A Novel Data Extraction Framework Using Natural Language Processing (DEFNLP) Techniques

Author

Tayyaba Hussain

Registration Number: 319554

Supervised by

Dr. Muhammad Usman Akram

Department of Computer & Software Engineering

College of Electrical & Mechanical Engineering (CEME)

National University of Science & Technology (NUST)

Islamabad

October, 2021

A Novel Data Extraction Framework Using Natural Language Processing (DEFNLP) Techniques

Author

Tayyaba Hussain

Registration Number: 319554

A thesis report submitted in partial fulfillment of the requirements
for the degree of MS in Software Engineering

Thesis Supervisor

Dr. Muhammad Usman Akram

Supervisor's Signature

____

Department of Computer & Software Engineering
College of Electrical & Mechanical Engineering (CEME)
National University of Science & Technology (NUST)
Islamabad

October, 2021

DECLARATION

I declare that this research work titled, "*A Novel Data Extraction Framework Using Natural Language Processing (DEFNLP) Techniques*" is my own work and it has not been submitted for evaluation anywhere else. All the material from other sources used in this report has been appropriately cited.

Student's Signature:

Tayyaba Hussain -319554

MS-19-CSE

____

PLAGIARISM REPORT

This thesis report has been checked for Plagiarism. Attached is the Turnitin report checked by Supervisor.

Student's Signature:

Tayyaba Hussain

319554

Signature of Supervisor:

COPYRIGHT STATEMENT

The student author owns the copyright of the text of this paper. Full or partial copies (in any form) can only be made according to the author's instructions and stored in the library of NUST College of Electrical and Mechanical Engineering (CEME). The librarian may be asked for details. This page must be part of any copy of the thesis made. No further copies may be made (in any way) without author's (written) permission.

All intellectual property rights given above belong to NUST College of Electrical and Mechanical Engineering (CEME) and unless otherwise agreed in advance, they must not be provided to third parties for use, without written permission. CEME will list the terms and conditions of any such agreement.

For further information on the terms of possible disclosure and use, please refer to the library of NUST College of Electrical and Mechanical Engineering (CEME), Rawalpindi.

ACKNOWLEDGEMENT

All praise and glory to Almighty Allah, for guiding me through this research work with each step and every new thought. I am thankful to my parents for their constant support and endless efforts. I am also very grateful to my supervisor Dr. Muhammad Usman Akram for his guidance, support and cooperation in my quest for knowledge. Along with my supervisor, I would like to acknowledge my entire thesis committee for their prudent suggestions. Finally, I would like to thank everyone who gave me valuable help, directly or indirectly with my dissertation.

.

TABLE OF CONTENTS

Abstract **11**

Chapter 1: Introduction **12**

Natural Language Processing (NLP) 12

Benefits of NLP 13

Disadvantages of NLP 14

NLP Tasks for Information Extraction 14

Motivation 16

Problem Statement 17

Aims & Objectives 17

Structure of Thesis 18

Chapter 2: Literature Review **19**

NLP Question Answering (QA) Models...19

Named Entity Recognition (NER) Methods..28

Research Gaps 33

Contribution 35

Chapter 3: Materials...**36**

Coleridge Initiative Dataset 36

Identifying The Datasets..36

Train Data DataType Information..37

Train Data Labels Description..37

Observations 38

Distribution of pub_title & dataset_title...40

Distribution of dataset_title & dataset_label...40

cleaned_label Visualization... 41

JSON Data Description 41

Tools & Languages...42

Chapter 4: Methodology.. **44**

Proposed Framework 44

Features of Data Extraction Framework Using Natural Language Processing Techniques (DEFNLP) 44

Phase I: Data Cleaning & Baseline Model...46

Phase II: QA Modelling & SpaCy..47
Phase III: Acronyms & Their Abbreviations Extraction 51
Chapter 5: Experimentation & Results 53
Experimentation 53
K-Means Visualization of Prediction Strings of All Phases........................54
Elbow Method for Optimal Number of Clusters......................................54
PCA sklearn Clustering Results ……………….....................................55
Extracting Top Features from Each Cluster………….............................56
Individual Performance of BERT QA Models...60
Performance Metrics of BERT Models Under DEFNLP.............................60
Chapter 6: Conclusion & Future Work 62
References 64

LIST OF FIGURES

Figure 1.1: Natural Language Processing (NLP) in Action 12

Figure 1.2: NLP Tasks 15

Figure 1.5: Activity Diagram of DEFNLP Process... 16

Figure 2.1: BERT Transfer and Adapt for Answer Sentence Selection [7] 20

Figure 2.2: BERT Overall Pre-Training and Fine-Tuning Procedures............ 22

Figure 2.3: Instances of NER [47] 28

Figure 3.2: Number of Duplicates Records in Dataset 39

Figure 3.4: Distribution of pub_title & dataset_title in Dataset 40

Figure 3.5: Relationship Between Dataset Title & Label 41

Figure 3.6: Top 100 Most Common Words in cleaned_label 41

Figure 4.1: Features of Data Extraction Framework Using Natural Language Processing Techniques (DEFNLP) 45

Figure 4.2: Excerpt of NER from Sentence... 48

Figure 4.3: Graphical Representation of Tagging Entities............................... 49

Figure 5.1: Datasets Stats Identified in Phase I, II & III 53

Figure 5.2: K-Means Algorithm 54

Figure 5.3: Elbow Method for Optimal Number of Clusters............................ 55

Figure 5.4: PCA sklearn Clustering Results... 56

Figure 5.5: Phase I Clustering Results.. 57

Figure 5.6: Phase II Clustering Results... 57

Figure 5.7: Phase III Clustering Results... 58

Figure 5.8: Datasets Aggregation Clustering Results.................................... 58

Figure 5.9: Aggregation Clustering Results.. 59

Figure 5.10: Individual Performance of BERT QA Models 60

LIST OF TABLES

Table 2.1: Summary of Different State of Art Used for Information Extraction. 31

Table 3.1: Training Data Exploration... 37

Table 3.2: Excerpt from Train Data.. 38

Table 3.3: Number of Duplicates in Datasets.. 38

Table 3.4: Single Publication with Multiple Dataset Title................................. 39

Table 3.5: Comparison of Section Titles of Two Publications............................ 42

Table 3.6: Session Matrices 42

Table 4.1: JSON Text Appended in Dataframe 46

Table 4.2: Datasets Identified Through Baseline Modelling 47

Table 4.3: Datasets Identified by Using Government Datasets 47

Table 4.4: Tokenized Sentences Having Specific Words 48

Table 4.5: Datasets Identified by Using BERT QA Model 50

Table 4.6: Datasets Identified by Matching Acronyms & Abbreviations............ 51

Table 5.1: Ablation Analysis of Different BERT QA Models Under DEFNLP ...61

ABSTRACT

Evidence through data is critical if government is to address many threats facing society, including; pandemics, climate change, Alzheimer's disease, child hunger, increasing food production, maintaining biodiversity, and addressing many other challenges. Yet much of the information about data necessary to inform evidence and science is locked inside publications. A new dataset is recently introduced, Coleridge Initiative - Show US the Data, to discover how the data is used for the public good. In this research, we demonstrate a general Data Extraction Framework Using Natural Language Processing Techniques (DEFNLP) which challenges data scientists to show how publicly funded data are used to serve science and society using Natural Language Processing (NLP) techniques and models. The proposed framework uses NLP libraries and techniques like SpaCy NER and different huggingface Question Answering (QA) models to predict the datasets used in publications after further processing, data and text mining. DEFNLP will enable government agencies and researchers to quickly find the information they need. Till now such issue having large dataset which belongs to numerous research areas has not been addressed. The proposed approach is domain independent and therefore can be applied to all kind of case studies and scenarios where data is extracted. Our methodology sets the state-of-the-art on this Coleridge dataset, reaching the impressive outcome of 0.654, which outperforms current state-of-the-art as compare to other frameworks. In terms of timing and performance, it has short timing and high performance as each epoch took around 5 minutes on average on a CPU with output size of 3.27kB.

Keywords— Big Data, Data Extraction, Data Mining, Named Entity Recognition (NER), Natural Language Processing (NLP), Question Answering (QA) Modelling, Text Mining

CHAPTER 1: INTRODUCTION

Data Extraction deals with retrieving information from data sources for later processing or storage. Data is gathered from a variety of sources or systems. The data can now be refined after it has been successfully extracted. Star Wars fan will be familiar with golden protocol droid robot. Well Star Wars might be set in a galaxy far away. The reality of having machines talk and response to us in a human like manner is already a reality, which keeps getting more and more realistic with every passing day. The people you ask for queries on websites, your smarter systems even call made over the internet, all that have one thing in common. None of them are actually human. Now you must be thinking if they are not human, how they mean it to sound and seem so human like? How they respond to us so intelligently? And how they so articular? This is the magic of NLP.

1.1. NATURAL LANGUAGE PPROCESSING (NLP)

What exactly is **Natural Language Processing (NLP)**? NLP is discipline of Artificial Intelligence (AI). As exhibit in figure 1.1, it allows machines to read, interpret, and deduce meaning from human languages. NLP merges computational linguistics in computer science to decipher language structure guidelines as well as create models capable of comprehending, breaking down, and separating important details from text and speech.

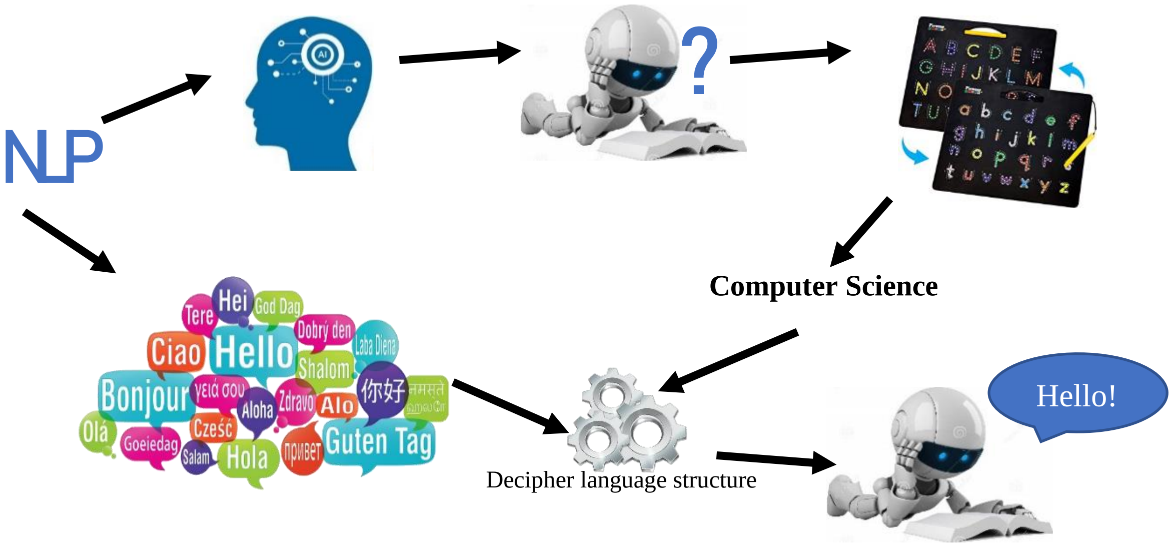


Figure 1.1: Natural Language Processing (NLP) in Action

1.2. BENEFITS OF NLP

NLP has many advantages; few are given below:

1. Users may ask queries about any topic and receive an immediate response within seconds.
2. The NLP system responds to queries in natural language.
3. The NLP system responds to questions precisely, with no unnecessary or unwanted information.
4. The amount of relevant information provided in the query improves the accuracy of the answers.
5. The NLP method allows computers to converse with individuals in their native language and automates a variety of language-related tasks.
6. Allows you to compare more language-based data to a human without worry and in an unbiased and consistent manner.
7. Creating a framework for a very unstructured data source
8. Every day people communicate with one another via public social media, which is possible through NLP.
9. Transferring fast quantity of publicly available data to one another. This information is especially useful in understanding human behavior and consumer tastes.
10. Data analysts and machine learning professionals utilize this information to educate robots how to mimic human language behavior.
11. Helps to save millions of dollars and tones of labor over time. As a result, we do not even need someone from the other end of the line.
12. NLP experts offer highly recruited salaries.

