UNIVERSITY OF CALIFORNIA, SAN DIEGO

Relation Extraction using Convolution Neural Networks for curation of GWAS catalog

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Computer Science

by

Ankit Goyal

Committee in charge:

Professor Chun-Nan Hsu, Chair
Professor Julian McAuley, Co-Chair
Professor Kamalika Chaudhuri

2016

The thesis of Ankit Goyal is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2016

DEDICATION

I dedicate this thesis to my family and friends, without whom this thesis or a master's degree would not have been possible. Their patience and support have been the most valuable contributions to my course.

EPIGRAPH

The scientists of today think deeply instead of clearly.
One must be sane to think clearly, but one can think deeply and be quite insane.

–Nikola Tesla

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisor Professor Chun-Nan Hsu for his constant and invaluable advice and support throughout my masters degree and thesis work. His immense knowledge and guidance helped me in conducting research and in the writing of this thesis.

Besides my advisor, I would like to thank my thesis committee members, Professor Julian McAuley and Professor Kamalika Chaudhuri, for taking out time to review my work and for their valuable comments.

This work was supported by the National Human Genome Research Institute (NHGRI) of the National Institutes of Health (NIH) under award number U01HG006894.

The work in Section 2.1 is a reprint of the material as it appears in "Natural Language Processing using Kepler Workflow System: First Steps" in Procedia Computer Science, Vol. 80. I am thankful to my co-authors Alok Singh, Shitij Bhargava, Daniel Crawl, Ilkay Altintas, and Chun-Nan Hsu for their inputs and support during the research work. The thesis author was the primary investigator and author of this paper.

Material from Chapter 2, 3, and 4 in part is currently being prepared for submission for the publication of material. The thesis author was the primary investigator and author of this material.

VITA

1991	Born in Gwalior, India
2008-2012	B.Tech., Computer Science and Engineering, Indian Institute of Technology, Jodhpur, India
2012-2015	Software Development Engineer, Microsoft Corporation, Hyderabad, India
2015-2017	M.S., Computer Science, University of California, San Diego, USA

PUBLICATIONS

Ankit Goyal, Alok Singh, Shitij Bhargava, Daniel Crawl, Ilkay Altintas, and Chun-Nan Hsu. "Natural Language Processing using Kepler Workflow System: First Steps." *Procedia Computer Science* 80 (2016): 712-721.

Ankit Goyal and Chun-Nan Hsu. "Relation Extraction from Biomedical Texts using Convolution Neural Networks" *SoCal Machine Learning Symposium*. Poster Presentation - 2016

ABSTRACT OF THE THESIS

Relation Extraction using Convolution Neural Networks for curation of GWAS catalog

by

Ankit Goyal

Master of Science in Computer Science

University of California, San Diego, 2016

Professor Chun-Nan Hsu, Chair
Professor Julian McAuley, Co-Chair

A crucial area of Natural Language Processing is information extraction, the study of the identification and extraction of concepts of interest (“genes”, “diseases”, etc.). This thesis proposes algorithms that extract relational information from biomedical text using machine learning techniques. In particular, the work presented here concerns with the identification of entity mentions from the given text which exhibits a semantic relationship among them and extraction of these entities for the curation of biomedical databases. One such database is the Genome-Wide Association Study (GWAS) catalog which is manually curated, literature-derived collection of all GWAS and is the center of our work.

This work presents a machine learning approach to natural language processing to automatically extract the information of GWAS catalog from a new biomedical text. We focus on characteristics of the population samples used in the experiments i.e. the experimental stage, the ethnicity groups of individuals and the size of the population pool. Our approach for relation extraction is based on convolutional neural networks with different filter sizes using already curated data from existing biomedical databases as training examples. Although these neural networks have been previously used for relation extraction and other natural language processing tasks, to the best of my knowledge they have never been applied to the problem of automatic data curation, and we focus primarily on developing a learning framework to deal with this issue specifically. We evaluated our approach by extracting the sample characteristics as tuple relations and achieved an improvement over the existing approach. Our neural network models were able to outperform an approach developed previously for the same task as a baseline.

Chapter 1

Introduction

The growth of digital information in today's world is staggering: from an estimated 2.6 EB[1] in 1986 to 15.8 EB in 1993 to 54.5 EB in 2000 and 295 EB in 2007 [16] and humanity is producing more and more data year at an astonishing rate. The continuing rapid development of the Internet makes it very easy to access this vast amount of data online quickly. Most of this information is present in the form of unstructured electronic text on the Web including newswire, blogs, communications, documents and so on. However, it is humanly impossible to read and comprehend a significant fraction of the available data and create *knowledge*: useful and actionable information.

To make this information easily understandable, the popular idea is to turn unstructured text into structured semantic content. However, the sheer volume and heterogeneity of data require us to have computers annotate all the data with the structure of our interest. To achieve this, we must use ideas and concepts from machine learning, information extraction, statistics, computer science, natural language processing and computer systems as our aid. The computer needs to know how to recognize a piece of text having a semantic property of interest in order to make a correct annotation. Thus, extracting semantic relations between the entities in a human language text is a crucial step towards natural language understanding applications.

The field of genomics is no exception to this phenomenon of information inflation. With the development of technology for DNA isolation and sequencing the amount of information in this domain is equivalent to the astronomical information available

[1] **EB**: 1 exabyte = 1000^6 bytes = 10^{18} bytes = 1 million terabytes (TB) = 1 billion gigabytes (GB)

with us today [41]. Large knowledge bases like *PubMed*[2] and *PubMed Central*[3] serve as bibliographic database for life sciences and biomedical information and usually derive data by curating from scientific literature. The automated curation of such databases is based on various machine learning and natural language processing techniques like text mining and information extraction.

1.1 Background

Relation extraction is the task of automatic extraction of structured information from unstructured or semi-structured machine-readable documents. This work involves the detection and classification of semantic relationship mentions within a set of artifacts from human language texts with the help of natural language processing (NLP). The relation extraction task can be divided into two steps: detecting if some entity pair mentions in the same sentence exhibit an association among them and classifying the detected relation instances into predefined classes. This task is involved with understanding the underlying syntactic and semantic structure of the language and also the relationships in question.

In our task, **relation** are semantic concepts that are true for a given set of entities. An **entity** may be a particular person, place, object or abstract idea which are unique. A relation r is a named tuple of the form (R, e_1, e_2) where each e_i is a distinct entity. Entity and relation mentions exist in sentences and documents. In this work, a **sentence** is a sequence of one or more tokens that expresses an idea and ends with a period symbol. A **token** is a sequence of characters separated by a whitespace, hyphen or a period. And a **document** is a sequence of one or more sentence that all relate to a coherent topic.

To identify the relation between a pair of entities automatically, it is necessary to skillfully combine lexical and sentence level clues from various syntactic and semantic structures in a sentence. This task is commonly characterized by a large body of linguistic analysis pipeline and knowledge resources to transform relation mentions into a rich representation which can be used by some statistical classifier. The linguistic analysis pipeline is usually *hand-designed* and comprises of various existing natural language

[2]**PubMed**: `https://www.ncbi.nlm.nih.gov/pubmed`
[3]**PubMed Central**: `https://www.ncbi.nlm.nih.gov/pmc/`

processing modules such as tokenization, part of speech tagging, parsing and so on. The knowledge resources, used for training and validating the classifier, are made up of annotated sentences with positive and negative relation examples. An example of relation extraction task is depicted in the Figure 1.1

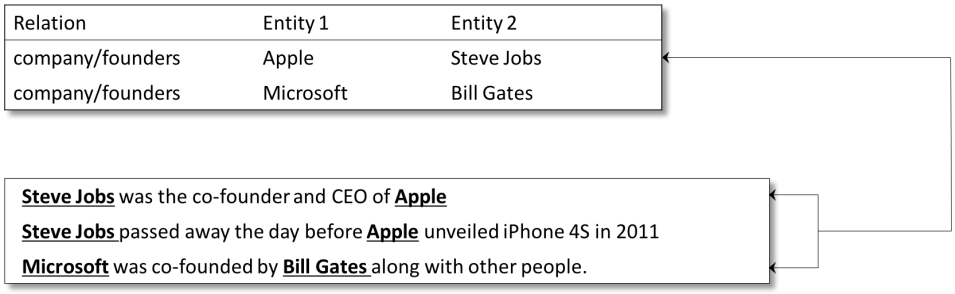

Figure 1.1: A depiction of relation extraction task showing positive and negative examples along with the annotated entities in the source text

1.2 The GWAS catalog

A genome-wide association study (GWAS) is an approach to detect genetic variations associated with particular disease or traits by scanning markers across the genomes of a large-scale sample of subjects in a high throughput manner. In less than a decade, GWAS studies have successfully produced discovery of new genetic associations which have led to the development of new strategies to diagnose, treat and prevent diseases. This increase in GWAS research calls for a database that allows researchers to query and search for previous results quickly. Such a database has been created and maintained online by the National Human Genome Research Institute (NHGRI)[4] called A Catalog of Published Genome-Wide Association Studies (Catalog of GWAS) [17]. This database provides a resource to overview scientific investigations, summarization of associated genetic sites and may help suggest genes that are responsible for phenotypic traits in humans.

The Catalog of GWAS was first released on November 25, 2008, with more than 5,000 entries available for search [46]. Since then, a large number of new GWAS articles

[4]**NHGRI**: `https://www.genome.gov/`

have been published, and the catalog is regularly updated by systematically selecting research articles reporting large-scale GWAS. NHGRI continues to update and curate the catalog regularly by a team of expert curators who manually select study-level fields of information from published GWAS and add them to the catalog. As of September 21, 2016, the Catalog of GWAS contains approximately 29,000 entries extracted from nearly 2,200 discrete articles covering more than 1500 diseases and traits.

