakanth Kavuluru, Anthony Rios, and Yuan Lu. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial Intelligence in Medicine*, 65(2):155–166, 2015.

[88] Eiru Kim, Sohyun Hwang, and Insuk Lee. Soynet: a database of co-functional networks for soybean glycine max. *Nucleic Acids Research*, page gkw704, 2016.

[89] Seonho Kim and Juntae Yoon. Link-topic model for biomedical abbreviation disambiguation. *Journal of biomedical informatics*, 53:367–380, 2015.

[90] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[91] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4447–4456, 2017.

[92] Patrik Koskinen, Petri Törönen, Jussi Nokso-Koivisto, and Liisa Holm. Pannzer: high-throughput functional annotation of uncharacterized proteins in an error-prone environment. *Bioinformatics*, 31(10):1544–1552, 2015.

[93] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[94] Maxat Kulmanov and Robert Hoehndorf. Evaluating the effect of annotation size on measures of semantic similarity. *Journal of biomedical semantics*, 8(1):7, 2017.

[95] Chaowang Lan, Qingfeng Chen, and Jinyan Li. Grouping mirnas of similar functions via weighted information content of gene ontology. *BMC bioinformatics*, 17(19):507, 2016.

[96] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[97] Liliana Laranjo, Adam G. Dunn, Huong Ly Tong, Ahmet Baki Kocaballi, Jessica A. Chen, Rabia Bashir, Didi Surian, Blanca Gallego, Farah Magrabi, Annie Y. S. Lau, and Enrico W. Coiera. Conversational agents in healthcare: a systematic review. *JAMIA*, 25(9):1248–1258, 2018.

[98] Juan J Lastra-Díaz, Ana García-Serrano, Montserrat Batet, Miriam Fernández, and Fernando Chirigati. Hesml: A scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. *Information Systems*, 66:97–118, 2017.

[99] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[100] Kyubum Lee, Sunwon Lee, Sungjoon Park, Sunkyu Kim, Suhkyung Kim, Kwanghun Choi, Aik Choon Tan, and Jaewoo Kang. BRONCO: biomedical entity relation oncology corpus for extracting gene-variant-disease-drug relations. *Database*, 2016, 2016.

[101] Kyubum Lee, Won-Ho Shin, Byounggun Kim, Sunwon Lee, Yonghwa Choi, Sunkyu Kim, Minji Jeon, Aik Choon Tan, and Jaewoo Kang. Hipub: translating pubmed and PMC texts to networks for knowledge discovery. *Bioinformatics*, 32(18):2886–2888, 2016.

[102] Yoong Keok Lee, Hwee Tou Ng, and Tee Kiah Chia. Supervised word sense disambiguation with support vector machines and multiple knowledge sources. In *Senseval-3: third international workshop on the evaluation of systems for the semantic analysis of text*, pages 137–140, 2004.

[103] Michael Levandowsky and David Winter. Distance between sets. *Nature*, 234(5323):34, 1971.

[104] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

[105] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2177–2185, 2014.

[106] Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1):198, 2017.

[107] Peng Li, Chao Huang, Yingxue Fu, Jinan Wang, Ziyin Wu, Jinlong Ru, Chunli Zheng, Zihu Guo, Xuetong Chen, Wei Zhou, et al. Large-scale exploration and analysis of drug combinations. *Bioinformatics*, 31(12):2007–2016, 2015.

[108] Xia Li, Qianghu Wang, Yan Zheng, Sali Lv, Shangwei Ning, Jie Sun, Teng Huang, Qifan Zheng, Huan Ren, Jin Xu, et al. Prioritizing human cancer microrna based on genes functional consistency between microrna and cancer. *Nucleic acids research*, 39(22):e153–e153, 2011.

[109] Ying Hong Li, Jing Yu Xu, Lin Tao, Xiao Feng Li, Shuang Li, Xian Zeng, Shang Ying Chen, Peng Zhang, Chu Qin, Cheng Zhang, et al. Svm-prot 2016: A web-server for machine learning prediction of protein functional families from sequence irrespective of similarity. *PloS one*, 11(8):e0155290, 2016.

[110] Dekang Lin et al. An information-theoretic definition of similarity. In *Icml*, volume 98, pages 296–304. Citeseer, 1998.

[111] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[112] Chenglin Liu, Jing Su, Fei Yang, Kun Wei, Jinwen Ma, and Xiaobo Zhou. Compound signature detection on lincs l1000 big data. *Molecular BioSystems*, 11(3):714–722, 2015.

[113] Min Liu, Weiming Shen, Qi Hao, and Junwei Yan. An weighted ontology-based semantic similarity algorithm for web service. *Expert Systems with Applications*, 36(10):12480–12490, 2009.

[114] Xiaoxia Liu, Zhihao Yang, Hongfei Lin, Michael Simmons, and Zhiyong Lu. Dignifi: Discovering causative genes for orphan diseases using protein-protein interaction networks. *BMC Systems Biology*, 11(3):23, 2017.

[115] Ying Liu, Bridget T McInnes, Ted Pedersen, Genevieve Melton-Meaux, and Serguei Pakhomov. Semantic relatedness study using second order co-occurrence vectors computed from biomedical corpora, umls and wordnet. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 363–372. ACM, 2012.

[116] Ying Liu, Bridget T. McInnes, Ted Pedersen, Genevieve Melton-Meaux, and Serguei V. S. Pakhomov. Semantic relatedness study using second order co-occurrence vectors computed from biomedical corpora, UMLS and wordnet. In *ACM International Health Informatics Symposium, IHI '12, Miami, FL, USA, January 28-30, 2012*, pages 363–372, 2012.

[117] Phillip W. Lord, Robert D. Stevens, Andy Brass, and Carole A. Goble. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283, 2003.

[118] Yuan Luo, Özlem Uzuner, and Peter Szolovits. Bridging semantics and syntax with graph algorithms - state-of-the-art of extracting biomedical relations. *Briefings in Bioinformatics*, 18(1):160–178, 2017.

[119] Pierre Marchal and Thierry Poibeau. Lexical knowledge acquisition: Towards a continuous and flexible representation of the lexicon. In *Workshop on Cognitive Knowledge Acquisition and Applications*. IJCAI, 2016.

[120] Sachin Mathur and Deendayal Dinakarpandian. Finding disease similarity based on implicit semantic similarity. *Journal of biomedical informatics*, 45(2):363–371, 2012.

[121] Gaston K Mazandu, Emile R Chimusa, and Nicola J Mulder. Gene ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery. *Briefings in bioinformatics*, 18(5):886–901, 2016.

[122] Gaston K Mazandu and Nicola J Mulder. Dago-fun: tool for gene ontology-based functional analysis using term information content measures. *BMC bioinformatics*, 14(1):284, 2013.

[123] Bridget T McInnes and Ted Pedersen. Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text. *Journal of biomedical informatics*, 46(6):1116–1124, 2013.

[124] Bridget T McInnes and Mark Stevenson. Determining the difficulty of word sense disambiguation. *Journal of biomedical informatics*, 47:83–90, 2014.

[125] X Meng, J Wang, C Yuan, X Li, Y Zhou, Ralf Hofestädt, and Ming Chen. Cancernet: a database for decoding multilevel molecular interactions across diverse cancer types. *Oncogenesis*, 4(12):e177, 2015.

[126] Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. The senseval-3 english lexical sample task. In *Proceedings of SENSEVAL-3, the third international workshop on the evaluation of systems for the semantic analysis of text*, 2004.

[127] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[128] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[129] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[130] Rashmi Mishra, Jiantao Bian, Marcelo Fiszman, Charlene R. Weir, Siddhartha Jonnalagadda, Javed Mostafa, and Guilherme Del Fiol. Text summarization in the biomedical domain: A systematic review of recent research. *Journal of Biomedical Informatics*, 52:457–467, 2014.

[131] George Montanez and Young-Rae Cho. Predicting false positives of protein-protein interaction data by semantic similarity measures. *Current Bioinformatics*, 8(3):339–346, 2013.

[132] Yoichi Murakami, Lokesh P Tripathi, Philip Prathipati, and Kenji Mizuguchi. Network analysis and in silico prediction of protein–protein interactions with applications in drug discovery. *Current opinion in structural biology*, 44:134–142, 2017.

[133] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[134] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10, 2009.

[135] Roberto Navigli. A quick tour of word sense disambiguation, induction and related approaches. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 115–129. Springer, 2012.

[136] Roberto Navigli, Kenneth C Litkowski, and Orin Hargraves. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 30–35. Association for Computational Linguistics, 2007.

[137] Steven Neale, Luís Gomes, Eneko Agirre, Oier Lopez de Lacalle, and António Branco. Word sense-aware machine translation: Including senses as contextual features for improved translation models. In *LREC*, 2016.

[138] Aurélie Névéol, Aude Robert, Francesco Grippo, Claire Morgand, Chiara Orsi, Laszlo Pelikan, Lionel Ramadier, Grégoire Rey, and Pierre Zweigenbaum. CLEF ehealth 2018 multilingual information extraction task overview: ICD10 coding of death certificates in french, hungarian and italian. In *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018.*, 2018.

[139] Minh Nam Nguyen, Tae Gyu Choi, Dinh Truong Nguyen, Jin-Hwan Kim, Yong Hwa Jo, Muhammad Shahid, Salima Akter, Saurav Nath Aryal, Ji Youn Yoo, Yong-Joo Ahn, et al. Crc-113 gene expression signature for predicting prognosis in patients with colorectal cancer. *Oncotarget*, 6(31):31674, 2015.

[140] Kristin K Nicodemus, Brita Elvevåg, Peter W Foltz, Mark Rosenstein, Catherine Diaz-Asper, and Daniel R Weinberger. Category fluency, latent semantic analysis and schizophrenia: a candidate gene approach. *Cortex*, 55:182–191, 2014.