NLP is also far more widespread than we know. We utilize it on a daily basis in seemingly minor circumstances. You don't know how to appropriately spell a word? Autocorrect has been restored. Want to know if your article has a red flag for copyright infringement? That's fine! A plagiarism checker like turnitin will scan the internet for published works that match your words line by line.

NLP appears to be quite interesting. We have a complex technological concept. It is actually rather simple to learn. We begin with a paper or an article to help our algorithm grasp what's going on. We must process the data in a way that the machine can understand. This is similar like teaching a kid to read for the first time.

1.3. DISADVANTAGES OF NLP

Below mentioned are some drawbacks of NLP:

1. Complex Question Language: If the query is poorly worded or confusing, the system may be unable to provide the appropriate answer.
2. Due to its limited characteristics, the system is built for a specific and specialized task; it is incapable of adapting to new domain and problems.
3. The NLP system lacks a user interface and capabilities that allow users to engage with the system further.

1.4. NLP TASKS FOR INFORMATION EXTRACTION

Following as exhibit in figure 1.4 are NLP tasks which are necessary to extract any kind of information:

1. Starting off by performing **Segmentation** which is to break the entire document down into its meaningful units. You can do this by segmenting the article along its punctuations like full stops '.” and commas “,”.
2. For algorithm to understand its sentences, we get the word in segments and to explain its meaning to a algorithm, so we break down our segments into its constitute words and store them, this is called **Tokenization** where each word is called **token**.
3. We can make the learning process faster by getting rid of not essential words which don’t have much meaning to our statement. And just to make our statement sound more cohesive these words such as ‘are’, ‘and’, ‘the’ are called **Stop Words**.
4. Another we have basic form of our document; we need to explain it to our machine. We first stood of by explaining some words like skipping, skips and skipped are the same word with

   added prefixes and suffixes. This is called **Stemming**.

5. We also identify the base words for different vertex like mood, gender etc. this is called **Lemmatization**. Stemming from the base word ‘Lemma’.
6. Now we explain the concepts of noun, verbs, articles or other parts of speech to the machine by adding these tags to our words. This is called part of **Speech Tagging**.
7. Next, **Named Entity Recognition (NER)** which introduce our machine to cultural references and every day names by flagging names of movies, important personalities or locations organizations, medical codes, time expressions, quantities, monetary values, percentages etc. that may appear in the document.

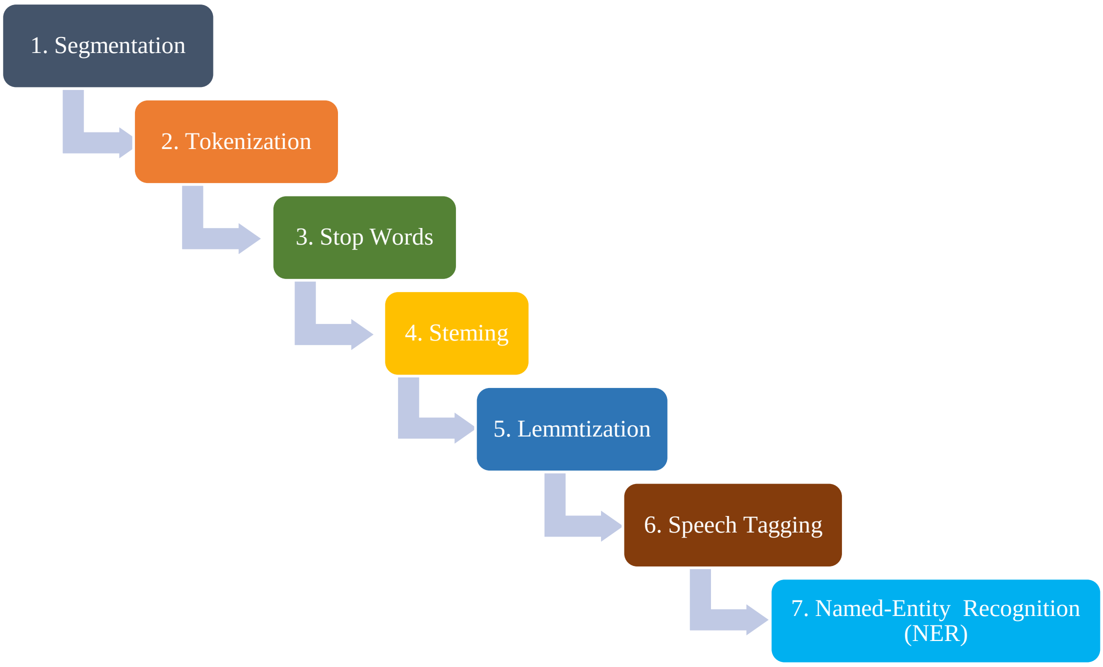


Figure 1.4: NLP Tasks

Once we have a base word in bags, we use ML algorithms like transfer learning, naïve bayes etc to teach our model human sentiment and speech. At the end of the day most of the techniques use NLP are simple Grammarly techniques that we have been taught in school. With the increasing demand for automated language solutions companies are looking for NLP experts to join them and prepare to offer highly recruited salaries as well.

1.5. MOTIVATION

This research urges data scientists to demonstrate how publicly financed data may be utilized to benefit research and society. Evidence derived from data is important if government is to handle the various risks confronting society, such as pandemics, climate change, Alzheimer's disease, child hunger, boosting food supply, preserving biodiversity, and addressing a variety of other issues. However, most of the information about data necessary to inform evidence and research is contained inside publications.

Now is the time for data scientists to help restore trust in data and evidence. In the United States, federal agencies are now mandated to show how their data are being used. The new Foundations of Evidence-based Policy making Act [19] requires agencies to modernize their data management. New Presidential Executive Orders [20] are pushing government agencies to make evidence-based judgements and decisions which must be based on the most up-to-date facts and science available. And the government is working to respond in an open and transparent way [2]. The overarching objective of this research is to identify the mention of datasets within scientific publications. Our predictions are short excerpts from the publications that appear to note as a dataset. We proposed a novel data extraction framework DEFNLP as a solution, shown in figure 1.5.

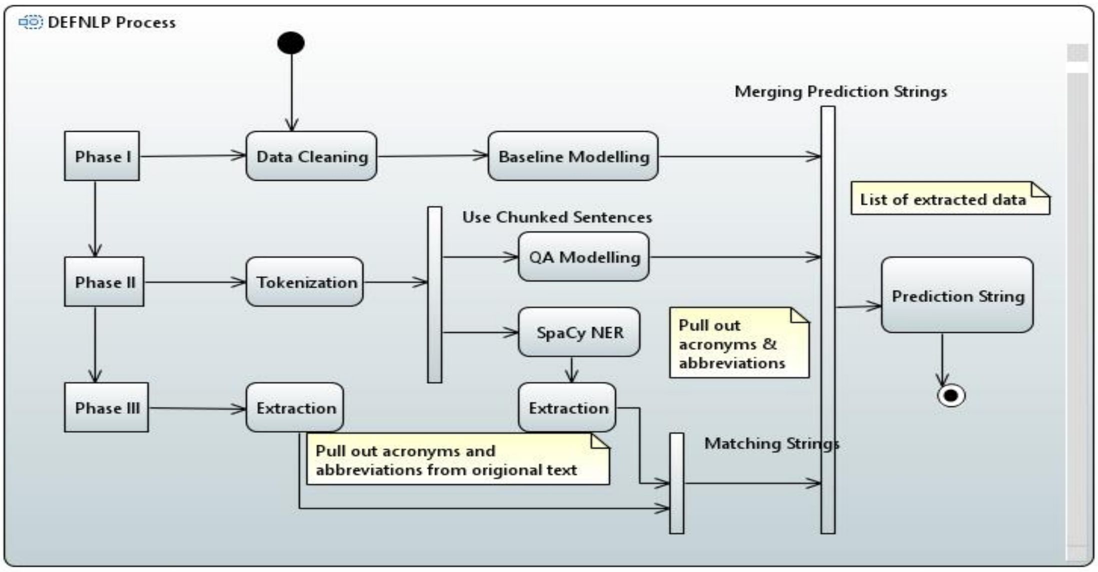


Figure 1.5: Activity Diagram of DEFNLP Process

The activity diagram demonstrates the DEFNLP process that this framework consists of NLP techniques and models. DEFNLP consists of three phases. Phase I consists of data cleaning and baseline modeling. Phase II consists of chunking text from large text, tokenization, SpaCy NLP library and different QA huggingface models. While Phase III consists of different extraction techniques like acronyms and abbreviation obtained in Phase II and matching of string.

1.6. PROBLEM STATEMENT

The research questions related to this research are as following:

a) Can natural language processing find the hidden-in-plain-sight data citations?

b) Can machine learning find the link between the words used in research articles and the data referenced in the article?

The government agencies are being pushing to make evidence-based judgements and decisions which must be based on the most up-to-date information and science. Its critical for government to get evidence through data to make wiser decisions and solutions about the challenges and problems facing by the society. For instance, current pandemic situation of covid-19, climate change, diseases, hunger, rise in food production etc. Hence, we proposed the solution to this problem to discover how the data is used for public good. This research also helps the government to respond in an open and transparent way [2] while making investments.

1.7. AIMS & OBJECTIVE

The overarching objective of this research is:

1. To identify the mention of datasets within scientific publications.
2. Switch from ad-hoc methods to automated ways to find out:
   a) What datasets are being used to solve problems?
   b) What measures are being generated?
   c) Which researchers are the experts?

1.8. STRUCTURE OF THESIS

The report is structured as follows:

Chapter 1 gives the introduction about proposed topic, aims, objectives and motivation.

Chapter 2 provides the review of state-of-the-art in the context of data extraction approaches using NLP techniques.

Chapter 3 gives the materials, including dataset and tools used for implementation.

Chapter 4 discusses the proposed deep learning methodology for reliable classification of leaf diseases. DEFNLP consists of three phases. Phase I consists of data cleaning and baseline modeling. Phase II consists of chunking text from large text using tokenization, SpaCy NLP library and different QA huggingface models. While Phase III consists of different extraction techniques like acronyms and abbreviation obtained in Phase II and matching of string.