Figure 1.2 shows an example entry in the Catalog of GWAS. Each entry represents an observed association reported in an article, specifying that an association between the genetic variant and a phenotype was observed from this study from an initial stage sample. This information is marked in database as *Strongest SNP*, *disease/ traits* and *Initial Sample Size* fields respectively. The entry also specifies that the observations were validated with a replication sample, given in the *Replication Sample Size* field and the whole study was conducted on a particular group of people, marked in the *Ethnicity* field. These data fields describe the characteristics of the population samples used in the GWAS studies and are the focus of this thesis. Other data fields include the information about where the genetic variant resides in the genome and statistical strength of the observation and hold importance from the biomedical research perspective.

Field	Value	Field	Value
PubMed ID	21764829	First Author	Png E
Date	7/15/2011	Journal	Hum Mol Genet
Study	A genome-wide association of Hepatitis-B vaccine response in an Indonesian population reveals multiple independent risk variants in the HLA region	Disease/Trait	Repsonse to HBV vaccine
Initial Sample Size	1683 Indonesian individuals	Reported Gene(s)	HLA-DR
Replication Sample Size	1931 Indonesian individuals	Mapped Gene(s)	BTNL2 – HLA-DR
Region	Chr 6:32389648	Strongest SNP	rs3135363
Context	Intergenic	Risk Allele	NR
p-Value	6.53E-22	Risk Allele Frequency	NR
OR or beta-coefficient	1.53	95% CI (text)	[1.35 – 1.74]
Platform (SNPs passing QC)	Illumina (455.508)	CNV	N

Figure 1.2: Example of an entry in the Catalog of GWAS showing different fields and their corresponding values from a sample text.

1.3 The Problem

Our goal is to automate the extraction of the study-level data fields from GWAS articles using a Machine Learning approach to Natural Language Processing. In this thesis, we primarily focus on the characteristics of the sample populations employed in the experiments, i.e. the experimental stage ("*initial*" or "*replication*"), the ethnicity groups of the individual involved and the size of the sample population pool. We aim to extract these information fields in the form of tuples <*stage, ethnicity*> and <*stage, sample size*> and therefore reach the final entries for the database. Collaborative work [22] has been performed for extracting other data fields towards a larger goal of obtaining all the information recorded in the Catalog of GWAS.

This task of automatic extraction of relational information is important as the curation of this knowledge base is primarily done by a team of experts with in-depth domain knowledge which is highly expensive and time-consuming. Various epidemiologists from NHGRI and more recently European Bioinformatics Institute (EBI)[5] manually curate the database on a weekly basis using the information from the published scientific literature. This operation primarily involves the professionals reading articles, extracting key findings and cross-referencing the data all of which can be prone to human errors. Also, as the volume of biological literature grows rapidly, it becomes increasingly difficult for these experts to keep pace with the published data thus creating a pressing need to assist curation with automated tools.

An example of matched results of the entry in the GWAS catalog with the actual passages in the text of the article is depicted in the Figure 1.3. These results can be expressed in the form of previously mentioned tuples which usually exhibit a semantic and/or lexical relationship between its elements. The process of extraction of such relations between the entity pairs from the text plays a vital role in information extraction and subsequent knowledge base population. These relation extraction tasks are based on text mining and information extraction which are derived from various machine learning and natural language processing strategies.

[5] **EBI**: `http://www.ebi.ac.uk`

Entity Type	Catalog of GWAS	Source Text
Disease / Trait	Response to HBV Vaccine	"hepatitis B vaccine response"
Initial Sample Size	1683	1. "We performed a two-stage genome-wide association study (GWAS) of antibody titer in 3614 hepatitis B vaccine recipients from Indonesian's Riau archipelago" 2. "In the first stage, following extensive quality control (QC) filtering, we analyzed 1683 vaccine recipients ..." 3. "After extensive QC filtering, we analyzed genotypes of 1706 SNPs in a second set of 1931 vaccine recipients from the same study. In this second stage, we replicated ..."
Replication Sample Size	1931	
Ethnicity	Indonesian	

Figure 1.3: Example of the curated data in the Catalog of GWAS entry matched to different passages in the text from source article

1.4 Our Approach

This thesis presents and implements a general approach to extraction of entity pairs which exhibit a given relationship from biomedical databases. The key idea is to use machine learning and natural language processing to understand the underlying structure of these relations using existing literature and extract the entity pairs from new content which adhere to this composition. Most existing methods leverage various tools of linguistic analysis like parsing, chunking, etc. to transform the relation mentions into some rich representation which can be used by some statistical classifier such as Support Vector Machines (SVM) [20] or Maximum Entropy (MaxEnt) [49]. Features used by such natural language processing tools are often hand-designed and are subjected to error propagation introduced by their imperfect quality.

This thesis targets an independent relation extraction system that both avoids complex feature engineering and minimizes the reliance on NLP modules, potentially alleviating error propagation and advancing our performance. We employ convolution neural networks (CNNs) that can extract n-grams from the raw sentences and induce an abstract representation which is capable of capturing underlying semantic and syntactic properties of words. Our method uses the tokenized words as input without any complicated pre-processing which are subsequently transformed into vectors by looking up word embeddings. These vectors are used to extract lexical and semantic features using a convolutional approach in the form of various feature maps. These maps are then used to create a final feature vector which is fed to a logistic regression function for final out-

put. This approach to natural language processing using convolution neural networks has been intensively studied but, to the best of our knowledge, never been applied to the problem of learning from curated data and knowledge base population.

We implement the approach as mentioned above and apply it to two problems of relation extraction from the biomedical literature. The first task is to extract pairs of stage (initial or replication) and ethnicity background of the study samples from the GWAS articles and the second task is to obtain pairs of the ethnicity and sample size from the same GWAS studies. Both these tasks use the already curated data from the Catalog of GWAS as the training examples. The existing curated data from the GWAS catalog is matched to text to create the training examples for our neural networks. However, as this matching is not trivial and involves various natural language processing tools, we will discuss it briefly in the subsequent chapters. Collaborative work, out of scope for this thesis, is also done on the task of annotating scientific literature using various crowd-sourcing techniques.

This approach of relation extraction is not limited to use for the GWAS Catalog. A huge number of different biomedical databases are available today, and many are similar to GWAS such that they contain data derived either through manual curation by teams of experts or structured information submitted by authors or researchers directly from published literature. Apart from the biomedical domain, this approach is equally applicable to other areas which can benefit from the extraction of structured information from freely available text. We intend our approach to be generalizable across these databases and to other relation extraction problems as well.

1.5 Related Work

Relation extraction is one of the most widely studied topics in Natural Language Processing. The task of relation extraction is to understand the relationship between a pair of entities from the text and predict new pairs exhibiting same association from other textual sources. There is considerable interest in automatic relation classification, both as an end in itself and as an intermediate step in a variety of natural language processing applications. Research in the field of relation extraction dates back as far as the

late 1970s. In 1975, Riesbeck and Schank [39] published work on ELI, an English Language Interpreter, which was able to produce structured representations of the semantic information in stories. In the mid-1980s, Andersen et. al. [1] launched a commercial data extraction product known as JASPER (Journalist's Assistant for Preparing Earnings Reports) which provided real-time financial news to traders. From 1987 until 1988, a series of competition-based conferences known as Message Understanding Conferences [11] led to the refinement of ideas, methods and evaluation strategies related to information extraction and natural language processing. In an effort to improve the performance of automated relation extraction systems, there was a fundamental shift in using various machine learning approaches for such tasks. With more development in the fields of statistics, machine learning, and natural language processing, researchers started applying different machine learning paradigms to the task of natural language processing.

1.5.1 Supervised Learning

Various supervised systems for relation extraction have been studied extensively since rich annotated linguistic resources have been released. In these supervised approaches, sentences (or part of sentences) in a corpus are either hand-labeled or heuristically annotated for the presence of entities which exhibit a relationship between themselves. A supervised relation extraction system has three steps: 1) data representation for labeled examples e.g. feature extraction for feature-based methods or extracting objects for kernel-based methods 2) train a classification model as the relation detector/classifier 3) apply the model as the relation extractor on the unseen relation mentions.

One of the most studied relation extraction tasks is the Automatic Content Extraction (ACE)[6] relation extraction evaluation sponsored by the U.S. government. In general objective, the ACE program is motivated by and addresses the same issues as the MUC program that preceded it and was convened by the NIST[7] from 1999 to 2008. ACE 2005 defined seven major entity types such as PER (Person), LOC (Location), ORG (Organization) and also identifies seven major relation types and more than 20 subtypes. ACE provides a large corpus which is manually annotated with entities, re-

[6] **ACE**: `http://www.itl.nist.gov/iad/mig/tests/ace`
[7] **NIST**: `http://www.nist.gov`

lations, events, and values. In this task, each relation mention expressing one of the predefined types is tagged with a pair of entity mentions appearing in the same sentence as its arguments. More details on ACE evaluation can be found on the official ACE website.

For data representation, state-of-the-art methods are either feature-based or kernel-based. Given a relation mention, a feature-based method extracts a rich list of structural, lexical, syntactic and semantic features and convert these extraction clues into feature vectors [13] [33]. In contrast, kernel-based methods represent each instance with an object such as augmented token sequences or a parse tree and use a carefully designed kernel to calculate their similarity [37] [7]. These approaches for data representation are practical because they leverage a large body of linguistic knowledge resources.

Different machine learning methods have been applied for relation extraction tasks. The two most popular ones are maximum entropy classifiers (MaxEnt) [49] and support vector machines (SVM) [20]. Other methods such as K-nearest neighbors algorithm [12] and Voted Perceptron learning algorithm [51] have also been utilized for this task of relation extraction. One major issue associated with these methods is the production of labeled training data which is expensive and in limited quantity. Also, as these relationships are labeled on a particular corpus, the resulting classifiers tend to be biased toward the corresponding text domain.

1.5.2 Distant Supervision

Distant supervision was first proposed by Craven and Kumlein [6] where they generate weakly labeled examples by aligning facts in the Yeast Protein Database into the articles that may establish the facts for training an extractor. This paradigm attempts to create training data automatically by leveraging large knowledge bases of facts and corpus. Since then, it has gained popularity in different domains. Bunescu and Mooney [3] treat each automatically-labeled relation mention as a labeled example and trains an extractor with supervised learning that tolerates incorrect labels of positive examples.