[141] Kristian Ovaska. Using semantic similarities and csbl. go for analyzing microarray data. In *Microarray Data Analysis*, pages 105–116. Springer, 2015.

[142] Ahmad P Tafti, Jonathan Badger, Eric LaRose, Ehsan Shirzadi, Andrea Mahnke, John Mayer, Zhan Ye, David Page, and Peggy Peissig. Adverse drug event discovery using biomedical literature: A big data neural network adventure. *JMIR Med Inform*, 5(4):e51, Dec 2017.

[143] Serguei Pakhomov, Bridget McInnes, Terrence Adam, Ying Liu, Ted Pedersen, and Genevieve B Melton. Semantic similarity and relatedness between clinical terms: an experimental study. In *AMIA annual symposium proceedings*, volume 2010, page 572. American Medical Informatics Association, 2010.

[144] Serguei VS Pakhomov, Greg Finley, Reed McEwan, Yan Wang, and Genevieve B Melton. Corpus domain effects on distributional semantic modeling of medical terms. *Bioinformatics*, 32(23):3635–3644, 2016.

[145] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[146] Christopher Parkinson and Shai Leib. Text editing with gesture control and natural speech, May 2 2017. US Patent 9,640,181.

[147] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.

[148] Maria Pelevina, Nikolay Arefyev, Chris Biemann, and Alexander Panchenko. Making sense of word embeddings. *arXiv preprint arXiv:1708.03390*, 2017.

[149] Hui Peng, Chaowang Lan, Yi Zheng, Gyorgy Hutvagner, Dacheng Tao, and Jinyan Li. Cross disease analysis of co-functional microrna pairs on a reconstructed network of disease-gene-microrna tripartite. *BMC bioinformatics*, 18(1):193, 2017.

[150] Jiajie Peng, Xuanshuo Zhang, Weiwei Hui, Junya Lu, Qianqian Li, Shuhui Liu, and Xuequn Shang. Improving the measurement of semantic similarity by combining gene ontology and co-functional network: a random walk based approach. *BMC systems biology*, 12(2):18, 2018.

[151] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[152] A. Pesaranghader, S. Muthaiyah, and A. Pesaranghader. Improving gloss vector semantic relatedness measure by integrating pointwise mutual information: Optimizing second-order co-occurrence vectors computed from biomedical corpus and umls. In *2013 International Conference on Informatics and Creative Multimedia*, pages 196–201, Sep. 2013.

[153] Ahmad Pesaranghader, Stan Matwin, Marina Sokolova, and Robert G Beiko. simdef: definition-based semantic similarity measure of gene ontology terms for functional similarity analysis of genes. *Bioinformatics*, 32(9):1380–1387, 2015.

[154] Ahmad Pesaranghader, Stan Matwin, Marina Sokolova, and Ali Pesaranghader. deepbiowsd: effective deep neural word sense disambiguation of biomedical text data. *Journal of the American Medical Informatics Association*, 26(5):438–446, 2019.

[155] Ahmad Pesaranghader and Saravanan Muthaiyah. Definition-based information content vectors for semantic similarity measurement. In *International Multi-Conference on Artificial Intelligence Technology*, pages 268–282. Springer, 2013.

[156] Ahmad Pesaranghader, Saravanan Muthaiyah, and Ali Pesaranghader. Improving gloss vector semantic relatedness measure by integrating pointwise mutual information: Optimizing second-order co-occurrence vectors computed from biomedical corpus and umls. In *2013 International Conference on Informatics and Creative Multimedia*, pages 196–201. IEEE, 2013.

[157] Ahmad Pesaranghader, Ali Pesaranghader, Stan Matwin, and Marina Sokolova. One single deep bidirectional lstm network for word sense disambiguation of text data. In *Advances in Artificial Intelligence: 31st Canadian Conference on Artificial Intelligence, Canadian AI 2018, Toronto, ON, Canada, May 8–11, 2018, Proceedings 31*, pages 96–107. Springer, 2018.

[158] Ahmad Pesaranghader, Ali Pesaranghader, and Norwati Mustapha. Word sense disambiguation for biomedical text mining using definition-based semantic relatedness and similarity measures. *International Journal of Bioscience, Biochemistry and Bioinformatics*, 4(4):280, 2014.

[159] Ahmad Pesaranghader, Ali Pesaranghader, and Azadeh Rezaei. Applying latent semantic analysis to optimize second-order co-occurrence vectors for semantic relatedness measurement. In *Mining Intelligence and Knowledge Exploration*, pages 588–599. Springer, 2013.

[160] Ahmad Pesaranghader, Ali Pesaranghader, and Azadeh Rezaei. Augmenting concept definition in gloss vector semantic relatedness measure using wikipedia articles. In *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, pages 623–630. Springer, 2014.

[161] Ahmad Pesaranghader, Ali Pesaranghader, Azadeh Rezaei, and Danoosh Davoodi. Gene functional similarity analysis by definition-based semantic similarity measurement of go terms. In *Canadian Conference on Artificial Intelligence*, pages 203–214. Springer, 2014.

[162] Ahmad Pesaranghader, Ali Pesaranghader, Azadeh Rezaei, and Danoosh Davoodi. Gene functional similarity analysis by definition-based semantic similarity measurement of GO terms. In *Advances in Artificial Intelligence - 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014, Montréal, QC, Canada, May 6-9, 2014. Proceedings*, pages 203–214, 2014.

[163] Ahmad Pesaranghader, Azadeh Rezaei, and Ali Pesaranghader. Adapting gloss vector semantic relatedness measure for semantic similarity estimation: An evaluation in the biomedical domain. In *Joint International Semantic Technology Conference*, pages 129–145. Springer, 2013.

[164] Ahmad Pesaranghader, Azadeh Rezaei, and Ali Pesaranghader. Adapting gloss vector semantic relatedness measure for semantic similarity estimation: An evaluation in the biomedical domain. In *Semantic Technology - Third Joint International Conference, JIST 2013, Seoul, South Korea, November 28-30, 2013, Revised Selected Papers*, pages 129–145, 2013.

[165] Ali Pesaranghader, Norwati Mustapha, and Ahmad Pesaranghader. Applying semantic similarity measures to enhance topic-specific web crawling. In *2013 13th International Conference on Intellient Systems Design and Applications*, pages 205–212. IEEE, 2013.

[166] Ali Pesaranghader, Ahmad Pesaranghader, Norwati Mustapha, and Nurfadhlina Mohd Sharef. Improving multi-term topics focused crawling by introducing term frequency-information content (tf-ic) measure. In *2013 International Conference on Research and Innovation in Information Systems (ICRIIS)*, pages 102–106. IEEE, 2013.

[167] Ali Pesaranghader, Herna Viktor, and Eric Paquet. Reservoir of diverse adaptive learners and stacking fast hoeffding drift detection methods for evolving data streams. *Machine Learning*, 107(11):1711–1743, 2018.

[168] Ali Pesaranghader and Herna L Viktor. Fast hoeffding drift detection method for evolving data streams. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 96–111. Springer, 2016.

[169] Ali Pesaranghader, Herna L Viktor, and Eric Paquet. A framework for classification in data streams using multi-strategy learning. In *International conference on discovery science*, pages 341–355. Springer, 2016.

[170] Catia Pesquita. Semantic similarity in the gene ontology. In *The Gene Ontology Handbook*, pages 161–173. Springer, 2017.

[171] Catia Pesquita, Daniel Faria, Hugo Bastos, André Falcao, and Francisco Couto. Evaluating go-based semantic similarity measures. In *Proc. 10th Annual Bio-Ontologies Meeting*, volume 37, page 38, 2007.

[172] Catia Pesquita, Daniel Faria, Hugo Bastos, António EN Ferreira, André O Falcão, and Francisco M Couto. Metrics for go based protein semantic similarity: a systematic evaluation. In *BMC bioinformatics*, volume 9, page S4. BioMed Central, 2008.

[173] Catia Pesquita, Daniel Faria, Andre O Falcao, Phillip Lord, and Francisco M Couto. Semantic similarity in biomedical ontologies. *PLoS computational biology*, 5(7):e1000443, 2009.

[174] Catia Pesquita, Delphine Pessoa, Daniel Faria, and Francisco Couto. Cessm: Collaborative evaluation of semantic similarity measures. *JB2009: Challenges in Bioinformatics*, 157:190, 2009.

[175] Mohammad Taher Pilehvar and Roberto Navigli. A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation. *Computational Linguistics*, 40(4):837–881, 2014.

[176] Damiano Piovesan, Manuel Giollo, Emanuela Leonardi, Carlo Ferrari, and Silvio CE Tosatto. Inga: protein function prediction combining interaction networks, domain assignments and sequence similarity. *Nucleic acids research*, 43(W1):W134–W140, 2015.

[177] Lutz Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761–767, 1998.

[178] Judita Preiss and Mark Stevenson. The effect of word sense disambiguation accuracy on literature based discovery. *BMC Med. Inf. & Decision Making*, 16(S-1):57, 2016.

[179] Shuye Pu, James Vlasblom, Andrei Turinsky, Edyta Marcon, Sadhna Phanse, Sandra Smiley Trimble, Jonathan Olsen, Jack Greenblatt, Andrew Emili, and Shoshana J Wodak. Extracting high confidence protein interactions from affinity purification data: At the crossroads. *Journal of proteomics*, 118:63–80, 2015.

[180] Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Alexandra M Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, et al. A large-scale evaluation of computational protein function prediction. *Nature methods*, 10(3):221, 2013.

[181] Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1156–1167, 2017.

[182] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[183] Hafeez Ur Rehman, Nouman Azam, JingTao Yao, and Alfredo Benso. A three-way approach for protein function classification. *PloS one*, 12(2):e0171702, 2017.