Chapter 5 discussed the experimentation including the setup used for implementation, quantitative results obtained, their discussion and benefits in comparison to other approaches.

Chapter 6 concludes the topic by suggesting some future work that is not under the scope of this research but can be implemented in future.

CHAPTER 2: LITERATURE REVIEW

In this section, a thorough but critical attempt is made to review existing state-of-the-art and to find out their shortcomings. All the literature discussed below is focused on various machine learning and NLP techniques and their advantages and drawbacks related to prediction of or search of interested information. The work done in available literature is classified into two categories of NLP tasks based on the methodology used by authors.

1. **NLP Question Answering (QA) Models**

   Different QA models such as SQUAD, BERT, ALBERT, ALBERT-large, ALBERT-xlarge, BIGBIRD, Longformer, TANDA, BIDAF, TAPAS, DistilBERT, RoBERTa etc.

2. **Named Entity Recognition (NER) Methods**

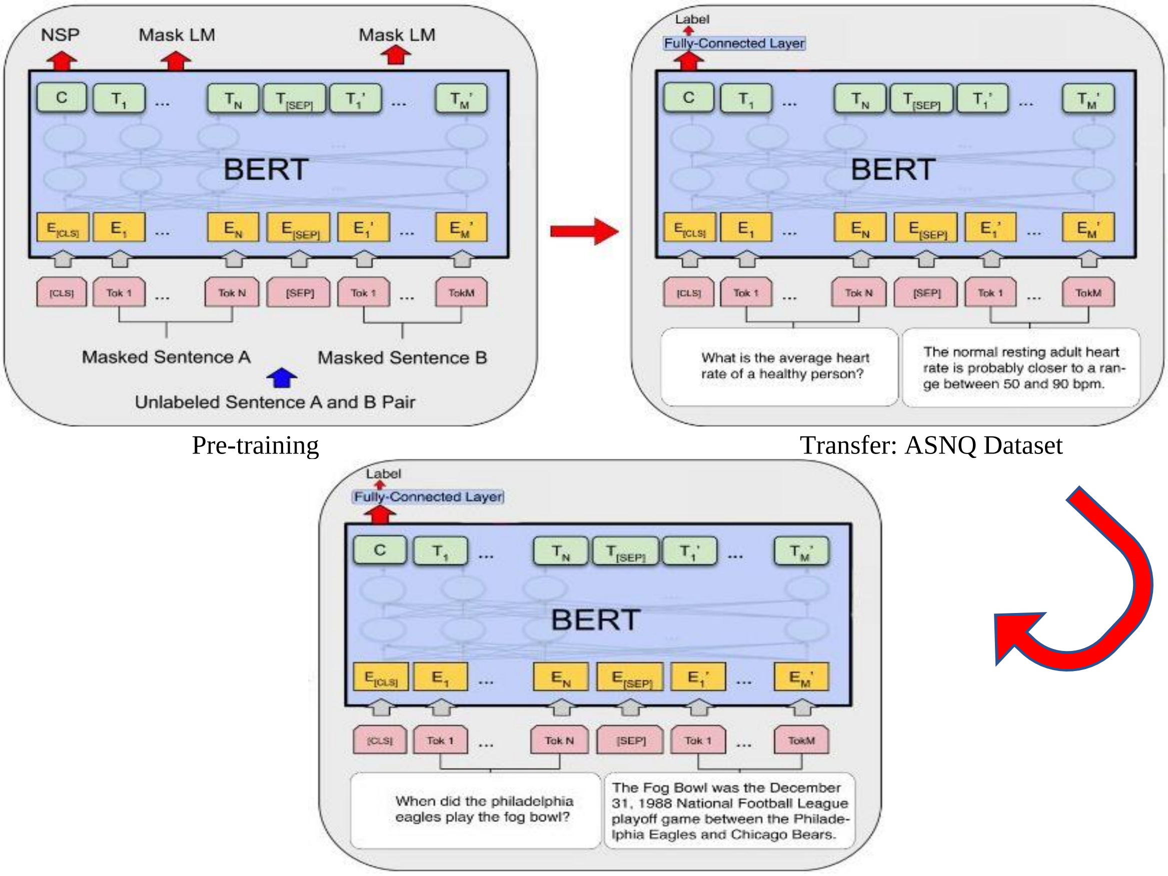
   Different text mining NLP tasks are performed to tag entities of text.

2.1. NLP QUESTION ANSWERING (QA) MODELS

Many authors [3], [4], [5] to [37] have utilized various machine learning question answering methods for searching purpose. First the query is processed by applying various techniques through a piece of text to find the best answer for a question. The main drawback of all these QA methods is that we need to extract informative and highly accurate answer prior to giving input to the model for prediction purpose. Extraction of such data requires extra effort for choosing the best techniques.

Hyunjin et al. [3] investigates sentence embedding models for ALBERT and BERT. The paper lacks the sentence embedding with massive text. There must be a need to evaluate sentence embedding with larger ALBERT frameworks, such as ALBERT-large and ALBERT- xlarge. BIGBIRD was proposed by Zaheer et al. [4]. The proposed sparse attention can manage sequences up to 8 times longer than what was previously achievable with equivalent technology. BIGBIRD dramatically improves performance on a variety of activities as a result of its capacity to handle lengthier text. Experimentally, BIGBIRD produces state-of-the-art results on a lot of NLP tasks, such as lengthy document categorization, but does not do as well on QA tests.

According to the authors [5], Longformer is a transformer-based paradigm which is customizable for handling lengthy documents and makes it simple to execute a large range of document-level NLP operations with no chunking the large input or complex architecture to integrate information across these chunks. The study lacks other pretraining objectives, especially for LED, increase the sequence length, and explore other tasks that might benefit from their model. The authors of [6] proposed a search framework for semantic information extraction that included NLP methods. The framework combines annotations from several NLP approaches like as parsing, NER, event recognition, and GDA recognition, and uses a region algebra framework to search over these annotations. Because there is duplication of annotations across the scoring queries, the scoring technique is not necessarily acceptable for the proposed framework.


NSP
Mask LM
Mask LM
C
T_1
T_N
$T_{[SEP]}$
T_1'
T_M'
BERT
$E_{[CLS]}$
E_1
E_N
$E_{[SEP]}$
E_1'
E_M'
[CLS]
Tok 1
Tok N
[SEP]
Tok 1
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Label
Fully-Connected Layer
What is the average heart rate of a healthy person?
The normal resting adult heart rate is probably closer to a range between 50 and 90 bpm.
When did the philadelphia eagles play the fog bowl?
The Fog Bowl was the December 31, 1988 National Football League playoff game between the Philadelphia Eagles and Chicago Bears.

Pre-training

Transfer: ASNQ Dataset

Adapt: Target Dataset

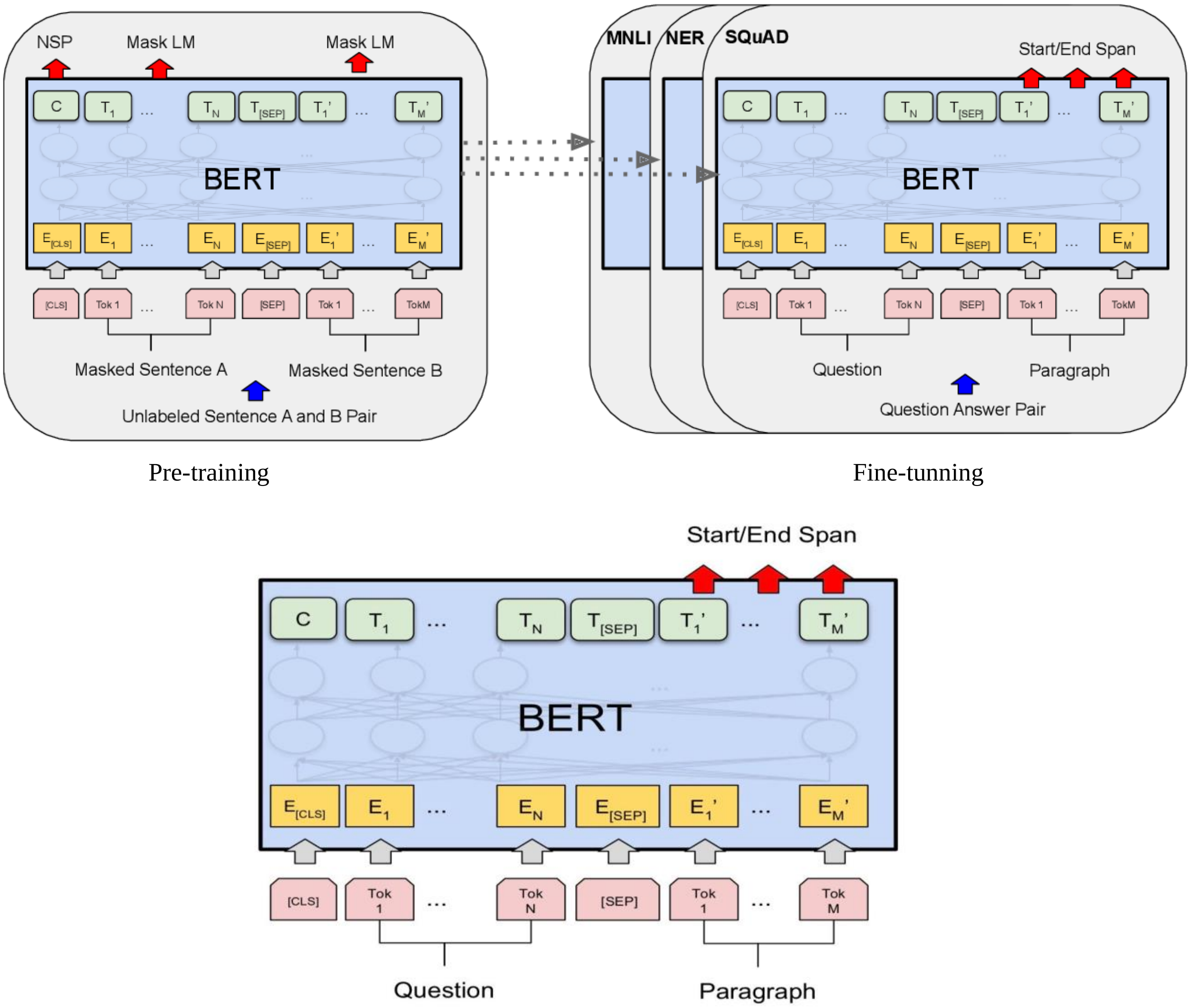
Figure 2.1: BERT Transfer and Adapt for Answer Sentence Selection [7]

As illustrated in figure 2.1, the authors of [7] proposed TANDA as a way for natural language problems for finetuning pretrained Transformer models. They accomplished this by fine-tuning a pre-trained framework as for a large and high-quality dataset before transferring it to a model for a general task. They then apply a second degree of fine-tuning to the transferred model to match it to the targeted domain. Their approach is useful for selecting answer sentences, which is a common inference job in Question Answering. They created a large-scale dataset from the Natural Questions dataset to enable for the transfer step. Future research on the TANDA method's applicability and generalization to other NLP issues might be quite intriguing. In AS2 context, it is intriguing to see if ASNQ could provide the similar advantages for tasks that are similar but obviously distinct, such as paraphrase or textual context, where the relationships among questions and answers differ from those found among members of text pairs. The authors of [8] employ the BERT approach to address a piece of the challenge of moderating activity on QA websites. They used BERT to predict 20 subjective or quality aspects of questions on QA websites in particular. Predicting subjective features is a difficult task for computers, and they showed that transfer learning from pre-trained transformers might help.