To provide a more accurate treatment of label noise while capturing the pair-level constraints, Riedel et. al. [38] proposes to use the Multiple-Instance Learning, which assumes that only at-least-one of the mentions for each tuple entity listed as hav-

ing a relation in the knowledge base, indeed has the target relation. Multi-Instance Multi-Label (MIML) learning [42] further improve it to allow a tuple of entities to have multiple relations. Further, Takamatsu et. al. [43] viewed the problem differently and proposed a method that estimated the probability of each pattern showing each relation, based on automatically labeled dataset. Their algorithm, however, either removes low-probability false positive matches or corrects high-probability false negative matches by filtering such mentions with their corresponding patterns.

Labeling noise can be reduced by using a more restricted labeling heuristic. For example, Wang et. al. [45] assume that only the first sentence in Wikipedia that contains a pair of related entities is a valid relation mention of the corresponding type. KYLIN [47] proposes three heuristics for labeling Wikipedia text with the infobox. Such heuristics are domain-specific and are not applicable in a more distant yet typical labeling scenario: align Freebase [8] to newswires.

Wang et. al. [45] uses distant supervision to improve supervised relation extraction. Their method starts with constructing relationship topics from the set of heuristically labeled examples (by distant supervision) using Diffusion Wavelets. They propose SVM kernel that encodes the background knowledge (a set of related topics) as a source for measuring the similarity between relation mentions. The resulting extraction algorithm improves on existing solutions in the Automatic Content Extraction (ACE) relation evaluation dataset.

Notwithstanding this progress in distant supervision, current performance is still quite modest and not satisfactory for practical use. For example, the system very recently described in Multi-Instance Multiple Label [42] learning achieves only a recall of 26.9 and a precision of 29.7 on a standard test set. The resulting instances from using distant supervision often suffer from low precision and semantic drifts and are suitable only for some domains.

1.5.3 Unsupervised Learning

Unsupervised relation extraction algorithms collect pairs of co-occurring entities as relation instances, extract features for these instances and then apply different cluster-

[8] Freebase: `https://developers.google.com/freebase/`

ing techniques to find the major relations in a corpus. These algorithms obtain a string of words between the entities in large amounts of text and clusters and simplify these word strings to produce relation-strings and rely upon tagging in advance, a predefined set of entity types, such as Person, Organization, and Location. The distributional hypothesis theory [14] reflects that the pair of entities that occur in related contexts tend to have similar relations which form the basis for various unsupervised relation extraction algorithms.

Among different such algorithms, Yoa et. al. [48] propose several generative models, broadly similar to Latent Dirichlet Allocation (LDA) for the task of relation extraction. One of their models learns fine-grained semantic classes as relation arguments, but they share the similar requirement of tagging coarse-grained argument types. Most of the unsupervised algorithms for relation extraction uses a quadratic clustering algorithm such as Hierarchical Clustering [15] or K-Means [4]. Hasegawa et. al. [15] adopted a hierarchical clustering method to cluster the context of entities and simply select the most frequent words int the contexts to represent the relation between those objects. Similarly, Chen et. al. [4] proposed a novel unsupervised method based on model order selection and discriminative label identification to address this problem

With the growth of digital information, the target domain is shifting towards the Web and new methods are being proposed without requiring predefined entity types. Kok and Domingos [27] propose Semantic Network Extractor (SNE) to extract concepts and relations. Based on the second-order Markov logic, SNE uses a bottom-up agglomerative clustering algorithm to cluster relation phrases and argument entities jointly. However, this method requires each entity and relation expression to belong to exactly one cluster and are unable to handle polysemous relations phrases.

Unsupervised approaches use enormous amounts of data and extract a very large number of relations, and the resulting associations may not be easy to map to the relations needed for a particular knowledge base.

1.6 Layout of Thesis

The remainder of this thesis is organized as follows: Chapter 2 describes the data preparation and preprocessing steps required to analyze the input text and use it as an input to the neural networks. Chapter 3 presents the general framework of the convolutional neural network that we are using for our task of relation extraction. Chapters 4 report the implementations of the approach and the results for the two information extraction problems described above along with various extensions to the neural network framework that we employed. Finally, Chapter 5 summarizes the results, describes the overall conclusions and explores future work.

The material in part from the Introduction is currently being prepared for submission for publication. The thesis author was the primary investigator and author of this material.

Chapter 2

Data Preprocessing

To successfully extract the characteristics of the sample populations in an article, which includes the experimental stage, the ethnicity groups of individuals involved and the size of sample population pool, it is necessary to obtain the text data of the article and prepare it for further processing and information extraction task. This information extraction.task involves steps like converting PDF documents to XML files, extracting textural information from XML file, recognition of suitable candidates for entities and transforming words into vectors using pre-trained embeddings. This chapter describes the preprocessing steps required to make sure that the source documents are ready for relation extraction using neural networks.

2.1 PDF to XML Conversion

Majority of the published scientific literature available today is in form of PDF documents with little or no free text associated with them. The PDF standard does not attempt to encode any semantic connection between the characters in a word or between paragraphs which makes the ease of extracting text an orthogonal issue to the file format. As such, the researchers wanting to do text mining on a large number of journals need to transcribe the PDF documents to text based formats like plain text, XMLs or HTMLs. To overcome this, we used an in-house document conversion pipeline [8] to extract textual information from PDF files and convert them to XML documents using different tools and platforms.

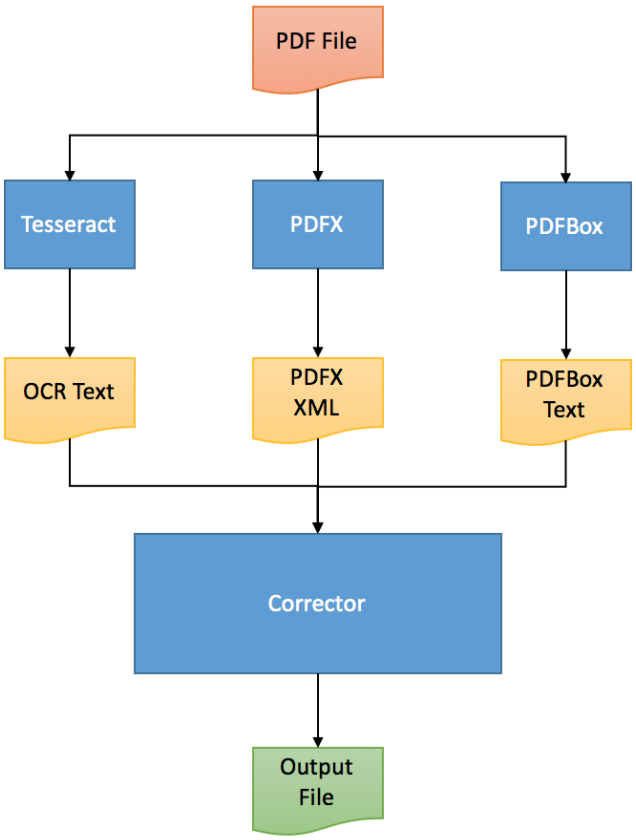

Figure 2.1: Flowchart diagram for PDF to XML conversion pipeline showing different text extraction modules followed by a corrector stage

An overview of the pipeline is depicted in Figure 2.1. In this workflow, we provide each PDF file to three different modules which run independently in parallel and extract text from PDF files using various extraction tools. PDFX is a web-service designed to reconstruct the logical structure of scholarly articles in PDF form and output them as an XML document. The final XML document depicts the input article's logical structure in terms of title, sections, tables, references, etc. and also links it to geometrical typesetting markers in the original PDF document, such as paragraph and column breaks. PDFBox is an open-source Java library which we are using to extract textual information from a PDF file and then using it for further processing. As a final step of text extraction, we use Tesseract which is a raw OCR tool with high character recognition accuracy and extracts text in a simple textual format without any formatting.

Using the outputs from the above three modules, we run the corrector stage, which manipulates and combines them to produce the final output. The corrector stage is a combination of multiple correctors where each corrector acts on the output from the previous one in a serialized manner. Various correctors that are run as a part of this stage perform the *Abstract Tile correction*, *Special Character correction* and *Reference Acknowledgement corrections* to produce an output in XML format. These correctors improve the overall quality of the final output file and remove the unwanted sections like metadata, etc. from the perspective of information extraction. The final XML file contains section-wise text according to source PDF file which is perfect for further text mining activities.

2.2 XML to Text Transcription

The full text of the articles to be used for relation extraction is obtained by traversing these XML files created through above PDF transcription engine. Some XML files were also collected as publicly-available NXMLs from PubMed Central, a free full-text archive of biomedical and life science literature. When traversing these XML files, text from several tags was ignored entirely as the content from those labels was known to be irrelevant to the task of extraction of population characteristics. These tags contained either metadata information related to the publication and not the actual content or were formatting tags for specifying the design of the article. A total of 22 such tags were identified and used for removal during extraction of text from the article. These fell into various categories like article metadata which included *article-id*, *journal-meta*, *copyright-statement*, etc. or formatting tags like *fpage*, *lpage* or other irrelevant tags.

The text obtained from the XMLs was further parsed to remove the elements that are less relevant to the task and clean the remaining text to make extraction simpler. Regular expression-based preprocessing is primarily geared at removing irrelevant or extraneous numbers that might lead to false positives for sample sizes, as well as normalizing the representations of numbers that may give rise to the correct values being missed. A total of 23 such patterns were parsed and removed or rewritten using regular expressions. Some of the examples include:

- *Removal of commas*: Commas that mark the thousandth, etc. digits of a number are removed. For example, "$12,696$" becomes "12696"
- *Mathematical or scientific notation*: Numbers that can be inferred to be irrelevant to sample size extraction based on the surrounding context are removed, such as "$p = 8.14 \times 10^{-05}$", "$3 \log 10$".
- *In-line references*: References within the text to other publications or elements within the same article are removed, such as "[10, 21]" and "Fig. 12".
- *Units and elements*: Descriptions of units or known elements are removed from the text, such as "1,020 SNPs", "23 mg/L"

2.3 Entity Detection

Once the source article is transcribed to plain text and is ready for the extraction of relevant information, we need to identify passages that may contain information related to the population characteristics. This task of passage identification should be inclusive in the sense that any candidate passage will be extracted and no relevant passage would be omitted. The identification of passages is primarily made by recognizing the entity mentions in the text potentially corresponding either to the ethnicity values or a sample size value (described in detail below) and extracting their surrounding passages. This task is particularly important for our case as we need to match the entries in the curated database with real mentions in the text which are not always exact string matches or even exact synonyms.