[184] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[185] Kirk Roberts, Halil Kilicoglu, Marcelo Fiszman, and Dina Demner-Fushman. Automatically classifying question types for consumer health questions. In *AMIA 2014, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 15-19, 2014*, 2014.

[186] Bernardino Romera-Paredes and Philip H. S. Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2152–2161, 2015.

[187] AKM Sabbir, Antonio Jimeno-Yepes, and Ramakanth Kavuluru. Knowledge-based biomedical word sense disambiguation with neural concept embeddings. In *Bioinformatics and Bioengineering (BIBE), 2017 IEEE 17th International Conference on*, pages 163–170. IEEE, 2017.

[188] Shouq A Sadah, Moloud Shahbazi, Matthew T Wiley, and Vagelis Hristidis. Demographic-based content analysis of web-based health-related social media. *J Med Internet Res*, 18(6):e148, Jun 2016.

[189] Lukasz Salwinski, Christopher S Miller, Adam J Smith, Frank K Pettit, James U Bowie, and David Eisenberg. The database of interacting proteins: 2004 update. *Nucleic acids research*, 32(suppl_1):D449–D451, 2004.

[190] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

[191] Martin H Schaefer and Luis Serrano. Cell type-specific properties and environment shape tissue specificity of cancer genes. *Scientific reports*, 6:20707, 2016.

[192] Andreas Schlicker, Francisco S Domingues, Jörg Rahnenführer, and Thomas Lengauer. A new measure for functional similarity of gene products based on gene ontology. *BMC bioinformatics*, 7(1):302, 2006.

[193] Andreas Schlicker, Jörg Rahnenführer, Mario Albrecht, Thomas Lengauer, and Francisco S Domingues. Gotax: investigating biological processes and biochemical activities along the taxonomic tree. *Genome Biology*, 8(3):R33, 2007.

[194] Didier Schwab, Jérôme Goulian, Andon Tchechmedjiev, and Hervé Blanchon. Ant colony algorithm for the unsupervised word sense disambiguation of texts: Comparison and evaluation. *Proceedings of COLING 2012*, pages 2389–2404, 2012.

[195] Beatriz Serrano-Solano, Antonio Díaz Ramos, Jean-Karim Hériché, and Juan AG Ranea. How can functional annotations be derived from profiles of phenotypic annotations? *BMC bioinformatics*, 18(1):96, 2017.

[196] Jose L Sevilla, Victor Segura, Adam Podhorski, Elizabeth Guruceaga, Jose M Mato, Luis A Martinez-Cruz, Fernando J Corrales, and Angel Rubio. Correlation between gene expression and go semantic similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 2(4):330–338, 2005.

[197] Zhiao Shi, Catherine K Derow, and Bing Zhang. Co-expression module analysis reveals biological processes, genomic gain, and regulatory mechanisms associated with breast cancer progression. *BMC systems biology*, 4(1):74, 2010.

[198] Jung Eun Shim, Tak Lee, and Insuk Lee. From sequencing data to gene functions: co-functional network approaches. *Animal Cells and Systems*, pages 1–7, 2017.

[199] Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David RJ Snead, Ian A Cree, and Nasir M Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE transactions on medical imaging*, 35(5):1196–1206, 2016.

[200] Xuebo Song, Lin Li, Pradip K Srimani, S Yu Philip, and James Z Wang. Measure the semantic similarity of go terms using aggregate information content. *IEEE/ACM transactions on computational biology and bioinformatics*, 11(3):468–476, 2014.

[201] Dhanya Sridhar, Shobeir Fakhraei, and Lise Getoor. A probabilistic approach for collective similarity-based drug–drug interaction prediction. *Bioinformatics*, 32(20):3175–3182, 2016.

[202] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[203] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385, 2015.

[204] Christopher Stokoe, Michael P Oakes, and John Tait. Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 159–166. ACM, 2003.

[205] Carlo Strapparava, Alfio Gliozzo, and Claudio Giuliano. Pattern abstraction and term similarity for word sense disambiguation: Irst at senseval-3. In *Proc. of SENSEVAL-3 Third International Workshop on Evaluation of Systems for the Semantic Analysis of Text*, pages 229–234, 2004.

[206] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[207] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.

[208] Kaveh Taghipour and Hwee Tou Ng. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *HLT-NAACL*, pages 314–323, 2015.

[209] Nasrin Taghizadeh and Hesham Faili. Automatic wordnet development for low-resource languages using cross-lingual wsd. *Journal of Artificial Intelligence Research*, 56:61–87, 2016.

[210] Jie Tan, Matthew Ung, Chao Cheng, and Casey S Greene. Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, volume 20, page 132. NIH Public Access, 2015.

[211] Zhixia Teng, Maozu Guo, Xiaoyan Liu, Qiguo Dai, Chunyu Wang, and Ping Xuan. Measuring gene functional similarity based on group-wise comparison of go terms. *Bioinformatics*, 29(11):1424–1432, 2013.

[212] Zhen Tian, Chunyu Wang, Maozu Guo, Xiaoyan Liu, and Zhixia Teng. Sgfsc: speeding the gene functional similarity calculation based on hash tables. *BMC bioinformatics*, 17(1):445, 2016.

[213] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[214] Fatemeh Vafaee, Daniela Rosu, Fiona Broackes-Carter, and Igor Jurisica. Novel semantic similarity measure improves an integrative approach to predicting gene functional associations. *BMC systems biology*, 7(1):22, 2013.

[215] David Vickrey, Luke Biewald, Marc Teyssier, and Daphne Koller. Word-sense disambiguation for machine translation. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, 2005.

[216] Vedrana Vidulin, Tomislav Šmuc, and Fran Supek. Extensive complementarity between gene function prediction methods. *Bioinformatics*, page btw532, 2016.

[217] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[218] Shibiao Wan, Man-Wai Mak, and Sun-Yuan Kung. Hybridgo-loc: mining hybrid features on gene ontology for predicting subcellular localization of multi-location proteins. *PLoS One*, 9(3):e89545, 2014.

[219] Guan Wang, Kenneth Jung, Rainer Winnenburg, and Nigam H. Shah. A method for systematic discovery of adverse drug events from clinical notes. *JAMIA*, 22(6):1196–1204, 2015.

[220] Haiying Wang, Francisco Azuaje, Olivier Bodenreider, and Joaquín Dopazo. Gene expression correlation and gene ontology-based similarity: an assessment of quantitative relationships. In *2004 Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 25–31. IEEE, 2004.

[221] James Z Wang, Zhidian Du, Rapeeporn Payattakool, Philip S Yu, and Chin-Fu Chen. A new method to measure the semantic similarity of go terms. *Bioinformatics*, 23(10):1274–1281, 2007.

[222] Jing Wang, Zihao Ma, Steven A Carr, Philipp Mertins, Hui Zhang, Zhen Zhang, Daniel W Chan, Matthew JC Ellis, R Reid Townsend, Richard D Smith, et al. Proteome profiling outperforms transcriptome profiling for coexpression based gene function prediction. *Molecular & Cellular Proteomics*, 16(1):121–134, 2017.

[223] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, and Ronald M. Summers. Tienet: Text-image embedding network for common thorax disease classification and reporting in chest x-rays. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9049–9058, 2018.

[224] Yue Wang, Kai Zheng, Hua Xu, and Qiaozhu Mei. Clinical word sense disambiguation with interactive search and classification. In *AMIA 2016, American Medical Informatics Association Annual Symposium, Chicago, IL, USA, November 12-16, 2016*, 2016.

[225] Yue Wang, Kai Zheng, Hua Xu, and Qiaozhu Mei. Interactive medical word sense disambiguation through informed learning. *JAMIA*, 25(7):800–808, 2018.

[226] David Warde-Farley, Sylva L Donaldson, Ovi Comes, Khalid Zuberi, Rashad Badrawi, Pauline Chao, Max Franz, Chris Grouios, Farzana Kazi, Christian Tannus Lopes, et al. The genemania prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic acids research*, 38(suppl_2):W214–W220, 2010.

[227] Christian X Weichenberger, Antonia Palermo, Peter P Pramstaller, and Francisco S Domingues. Exploring approaches for detecting protein functional similarity within an orthology-based framework. *Scientific reports*, 7(1):381, 2017.

[228] Yorick Wilks and Mark Stevenson. The grammar of sense: Is word-sense tagging much more than part-of-speech tagging? *arXiv preprint cmp-lg/9607028*, 1996.

[229] Xiaomei Wu, Erli Pang, Kui Lin, and Zhen-Ming Pei. Improving the measurement of semantic similarity between gene ontology terms and gene products: insights from an edge-and ic-based hybrid method. *PloS one*, 8(5):e66745, 2013.

[230] Xiaomei Wu, Lei Zhu, Jie Guo, Da-Yong Zhang, and Kui Lin. Prediction of yeast protein–protein interaction network: insights from the gene ontology and annotations. *Nucleic acids research*, 34(7):2137–2150, 2006.

[231] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[232] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[233] Tao Xu, LinFang Du, and Yan Zhou. Evaluation of go-based functional similarity measures using s. cerevisiae protein interaction and expression profile data. *BMC bioinformatics*, 9(1):472, 2008.

[234] Haixuan Yang, Tamás Nepusz, and Alberto Paccanaro. Improving go semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty. *Bioinformatics*, 28(10):1383–1389, 2012.

[235] Yang Yang and Bao-Liang Lu. Protein subcellular multi-localization prediction using a min-max modular support vector machine. *International Journal of Neural Systems*, 20(01):13–28, 2010.

[236] Yang Yang, Zhuangdi Xu, and Dandan Song. Missing value imputation for microrna expression data by using a go-based similarity measure. In *BMC bioinformatics*, volume 17, page 10. BioMed Central Ltd, 2016.

[237] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[238] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1480–1489, 2016.

[239] Antonio Jimeno Yepes. Word embeddings and recurrent neural networks based on long-short term memory nodes in supervised biomedical word sense disambiguation. *Journal of Biomedical Informatics*, 73:137–147, 2017.