In [9], the authors describe the SQUAD, a large reading comprehension dataset based on Wikipedia articles with crowdsourced question-answer pairs. SQuAD provides a variety of question-and-answer forms. The performance of their logistic regression model (51.0 percent F1) compared to the human F1 of 86.8 % demonstrates that there is still a lot of opportunity for development. BERT, a novel language representation paradigm, was created by the inventors of [10]. By depending on both the left and right contexts simultaneously BERT is intended to pre-train deep bidirectional models from unsupervised learning text data at all levels. Figure 2.2 shows BERT pretraining and finetuning techniques. The similar designs are used in pre-training and fine-tuning, aside from the output layers. The similar pre-trained model parameters are utilized, are used to start models for multiple downstream activities. All parameters are fine-tuned, during fine-tuning. Special symbol [CLS] that appears before each input example, and [SEP] is a unique separator token (for example, separating questions/answers).

As a result, the pre-trained BERT modeling may indeed be fine-tuned only with one new output layer to produce state-of-the-art models for such a range of tasks, such as QA and language inference, without requiring major task-specific architectural changes. It achieves novel state-of-the-art results on eleven NLP tasks containing an increase in the GLUE score to 80.5 % (a 77 %-point absolute improvement), an increase in MultiNLI accuracy to 86.7 % (a 4.6 % absolute enhancement), an increase in the

SQuAD v1.1 QA Test F1 to 93.2 %, a 1.5 point absolute improvement, and an in 5.1 point absolute enhancement.



Pre-training

Fine-tunning

Figure 2.2: BERT Overall Pre-Training and Fine-Tuning Procedures

The authors of [11] described BIDAF, a multi-stage hierarchical method which employs a bidirectional attention flow technique for query-aware context representation that does not rely on early summarizing. Experiments indicate that their model outperforms the competition in the SQuAD dataset and the CNN/DailyMail cloze test. The ablation analysis demonstrates the significance of each component in their model. Model is learning, reveal by visualization and conversations, by attending the correct locations in the particular text, to develop an appropriate representation for MC and answer complicated problems. The research gap is to extend their technique such that many hops of the attention layer are incorporated. Reformer [12] combines the modelling capability of a Transformer with an architecture that can perform lengthy sequences rapidly and with little memory use, even for models with several layers. They think that by doing so, large, fully parameterized Transformer models will become more common and accessible. Furthermore, the Reformer's capacity to handle lengthy sequences allows it to be used on a variety of generative tasks. The Reformer may expand the capabilities of Transformer models to other areas such as time-series forecasting, music, picture, and video production, in addition to creating very lengthy coherent text.

TAPAS, a paradigm for question answering over tables that eliminates the production of logical forms, was introduced by the authors of this study [13]. TAPAS successfully recovers masked words and table cells after pre-training across huge data sets of text-table pairings. They also proved that the model can fine-tune on semantic parsing datasets with just limited supervision and a differentiable end-to-end recipe. According to the results, TAPAS outperforms or competes with advanced semantic parsers. They plan to improve the model in the future to represent a database with several tables as context, as well as to handle large tables well. DistilBERT, a general-purpose pretrained form of BERT that is 40% smaller and 60% faster while preserving 97% of language comprehension skills, was demonstrated by the authors of [14]. They proved that distillation may be used to successfully train a general-purpose language model, and they investigated the various components utilizing ablation research. They also demonstrated DistilBERT's viability as a solution for edge applications.

In [15], the authors provide offer a replication study of BERT pre-training, including an in-depth examination of the impact of hyperparameter tweaking and training set size. They demonstrate that BERT was significantly undertrained and propose RoBERTa, a better recipe for training BERT models which could match or exceed all post-BERT techniques. The modifications are straightforward: (1) They train the model for a longer period of time, with larger batches and more data; (2) they remove the next sentence prediction target; (3) they train on longer sequences; and (4) they dynamically alter the masking pattern used on the training data. In order to compensate for training set size impacts, they additionally gather a large new dataset (CC-NEWS) that is equivalent in size to current privately used datasets. The authors of this [16] paper demonstrate that retrieval may be done successfully using dense representations alone, with embedding learnt with a simple dual encoder architecture, from a limited number of questions and paragraphs. When tested on a variety of open-domain QA datasets, their dense retriever surpasses a powerful Lucene BM25 framework by 9 % - 19 % ultimate in terms of top-20 passage retrieval accuracy, assisting their end-to-end QA system in establishing new state-of-the-art on a variety of open domain QA benchmarks.

In this [21] study, authors investigate the landscape of transfer learning approaches for NLP by providing a single framework that translates all text-based language problems into a text-to-text format. Their extensive study compares pretraining objectives, architectures, unsupervised learning data sets, transfer techniques, and other variables on hundreds of language comprehension tests. They achieve state-of-the-art outcomes on several benchmarks, by combining research findings with scale and their modern "Colossal Clean Crawled Corpus" such as summarization, question answering, text classification, and more. The authors of [22] provide a unique strategy for generating such benchmarks systematically by maximizing composite divergence while maintaining low atom difference among train and test sets, for establishing compositional generalization benchmarks, they statistically compared to prior techniques. They use to assess the compositional generalization abilities of three machine learning architectures, which they provide a large and realistic natural language question answering dataset created using this method. They learn that they cannot generalize compositionally and that there is an unexpected negative link between accuracy and compound divergence. The authors of [22] are curious about the performance of existing architectures in the realm of compositionality benchmarks on the end-to-end task in CFQ that expects a natural language answer to a natural language query. There is also a need to broaden the technique to include a larger range of language comprehension subsets, such as the usage of unclear constructions, quantification, negations, new languages, comparatives and other vertical domains.

In this article, the authors of [23] proposed framework of hybrid generation because of access to parametric and non-parametric memory. They demonstrated that their RAG models produce cutting-edge outcomes in open-domain QA. They discovered RAG to be more truthful and precise. They investigated the learnt retrieval component thoroughly, verifying its efficacy, and demonstrated how the retrieving index may be hot-swapped to modify the model with no retraining. By combining generative pretraining with exclusionary fine-tuning, the authors of [24] described a method for achieving high natural language understanding using a single task-agnostic model. Their model attains major world knowledge and skills in order to process long-range interconnections, by pretraining on a diverse corpus with long stretches of adjacent text, which is then successfully transferred to trying to solve discriminating tasks such as QA, entailment determination, semantic similarity assessment and text classification, significantly improving the state-of-the-art on 9 of the 12 analyzed tasks. To increase the performance on discriminative tasks, ML research has long focused on unsupervised pre-training. Their [24] work showed that large performance increases are feasible, and it suggests which models (Transformers) and data types text with long-term relationships work well with this technique. For both natural language understanding and other domains, they believe that this will pave the way for future study into unsupervised learning, allowing them to better understand how and when unsupervised learning works.

The authors of [25] method combines a bigram hashing and TF-IDF matching search component with a multi-layer recurrent neural network framework trained to recognize answers in Wikipedia passages. Their tests on several publicly available QA datasets shows that (i) both models are super competitive in comparison to their counterparts and (ii) multitask learning with faraway supervision over their combo is an efficient full system on this difficult task. Future efforts should focus on improving their DrQA system. There are two obvious approaches: (i) incorporate the fact that Document Reader directly aggregates across multiple documents and paragraphs in the training, as it now trains on paragraphs individually and (ii) undertake end-to-end training across Document Reader pipelines and the Document Retriever, rather than independent systems. The authors of [26] investigated the topic of directly reading texts to answer questions, focusing on the difference between such direct approaches and employing human-annotated or automatically produced KBs. They introduced Key-Value Memory Networks, a novel model that bridges this gap by outperforming many existing techniques on two datasets, WIKIMOVIES and WIKIQA. However, there is still a performance difference. Future efforts should be made to narrow this gap even more. These models might also be used to store and access memories for other tasks and future research should test them in different domains, such as in a complete dialogue context.

This article [27] describes a MemNNs implementation for large-scale easy QA. Their results show that, when properly trained, MemNNs can handle natural language and a very large memory (millions of entries), and therefore can achieve state-of-the-art results on the popular benchmark WebQuestions. In order to get closer to human task transfer capabilities, future studies shall strive to reduce the quantity of supervision necessary, and also the number of training instances required to complete a new task. Such that, there is no known generic i.e., non-hand engineered technique which solves tasks in a semi supervised situation using just 1000 or less training instances. Furthermore, they believe that a feedback cycle of designing increasingly difficult challenges and developing algorithms to solve them would lead to new research areas. The authors of this [28] paper present a Wikidata knowledge-based question-answering system. The system converts a natural language inquiry into a SPARQL query, the performance of which yields a response. The suggested KBQA system can answer difficult questions requiring logical reasoning. The framework is the first solution to the LC-QUAD2.0 dataset and they tested its execution for various sorts of questions in LC-QUAD2.0.

This paper [29] introduces the Simple Recurrent Unit (SRU) which is a customizable recurrent architecture which is as speedy as feed-forward and convolutional units. The authors demonstrate the efficacy of SRU on a variety of NLP tasks. In terms of classification and QA datasets, SRU outperforms cuDNN-optimized LSTM by 5–9x and outperforms LSTM and convolutional models. By integrating SRU into the design, they also achieve an average 0.7 BLEU increase in translation over the Transformer model. The authors of this [30] article examines and evaluate a number of aspects that might influence the performance of domain language models. They discover that a language model tailored to a certain domain and application works well. This suggests that there is no master model that can "do it all," at least not well enough as a focused model. The model's size is a secondary consideration.

The authors of [31] presents OpenNRE, as a relation extraction tool that is open and flexible. OpenNRE strikes a balance between system encapsulation, operational efficiency, model flexibility, and usability. Custom model training and rapid model validation are made simple with OpenNRE. Some experimental findings also show that the OpenNRE models are efficient and effective, achieving equivalent or even greater performance than the original publications. Furthermore, an online solution is provided for addressing real-time extraction requirements without the need for training and deployment. Researchers of [32] present UNILM, a unified pre-training model that is jointly optimized for various LM targets with common parameters. The combination of bidirectional, unidirectional, and sequence-to-sequence LMs enables the pre-trained UNILM to be finetuned for both NLG and NLU tasks. On the GLUE benchmark and two question answering datasets, the results of their experiments demonstrate that their method surpasses BERT. Furthermore, on five NLG datasets, UNILM outperforms prior state-of-the-art models: CNN/DailyMail and Gigaword unsupervised learning summarization, SQuAD question creation, CoQA generative QA, and DSTC7 dialogue response generation.