After identification of the passages, we need to detect the candidates for ethnicity groups of genome-wide association studies' populations, the size of the population involved and stage of the survey which would serve as the base for the relation extraction tasks. This operation is not required in case of annotated text where the candidate entities are already tagged using different crowd-sourcing methods as discussed in previous sections. However, in our case, where we are using entries from the curated database as training examples for our system, we need to correctly identify these entities by matching them from curated entries to mentions in the text. These selection criteria and entity

preparation are achieved by studying the properties associated with these objects and also using the "extraction guideline" [18] as provided by the curation teams NHGRI and EBI. The process of identification of these candidate entities was completed under collaborative work [44] and is discussed below:

- *Ethnicity*: We constructed the dictionary of ethnicity mappings using a multitude of terms that can refer to the ethnicity of an individual, including the country of origin (e.g. "Germany"), the specific ethnicity group (e.g. "European"), an adjectival for the country (e.g. "German") and similar sets of terms for cities and other regions with this knowledge being based on the CIA World Factbook [1]. The final dictionary comprises 449 terms that map to 14 top-level ethnicity groups which cover a majority of the mentions in text.

- *Sample Size*: A typical GWAS article may contain hundreds of numeric values and considering each instance of a numeric value a potential candidate for sample size leads to a huge increase in the number of false positives which renders the dictionary tagging approach ineffective. We therefore use a Conditional Random Field model to tag instances of numeric values in text that appear to correspond an experimental sample based on multiple syntactic and semantic features of the surrounding textual content. The output of the CRF tagger was a set of tokens in text which was most likely to correspond to the sizes of the experimental samples in the GWAS.

- *Stage*: A rule-based classifier was modified for identification of stage entities to use the presence of stage-specific words to make its prediction (e.g., "discovery" or "first stage" for the initial stage, or "follow-up" or "second stage" for replication). These classifier was used to tag different tokens in the input text as a candidate for the stage mentions in the relationship tuples.

[1] CIA World Factbook: `http://www.cia.gov/library/publications/the-world-factbook`

2.4 Word Representation

After detecting entity pairs for relation mentions in the text, we need to encode the text in a form which is easily understandable by neural networks. This encoding is required because convolutional neural networks were originally developed for image data which is fixed-sized, low-dimensional and dense whereas text documents are variable-sized, high-dimensional and sparse if represented by a sequence of one-hot vectors. Therefore, in most of CNN studies on text, the words in sentences are first converted to low-dimensional word vectors which are often obtained by some other method similar to language modeling from an additional large corpus. We are using *word2vec* [29] tool which is a two-layer neural network model to produce word embeddings according to their linguistic context. The word embeddings created using word2vec can capture different degrees of similarity between the phrase and various semantic, and syntactic patterns can be reproduced using vector arithmetic. For this thesis, we are using word vectors that were induced using the *word2vec* tool from over 5.5 Billion tokens which were derived using a combination of all the abstracts from the PubMed publications and all the full-text documents from the PubMed Central Open Access subset. These literature sources effectively cover the entire available biomedical domain scientific research and are a useful resource for this area of study.

Apart from word embeddings, we also need a way to encode the entity candidates for the relation mentions in the text. Apparently, it is not possible to capture the structural information (like shortest dependency path between the entity pairs) only through word embeddings. For this purpose, we also use position embeddings in the case of relation extraction problems. A positional embedding can be defined as the combination of the relative distances of the current word to each of the candidate entities e_1 and e_2. For each word, we define the relative distance between the word and a candidate entity as the number of words between them and are represented using a vector of dimensionality two which, therefore, creates a positional embedding for each word a vector of size 4. A sample raw sentence with marked entities and its transformation into vector encoding using both word and positional embeddings is depicted in Figure 2.2

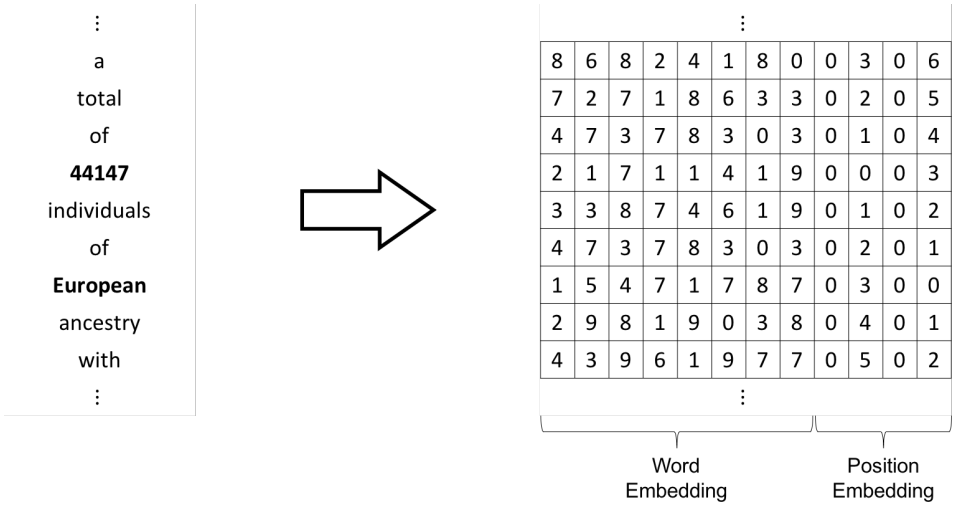

Figure 2.2: Example input text snippet with its transformation into a matrix representation using word and position embeddings

The work in Section 2.1 is a reprint of the material as it appears in "Natural Language Processing using Kepler Workflow System: First Steps" in Procedia Computer Science, Vol. 80. I am thankful to my co-authors Alok Singh, Shitij Bhargava, Daniel Crawl, Ilkay Altintas, and Chun-Nan Hsu for their inputs and support during the research work. The thesis author was the primary investigator and author of this paper.

Material from Chapter 2 in part is currently being prepared for submission for the publication of material. The thesis author was the primary investigator and author of this material.

Chapter 3

Neural Network Framework

Once the preprocessing stage is complete, and the source documents are ready for the main task, we feed them to a neural network for extraction of relational entities for GWAS population characteristics. We depend on Convolutional Neural Networks (CNNs) to detect the underlying semantic and syntactic relationship between these objects and extract the pairs from the new articles accordingly. This chapter describes the framework of our neural networks and their working to produce the final output.

3.1 Convolution Neural Networks

In machine learning, a convolutional neural network (CNN) is a type of feed-forward artificial neural network which was primarily designed to overcome the problems associated with image processing while taking inspiration from neurobiology. Yann LeCunn et. al. tried to capture the organization of neurons in the visual cortex of the cat, which at that time was known to consist of maps of local receptive fields that decreased in granularity as the cortex moved anteriorly [21]. These neural networks utilize multiple layers of convolving filters that are applied to the local regions to recognize the underlying patterns in the input data. As opposed to multi-layer perceptron architecture, convolutional neural networks have certain distinguishing characteristics like local connectivity, replicated filters with shared weights structure and so on which allow these systems to achieve better generalization on vision problems. These features have made CNNs especially favorable for image and speech processing tasks.

The architecture of a convolutional neural network (CNN) is formed by stacking a multiple number of distinct layers which transform the input volume into an output result through a differentiable function. A typical CNN commonly uses a set of separate layers which perform individual operations and contribute to the overall function of transformation. The core building block of any CNN is the *convolution layer* where different filters convolve across the width and height of the input volume producing a feature map for that filter. A filter consists of a layer of connection weights, with the input being the size of a small 2-dimensional image patch and the output being a single unit. Since this filter is repeatedly applied, the resulting connectivity looks like a series of overlapping receptive fields which map to a matrix of the feature outputs. The network thus learns the filters that activate according to some particular feature in the input.

Another important concept of CNNs is *pooling layer* which is a kind of non-linear down-sampling operation and provides a form of translation invariance. The function of the pooling layer is to progressively reduce the spatial size of the representation to reduce the total number of parameters and overall computation in the network which therefore also controls the amount of overfitting. Several non-linear functions exist which can be used to implement the pooling operation among which max pooling and average pooling are the most common. A typical CNN architecture may have multiple layers with alternating convolutional and pooling layers which eventually extracts most important features from the input image. By applying such a subsampling layer in between convolutional layers, we develop the spatial abstractness with the increasing feature abstractness.

After several processing layers, the high-level reasoning in the neural network is done via *fully-connected layer* which has full connections to all the activations in the previous layer and culminates into a decision function in the form of a softmax or logistic operation. While training of the whole network takes quite some time, the convolutional neural network learns much faster than a standard feed-forward neural network and performs quite well in comparison to the previous neural network paradigms. A typical architecture of convolutional neural networks showing these layers is depicted in Figure 3.1

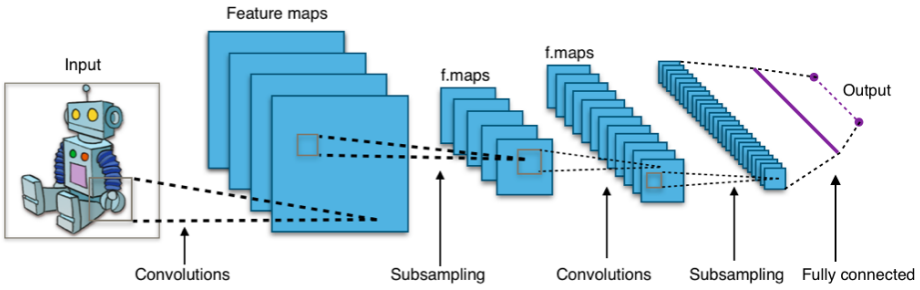

Figure 3.1: A typical CNN architecture used for image processing showing multiple layers of convolutions function, each of which produces various feature maps, followed by subsampling layers with a final fully-connected layer.