[240] Bin Yu, Lifeng Lou, Shan Li, Yusen Zhang, Wenying Qiu, Xue Wu, Minghui Wang, and Baoguang Tian. Prediction of protein structural class for low-similarity sequences using chous pseudo amino acid composition and wavelet denoising. *Journal of Molecular Graphics and Modelling*, 76:260–273, 2017.

[241] Guoxian Yu, Wei Luo, Guangyuan Fu, and Jun Wang. Interspecies gene function prediction using semantic similarity. *BMC systems biology*, 10(4):121, 2016.

[242] Guoxian Yu, Hailong Zhu, Carlotta Domeniconi, and Jiming Liu. Predicting protein function via downward random walks on a gene ontology. *BMC bioinformatics*, 16(1):271, 2015.

[243] Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. Semi-supervised word sense disambiguation with neural models. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1374–1385, 2016.

[244] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3010–3019, 2017.

[245] Shu-Bo Zhang and Jian-Huang Lai. Exploring information from the topology beneath the gene ontology terms to improve semantic similarity measures. *Gene*, 586(1):148–157, 2016.

[246] Shu-Bo Zhang and Qiang-Rong Tang. Protein–protein interaction inference based on semantic similarity of gene ontology terms. *Journal of theoretical biology*, 401:30–37, 2016.

[247] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[248] Chenguang Zhao and Zheng Wang. Gogo: An improved algorithm to measure the semantic similarity between gene ontology terms. *Scientific reports*, 8(1):15107, 2018.

[249] Zhi Zhong and Hwee Tou Ng. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83. Association for Computational Linguistics, 2010.

[250] Zhi Zhong and Hwee Tou Ng. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 273–282. Association for Computational Linguistics, 2012.

[251] Quan Zou, Jinjin Li, Li Song, Xiangxiang Zeng, and Guohua Wang. Similarity computation strategies in the microrna-disease network: a survey. *Briefings in functional genomics*, page elv024, 2015.

Appendix A

Long Short-Term Memory (LSTM)

A standard fully-connected Deep Neural Networks (DNN) is simply a multi-layer perceptron (MLP) with many hidden layers between its input and output. Next to nonlinear transformations, the most important advantage of DNNs is their multilevel distributed representation of input. DNNs also do not require detailed assumptions about the input data distribution, a trait to successfully exploit large amounts of data without lapsing into overtraining. Attributed to their topological structure, however, they are lacking in modeling sequence data properly.

Recurrent neural networks (RNNs), shown in Figure A.1, are a DNN containing a self-connected hidden layer designed to resolve the shortcoming of traditional DNNs. One benefit of the recurrent connection is that a memory of previous inputs remains in the network's internal state through mapping real-valued input sequences to real-valued output sequences. As a result, RNNs can exhibit dynamic temporal behavior by accessing to the past context. Context, i.e. the sequence of preceding and succeeding terms that come before and after one target ambiguous term, plays an important role in an accurate disambiguation. Nonetheless, classical RNNs themselves have issues with long-range dependencies as the gradient either explodes or vanishes too quickly during backpropagation.

LSTMs address the vanishing gradient problem in RNNs by incorporating gating functions into their state dynamics (Figure A.2).[10]

Each LSTM network maintains a hidden vector $\boldsymbol{h}$ and a memory cell vector $\boldsymbol{c}$ responsible for controlling state updates and outputs. An LSTM block at time step t takes $\boldsymbol{x}_t, \boldsymbol{h}_{t-1}$ and $\boldsymbol{c}_{t-1}$ as input and produces $\boldsymbol{h}_t$ and $\boldsymbol{c}_t$ via the following formulas (Equation (.1))[1]:

[1]In the equations, $\cdot$ means matrix multiplication, $\odot$ implies element-wise product of vectors, variable in lowercase represents a vector, and uppercase letters denote matrices; σ is the element-wise sigmoid function.

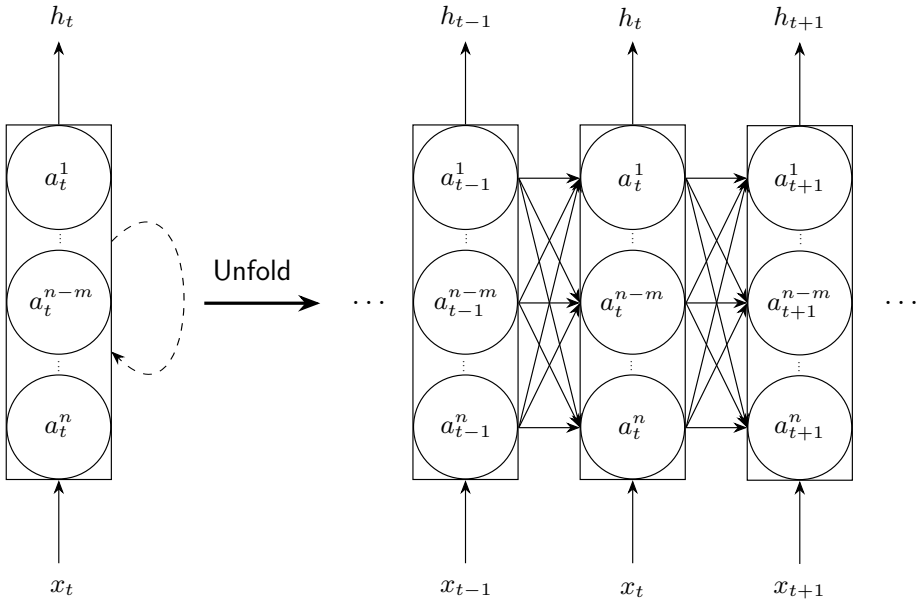

Figure A.1: Recurrent neural network architecture.
From one RNN block to another connections between their units form a directed graph along a sequence. Input $\boldsymbol{x}_i$ and output $\boldsymbol{h}_i$ of each block can be either a scalar or a vector while they stay homogeneous throughout the sequence.

$$
\begin{aligned}
\tilde{\boldsymbol{c}}_t &= \tanh\left(\boldsymbol{W}_c \cdot [\boldsymbol{x}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_c\right) \\
\boldsymbol{i}_t &= \sigma\left(\boldsymbol{W}_{i\cdot} [\boldsymbol{x}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_i\right) \\
\boldsymbol{f}_t &= \sigma\left(\boldsymbol{W}_f \cdot [\boldsymbol{x}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_f\right) \\
\boldsymbol{o}_t &= \sigma\left(\boldsymbol{W}_{o\cdot} [\boldsymbol{x}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_o\right) \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh\left(\boldsymbol{c}_t\right)
\end{aligned}
\tag{.1}
$$

where $\tilde{\boldsymbol{c}}_t$ is the self-recurrent which is equal to standard RNN, $\boldsymbol{i}_t, \boldsymbol{f}_t$ and $\boldsymbol{o}_t$ are the input, forget, and output gates activation vector, $\boldsymbol{c}_t$ is the memory cell vector, $\boldsymbol{W}_i, \boldsymbol{W}_f, \boldsymbol{W}_c, \boldsymbol{W}_o$ are the weight matrices of the input signal (i.e. $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$) with respect to the gatesand the memory cell, and $\boldsymbol{b}_i, \boldsymbol{b}_f, \boldsymbol{b}_c$ and $\boldsymbol{b}_o$ denote the bias vectors. The initial value for $\boldsymbol{c}_0$ and $\boldsymbol{h}_0$ is 0.

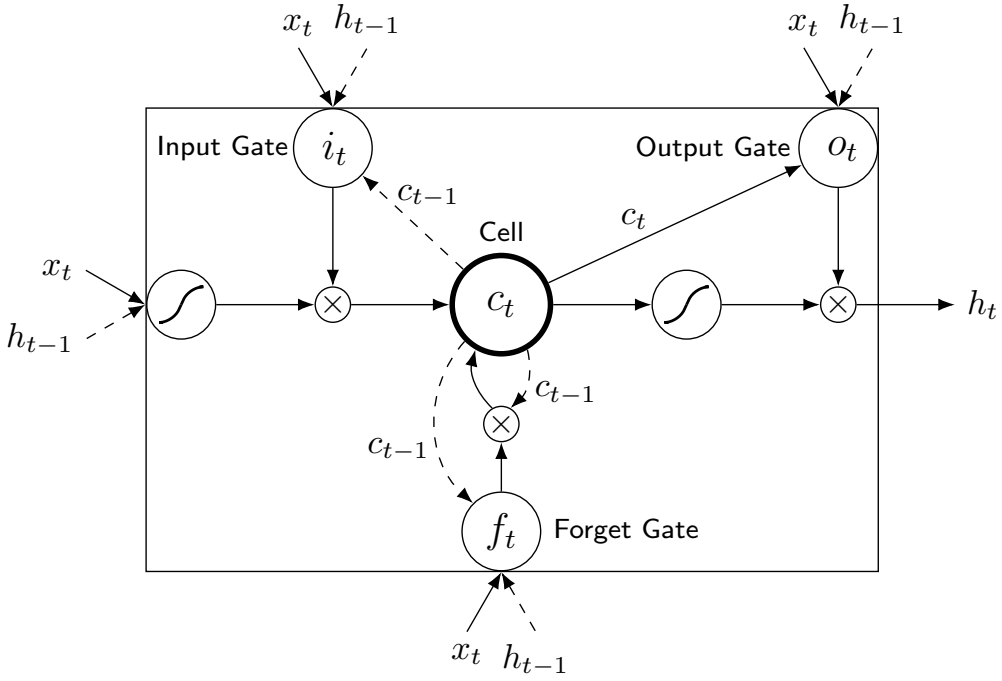

Figure A.2: Long Short-Term Memory network architecture.
For clarity, a single memory cell is shown at time step t. The LSTM blocks are composed of three multiplicative gates: an input gate, a forget gate and an output gate, which in turn control the proportion of input information transferred to a memory cell, the proportion of historical information from the previous state to remember/forget, and the proportion of output information to pass on to the next time step. The output $\boldsymbol{h}_i$ goes to every unit in the next layer.