The authors of [33] demonstrate how to train a knowledge retriever unlabeled data by utilizing backpropagating through a retrieval phase that analyses millions of pages using masked language modelling as the learning signal. The authors illustrate the efficacy of REALM by finetuning it on the difficult job of Open-QA. On three popular Open-QA benchmarks, they compare their approach to state-of-the-art frameworks for both implicit as well as explicit information storage, and discover that it outperforms prior approaches were outperformed by this method by a substantial edge i.e. 4-16 % absolute accuracy, while also giving qualitative improvements such as modularity and interpretability. The authors of this [34] study introduced the Recurrent Entity Network, a novel model that represents a promising step toward the first aim. The model is able to monitor the world state properly while reading text tales, allowing it to set based on a real-world dataset, the competitive benchmark of story interpretation, by becoming the first model to answer them all. They also demonstrated that their model can capture basic dynamics over extended periods and compete on a real world dataset.

The authors of this [35] paper demonstrated by utilizing backpropagation, a neural network with a recurrent attention mechanism and an explicit memory for accessing the memory can be effectively trained on a variety of tasks ranging from QA to language modelling. It outperforms tweaked RNNs and LSTMs of equivalent difficulty on language modelling tasks. In both experiments, expanding the number of memory hops enhances the performance. The framework has been unable to compete against memory networks trained with heavy supervision, and both failed a number of the 1k QA tests.

Additionally, seamless lookups may not expand effectively when a bigger amount of memory is required. They intend to investigate multiscale ideas of attention or hashing in these contexts. The LAMB optimizer, which allows adaptive elementwise updating and layer wise learning rates, is proposed in this [36] article. LAMB is also a general-purpose optimizer that can handle both small and big batches. They also presented theoretical analyses for the LAMB optimizer, emphasizing the situations in which it outperforms traditional SGD. LAMB outperforms current optimizers in a wide range of situations. The authors were able to reduce the BERT training duration from three days to about 76 minutes by utilizing LAMB to increase the batch size of BERT pre-training to 64K without compromising accuracy.

2.2. NAMED ENTITY RECOGNITION (NER) METHODS

Named-entity recognition (NER) is a subfield of extracting data which attempts to discover and categories named entities referenced in unsupervised learning text into predefined categories as shown in figure 2.3 person names, date and time, organizations, medical codes, quantities, monetary values, locations, percentages, and so forth. The biggest drawback of NER is data, entity, domain, task, language etc. related issues. Let us get this point through below mentioned literature review.

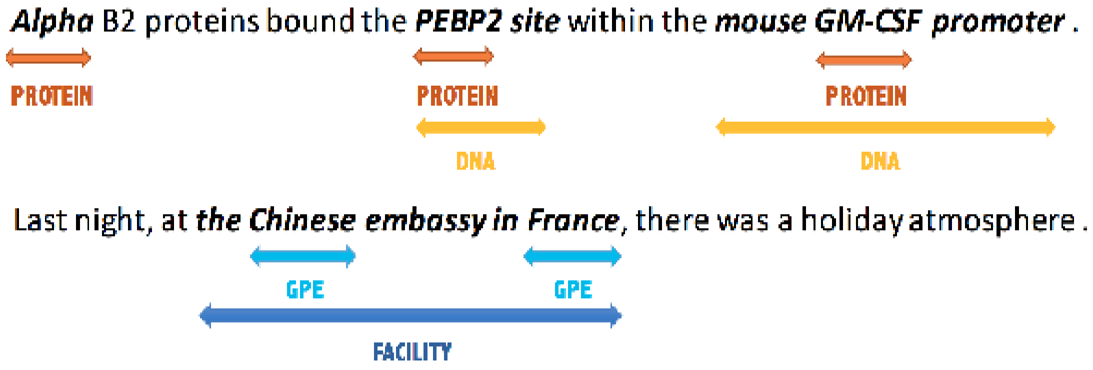

Alpha B2 proteins bound the PEBP2 site within the mouse GM-CSF promoter .
PROTEIN
PROTEIN
PROTEIN
DNA
DNA
Last night, at the Chinese embassy in France, there was a holiday atmosphere .
GPE
GPE
FACILITY

Figure 2.3: Instances of NER [47]

In this [37] work, the author introduced BioBERT, a language representation framework that has been pretrained for biomedical text mining. They demonstrated the effectiveness of pretraining BERT on biomedical corpora critical for its application in the biomedical sector. BioBERT beats prior models and QA with minimum task-specific architectural change on biomedical text mining tasks such as NER (for clinical notes, human phenotype-gene RE, and clinical temporal RE). In this [38] study, authors compare document-level features in two typical NER architectures frequently used in the literature, namely “fine-tuning” and “feature-based LSTM-CRF.” They assess several hyperparameters for document-level characteristics such as context window size and document-locality enforcement. The authors describe experiments from which they draw suggestions for modelling document context, as well as new state-of-the-art scores on various CoNLL-03 benchmark datasets. Approach is embedded into the FLAIR modelling to make replication of their studies easier.

The authors of this [39] study presented HunFlair, a multi-entity NER tagger incorporated into the largely used NLP methodology, Flair. By an average of 7.26 points per point above the next best tool, HunFlair surpasses other state-of-the-art independent NER programmes. It could be installed using just a single command and is used with only four lines of code. Researchers present two new neural architectures in this [40] paper: one is based on bi-directional LSTMs and conditional random fields, while another uses a transition-based method influenced by shift-reduce parsers to generate and label segments. Their methods depend on two types of word information: character-based representation of words from supervised corpora and unsupervised word representations from unlabeled corpora.

These models achieve cutting-edge NER accuracy in four languages while avoiding the use of language-specific information. The authors of this article [41] suggested a simple and generic semi-supervised technique for augmenting token representations in sequence tagging models using pre-trained neural language frameworks. In two prominent datasets for NER and Chunking, their approach considerably outperforms existing state-of-the-art models. According to their findings, including a backward LM to standard forward LMs increases the performance on a regular basis. Even if the LM is learned on unsupervised learning from a different domain or if the baseline model is built on a high variety of labelled instances, the suggested approach is robust. Two neural architectures for sequence tagging are shown in this work [42]. A major feature of their models is that they explicitly build and label chunks of input to represent output label dependency, using either a straightforward CRF architecture or a transition based architecture. They also utilized pre-trained word representations as well as "character-based" models which include morphological and orthographic information are used, which is critical for success. Dropout is used to avoid the researcher from being overly dependent on one particular class.

The authors of this [43] study offer a unique neutral network architecture which automatically advantages from both word- and character-level interpretations by combining bi-directional LSTM, CNN, and CRF. It is suitable to a broad range of sequence labelling tasks, because method is genuinely end-to-end, no feature extraction or data preprocessing is required. The authors test the algorithm on two data sets, the Penn-Treebank WSJ corpus for POS tagging and the CoNLL 2003 corpus for named entity identification (NER), for two sequence labelling tasks. With 97.55 % POS tagging accuracy and 91.21 % F1 for NER, they get state-of-art efficiency on both datasets. On tweets, the performance of typical NLP techniques is substantially reduced. This [44] article tackles this problem by rebuilding the NLP pipeline from POS tagging to NER. When compared to the Stanford NER system, their innovative T-NER system doubles F1 score. T-NER achieves this performance by using the redundancy inherent in tweets and utilizing LabeledLDA to use of Freebase dictionaries as a source of remote supervision. LabeledLDA surpasses co-training, boosting F1 by 25 % across 10 commonly encountered object types.

Authors of this [45] paper, propose a single trainable NER framework which gets new state-of-the-art performance on 7 public biomedical benchmarks with no need for requiring heavy contextual embeddings like BERT by reimplementing a deep learning architecture based on Bi-LSTM-CNN-Char on top of Apache Spark. This includes raising BC4CHEMD to 93.72% (a 4.1% increase), Species 800 to 80.91% (a 4.6% increase), and JNLPBA to 81.29% (5.2 % gain). This [46] study offers a hybrid semi-Markov conditional random field (HSCRF) framework for neural sequence labelling, wherein word level labels are used to generate SCRFs segment scores. In addition, the techniques for concurrently training and decoding CRF and HSCRF output layers are described. The experimental findings on the CoNLL 2003 English NER task proved the efficacy of the suggested HSCRF model, that reached state-of-the-art performance.

Researchers renormalize the NER problem as an MRC QA task in this [47] article. This elaboration has two major advantages: (1) the query can address overlapping or layered entities; and (2) the query encodes substantial previous information regarding the entity category to retrieve. Both on nested and flat NER datasets, the suggested approach yields SOTA findings, indicating its efficacy. To handle a specific sequence labelling issue, in this [48] study, researchers offer a deep neural network model, NER. The model is divided into three sub-networks to effectively leverage character level and capitalization characteristics as well as word-level contextual representation. The model was assessed in Russian, Vietnamese, English, and Chinese to demonstrate its ability to generalize to many different languages. Obtained state-of-the-art results: 91.10 %, 94.43 %, 91.22 % and Gareev's dataset, VLSP-2016, CoNLL-2003, and MSRA datasets, respectively, yielded 92.95 % of F-Measure.

Iterated dilated convolutional neural networks, rapid token encoders that effectively gather wide context without sacrificing resolution, are presented by researchers of [49]. These give significant increases in sequence labelling speed, especially when processing complete documents at once. Allowing the framework to properly identify named entities, the authors of this [50] work employ principles using graph-based dependency parsing to offer their framework with a global perspective of the input via a biaffine framework that evaluates pairings of start and end tokens in a sentence that they explore all spans. They demonstrate that the framework performs well for both nested and flat NER by evaluating it on eight corpora and attaining SoTA performance on all of them, with accuracy improvements of up to 2.2 %.

Summary of above mentioned few major techniques used in literature, which break the ice to work on this research is given in tabular form in Table 2.1.