3.2 Applications to Natural Language Processing

Recently, convolutional neural networks have begun to overtake traditional sparse, linear models for natural language processing as they automatically learn the features from the sentences and minimize the dependence on external toolkits and resources. These models have been shown to be useful for various natural language processing problems and have achieved excellent results in semantic parsing [10], sentence modeling [25], sentence classification [26] and other traditional NLP tasks. Besides comprising different robust classifiers as part of their architecture, these neural networks are also used to train a neural language model which can generate sentences word by word [40]. Several models of convolutional neural networks have been applied to diverse content ranging from long-form texts (like movie reviews) to short texts (like tweets) and have shown good performance in each of these domains.

The main difference in using these neural networks for natural language processing tasks from image processing functions concerns the input to these nets. We begin by converting a tokenized text snippet, of size s, to a matrix whose each row of is a word vector representation for a token in the text. These word vector representations might be outputs from trained *word2vec* [29] or *GloVe* [35] models with a fixed dimensionality, d, of the vectors. Another approach is to use one-hot vectors that index the word into a vocabulary and these vectors are updated accordingly while training the whole network [24]. The various word embedding approaches can capture multiple different degrees of

similarity between the words which can be used to reconstruct the linguistic contexts of these words. We can effectively treat the text matrix as an 'image' of size $s \times d$ (with a predefined s) and perform convolutions on it through linear filters.

As each row of this matrix represents a discrete token, the breadth of the convolving filters are identical to the dimensionality of the word vectors (i.e. d). Each filter slides over the original input matrix and for every position, we compute element-wise multiplication between the two matrices (the input text matrix and the filter matrix) and add the multiplications output to get the final integer which forms a single element of the 'feature map'. It is important to note that a filter, of size $d \times h$, acts as feature detector from the original input by extracting features in a way similar to (but not limited to) multiple h-grams from the sentence and represent them in a more compact way using feature maps. These features maps are then sub-sampled to continue with a combination of the most important features. The final set of features is fed to a decision function using a fully connected layer to produce the final output as desired. Figure 3.2 denotes a typical architecture of convolutional neural networks for sentence classification tasks [5] which also forms the basis of our framework as discussed in next section.

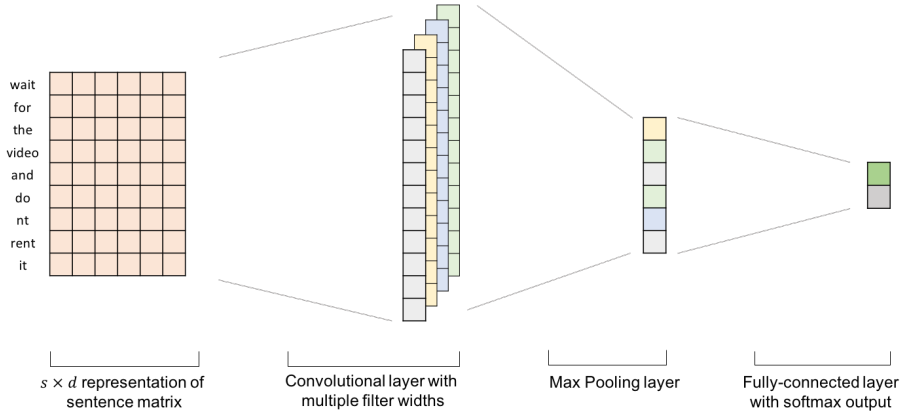

Figure 3.2: Illustration of a Convolutional Neural Network (CNN) architecture for sentence classification with a fixed-size input vector matrix followed by a convolutional, a pooling, and a fully-connected layer.

3.3 Our Framework

As mentioned in the previous section, the architecture for our convolutional neural network for the task of relation extraction is based on an approach to sentence classification by Collobert et. al. [5]. The model proposed by us takes as input the raw sentence snippets with annotated entity mentions and produces a final boolean output indicating if the tagged items do (or do not) exhibit the corresponding relationship between them. The input text comprising of varying-sized words are first transformed into fixed length vectors which are understandable to the network using word and position embeddings. These vectors are subsequently fed into a convolutional layer where multiple filters convolve over the input to extract various related features in the form of feature maps. These feature maps are pooled using a max function to create a feature vector of most important features in the input sentence. The logistic function finally uses this feature vector in the fully-connected layer to produce the final output as desired. The entire architecture of our convolutional neural network is depicted in Figure 3.3 while we discuss each layer in more details below.

The input to the convolutional neural networks should have a form on which convolution function could be performed to extract various underlying feature elements. Our relation extraction system comprises of a couple of raw sentences snippets marked with the two entities to be extracted which are annotated using methods described in previous sections. We begin by transforming each word x_i into a vector e_i of dimensionality 8 using the pre-trained word embeddings table. Along with this, in order to embed the positions of the two entity heads as well as other words in the input text into the representation, the relative distance of each word x_i from the two entity heads $i - i_1$ and $i - i_2$ are also mapped into real-valued vectors d_{i_1} and d_{i_2}, each of which is of size 2. These word embeddings e_i and position embeddings d_1 and d_2 are concatenated together into a single vector $\mathbf{x_i}$, of dimensionality $d = 12$, to represent the word x_i. We thus create an input text matrix $\mathbf{A} \in \mathbb{R}^{s \times d}$ where each row is the vector representation of a word as above and we can effectively treat this matrix as an 'image' to be convolved through the linear filters.

In the next step, the matrix $\mathbf{A}$ representing the input relation mention is fed into the convolutional layer to extract higher level features. A convolution operation involves

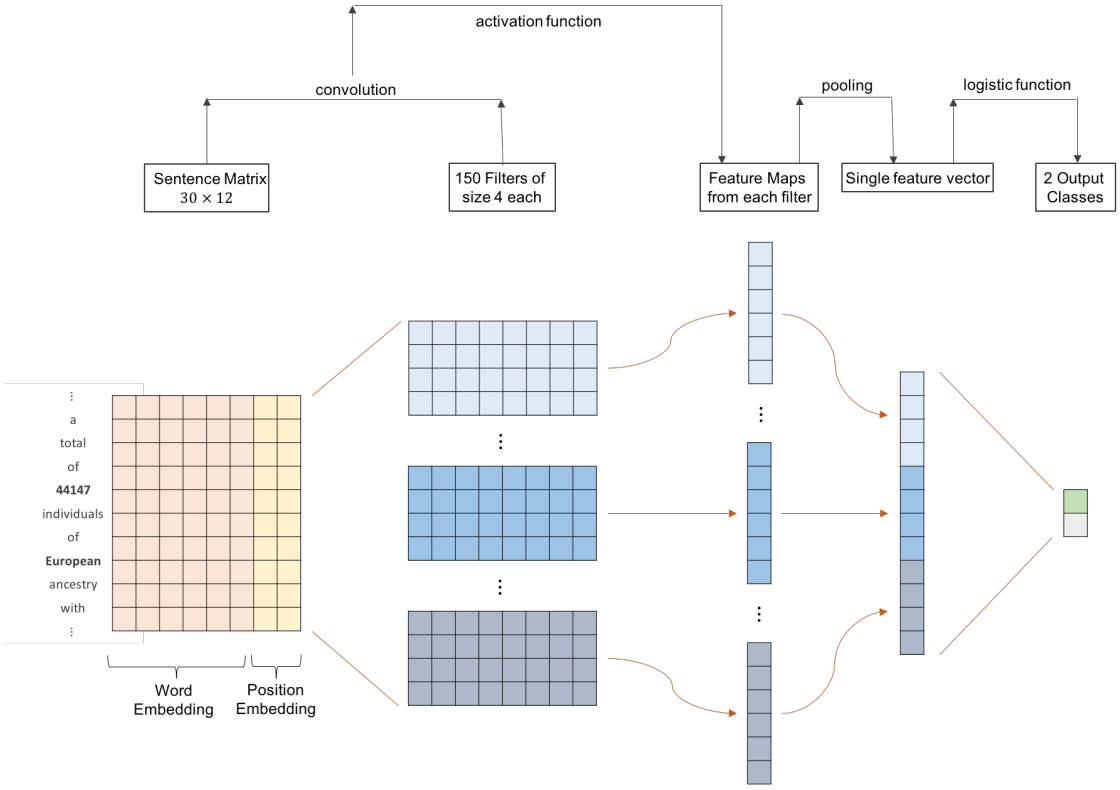

Figure 3.3: Illustration of our architecture for the Convolutional Neural Network with different stages of processing as per configuration mentioned in Table 3.1

a *filter* w $\in \mathbb{R}^{h \times k}$, which is applied to a window of h words to produce a new feature. As the rows of the input 'image' matrix represent discrete words, it is reasonable to use filters with widths equal to the dimensionality of the word vectors (i.e. d). Thus, with each filter of size $h \times d$, we simply vary the height (h) of the filter which represents the number of adjacent rows considered jointly. Such a filter, parametrized by the weight vector $\mathbf{w}$, is repeatedly applied over the sub-matrices of $\mathbf{A}$ to produce output sequence $\mathbf{o} \in \mathbb{R}^{s-h+1}$ for the convolution operator.

$$o_i = \mathbf{w} \,.\, \mathbf{A}[i : i+h-1]$$

where $i \in \{1 \ldots (s-h+1)\}$ and . is the dot product between the sub-matrix and the filter (a sum over element-wise multiplications). One may use multiple filters for the same height, also known as the *region size* of the filter, to learn complementary features from the same regions. We add a bias term $b \in \mathbb{R}$ and an *activation function* f to each o_i, to create a *feature map* $\mathbf{c} \in \mathbb{R}^{s-h+1}$ for this filter:

$$c_i = f(o_i + b)$$

where the activation function f is a non-linear, continous function such as *tanh* or *sigmoid* that transforms a set of input signals into an output signal using a non-linear decision boundary.