Standard RNNs and LSTMs however have restrictions as the future input information can not be reached from the current state. To resolve this issue, a Bidirectional LSTM, depicted in Figure A.3, fuses two reversed unidirectional LSTMs. For WSD this means we are able to encode information of both preceding and succeeding terms.

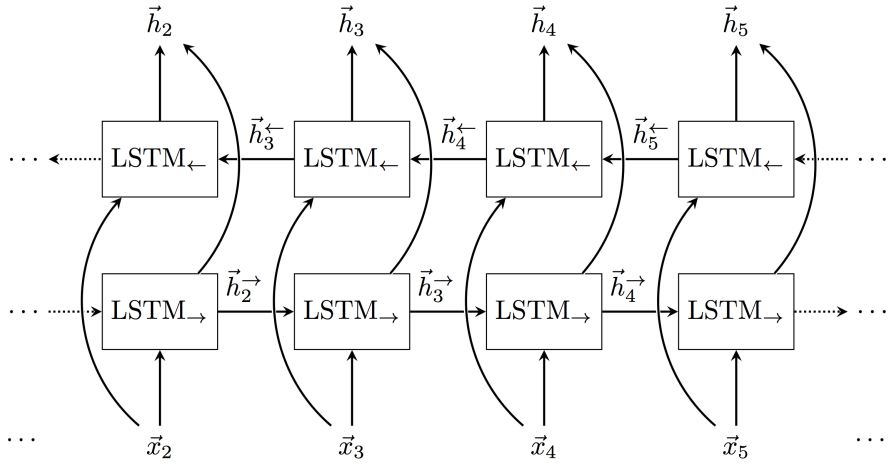

Figure A.3: Bidirectional Long Short-Term Memory network architecture.

Appendix B

Macro Accuracy and Micro Accuracy

Macro- and micro-accuracy will compute slightly different things, and thus their interpretation differs. A macro-accuracy will compute the accuracy independently for each class and then take the average (hence treating all classes equally), whereas a micro accuracy will aggregate the contributions of all classes to compute the average accuracy. Assume we have two confusion matrices for two sample classifiers shown in Figure B.1:

	SAMPLE 1	*True*	
Predicted		**Class A**	**Class B**
	Class A	10	30
	Class B	20	40

SAMPLE 2	**Class 1**	**Class 2**	**Class 3**
Class 1	1	4	7
Class 2	2	5	8
Class 3	3	6	9

Figure B.1: Confusion matrices of two sample classifiers

Sample 1: accuracy = (10 + 40) / (10 + 20 + 30 + 40) = 0.5
Sample 2: accuracy = (1 + 5 + 9) / (1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9) = 0.33
Macro-accuracy (as a whole) = (0.5 + 0.33) / 2 = 0.42
Micro-accuracy (as a whole) = ((10+40) + (1+5+9)) / ((10+20+30+40) + (1+2+3+4+5+6+7+8+9)) = 0.45

In our first experiment we have 203 confusion matrices each for one ambiguous term (it is in each round of validation in a 10-fold validation experiment). In the second experiment the number of confusion matrices is 20 (or 21) in a validation round; each matrix for one unseen ambiguous term (this number depends on our

validation number between 1 to 10).

Appendix C

Results For 203 Ambiguous Terms

Table C.1 represents the average accuracy of the 10 validations for each ambiguous term (in deepBioWSD$_{\textit{pre-trained unsupervised w/ sense embdgs}}$).

Table C.1: Disambiguation Accuracy of 203 terms using deepBioWSD

#	Ambiguous term	Senses (representatives terms)	ACC%
1	veterinary	(1) Veterinary Assistant, (2) Veterinary Medicine	59.6
2	Heregulin	(1) NRG Proteins, (2) NRG1 Protein	61.8
3	Borrelia	(1) Lyme Borreliosis, (2) Borrelia	71.8
4	CI	(1) Ivory Coast, (2) Chile	74
5	Hybridization	(1) Genetic Hybridization, (2) Nucleic Acid Hybridization	75.9
6	B-Cell Leukemia	(1) B-Cell Lymphocytic Leukemia, (2) Chronic B-Lymphocytic Leukemia	78.5
7	HGF	(1) Hybridoma Growth Factor, (2) Hepatocyte Growth Factor	80.3
8	Murine sarcoma virus	(1) Mouse Sarcoma Viruses, (2) Moloney Sarcoma Virus	81.8
9	HHV 8	(1) Kaposi Sarcoma-Associated Herpesvirus, (2) Kaposi Sarcoma	84.5
10	EGG	(1) Ovum, (2) Egg (food product)	84.6
11	Ion	(1) Iontophoreses, (2) Ions	85.8
12	RBC	(1) Red Blood Cell, (2) Red Blood Cell Count	86.1
13	rDNA	(1) Recombinant DNA, Ribosomal DNA	86.2
14	Exercises	(1) Rehabilitation Exercise, (2) Physical Activity	88
15	lens	(1) Lens (device), (2) Lens Disease, (3) Eye Lens	88
16	WT1	(1) WT1 Protein, (2) Wilm's Tumor	88.5
17	Erythrocytes	(1) Red Blood Corpuscle, (2) Red Blood Cell Count	88.6
18	Adrenal	(1) Epinephrine, (2) Adrenal Gland	88.9
19	tomography	(1) Tomography, (2) Computed X Ray Tomography	88.9
20	BR	(1) Brazil, (2) Bromide	89.4
21	Digestive	(1) Ailmentary System, (2) Digestion	89.4
22	Phosphorylase	(1) Glucan Phosphorylase, (2) Glycogen Phosphorylase (muscle form)	89.5
23	Lupus	(1) Lupus Vulgaris, (2) Systemic Lupus Erythematosus, (3) Discoid Lupus	89.9
24	Pleuropneumonia	(1) Pleuropneumonia, (2) Haemobartonella	89.9
25	Ca	(1) Calcium, (2) California, (3) Cornu Ammonis, (4) Canada	90.3
26	Gamma-Interferon	(1) Recombinant Interferon-Gamma, (2) Type II Interferon	90.3
27	CH	(1) China, (2) Switzerland	90.6
28	FAS	(1) Fatty Acid Synthases, (2) Fetal Alcohol Syndrome	90.8
29	Staph	(1) Staphylococcal Infection, (2) Staphylococcus (organism)	90.8
30	TEM	(1) Transmission Electron Microscopy, (2) Triethylene Melamine	90.9
31	Glycoside	(1) Cardiac Glycosides, (2) Glycosides	91.3
32	Coffee	(1) Caffea (plant), (2) Coffee (drink)	91.6
33	posterior pituitary	(1) Posterior Pituitary Hormones, (2) Posterior Pituitary Gland	91.6
34	MAF	(1) Macrophage Activating Factors, (2) Maf Proteins	92.2
35	NEUROFIBROMATOSIS	(1) Neurofibromatosis Syndrome, (2) Neurofibromatosis 1 Gene	92.2
36	Phosphorus	(1) Dietary Phosphorus, (2) Phosphorus (non-metal element)	92.7
37	Potassium	(1) Potassium (an element), (2) Dietary Potassium	92.8
38	Strep	(1) Streptococcus (organism), (2) Streptococcal Infection	92.8
39	STEM	(1) Plant Stem, (2) Scanning Transmission Electron Microscopy	93
40	WBS	(1) Williams-Beuren Syndrome, (2) Beckwith-Wiedemann Syndrome	93
41	Leishmaniasis	(1) Leishmania Vaccines, (2) Leishmania Infection	93.1
42	Torula	(1) Cryptococcus neoformans Infection, (2) Cryptococcus (organism)	93.2
43	Familial Adenomatous Polyposis	(1) APC Gene, (2) Familial Polyposis Syndrome	94.1
44	Arteriovenous Anastomoses	(1) Surgical Arteriovenous Shunt, (2) Arteriovenous Anastomose	94.5
45	Milk	(1) Mammary Gland Milk, (2) Mother's milk	94.6
46	Malaria	(1) Malaria Vaccines, (2) Malaria (infection)	94.7
47	Nurse	(1) Nurse, (2) Breast Feeding	94.8
48	TMJ	(1) Temporomandibular Joint Syndrome, (2) Temporomandibular Joint Structure	94.9
49	Laryngeal	(1) Larynx, (2) Artificial Larynx	95.2
50	Haemophilus ducreyi	(1) Chancroid, (2) Haemophilus ducreyi	95.9