Table 2.1: Summary of Different Literature Reviewed for Information Extraction

Paper	Year	Main Contribution	Research Gaps
[3]	2021	Sentence embedding models for BERT & ALBERT	Lacks large text embedding ALBERT-large & ALBERT-xlarge
[4]	2011	1. Authors proposed that BIGBIRD handle sequences of up to 8x length. 2. Capability to handle larger amounts of text BIGBIRD significantly increases performance on a variety of jobs.	BIGBIRD not score much better for QA like tasks

[5]	2019	Authors proposed Longformer for processing long documents without chunking and complex architecture	Lacks LED, increase the sequence length, and explore other tasks that might benefit from their model
[6]	2011	Described a search architecture that incorporates annotations from different NLP approaches like as parsing, NER, event recognition, and GDA recognition, as well as area algebra searches.	There is no acceptable scoring technique since there is annotation duplication across the scoring questions.
[7]	2020	TANDA as a means of fine-tuning pre-trained agents' Natural language task transformer models	It would be intriguing to investigate when ASNQ can provide the same advantages for tasks that are similar but obviously distinct, such as paraphrase or textual import, in which the connections identified between members of text pairings differ from those discovered between questions and answers.
[13]	2020	TAPAS, a paradigm for question answering over tables that eliminates the production of logical forms, was introduced by the authors of this study	Modelling a framework to represent a database with several tables as context, as well as to handle huge tables well.
[23]	2020	To having access to parametric and non-parametric memory, authors in this article proposed hybrid generation models.	This discovery offers up new paths for research into how parametric and non-parametric memory combine and how to integrate them most efficiently, with potential applications to a broad range of NLP tasks.
[24]	2018	(1) Through generative pre-training and discriminative fine-tuning, introduced single task-agnostic model (2) Work suggests that achieving significant performance gains is indeed possible. Offers hints as to what Transformers or models and data sets work best with the approach	To better understand how and when unsupervised learning works.
[25]	2017	Open domain QA (1) Bigram hashing and TF-IDF matching both modules are very competitive in comparison to their counterparts (2) Multitask learning with remote supervision is an effective complete system for this challenging work.	Absence to perform end-to-end sentence tokenization or chunking to extract data of interest

[27]	2015	Describes a MemNNs implementation for large-scale easy QA.	To get closer to human task transfer capabilities, the number of training examples necessary to accomplish a new task must be reduced.
[31]	2019	Offer OpenNRE, an open and extensible relation extraction tools.	Requirement of long-term maintenance in the future to address issues and fulfil new requests
[43]	2016	The authors of this study presented a neural network design for sequence tagging. It is an end-to-end framework which does not rely on task-specific capabilities, feature engineering, or data pre-processing.	Applying the approach to data from other domains, including such social media, would be an intriguing direction (Twitter and Weibo).
[47]	2020	This research has two major advantages: (1) the query can address overlapping or layered entities; and (2) the query encodes substantial previous information about the entity category to retrieve.	Should investigate variants of model architecture.
[48]	2019	The researchers presented a deep hybrid neural network framework that employs three sub-networks to completely utilize the main input characteristics, following by a CRF layer to record the implicit limitations on the sequence of output tags.	(1) Using capitalized sequences that have been transformed from raw input phrases can improve tagging accuracy. The increasing accuracy heavily dependent on features of the language. (2) Cannot manage large enough datasets.
[49]	2017	Iterated dilated convolutional neural networks, rapid token encoders that effectively gather wide context without sacrificing resolution, are presented by researchers of [49]. These give significant increases in sequence labelling speed, especially when processing complete documents at once.	There is a need to expand research to NLP tasks with higher structured output like parsing.

2.3. RESEARCH GAPS

In most of the work, question answering is performed. To yield solid and promising results with NLP, the dataset size must be large enough to capture all the possible conditions found in practice irrespective of transfer learning and augmentation techniques. Studies have shown that having too few samples in the input dataset results in high error rates most of the times. Although in above literature, NLP techniques have been applied to classify and predict the desired information from textual information, there are some drawbacks as given below:

1. Need to deduce a solution for such problems where the large textual dataset contains variety of data, from different domains.
2. Lacks [1] the sentence embedding with large text. Modelling of framework which outperforms in QA tasks [4].
3. Labels segments using transfer-based learning approach.
4. Modelling a framework [13] to represent a database with several tables as context, as well as to handle huge tables well.
5. It may be worthwhile to examine if the two factors could be simultaneously pre-trained from beginning, with either a denoising aim identical to BERT or with different objective. In [23] work offers up new study avenues on how parametric and non-parametric memories interact and how to most efficiently combine them, with promise for application to a broad range of NLP tasks.
6. To better understand how and when unsupervised learning works [24].
7. To better understand which models and datasets work best, for a particular problem.
8. Perform end-to-end sentence tokenization or chunking having the data of interest [25].
9. Need to develop such a framework which should be general or apply to all kind of domains [26]. Because a good framework does not require any domain or task-specific expertise, it is straightforward to adapt to various areas [43].
10. To get closer to human task transfer capabilities, the number of training examples necessary to accomplish a new task must be reduced [27].
11. Performance is independent of model size; hence model size is not primary factor.
12. Requirement of long-term maintenance in the future to address issues and fulfil new requests [31].
13. Lack of structural knowledge to learn how to decide what entities are informative.
14. Scale the batch size of model for massive datasets without losing accuracy and performance degradation, thereby, reducing the model training time.

15. Build small training datasets because its difficult in making a large enough dataset for training.
16. Need of smaller, faster, cheaper and lighter model. How data scientists could help government agencies for being pushing to make evidence-based decisions based on the most up-to-date data available and science. Previous research has shown that it is possible to develop algorithms to automate the search and discovery of references to data. Now, we have developed the best approaches to identify critical datasets used in scientific publications.

2.4. CONTRIBUTION

Based on the extensive review of literature and to overcome all the above-mentioned shortcomings and drawbacks, the aim of this research is to use the new dataset “Coleridge Initiative- Show Us the Data [1]," which contains 14316 examples of research publications or papers fostered the new research in search or QA containing substantial number of sample large textual document to capture high level accuracy and to apply transfer learning to classify datasets. In proposed framework, we have use natural language processing (NLP) to automate the discovery of how scientific data are referenced in publications. Utilizing the full text of scientific publications from numerous research areas gathered from CHORUS [18] publisher members and other sources, we identify datasets that the publications' authors used in their work. Our framework also intuits that a larger model size does not always imply greater performance on a cross-domain benchmark challenge.

Proposed methodology provides long-term maintenance in the future to address issues and fulfil new requests because it can not only increase the performance as compared to other benchmark transfer learning models. Further to remove the load, **DEFNLP** comes up as a state of art approach which generally consider text at sentence level and thus don’t model information that crosses sentence context or barriers by using **NER** to locate the desired information.

CHAPTER 3: MATERIALS

3.1. COLEREIDGE INNITIATIVE DATASET:

Research is carried out using the Kaggle platform by participating in [1] Coleridge Initiative - Show US the Data competition to identify the mentioned datasets within scientific publications and not just to match known dataset strings but to generalize to datasets that have never been seen before using NLP and statistical techniques. This problem challenges data scientists to automate NLP approaches that enable government agencies and researchers to quickly find the information they need. We are provided with 4 main pieces of data:

1. **Train Data:** Containing all the metadata of the publications, such as their title and the dataset they utilize.
2. Actual publications that are referenced in train data in **JSON** format.
3. **Test Data:** Containing the actual publications that will be used for testing purposes (thus, with no ground truth CSV file available).
4. sample_submission: Containing all the publications IDs in the test set, for which we'll have to populate the prediction column.

3.2. IDENTIFYING THE DATASETS (PREDDICTION STRINGS):

1. Short excerpts from the publications that appear to note as dataset.
2. Each publication's predicted strings are **sorted alphabetically** and processed in that order. Any score ties are settled in this manner.
3. For each publication Id in the test set, must predict excerpts (multiple excerpts divided by a pipe '|' character) for **PredictionString** variable. The file should contain a header i.e., Id and PredictionString and must be in correct format.
4. Predictions that more accurately match the precise words used to identify the dataset will score higher.
5. Generalize to datasets that have never been seen before using NLP and statistical techniques.

6. Not all datasets have been identified in train, so these unidentified datasets have been used as a portion of the public test labels. These should serve as guides for the difficult task of labeling the private test set.

3.3. TRAIN DATA DATATYPE INFORMATION

1. There are 5 columns in the dataset.
2. There are 19661 rows in the dataset.
3. All columns are categorical no numeric columns.
4. Not any null values.
5. There 45 dataset titles but 130 dataset labels. Meaning that there are some datasets that has multiple labels as described in Table 3.1.

Table 3.1: Training Data Exploration

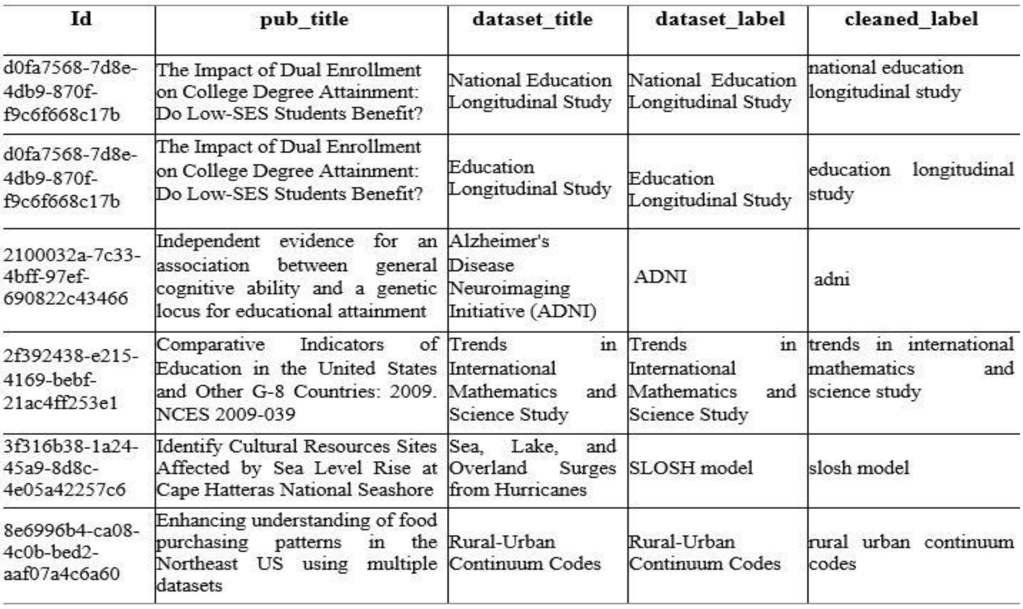
Label	Count	Unique	Top	Freq
Id	19661	14316	170113f9-399c-489e-ab53-2faf5c64c5bc	22
pub_title	19661	14271	Science and Engineering Indicators 2014	22
dataset_title	19661	45	Alzheimer's Disease Neuroimaging Initiative (ADNI)	6144
dataset_label	19661	130	ADNI	3673
cleaned_label	19661	130	adni	3673

3.4. TRAIN DATA LABELS DESCRIPTION

As shown in table 3.2, training data contains five categorical labels which are describe as follows:

1. **id** - publication id - note that there are multiple rows for some training documents, indicating multiple mentioned datasets.
2. **pub_title** - title of the publication (a small number of publications have the same title).
3. **dataset_title** - the title of the dataset that is mentioned within the publication.
4. **dataset_label** - a portion of the text that indicates the dataset.
5. **cleaned_label** - the dataset_label, as passed through the clean_text function from the Evaluation page.

Table 3.2: Excerpt from Train Data



Id	pub_title	dataset_title	dataset_label	cleaned_label
d0fa7568-7d8e-4db9-870f-f9c6f668c17b	The Impact of Dual Enrollment on College Degree Attainment: Do Low-SES Students Benefit?	National Education Longitudinal Study	National Education Longitudinal Study	national education longitudinal study
d0fa7568-7d8e-4db9-870f-f9c6f668c17b	The Impact of Dual Enrollment on College Degree Attainment: Do Low-SES Students Benefit?	Education Longitudinal Study	Education Longitudinal Study	education longitudinal study
2100032a-7c33-4bff-97ef-690822c43466	Independent evidence for an association between general cognitive ability and a genetic locus for educational attainment	Alzheimer's Disease Neuroimaging Initiative (ADNI)	ADNI	adni
2f392438-e215-4169-bebf-21ac4ff253e1	Comparative Indicators of Education in the United States and Other G-8 Countries: 2009. NCES 2009-039	Trends in International Mathematics and Science Study	Trends in International Mathematics and Science Study	trends in international mathematics and science study
3f316b38-1a24-45a9-8d8c-4e05a42257c6	Identify Cultural Resources Sites Affected by Sea Level Rise at Cape Hatteras National Seashore	Sea, Lake, and Overland Surges from Hurricanes	SLOSH model	slosh model
8e6996b4-ca08-4c0b-bed2-aaf07a4c6a60	Enhancing understanding of food purchasing patterns in the Northeast US using multiple datasets	Rural-Urban Continuum Codes	Rural-Urban Continuum Codes	rural urban continuum codes

Prediction Submission

A sample_submission file must be submitted in the correct format.