The feature maps produced using the above convolution operation are designed to encode the semantic information in the structure of the input sentence snippets. Every filter map may contain one or more local features from the text and would also denote the confidence in the presence of that feature. Each filter of a given region size captures a different structure from the text by virtue of its weight matrix $\mathbf{w}$. Also, the convolution approach provides the transitional invariance which ensures that each feature would be captured by its corresponding filter irrespective of its position (beginning, middle or end) in the input sentence. Along with this, the activation function processes only those features which have a significant presence in the text thus avoiding extra computation

time. These multiple filters create multiple feature maps which are then used as an input to the pooling function which forms the next layer.

The pooling layer reduces the overall number of feature maps produced by the convolution operation of filters with the input sentence in the previous layer. This action is vital as it reduces the huge number of feature maps thus lessen the complexity in further layers. Also, some of the feature maps might be irrelevant or not as important as other features maps and could be safely ignored. In some cases, some features maps could be attained indirectly by a combination of other similar maps and thus would be of less value. We, therefore, sub-sample the features using the 1-max pooling operation over all the feature maps produced from convolution operation. This creates a final univariate feature vector of a smaller dimensionality from a large number of feature maps that are generated by the convolution operation. This process ensures that the most important features are not ignored and are carried forward in further operations. It is, therefore, safe to assume that the pooling layer reduces the dimensionality of each feature map but retains the most important information.

In the final layer, the feature vector generated from the pooling layer using the feature maps from the convolutional layer are fed into a fully connected layer and translated into votes. In our case, we only have to decide between two categories: if the tagged entities exhibit a relation or not. The fully connected layer is a traditional Multi-Layer Perceptron that uses a logistic regression function in the output layer. The term "Fully Connected" implies that every unit in the previous layer is connected to every unit on the next layer. The purpose of this layer is to use the high-level features from the convolutional and pooling layers for classifying the input sentence with annotated entity mentions into its truth class based on the training dataset.

3.4 Framework Extensions

The architecture of the convolutional neural network discussed in the previous section accounts for the basic configuration that we are using for the task of relation extraction, and we first consider its performance on the given dataset. This configuration is primarily based on the work of Kim et. al [26] and is described in Table 3.1. We

further tweak with this setup and try different configurations to improve the performance of our task further and also understand the effect of different architecture settings in neural networks. To this end, we hold all other settings constant (as per Table 3.1) and vary only the component of interest. For each configuration that we consider, we perform the same experiments as the base configuration and report the results for the same. We briefly describe here the different configuration changes to the network and how they might affect the overall performance.

Table 3.1: Primary configuration of our architecture for convolutional neural network for the task of relation extraction

Description	Values
Input Word Vectors	*word2vec*
Filter Region Size	4
Feature Maps	150
Activation Function	*tanh*
Pooling Function	1-max Pooling
Dropout Rate	0.5

Different Filter Region Sizes We begin by exploring the effect of filter region size when using only one filter size and set the number of feature maps for this filter to 150 (as per basic configuration). Different filter region sizes would extract the feature maps from the sentences by providing different windows sizes for convolution operation over the input text. We consider the region sizes of 1, 3, 4, 5, 7, 8 and 10 and study the effect of changing the region size on the performance of our neural network. One primary expectation was that this effect of changing the region size of the filter is dependent on the dataset and each dataset would have its optimal filter region size. For a dataset, with small to medium sentences (similar to many pieces of biomedical literature), we anticipat that the optimal region size should be between 4 and 7. However, for datasets comprising longer sentences the optimal region size may be larger.

Multiple Filter Size Combinations We also explore the effect of combining different filter region sizes, while keeping the number of feature maps for each filter size fixed at 150. The combination of multiple sizes enables us to consider different filter region

sizes and capture features which might get missed while using a single region size. We anticipate that combining several filters with different region sizes close to the optimal single region size might improve performance, but adding region sizes far from the optimal range may hurt performance. We, therefore, combine several different filter region sizes close to this optimal range and compared this to approaches that use region sizes outside this range. We consider a combination of 3 filters with increasing filter sizes and came up with different sets of filters with region sizes [2, 3, 4], [3, 4, 5], [3, 5, 8] and [5, 7, 8].

Activation Function We consider different activation functions in the convolution layer for our neural network including ReLU, hyperbolic tangent (*tanh*) and Sigmoid function. We anticipate that varying these activation function might not have drastic effects on the performance of the neural network and any of these activation functions may have the best performance depending on other parameters in the network. However, the performance of the *tanh* function may be due to its zero centering property as compared to sigmoid or ReLU function. Also, ReLU function, which has merits of non-saturating form, has been observed to accelerate the convergence of filter weights during the training stage. All the results of experiments related to activation functions would also be dependent on the domain and problem under discussion and not universal for convolutional neural networks.

Along with these variations, we also experimented with other parameters like the number of filter maps for each value of region size and the pooling function used for down-sampling the filter maps in the pooling layer. These, however, did not have much effect on the overall performance of the network, and we did not pursue them further. Again, this might be specific to our domain of biomedical literature and would not stand true for other problems and dataset. Also, in the case of multiple filter sizes, same pooling operation was applied on each filter maps to ensure the uniformity of results. Figure 3.4 depicts the overall architecture of our convolutional neural network with filters of multiple region sizes [3, 4, 5].

Material from Chapter 3 in part is currently being prepared for submission for the publication of material. The thesis author was the primary investigator and author of

this material.

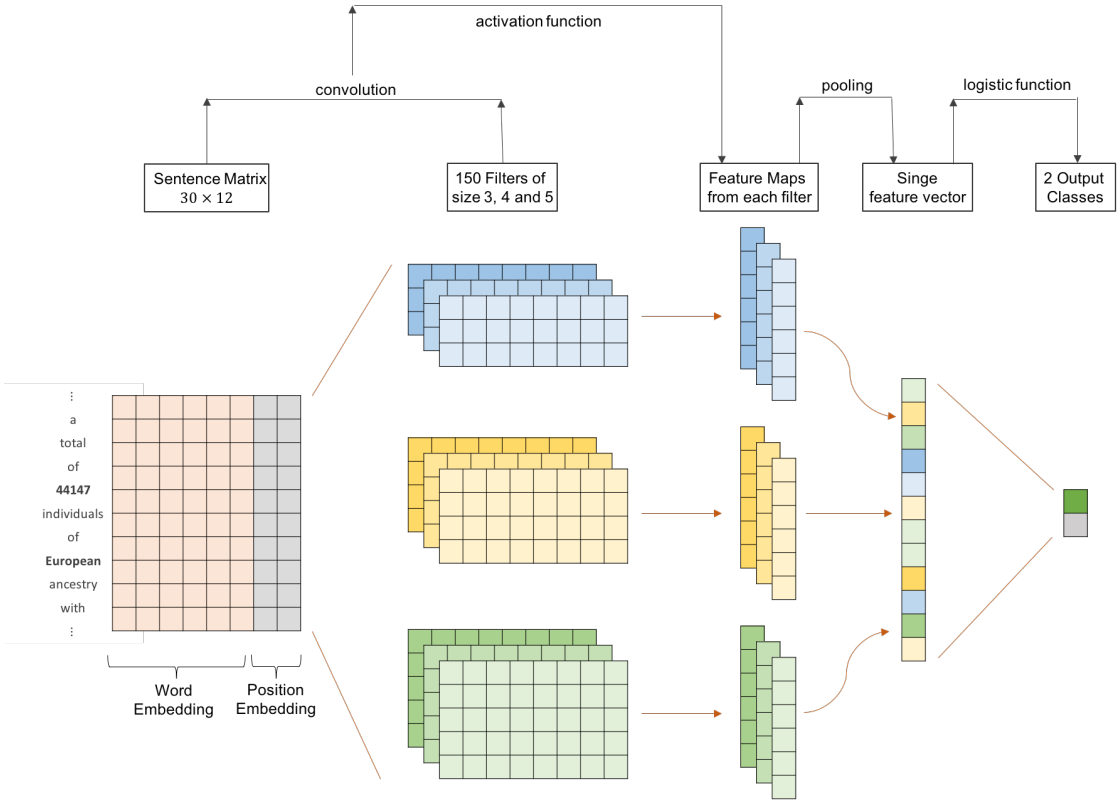

Figure 3.4: Updated architecture of our Convolutional Neural Network with a combination of multiple filter region sizes and illustration of different stages of processing in the overall model.

Chapter 4

Experiments and Results

This chapter extends upon the information extraction task using convolutional neural networks as described in the previous sections. We primarily aim to extract the relational information from preprocessed biomedical texts in the form of tuples of <*stage, ethnicity*> and <*stage, sample size*> and therefore, reach the final entries for the GWAS catalog. The extraction of these semantic data is executed as two separate tasks with Task 1 as identification of pairs of stage and ethnicity groups and Task 2 being the identification of tuples of stage and sample size. These tasks are further described in Figure 4.1 with the help of an example. This chapter focuses on the experiments undertaken for these tasks and the comparison of the results for the same. We begin with a description of our experiment strategy including the information on the dataset used, training and testing process and hyper-parameters settings for the neural network. Following this, we report our experiments for the two tasks using different neural networks (as described in the previous chapter) and their corresponding results.