#	Ambiguous term	Senses (representatives terms)	ACC%
51	Yellow Fever	(1) Yellow Fever (infectious disease), (2) Yellow Fever Vaccine	95.9
52	MCC	(1) MCC Gene, (2) Merkel Cell Carcinoma	96
53	Medullary	(1) Adrenal Medulla, (2) Medulla Oblongata	96
54	Crown	(1) Tooth Crown, (2) Dental Prosthetic Crown	97
55	Parotitis	(1) Epidemic Parotitis, (2) Parotiditides	97.1
56	Cement	(1) Dental Adhesive, (2) Dental Cementum	97.2
57	Semen	(1) Plant Zygote, (2) Seminal Plasma	97.2
58	Schistosoma mansoni	(1) Schistosoma mansoni (organism), (2) Schistosoma mansoni Infection	97.4
59	Moles	(1) Family Talpidae, (2) Skin mole	97.7
60	DE	(1) Delaware, (2) Germany	98
61	Platelet	(1) Platelet Count, (2) Blood Platelet	98
62	Fe	(1) Dietary Iron, (2) Iron (metallic element)	98.4
63	Nursing	(1) Nursing, (2) Breast Feeding	98.4
64	Astragalus	(1) Astragalus Bone, (2) Astragalus Plant	98.5
65	Brucella abortus	(1) Bovine Brucellosis, (2) Brucella melitensis biovar abortus	98.5
66	Cold	(1) Cold Temperature, (2) Chronic Obstructive Lung Disease, (3) Common Cold	98.5
67	Hip	(1) Coxa, (2) Ischium	98.6
68	CIS	(1) Carcinoma in Situ, (2) Commonwealth of Independent States	99.5
69	Ice	(1) Methamphetamine, (2) Frozen Water, (3) Interleukin 1 Converting Enzyme	99.5
70	Pneumocystis	(1) Pneumocystis Pneumonia, (2) Pneumocystis	99.5
71	SARS-associated coronavirus	(1) Severe Acute Respiratory Syndrome (infection), (2) SARS Associated Coronavirus (virus)	99.5
72	Cell	(1) Cell, (2) Cellular Phone	99.6
73	Polymyalgia Rheumatica	(1) Forestier-Certonciny Syndrome, (2) Giant Cell Arteritis	99.6
74	TAT	(1) Thematic Apperception Test, (2) TAT Gene, (3) TAT Protein	99.6
75	HIV	(1) HIV Infection, (2) AIDS Virus	99.7
76	AA	(1) Amino Acids, (2) Alcoholics Anonymous	99.9
77	Ala	(1) L-Isomer Alanine, (2) Delta-Aminolevulinic Acid, (3) Alpha Linolenic Acid	99.9
78	Cardiac pacemaker	(1) Sinoatrial Node, (2) Artificial Pacemaker	99.9
79	Cholera	(1) Vibrio cholerae Infection, (2) Cholera Vaccines	99.9
80	Compliance	(1) Index of Expandability, (2) Patient Adherence	99.9
81	Cortical	(1) Kidney Cortex, (2) Cerebral Cortex, (3) Adrenal Cortex	99.9
82	DDD	(1) Mitotane, (2) Dichlorodiphenyldichloroethane	99.9
83	eCG	(1) Equine Gonadotropins, (2) Electrocardiography	99.9
84	Eels	(1) Electron Energy Loss Spectroscopy, (2) Anguilliformes	99.9
85	Hemlock	(1) Tsuga (coniferous tree), (2) Hemlock (poisonous plant)	99.9
86	Iris	(1) Eye Iris, (2) Iris Plant	99.9
87	LABOR	(1) Obstetric Labor, (2) Working	99.9
88	Lactation	(1) Milk Secretion, (2) Breast Feeding	99.9
89	lymphogranulomatosis	(1) Boeck's Sarcoid, (2) Hodgkin's Granuloma	99.9
90	MBP	(1) Myelin Basic Protein, (2) Mannan Binding Protein	99.9
91	NBS	(1) Neuroblastoma, (2) Seemanova Syndrome II	99.9
92	Projection	(1) Forecasting, (2) Mental defence through projection	99.9
93	Radiation	(1) Radiation, (2) Radiotherapy	99.9
94	Respiration	(1) Respiration, (2) Cellular Respiration	99.9
95	Retinal	(1) Vitamin A Aldehyde, (2) Retina	99.9
96	SARS	(1) Severe Acute Respiratory Syndrome (infection), (2) SARS Associated Coronavirus (virus)	99.9
97	Sodium	(1) Sodium (metallic element), (2) Dietary Sodium	99.9
98	Synapsis	(1) Synapse, (2) Chromosome Pairing	99.9
99	THYMUS	(1) Thymus Extract, (2) Thymus Gland, (3) Thymus Plant	99.9
100	Tolerance	(1) Drug Tolerance, (2) Immune Tolerance	99.9

#	Ambiguous term	Senses (representatives terms)	ACC%
101	TPO	(1) Thrombopoietin, (2) Iodotyrosine Deiodase	99.9
102	ADA	(1) American Dental Association, (2) Adenosine Deaminase	100
103	ADH	(1) Alcohol Dehydrogenase, (2) Arginine Vasopressin	100
104	ADP	(1) Adenosine Diphosphate, (2) Automatic Data Processing	100
105	ALS	(1) Antilymphocyte Serum, (2) Amyotrophic Lateral Sclerosis	100
106	ANA	(1) American Nurses Association, (2) Antinuclear Antibody	100
107	BAT	(1) Brown Adipose Tissue, (2) Chiroptera	100
108	BLM	(1) Bloom Syndrome, (2) Bleomycin	100
109	BPD	(1) Bronchopulmonary Dysplasia, (2) Borderline Personality Disorder	100
110	BSA	(1) Body Surface Area, (2) Bovine Serum Albumin	100
111	BSE	(1) Bovine Spongiform Encephalitis, (2) Breast Self-Examination	100
112	CAD	(1) Computer Assisted Diagnosis, (2) Coronary Artery Disease	100
113	Callus	(1) Bony Callus, (2) Skin Callus	100
114	CAM	(1) Cell Adhesion Molecules, (2) Chorioallantoic Membrane	100
115	CCD	(1) Central Core Disease, (2) Cleidocranial Dysostosis	100
116	CCl4	(1) CCL4 Chemokine, (2) Carbon Tetrachloride	100
117	CDA	(1) Cladribine, (2) Congenital Dyserythropoietic Anemia	100
118	CDR	(1) Deoxycytidine, (2) Immunoglobulin Hypervariable Region	100
119	Cilia	(1) Cilium, (2) Eyelashes	100
120	CLS	(1) Coffin-Lowry Syndrome, (2) Capillary Leak Syndrome	100
121	CNS	(1) Clinical Nurse Specialist, (2) Central Nervous System	100
122	Cortex	(1) Cerebral Cortex, (2) Adrenal Cortex Disease	100
123	CP	(1) Cerebral Palsy, (2) Corynebacterium acnes, (2) Cleft Palate	100
124	CPDD	(1) Calcium Pyrophosphate Deposition Disease, (2) cis-Diamminedichloroplatinum	100
125	Crack	(1) Crack Cocaine, (2) Tooth Fracture	100
126	CRF	(1) Chronic Renal Failure, (2) Corticotropin Releasing Hormone	100
127	cRNA	(1) Nurse Anesthetist, (2) Complementary RNA	100
128	CTX	(1) Cerebral Cholesterinoses, (2) Cyclophosphamide	100
129	DAT	(1) Alzheimer's Disease, (2) DAT Dopamine Transporter	100
130	DBA	(1) Diamond Blackfan Anemia, (2) DBA Mice	100
131	dC	(1) District of Columbia, (2) Cytosine Deoxyriboside	100
132	DDS	(1) Diaminodiphenylsulfone, (2) Denys Drash Syndrome, (3) Drug Delivery System	100
133	DI	(1) Diabetes Insipidus, (2) Ploidy	100
134	DON	(1) Nurse Administrator, (2) Diazooxonorleucine	100
135	drinking	(1) Drinking (liquid consumption), (2) Alcohol Drinking	100
136	EM	(1) Electron Microscopy, (2) Estramustine	100
137	EMS	(1) Emergency Medical Service, (2) Ethylmethane Sulfonate	100
138	Epi	(1) Epinephrine, (2) Epirubicin	100
139	ERP	(1) Evoked Potential, (2) Endoscopic Retrograde Cholangiopancreatography	100
140	ERUPTION	(1) Skin Rash, (2) Tooth Eruption	100
141	FA	(1) Folic Acid, (2) Fanconi Anemia	100
142	Fish	(1) Fishes, (2) Fluorescent in Situ Hybridization	100
143	Follicle	(1) Hair Follicle, (2) Ovarian Follicle	100
144	Follicles	(1) Hair Follicle, (2) Ovarian Follicle	100
145	FTC	(1) United States Federal Trade Commission, (2) Follicular Thyroid Carcinoma	100
146	GAG	(1) gag Gene, (2) Glycosaminoglycans	100
147	Ganglion	(1) Ganglia, (2) Ganglionic Cyst	100
148	Gas	(1) Gases, (2) Flatulence	100
149	HCl	(1) Hairy Cell Leukemia, (2) Hydrochloric Acid	100
150	HPS	(1) Hantavirus Infection, (2) Hermanski Pudlak Syndrome	100