1. **Id** - publication id as mentioned in test data.
2. **PredictionString** - To be filled with equivalent of cleaned_label of train data.

3.5. OBSERVATIONS

The following are some observations which we deduce after the experimentation with our dataset.

1. The Train dataset has 19,661 counts but only 14,316 unique 'Id' in the dataset which means that there are some publications that are using multiple datasets. That's why that id is repeating as shown in Table 3.2.

Table 3.3: Number of Duplicates in Datasets

Labels	Number of Duplicates
Id	5345
pub_title	5419
dataset_title	19616
dataset_label	19531
cleaned_label	19531

The train dataset has 19,661 counts but only 14,316 unique 'Id' in the dataset. This means some 'Id' are duplicates, meaning some 'Id' use multiple datasets. The 'pub_title' has 19,661 counts but has only 14,271 unique titles. This means some 'pub_title' are duplicates. There are less 'pub_title' counts than 'Id' counts, meaning some 'pub_title' has multiple 'Id'. The 'dataset_title' has 19,661 counts but has only 45 unique titles. This means some 'dataset_title' are used many times by different publications. The 'dataset_label' has 19,661 counts but has only 130 unique labels. This means some 'dataset_label' are duplicates. There are less 'dataset_title' counts than 'dataset_label', meaning some 'dataset_title' are labeled differently by different publications. All above mentioned details could also be visualize through figure 3.2.

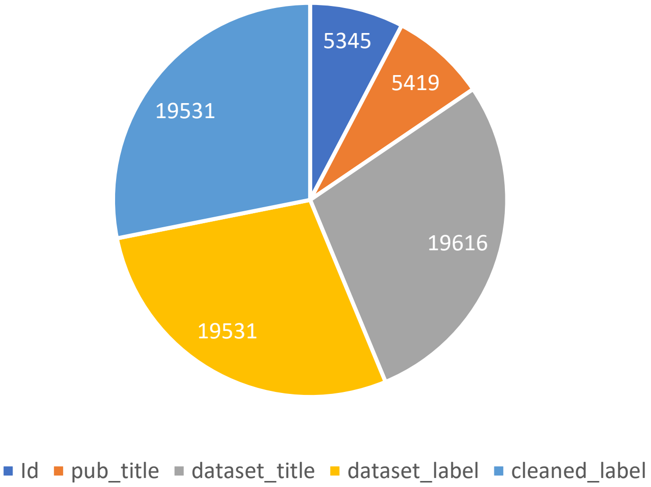


Figure 3.2: Number of Duplicate Records in Dataset

2. Single publication i.e., pub_title is using multiple datasets as shown in Table 3.3.
3. There is no one to one mapping of id and pub_title. Meaning that there are cases when two different publications (from two different authors) have same title.

Table 3.4: Single Title with Multiple Publications

Sr No.	Id	pub_title	dataset_title
25	5fa574e1-b2b8-4e12-a55b-efe965abd28c	Teachers and the Gender Gaps in Student Achievement	Beginning Postsecondary Student
26	e32d24d8-bc41-401c-b49d-99b93c2d4533	Teachers and the Gender Gaps in Student Achievement	Survey of Earned Doctorates

3.4.1. Distribution of pub_title & dataset_title

1. No. of different labels according to publication Id: **133**
2. We have duplicate ids but we are not going to remove them from data because one research paper may consist of more than one dataset. Anyhow duplicate publications with repetitive number are shown in Figure 3.4.

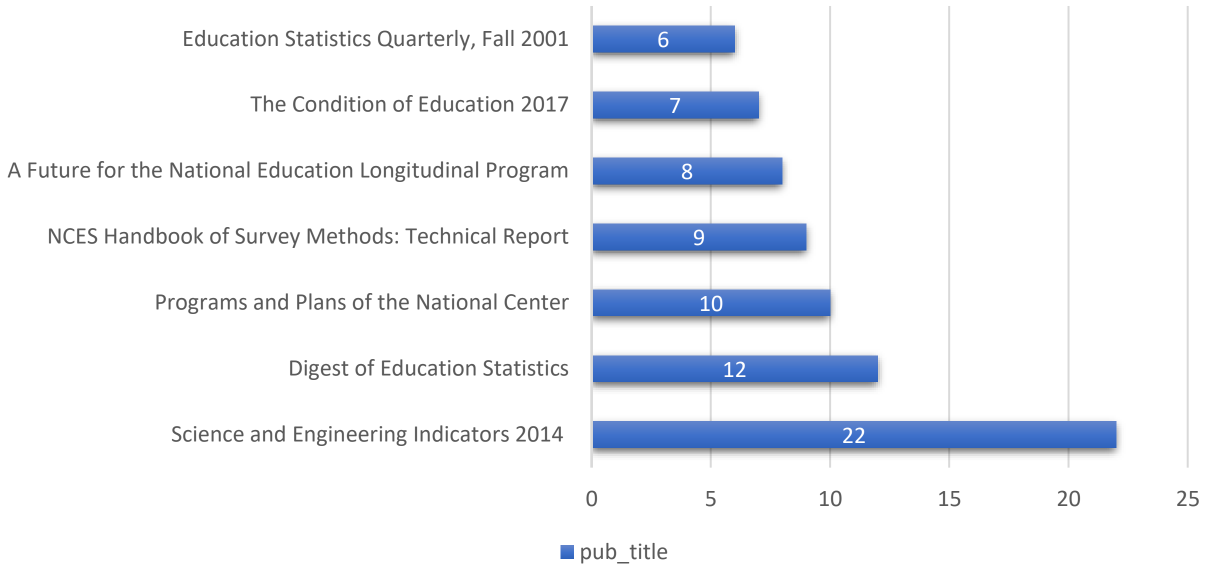


Figure 3.4: Distribution of pub_title & dataset_title in Dataset

3.4.2. Distribution of dataset_title & dataset_label

The labels, dataset_title and dataset_label both are somehow related to each other or are same. As it is clear from Figure 3.5, that there is high distribution of dataset_title Alzheimer's Disease Neuroimaging Initiative (ADNI) i.e., 6144 which is also included in dataset_label as ADNI with 2400 repetitions. Rural-Urban Continuum Codes have equal distribution in dataset_title and dataset_label i.e., 490. From the bar chart it is also clear that in both labels Alzheimer's Disease Neuroimaging Initiative (ADNI) and ADNI have the greatest distribution from rest in dataset.

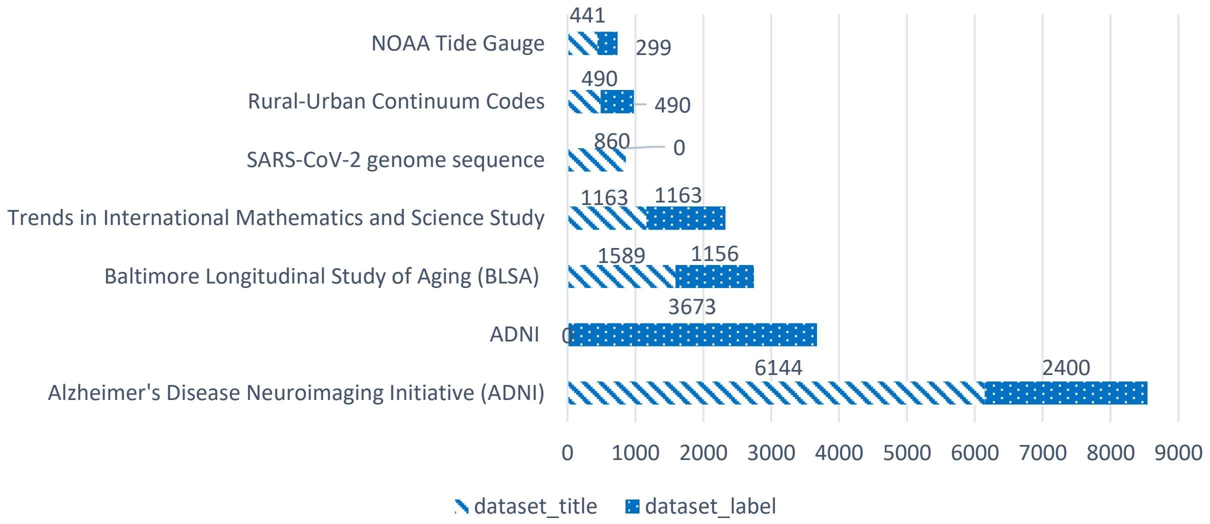


Figure 3.5: Relationship Between Dataset Title & Label

3.4.3. cleaned_label Visualization

Cleaned_label is nothing but the clean form (without stop words and symbols) in lowercase of labels, dataset_title and dataset_label, through which get the prediction strings in cleaned text form as shown in Figure 3.6.

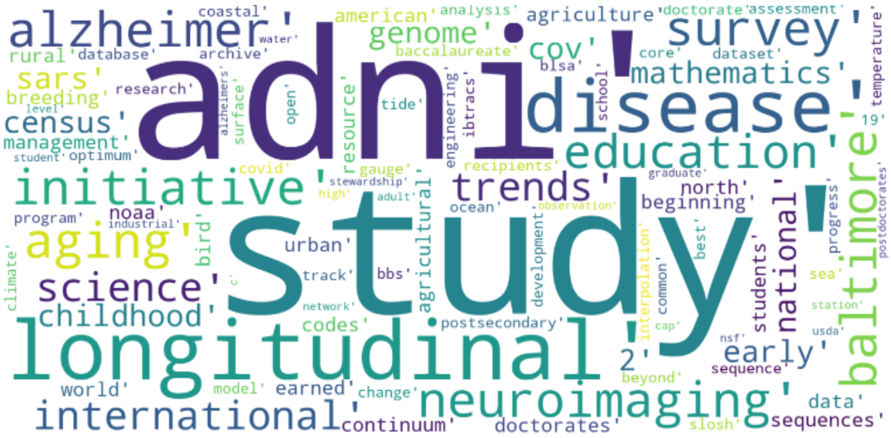
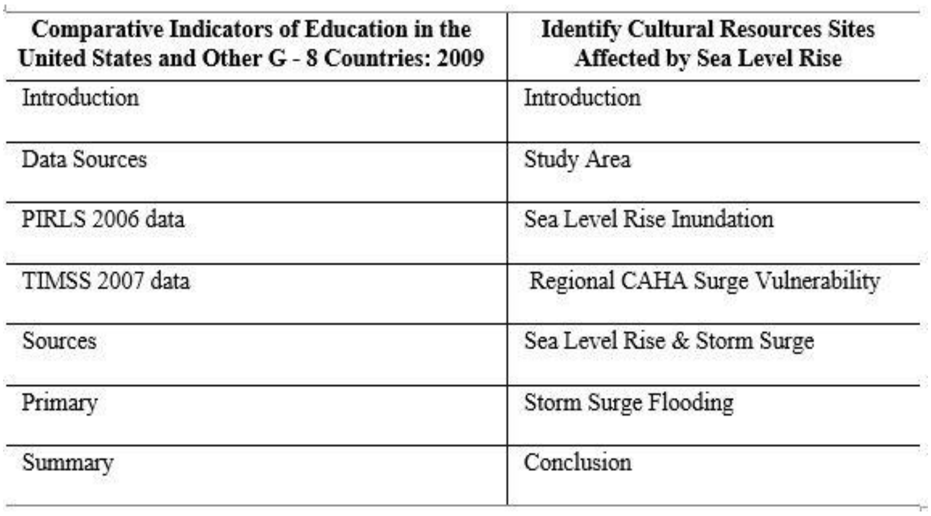


Figure 3.6: Top 100 Most Common Words in cleaned_label

3.4.4. JSON Data Description

1. JSON format, broken up into sections with section titles.
2. Examining the section titles in each json file.
3. Some json files have different titles, few are common in some as shown in Table 3.4. The section_title "Introduction" is common in both the publications while rest are different.