4.1 Experiments Metrics

For each task described previously, we follow a pipeline structure where the computation begins with an input document and finally produces the relational tuple as required for the operation. The PDF file used as input goes through a series of pre-processing steps like PDF-to-XML conversion, XML-to-Text transcription, entity tagging and vector creation as described in Chapter 2. These tagged entity candidates

Task	Tuple	Output	Source Text
Task 1	<*stage, ethnicity*>	<*first, Indonesian*> <*second, Indonesian*>	1. "We performed a two-stage genome-wide association study (GWAS) of antibody titer in 3614 hepatitis B vaccine recipients from Indonesian's Riau archipelago" 2. "In the first stage, following extensive quality control (QC) filtering, we analyzed 1683 vaccine recipients ..." 3. "After extensive QC filtering, we analyzed genotypes of 1706 SNPs in a second set of 1931 vaccine recipients from the same study. In this second stage, we replicated ..."
Task 2	<*stage, sample size*>	<*first, 1683*> <*second, 1931*>	

Figure 4.1: An illustration of the two relation extraction tasks along with example tuples extracted from the source text.

(in the form of vectors) are used to train the neural network model with truth values extracted from the existing entries in GWAS catalog. Therefore, we need to create the training data using biomedical literature which have been already curated and added to the catalog. The two relation extraction tasks follow a shared data preprocessing pipeline but have separate neural networks as the underlying structure in the text for these tasks could be very different from each other. These neural networks are similar in the overall architecture but are trained and run independently to avoid the overlap of one task with another.

As already discussed, our dataset for the experiments is created using the articles which are already curated and have been entered into the catalog of GWAS. This data is available freely in form of a spreadsheet and can be obtained from the NHGRI page[1]. We selected the articles that satisfied the following criteria:

- *Curated data available*: 2,185 PubMed articles were curated with the data available to start with.
- *NXMLs or PDFs available*: We used NXML version of the articles if they are available through PubMed Central. These versions have high-quality text and we can ignore the PDF-to-XML conversion step while using these files. Otherwise, we transcribed PDF versions of the remaining articles to text. This excludes 324 articles and leaves 1,861 articles.
- *No missing values or "NR"*: The characteristics of the samples are available for whichever stage is mentioned in the article, and the curated data contains no blank

[1] `https://www.genome.gov/26525384/`

entries. Also, excluded are those articles for which curators were unable to find a conclusive ethnicity group for the sample and the entries state "NR" ("not reported"). This step leaves us with 1,674 articles.

- *Ethnicty metions in Text*: Terms that corresponds to ethnic groups must be available in text (and not inferred from affiliations of authors, for example), leaving 1,130 articles.

- *Sample Size mentions in Text*: The sample size is present in the text as a number and not inferred from some other description. Also, it is important for the value of sample size to be present in the textual context.

- *Does not contain Errors*: The articles were excluded for which the curated data was found to contain errors in their entries.

The final dataset for the Task 1 consists of 1,311 articles, comprising of 2,357 <*stage, ethnicity*> tuples. Similarly, for Task 2 only 409 articles comprising of 657 <*stage, sample size*> tuples were used for the training of neural network. Along with these positive examples, there were multiple negative examples that were automatically created by the imperfections while tagging the entity mentions in the input text. Out of these, an intersection of 92 articles and 300 tuples articles comprised of the dataset for evaluation of the trained networks.

Before the training process, these dataset articles along with the tagged entities need to be converted to a form that is understood by the neural networks. As discussed in previous sections, the pre-trained word embeddings[2] [36] were created using the *word2vec* tool on a huge corpus of biomedical literature and have a dimensionality of 8. Also, the positional embeddings of size 2 were used to represent each candidate mentions and its location with respect to other tokens in the text. We entered a set of 30 words along with the marked entities to the network for the processing. The input to the system was, therefore, a matrix of size 12×30 and was treated as an "image" for the convolution process.

The dataset for the two tasks is used to train their corresponding neural networks using the Stochastic Gradient Descent (SGD) method. For each of the tasks, five-fold

[2] `http://bio.nlplab.org/`

article based cross-validation (5-fold CV) is performed during the training phase. The articles in the dataset are shuffled randomly, and each fold utilizes all the tuples belonging to 80% of the articles in the dataset as training data, and the tuples in the remaining 20% of the articles as validation data. The gradients for each neuron were computed using the back-propagation of errors through multiple layers and the filter weights were adjusted in each iteration accordingly. Each training iteration was done with shuffled mini-batches of size 50 and the AdaDelta update rule [50] with a dropout rate of 0.5. For all the experimental runs, we used *tanh* for the non-linear activation and 150 filters for each window size in the model.

To assess the performance of our system, we compare the results on the evaluation dataset with the previous work in the same domain [44]. The key idea of this work was to use a *cost-sensitive* learning algorithm to learn from the training examples weighted by an estimated reliability of each example. This approach involved extracting a huge number of token-based, context-based and other features from the text which were entered into a committee of weak classifiers. These classifiers, varying from simple binary classifiers to rule-based-classifiers, were used to estimate the cost to be assigned to each training instance. These costs were then used to train a L_2-regularized, linear support vector machine (SVM) to classify each example as a positive or negative instance. Overall, it can be surmised that our baseline model was based on extensive feature modeling and engineering using different natural language modules, and an SVM classifier for the mentioned information extraction task.

The result for each task is then collected to obtain the corresponding tuples for all the articles in the evaluation dataset. These results are compared against the curated data and F1 score calculated in the standard way:

- If a tuple in the result for a specific article is present in the curated data for that article, it is considered a true positive (TP); otherwise, it is considered a false positive (FP)
- If a tuple in the curated data for a specific article does not have a counterpart in the extracted results, it is considered a false negative (FN).

Using this we calculate the precision, recall and F-1 score for each task and compare it against the baseline method for the same dataset. These results are discussed

in further details for different neural networks in the upcoming sections.

4.2 Default Parameters

We measure the effectiveness of our architecture for the task of relation extraction by running experiments for each task as described in the previous sections. We start with designing our neural network model with the default parameters as listed in Table 3.1 and evaluate them for our tasks. The main idea is to understand how well our primary system would perform in comparison to the previous existing models. The outcome from these runs guides us to add extensions and change settings in the model to improve the performance further.

We train our models for the two tasks using their respective datasets using the stochastic gradient descent methods as mentioned in the previous section. After this, we evaluate these models using the test dataset to extract pairs of <*stage, ethnicity*> for Task 1 and <*stage, sample size*> for Task 2 as required. These results are summarized in Table 4.1 with the corresponding baselines for each of these tasks. It is evident from the table that we are not performing significantly better in comparison to the existing systems, especially for Task 1 where the F-1 scores are exactly same as before.

However, these results show that our neural network model successfully captures the underlying semantic and syntactic structure of the text and can be used for the relation extraction task to perform equally good as the existing models. Also, as current results are based on a model with default parameters, we believe that extending our framework would lead to better results for our tasks. Having established a baseline performance for the convolutional neural networks for these jobs, we consider the effect of different architecture decisions and hyperparameters settings in next sections.

4.3 Filter Sizes Extension

Although the results with basic parameters are quite good and comparable to the existing system, they can be further enhanced by extending the current neural network model. One way to achieve this is to change the filter region sizes and study its effect

Table 4.1: Performance of our model with basic configuration (Neural Network Model) as listed in Table 3.1 for each of the two tasks and its comparison against the previous work (Baseline Model).

Task Description	Precision	Recall	F-1 Score
Task 1			
Baseline Model	0.74	0.77	0.75
Neural Network Model	0.73	0.79	0.75
Task 2			
Baseline Model	0.56	0.74	0.64
Neural Network Model	0.68	0.70	0.69

on the overall performance of the model. As already discussed in previous sections, this would modify the quality of our model as different filter sizes would result in different windows of text being captured in the convolutional layer, from which subsequent features are extracted. We consider region sizes of 1, 3, 4, 5, 7, 8 and 10 while keeping the other parameters fixed and record the results of our experiments for both tasks. These results are reported in Table 4.2 for the first task of extraction of <*stage, ethnicity*> tuples and Table 4.3 for second task of extractions of pairs of <*stage, sample size*>.

From these results, one can see that each task has its optimal filter region size. The table also suggests that a reasonable range for relation extraction tasks might be from 5 to 7 for our domain of biomedical literature. The filter size extension is not only able to achieve a much higher degree of recall, but this improvement is also accompanied by an increase in the overall precision as well. We can use these optimal filter region sizes to improve further the performance of our model using another extension as described in next section.

4.4 Filter Combinations Extension

The performance of our neural network model is highly improved upon changing the filter region sizes in comparison to default parameters or baseline models. This development motivated us to combine multiple filter sizes into a single architecture to capture the maximum possible combination of features in a single unified neural network model. As such, we explored the effect of combining different filter region sizes,

Table 4.2: Performance of our convolutional neural network with different filter region sizes for Task 1

Filer Size	Precision	Recall	F-1 Score
Baseline	0.74	0.77	0.75
1	0.52	0.66	0.58
3	0.71	0.77	0.73
4	0.73	0.79	0.75
5	0.74	0.80	0.76
7	**0.77**	**0.82**	**0.79**
8	0.73	0.79	0.75
10	0.69	0.73	0.71

Table 4.3: Performance of our convolutional neural network with different filter region sizes for Task 2

Filer Size	Precision	Recall	F-1 Score
Baseline	0.55	0.74	0.63
1	0.42	0.57	0.48
3	0.55	0.70	0.61
4	0.58	0.73	0.65
5	**0.62**	**0.77**	**0.69**
7	0.60	0.79	0.68
8	0.57	0.77	0.66
10	0.54	0.72	0.62

while keeping all the other parameters fixed as before. The number of filter sizes was also increased such that each of the region sizes had 150 filters as before. The result of the experiments conducted with these multiple filter sizes are recorded in Table 4.4 and 4.5 for Task 1 and Task 2 respectively.

As can be directly inferred from the two tables, the optimal combination of filters is different from the two tasks which are similar to the observation from last experiments. Also, for each of the tasks, it can be noted that the combination of various filters with region sizes akin to the optimal single region size improves the performance drastically in comparison to adding region sizes far from the optimal range. The best results we are able to achieve for our neural network model are an F-1 score of 0.85 for extraction of <*stage, ethnicity*> tuples and an F-1 score of 0.74 to extract pairs of <*stage, sample size*>.