#	Ambiguous term	Senses (representatives terms)	ACC%
151	HR	(1) Croatia, (2) Heart Rate	100
152	IA	(1) Iowa, (2) Intra-Arterial Injection	100
153	INDO	(1) Indonesia, (2) Indomethacin	100
154	IP	(1) Immune Precipitation, (2) Bloch-Siemens Syndrome	100
155	ITP	(1) Inosine Triphosphate, (2) Autoimmune Thrombocytopenia	100
156	JP	(1) Aggressive Periodontitis, (2) Japan	100
157	Language	(1) Programming Language, (2) Natural Language	100
158	Lawsonia	(1) Lawsonia Plant, (2) Lawsonia Bacteria	100
159	MHC	(1) Myosin Heavy Chain, (2) Histocompatibility Complex	100
160	MRS	(1) Melkersson-Rosenthal Syndrome, (2) Magnetic Resonance Spectroscopies	100
161	NM	(1) Nitrogen Mustard, (2) New Mexico	100
162	NPC	(1) Niemann Pick's Disease, (2) Nuclear Pore	100
163	OCD	(1) Obsessive Compulsive Disorder, (2) Osteochondritis Dissecans	100
164	OH	(1) Hydroxyl Radical, (2) Ohio	100
165	Orf	(1) Protein Coding Region, (2) Orf Virus Infection	100
166	ORI	(1) Origin of Replication, (2) Office of Research Integrity	100
167	PAC	(1) Premature Atrial Contraction, (2) P1-Derived Artificial Chromosome	100
168	PAF	(1) Progressive Autonomic Failure, (2) Platelet Activating Factor	100
169	PCA	(1) Principal Component Analyses, (2) Posterior Cerebral Artery,(3) Patient Controlled Analgesia, (3) Patient Controlled Analgesia, (4) Passive Cutaneous Anaphylaxis, (5) p-Chloroamphetamine	100
170	PCB	(1) Polychlorinated Biphenyls, (2) Procarbazine	100
171	PCD	(1) Apoptosis, (2) Kartagener's Syndrome	100
172	PCP	(1) Pneumocystis carinii Pneumonia, (2) Pentachlorophenol, (3) Phencyclidine	100
173	PEP	(1) Phosphoenolpyruvate, (2) Peplomycin	100
174	PHA	(1) Kidney Bean Lectins, (2) Pelger Huet Anomaly	100
175	Pharmaceutical	(1) Dosage Form, (2) Pharmacy	100
176	pI	(1) Isoelectric Point, (2) Mitotic Index	100
177	Plague	(1) Yersinia pestis Infection, (2) Plague Vaccine	100
178	Plaque	(1) Senile Plaque, (2) Dental Plaque	100
179	POL	(1) pol Gene, (2) Poland	100
180	PR	(1) Puerto Rico, (2) Progesterone Receptor	100
181	PVC	(1) Premature Ventricular Complex, (2) Polyvinyl Chloride	100
182	RA	(1) Refractory Anemia, (2) Radium, (3) Rheumatoid Arthritis	100
183	RB	(1) Retinoblastoma, (2) Rubidium	100
184	Root	(1) Plant Root, (2) Tooth Root	100
185	RSV	(1) Rous sarcoma virus, (2) Respiratory Syncytial Virus	100
186	SCD	(1) Sudden Cardiac Death, (2) Sickle Cell Anemia	100
187	sex factor	(1) Sex Factor (in studies), (2) Bacterial Sex Factor	100
188	SLS	(1) Sjogren Larsson Syndrome, (2) Sodium Lauryl Sulfate	100
189	SPR	(1) Substance P Receptor, (2) Surface Plasmon Resonance	100
190	SS	(1) Synovial Sarcoma, (2) Sweet's Syndrome	100
191	Sterilization	(1) Reproductive Sterilization, (2) Sterilization (disinfectant)	100
192	Tax	(1) Paclitaxel, (2) Taxes	100
193	TLC	(1) Total Lung Capacity, (2) Thin Layer Chromatography	100
194	TMP	(1) Trimethoprim, (2) Thymidylic Acid	100
195	TNC	(1) Tenascin-C, (2) Troponin-C	100
196	TNT	(1) Troponin-T, (2) Trinitrotoluene	100
197	TPA	(1) Phorbol Myristate Acetate, (2) Tissue Plasminogen Activator	100
198	TRF	(1) Thyrotropin Releasing Factor, (2) T-Cell Replacing Factor	100
199	TSF	(1) Thrombocytopoiesis Stimulating Factor, (2) T-Cell Stimulating Factor	100
200	TYR	(1) Tyrosine, (2) Tyrosinase	100
201	US	(1) United States, (2) Ultrasonography	100
202	Ventricles	(1) Heart Ventricle, (2) Cerebral Ventricle	100
203	Wasp	(1) WASP Protein, (2) Wasp (animal)	100
Average			**96.82**

Appendix D

Copyright Permissions

This appendix includes the copyright forms for our publications in:

1. **Journal Paper:** "*simDEF: definition-based semantic similarity measure of gene ontology terms for functional similarity analysis of genes*" (Bioinformatics, Oxford, England) [153]

2. **Conference Proceeding:** "*One single deep bidirectional LSTM network for word sense disambiguation of text data*" (In Advances in Artificial Intelligence: 31st Canadian Conference on Artificial Intelligence, Canadian AI 2018, Toronto, ON, Canada) [157]

3. **Journal Paper:** "*deepBioWSD: effective deep neural word sense disambiguation of biomedical text data*" (JAMIA: Journal of the American Medical Informatics Association) [154]

OXFORD UNIVERSITY PRESS

Robert Beiko ▾

Display Archive Copyright Agreement

To pay any relevant charges relating to your paper , or to claim or purchase issues, please click [Proceed] below.

Back

Proceed

Licence to Publish

OXFORD JOURNALS
OXFORD UNIVERSITY PRESS

Journal:	Bioinformatics
DOI:	10.1093/bioinformatics/btv755
Title:	simDEF: Definition-based Semantic Similarity Measure of Gene Ontology Terms for Functional Similarity Analysis of Genes

Standard Licence

You hereby grant to Oxford University Press an exclusive licence for the full period of copyright throughout the world:

to publish the final version of the Article in the above Journal, and to distribute it and/or to communicate it to the public, either within the Journal, on its own, or with other related material throughout the world, in printed, electronic or any other format or medium whether now known or hereafter devised;

to make translations and abstracts of the Article and to distribute them to the public;

to authorize or grant licences to third parties to do any of the above;

to deposit copies of the Article in online archives maintained by OUP or by third parties authorized by OUP.

You authorize us to act on your behalf to defend the copyright in the Article if anyone should infringe it and to register the copyright of the Article in the US and other countries, if necessary.

In the case of a multi authored article, you confirm that you are authorized by your co-authors to enter the licence on their behalf.

You confirm to OUP that the Article

is your original work;

has not previously been published (in print or electronic format), is not currently under consideration by another journal, or if it has already been submitted to other journal, it will be immediately withdrawn;

will not be submitted for publication to any other journal following acceptance in the above Journal; and

OUP will be the first publisher of the Article.

You warrant to OUP that

no part of the Article is copied from any other work,

you have obtained ALL the permissions required (for print and electronic use) for any material you have used from other copyrighted publications in the Article; and

you have exercised reasonable care to ensure that the Article is accurate and does not contain anything which is libellous, or obscene, or infringes on anyone's copyright, right of privacy, or other rights.

Further Information

(Full details of OUP's publication rights policies, including author rights can be found at http://www.oxfordjournals.org/access_purchase/publication_rights.html)

Author Self-Archiving Policy

On publication of your Article in the Journal you are not required to remove any previously posted ORIGINAL VERSIONS from your own personal website or that of your employer or free public servers of articles in your subject area, provided (1) you include a link (url) to the VERSION OF RECORD on the Journal’s website; AND (2) the Journal is attributed as the original place of publication with the correct citation details given.

You may post the ACCEPTED MANUSCRIPT of the Article (but not the published version itself) onto

your own website, your institution's website and in institutional or subject-based repositories, PROVIDED THAT it is not made publicly available until 12 MONTHS after the online date of publication, and that: (1) you include a link (url) to the VERSION OF RECORD on the Journal's website; (2) the Journal is attributed as the original place of publication with the correct citation details given.

Notes

Author's Original Version: an unrefereed manuscript version of the article, as submitted for review by a journal

Accepted Manuscript: the final draft author manuscript, as accepted for publication by a journal, including modifications based on referees' suggestions but before it has undergone copyediting and proof correction

Free Link to Published Article

On publication of your article, you will receive a URL, giving you access to the published article on the Journal website, and information on use of this link.

Educational Use

You may use the Article within your employer's institution or company for educational or research purposes only, including use in course-packs, as long as: (1) you do not use it for commercial purposes or re-distribution outside of the institution/company; (2) you acknowledge the Journal as the original place of publication with the correct citation details given.

'Robert Beiko' signed this copyright agreement on 2015-12-22 18:19:56 GMT.

View as PDF

Print this copyright agreement

Oxford Journals | My Account | Author Services https://secureacademic.oup.com/licensing/display_archive.html

Reader

OXFORD
UNIVERSITY PRESS

Registered VAT number: GB 125 50 67 30 | Copyright © 2016

Consent to Publish

Lecture Notes in Computer Science

Springer

Title of the Book or Conference Name: The 31st Canadian Conference on Artificial Intelligence

Volume Editor(s): Ebrahim Bagheri, Jackie Cheung

Title of the Contribution: One Single Deep Bidirectional LSTM Network for Word Sense Disambiguation of Text Data

Author(s) Name(s): Ahmad Pesaranghader, Stan Matwin, Marina Sokolova, Ali Pesaranghader

Corresponding Author's Name, Address, Affiliation and Email: Ahmad Pesaranghader, Institute for Big Data Analytics, Halifax, NS B3H 4R2, Canada, ahmad.pgh@dal.ca

When Author is more than one person the expression "Author" as used in this agreement will apply collectively unless otherwise indicated.

§ 1 Rights Granted

Author hereby grants and assigns to Springer International Publishing AG, Cham (hereinafter called Springer) the exclusive, sole, permanent, world-wide, transferable, sub-licensable and unlimited right to reproduce, publish, distribute, transmit, make available or otherwise communicate to the public, translate, publicly perform, archive, store, lease or lend and sell the Contribution or parts thereof individually or together with other works in any language, in all revisions and versions (including soft cover, book club and collected editions, anthologies, advance printing, reprints or print to order, microfilm editions, audiograms and videograms), in all forms and media of expression including in electronic form (including offline and online use, push or pull technologies, use in databases and networks for display, print and storing on any and all stationary or portable end-user devices, e.g. text readers, audio, video or interactive devices, and for use in multimedia or interactive versions as well as for the display or transmission of the Contribution or parts thereof in data networks or seach engines), in whole, in part or in abridged form, in each case as now known or developed in the future, including the right to grant further time-limited or permanent rights. For the purposes of use in electronic forms, Springer may adjust the Contribution to the respective form of use and include links or otherwise combine it with other works. For the avoidance of doubt, Springer has the right to permit others to use individual illustrations and may use the Contribution for advertising purposes.

The copyright of the Contribution will be held in the name of Springer. Springer may take, either in its own name or in that of copyright holder, any necessary steps to protect these rights against infringement by third parties. It will have the copyright notice inserted into all editions of the Contribution according to the provisions of the Universal Copyright Convention (UCC) and dutifully take care of all formalities in this connection in the name of the copyright holder.