Table 4.4: Performance of our convolutional neural network with multiple filter region size combinations for Task 1

Filer Size	Precision	Recall	F-1 Score
Baseline	0.74	0.77	0.75
7	0.77	0.82	0.79
[2, 3, 4]	0.80	0.84	0.82
[3, 4, 5]	0.83	0.85	0.84
[3, 5, 8]	**0.84**	**0.87**	**0.85**
[5, 7, 8]	0.83	0.84	0.83

Table 4.5: Performance of our convolutional neural network with multiple filter region size combinations for Task 2

Filer Size	Precision	Recall	F-1 Score
Baseline	0.55	0.74	0.63
5	0.62	0.77	0.69
[2, 3, 4]	0.66	0.78	0.71
[3, 4, 5]	**0.69**	**0.81**	**0.74**
[3, 5, 8]	0.68	0.79	0.73
[5, 7, 8]	0.65	0.76	0.70

Material from chapter 4 in part is currently being prepared for submission for the publication of material. The thesis author was the primary investigator and author of this material.

Chapter 5

Conclusions and Future Work

A large number of curated biomedical databases available in the public domain provides an unprecedented opportunity to train natural language processing systems to comprehend biomedical publications. In this thesis, we describe an approach to two such information extraction tasks for the Catalog of Genome-Wide Association Studies (GWAS): extraction of tuples of the form <*stage, ethnicity*> and <*stage, sample size*> where stage refers to the specific experimental stage of the GWAS, ethnicity to the ethnic groups of populations involved, and size to the size of the population pool. Our approach uses convolutional neural networks to decipher the underlying structure in the source text and perform information extraction tasks.

5.1 Overall Conclusions

The results show that our approach is effective and outperforms alternative conventional intensive feature-engineered approaches by reaching a F1 score of 0.85 for extracting relations of the form <*stage, ethnicity*> and 0.74 for relations of the type <*stage, size*>. The generality of our approach also leads us to conclude that they can be used for a variety of applications and specifically to the automated curation of biomedical databases.

5.2 Future Work

Although the current results look promising and provide us with a better way for information extraction in comparison to current methods, there are couple of ways we can build on this work to further our work.

- *Extraction of other entities*: We can further use these neural network model to extract other entities for the overall curation of the GWAS catalog like disease/traits, p-values, etc. This is serve as a single system to extract all the fields for the database and avoid us the hassle of using different models for different scenarios.

- *Extension to long tuple extraction*: Currently we are extracting only pair of entries from the text for the curation of database which leads to multiple networks which needs to be independently trained and maintained. A better way could be to design a model which could directly extract the larger tuples like triplets of <*stage, ethnicity, sample size*>

- *Annotation Quality*: One major restriction in current method is tagging of the candidate entries for the relation mentions in the text. If there is an inherent error in the tagger, then the neural network model won't be able to perform as desired. Collaborative work is being undertaken currently to merge these tagging systems with simple crowd-sourcing methods.

Bibliography

[1] Peggy M Andersen, Philip J Hayes, Alison K Huettner, Linda M Schmandt, Irene B Nirenburg, and Steven P Weinstein. Automatic extraction of facts from press releases to generate news stories. In *Proceedings of the third conference on Applied Natural Language Processing*, pages 170–177. Association for Computational Linguistics, 1992.

[2] Shitij Bhargava. Learning from the Catalog of GWAS to Extract Population Characteristics. Master's thesis, University of California, San Diego, 2015.

[3] Razvan C Bunescu and Raymond Mooney. Learning to extract relations from the web using minimal supervision. In *Annual meeting - Association for Computational Linguistics*, volume 45, page 576, 2007.

[4] Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Unsupervised feature selection for relation extraction. In *Proceedings of IJCNLP*, 2005.

[5] 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.

[6] Mark Craven, Johan Kumlien, et al. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, pages 77–86, 1999.

[7] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423. Association for Computational Linguistics, 2004.

[8] Ankit Goyal, Alok Singh, Shitij Bhargava, Daniel Crawl, Ilkay Altintas, and Chun-Nan Hsu. Natural Language Processing Using Kepler Workflow System: First Steps. *Procedia Computer Science*, 80:712 – 721, 2016. International Conference on Computational Science 2016, {ICCS} 2016, 6-8 June 2016, San Diego, California, {USA}.

[9] Malcom W. Greaves. Relation Extraction using Distant Supervision, SVMs and Probabilistic First Order Logic. Master's thesis, Carnegie Mellon University, 2014.

[10] Edward Grefenstette, Phil Blunsom, Nando de Freitas, and Karl Moritz Hermann. A deep architecture for semantic parsing. *arXiv preprint arXiv:1404.7296*, 2014.

[11] Ralph Grishman and Beth Sundheim. Message Understanding Conference-6: A Brief History. In *COLING*, volume 96, pages 466–471, 1996.

[12] Ralph Grishman, David Westbrook, and Adam Meyers. NYU's English ACE 2005 system description. *ACE*, 5, 2005.

[13] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on Association for Computational Linguistics*, pages 427–434. Association for Computational Linguistics, 2005.

[14] Zellig S Harris. Distributional Structure. *Word*, 10(2-3):146–162, 1954.

[15] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 415. Association for Computational Linguistics, 2004.

[16] Martin Hilbert and Priscila López. The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025):60–65, 2011.

[17] Lucia A Hindorff, Heather A Junkins, PN Hall, JP Mehta, and TA Manolio. A Catalog of published Genome-wide Association Studies. 2011.

[18] Lucia A. Hindorff, Jacqueline A. L. MacArthur, Joannella Morales, Emily H. Bowler, Peggy Hall, Kent Klemm, Heather A. Junkins, Tony Burdett, Danielle Welter, Teri A. Manolio, and Helen Parkinson. Comprehensive curation and visualization of ethnicity information from published genome-wide association studies (GWAS): an improved GWAS catalog. 2014.

[19] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics, 2011.

[20] Gumwon Hong. Relation extraction using support vector machine. In *International Conference on Natural Language Processing*, pages 366–377. Springer, 2005.

[21] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.

[22] Suvir Jain, R Kashyap, Tsung-Ting Kuo, Shitij Bhargava, Gordon Lin, and Chun-Nan Hsu. Weakly supervised learning of biomedical information extraction from curated data. *BMC Bioinformatics*, 17(1):1, 2016.

[23] Jing Jiang and ChengXiang Zhai. A Systematic Exploration of the Feature Space for Relation Extraction. In *HLT-NAACL*, pages 113–120, 2007.

[24] Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*, 2014.

[25] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[26] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[27] Stanley Kok and Pedro Domingos. Extracting semantic networks from text via relational clustering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 624–639. Springer, 2008.

[28] Martin Krallinger, Alexander Morgan, Larry Smith, Florian Leitner, Lorraine Tanabe, John Wilbur, Lynette Hirschman, and Alfonso Valencia. Evaluation of text-mining systems for biology: overview of the Second BioCreative community challenge. *Genome biology*, 9(2):1, 2008.

[29] 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.

[30] Bonan Min. *Relation Extraction with Weak Supervision and Distributional Semantics*. PhD thesis, New York University, 2013.

[31] Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. Towards large-scale unsupervised relation extraction from the web. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(3):1–23, 2012.

[32] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.

[33] Thien Huu Nguyen and Ralph Grishman. Employing Word Representations and Regularization for Domain Adaptation of Relation Extraction. In *Association for Computational Linguistics*, pages 68–74, 2014.

[34] Thien Huu Nguyen and Ralph Grishman. Relation Extraction: Perspective from Convolutional Neural Networks. In *Proceedings of NAACL-HLT*, pages 39–48, 2015.

[35] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global Vectors for Word Representation. In *EMNLP*, volume 14, pages 1532–43, 2014.

[36] Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. Distributional semantics resources for biomedical text processing. *Proceedings of Languages in Biology and Medicine*, 2013.

[37] Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 697–704. Association for Computational Linguistics, 2008.

[38] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010.

[39] Christopher K Riesbeck and Roger C Schank. Comprehension by Computer: Expectation-Based Analysis of Sentences in Context. Technical report, DTIC Document, 1976.

[40] Le Hai Son, Alexandre Allauzen, and François Yvon. Continuous space translation models with neural networks. In *Proceedings of the 2012 conference of the north American chapter of the Association for Computational Linguistics: Human language technologies*, pages 39–48. Association for Computational Linguistics, 2012.

[41] Zachary D Stephens, Skylar Y Lee, Faraz Faghri, Roy H Campbell, Chengxiang Zhai, Miles J Efron, Ravishankar Iyer, Michael C Schatz, Saurabh Sinha, and Gene E Robinson. Big Data: Astronomical or Genomical? *PLoS Biology*, 13(7):1–11, 2015.

[42] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics, 2012.

[43] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 721–729. Association for Computational Linguistics, 2012.

[44] Kashyap R. Tumkur. Learning from the Catalog of GWAS to Extract Population Characteristics. Master's thesis, University of California, San Diego, 2015.

[45] Chang Wang, James Fan, Aditya Kalyanpur, and David Gondek. Relation extraction with relation topics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1426–1436. Association for Computational Linguistics, 2011.

[46] Danielle Welter, Jacqueline MacArthur, Joannella Morales, Tony Burdett, Peggy Hall, Heather Junkins, Alan Klemm, Paul Flicek, Teri Manolio, Lucia Hindorff, et al. The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic Acids Research*, 42(D1):D1001–D1006, 2014.

[47] Fei Wu and Daniel S Weld. Autonomously semantifying wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 41–50. ACM, 2007.

[48] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics, 2011.

[49] Lin Yao, Chengjie Sun, Xiaolong Wang, and Xuan Wang. Multi-class Relationship Extraction from Biomedical Literature Using Maximum Entropy. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*, pages 551–554. IEEE, 2010.

[50] Matthew D Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[51] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106, 2003.

[52] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal*, pages 17–21, 2015.

[53] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. Relation Classification via Convolutional Deep Neural Network. In *COLING*, pages 2335–2344, 2014.