§ 2 Regulations for Authors under Special Copyright Law

The parties acknowledge that there may be no basis for claim of copyright in the United States to a Contribution prepared by an officer or employee of the United States government as part of that person's official duties. If the Contribution was performed under a United States government contract, but Author is not a United States government employee, Springer grants the United States government royalty-free permission to reproduce all or part of the Contribution and to authorize others to do so for United States government purposes.

If the Contribution was prepared or published by or under the direction or control of Her Majesty (i.e., the constitutional monarch of the Commonwealth realm) or any Crown government department, the copyright in the Contribution shall, subject to any agreement with Author, belong to Her Majesty.

If the Contribution was created by an employee of the European Union or the European Atomic Energy Community (EU/Euratom) in the performance of their duties, the regulation 31/EEC, 11/EAEC (Staff Regulations) applies, and copyright in the Contribution shall, subject to the Publication Framework Agreement (EC Plug), belong to the European Union or the European Atomic Energy Community.

If Author is an officer or employee of the United States government, of the Crown, or of EU/Euratom, reference will be made to this status on the signature page.

§ 3 Rights Retained by Author

Author retains, in addition to uses permitted by law, the right to communicate the content of the Contribution to other scientists, to share the Contribution with them in manuscript form, to perform or present the Contribution or to use the content for non-commercial internal and educational purposes, provided the Springer publication is mentioned as the

original source of publication in any printed or electronic materials. Author retains the right to republish the Contribution in any collection consisting solely of Author's own works without charge subject to ensuring that the publication by Springer is properly credited and that the relevant copyright notice is repeated verbatim.
Author may self-archive an author-created version of his/her Contribution on his/her own website and/or the repository of Author's department or faculty. Author may also deposit this version on his/her funder's or funder's designated repository at the funder's request or as a result of a legal obligation. He/she may not use the publisher's PDF version, which is posted on SpringerLink and other Springer websites, for the purpose of self-archiving or deposit. Furthermore, Author may only post his/her own version, provided acknowledgment is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be provided by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]". The DOI (Digital Object Identifier) can be found at the bottom of the first page of the published paper.
Prior versions of the Contribution published on non-commercial pre-print servers like ArXiv/CoRR and HAL can remain on these servers and/or can be updated with Author's accepted version. The final published version (in pdf or html/xml format) cannot be used for this purpose. Acknowledgment needs to be given to the final publication and a link must be inserted to the published Contribution on Springer's website, by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]".
Author retains the right to use his/her Contribution for his/her further scientific career by including the final published paper in his/her dissertation or doctoral thesis provided acknowledgment is given to the original source of publication. Author also retains the right to use, without having to pay a fee and without having to inform the publisher, parts of the Contribution (e.g. illustrations) for inclusion in future work, and to publish a substantially revised version (at least 30% new content) elsewhere, provided that the original Springer Contribution is properly cited.

§ 4 Warranties

Author warrants that the Contribution is original except for such excerpts from copyrighted works (including illustrations, tables, animations and text quotations) as may be included with the permission of the copyright holder thereof, in which case(s) Author is required to obtain written permission to the extent necessary and to indicate the precise sources of the excerpts in the manuscript. Author is also requested to store the signed permission forms and to make them available to Springer if required.
Author warrants that he/she is entitled to grant the rights in accordance with Clause 1 "Rights Granted", that he/she has not assigned such rights to third parties, that the Contribution has not heretofore been published in whole or in part, that the Contribution contains no libelous statements and does not infringe on any copyright, trademark, patent, statutory right or proprietary right of others, including rights obtained through licenses; and that Author will indemnify Springer against any costs, expenses or damages for which Springer may become liable as a result of any breach of this warranty.

§ 5 Delivery of the Work and Publication

Author agrees to deliver to the responsible Volume Editor (for conferences, usually one of the Program Chairs), on a date to be agreed upon, the manuscript created according to the Springer Instructions for Authors. Springer will undertake the reproduction and distribution of the Contribution at its own expense and risk. After submission of the Consent to Publish form Signed by the Corresponding Author, changes of authorship, or in the order of the authors listed, will not be accepted by Springer.

§ 6 Author's Discount

Author is entitled to purchase for his/her personal use (directly from Springer) the Work or other books published by Springer at a discount of 40% off the list price as long as there is a contractual arrangement between Author and Springer and subject to applicable book price regulation. Resale of such copies or of free copies is not permitted.

§ 7 Governing Law and Jurisdiction

This agreement shall be governed by, and shall be construed in accordance with, the laws of Switzerland. The courts of Zug, Switz risdiction.

Correspon responsibility for releasing this material on behalf of any and all Co-authors.

Signature **Date:** 22 Feb 2018

I'm an employee of the US Government and transfer the rights to the extent transferable (Title 17 §105 U.S.C. applies)

I'm an employee of the Crown and copyright on the Contribution belongs to Her Majesty

I'm an employee of the EU or Euratom and copyright on the Contribution belongs to EU or Euratom

16.02.2016
10:50

OXFORD UNIVERSITY PRESS

ahmad.pgh@dal.ca ▾

Display Copyright Agreement

Finish

Licence to Publish

OXFORD JOURNALS
OXFORD UNIVERSITY PRESS

Journal:	Journal of the American Medical Informatics Association
DOI:	10.1093/jamia/ocy189
Title:	deepBioWSD: Effective Deep Neural Word Sense Disambiguation of Biomedical Text Data

Standard Licence

You hereby grant to Oxford University Press an exclusive licence for the full period of copyright throughout the world:

to publish the final version of the Article in the above Journal, and to distribute it and/or to communicate it to the public, either within the Journal, on its own, or with other related material throughout the world, in printed, electronic or any other format or medium whether now known or hereafter devised;

to make translations and abstracts of the Article and to distribute them to the public;

to authorize or grant licences to third parties to do any of the above;

to deposit copies of the Article in online archives maintained by OUP or by third parties authorized by OUP.

You authorize us to act on your behalf to defend the copyright in the Article if anyone should infringe it and to register the copyright of the Article in the US and other countries, if necessary.In the case of a multi authored article, you confirm that you are authorized by your co-authors to enter the licence on their behalf.

You confirm to OUP that the Article:

is your original work;

has not previously been published (in print or electronic format), is not currently under consideration by another journal, or if it has already been submitted to other journal, it will be immediately withdrawn;

will not be submitted for publication to any other journal following acceptance in the above Journal; and

OUP will be the first publisher of the Article.

You warrant to OUP that:

no part of the Article is copied from any other work,

you have obtained ALL the permissions required (for print and electronic use) for any material you have used from other copyrighted publications in the Article; and

you have exercised reasonable care to ensure that the Article is accurate and does not contain anything which is libellous, or obscene, or infringes on anyone's copyright, right of privacy, or other rights.

RETAINED RIGHTS

Abstract and Citation information - Authors may reuse the Abstract and Citation information (e.g. Title, Author name, Publication dates) of their article anywhere at any time including social media such as Facebook, blogs and Twitter, providing that where possible a link is included back to the article on the OUP site. Preferably the link should be, or include, the Digital Object Identifier (DOI) which can be found in the Citation information about your article online.

Author's Original Version - Authors may reuse the Author's Original Version (AOV) anywhere at any time, providing that once the article is accepted they provide a statement of acknowledgement, and that once the article has been published this acknowledgement is updated to provide details such as the volume and issue number, the DOI, and a link to the published article on the journal's website

Accepted Manuscript - On first online publication authors may immediately upload their Accepted Manuscript (AM) to their own personal webpage, or their institutional repository on the proviso that it is not made publically available until **12 months** after the online date of publication. After the embargo period, authors may make their AM publicly available through their institutional repository or other non-commercial repositories PROVIDED THAT (1) you include a link (URL) to the VERSION OF RECORD on the journals' website; and (2) the Journal is attributed as the original place of publication with the correct citation details given. AM may not be uploaded to commercial websites or repositories, unless the website or repository has signed a licensing agreement with OUP.

Version of Record - The Version of Record (VOR) as it appears in the journal following copyediting and proof correction may not be deposited by authors in institutional repositories or posted to third party websites unless the repository or website has signed an agreement with OUP allowing posting.

Free link to published article - On publication, authors will be sent an online offprint link allowing access to their article on the OUP website without subscription. This link may be shared directly with interested colleagues, but is not intended for mass distribution through, repositories, or social media. If you wish to share links or publicize your article we would ask that you instead distribute a link to the abstract of the article.

Funding bodies - Please be aware that you are responsible for all funding agency compliance and the accuracy of information you provide in relation to your article. OUP and/or the controlling Learned Society shall not be responsible for checking that funding agency requirements have been complied with.

Educational use - You may use the Article within your employer's institution or company for educational or research purposes only, including use in course-packs, as long as: (1) you do not use it for commercial purposes or re-distribution outside of the institution/company; (2) you acknowledge the Journal as the original place of publication with the correct citation details given.

Further information - details of OUP's publication rights policies, including author rights can be found
at https://academic.oup.com/journals/pages/access_purchase/rights_and_permissions/publication_rights

' ahmad.pgh@dal.ca' signed this copyright agreement on 2018-12-19 15:01:15 GMT.

View as PDF Print this copyright agreement

Contact Us
Help
Ordering
Shipping
Returns

Privacy policy
Cookie policy
Legal notices
Site map
Accessibility
Get Adobe Reader

US Customer Services
Tel: +1-800-852-7323 (toll free)
Email: jnlorders@oup.com

UK Customer Services
(09:00-17:00 GMT, Monday to Friday)
Tel: +44(0)1865 353907
Email: jnls.cust.serv@oup.com

OXFORD
UNIVERSITY PRESS

Registered VAT number: GB 125 50 67 30 | Copyright © 2016