Table 3.5: Comparison of Section Titles of Two Publications



Comparative Indicators of Education in the United States and Other G - 8 Countries: 2009	Identify Cultural Resources Sites Affected by Sea Level Rise
Introduction	Introduction
Data Sources	Study Area
PIRLS 2006 data	Sea Level Rise Inundation
TIMSS 2007 data	Regional CAHA Surge Vulnerability
Sources	Sea Level Rise & Storm Surge
Primary	Storm Surge Flooding
Summary	Conclusion

3.6. TOOLS AND LANGUAGES

Kaggle is used as tool in this research for implementing proposed methodology with the specifications as exhibit in Table 3.5, it provides free CPU and GPU support and RAM up to 13 GB max and 16 GB max respectively with the disk space of 73.1 GB Max. For faster access, more storage space and to avoid clash and timeout during processing, two notebooks has been created. Since the size of dataset that is used in this research was very large i.e., there were about 14,316 publications in JSON format that requires more RAM for processing. Another reason to create two notebooks is that we are bound to used only CPU accelerator because of competition requirements, though GPU is much faster in processing data. Also, the same virtual machine can be used continuously for at most 9 hours.

Table 3.6: Session Matrices

ACCELERATOR	RAM	DISK
CPU	13 GB Max	73.1 GB Max
GPU	16 GB Max	73.1 GB Max

Kaggle also supports Python that is used as a scripting language in this research. Python NLP and machine learning packages can be imported into the Kaggle notebook by using just a single line of code instead of downloading and installing them separately thereby saving a lot of time. Some of the packages used include the following:

- Keras with TensorFlow backend
- Numpy and pandas
- Matplotlib for plotting
- csv for reading csv files containing labels and metadata for the training set
- JSON for reading research publications containing the full text of the training set's publications in JSON format, broken into sections with section title
- Importing NLP packages include nltk, SpaCy, WordCloud, STOPWORDS, AbbrX
- For modelling import torch, AutoTokenizer, AutoModelForQuestionAnswering
- Importing seaborn, glob, tqdm, gc, collections, re, spikex etc.

CHAPTER 4: METHODOLGY

After exploratory data analysis, we move forward towards featuring of our proposed framework. Data Extraction Framework Using Natural Language Processing Techniques (**DEFNLP**), build just such an open and transparent approach. The findings demonstrate how public data is utilized in science and assist the government in making more wiser and transparent public investments. It helps move researchers and governments from using ad-hoc methods to automated ways of finding out what datasets are being used to solve problems, what measures are being generated, and which researchers are the experts.

4.1. PROPOSED FRAMEWORK

In proposed framework **DEFNLP**, we have use NLP techniques to automate the discovery of how scientific data are referenced in publications. Utilizing the full text of scientific publications from numerous research areas gathered from [18] CHORUS publisher members and other sources, we have identified data sets that the publications' authors used in their work. Our predictions are short excerpts from the publications that appear to note as dataset. Predictions that more accurately match the precise words used to identify the dataset within the publication, score higher. Predictions are cleaned using the clean_text function to ensure proper matching.

Publications are provided in JSON format, broken up into sections with section titles. The goal in this research is not just to match known dataset strings but to generalize to datasets that have never been seen before using NLP and statistical techniques. A percentage of the public test set publications are drawn from the training set - not all datasets have been identified in train, so these unidentified datasets have been used as a portion of the public test labels. These should serve as guides for the difficult task of labeling the private test set.

4.2. Features of Data Extraction Framework Using Natural Language Processing Techniques (DEFNLP)

DEFNLP is simplified into three phases for identifying the datasets as prediction strings. Following the concept of transfer learning of machine learning as shown in figure 4.1, we extract the desired datasets by merging prediction strings obtain in phase 1, 2 and 3 respectively.

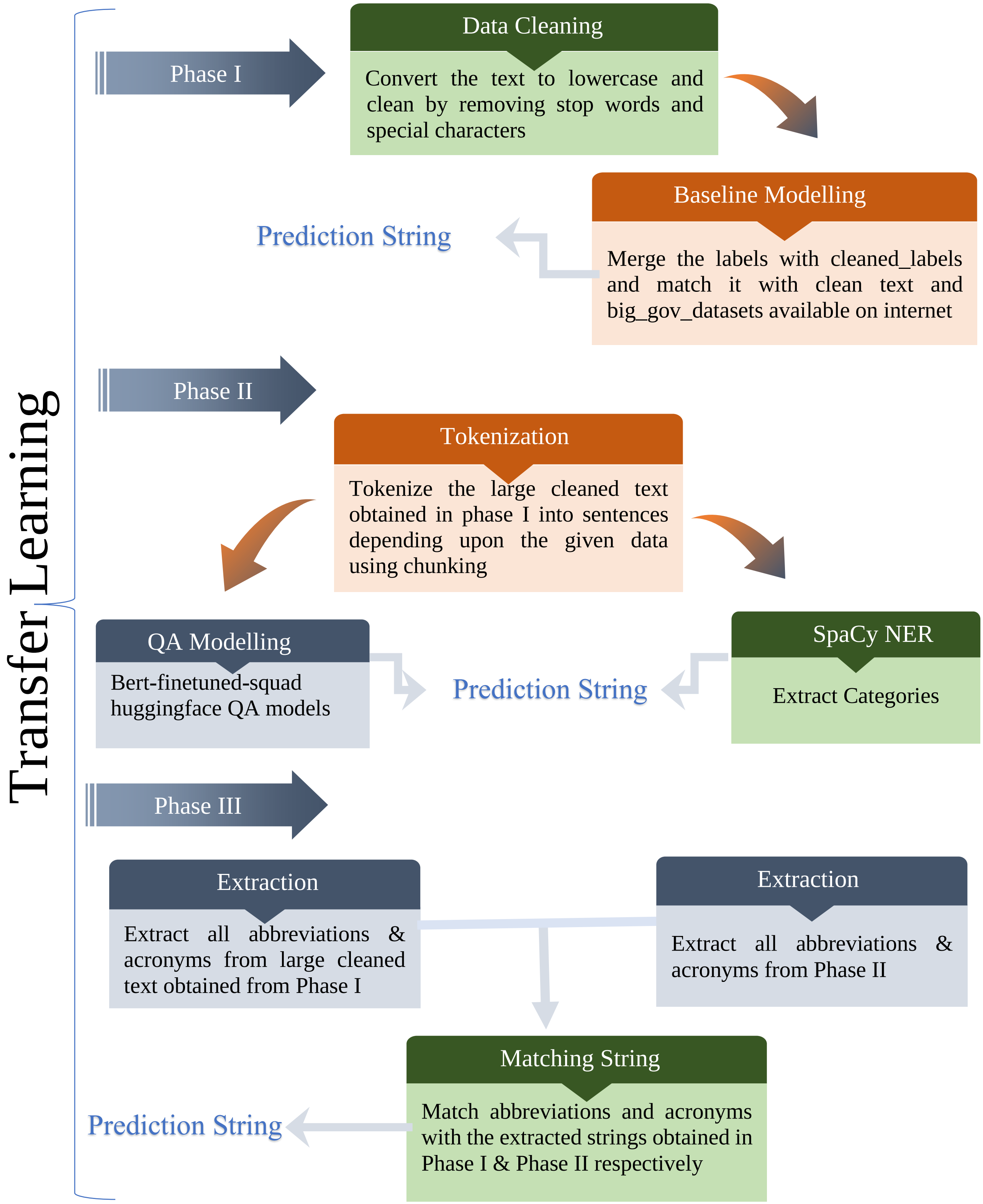

Transfer Learning
Phase I
Data Cleaning
Convert the text to lowercase and clean by removing stop words and special characters
Baseline Modelling
Merge the labels with cleaned_labels and match it with clean text and big_gov_datasets available on internet
Prediction String
Phase II
Tokenization
Tokenize the large cleaned text obtained in phase I into sentences depending upon the given data using chunking
QA Modelling
Bert-finetuned-squad huggingface QA models
Prediction String
SpaCy NER
Extract Categories
Phase III
Extraction
Extract all abbreviations & acronyms from large cleaned text obtained from Phase I
Extraction
Extract all abbreviations & acronyms from Phase II
Matching String
Match abbreviations and acronyms with the extracted strings obtained in Phase I & Phase II respectively
Prediction String

Figure 4.1: Features of Data Extraction Framework Using Natural Language Processing Techniques (DEFNLP)

Phase I: Data Cleaning & Baseline Modelling

1. In phase I as a baseline modelling, we merge all text from json files with respect to their ids of publications into train as well as sample_submission files data.
   - Merge all text from json into train with column name" text" in clean format by using the **clean_text** function for matching purposes which converting the text in lowercase and removing special characters, emojis and multiple spaces etc.
   - We have our clean **text** appended in our train dataframe.
   - Similarly, appending the text for the sample_submission, as shown in Table 4.1

Table 4.1: JSON Text Appended in Dataframe

Id	Prediction String	text
2100032a-7c33-4bff-97ef-690822c43466	NaN	cognitive deficits reduced educational ach...
2f392438-e215-4169-bebf-21ac4ff253e1	NaN	report describes how the education system...
3f316b38-1a24-45a9-8d8c-4e05a42257c6	NaN	cape hatteras national seashore caha locate...
8e6996b4-ca08-4c0b-bed2-aaf07a4c6a60	NaN	a significant body of research has been conduc..

2. We make the learning process faster by getting rid of not essential words which don't have much meaning to our statement. And just to make our statement sound more cohesive stop words such as 'are', 'and', 'the' has been removed. We speed up the learning process by eliminating non-essential words that are not much significance to our statement. And, to make our statement seem more coherent, stop words like 'are,' 'and,' and 'the' have been deleted.

3. In phase 1, we have done **data cleaning** by converting large text to lowercase, remove stop words, special characters, emojis and multiple spaces by using simple regular expressions and functions.

4. After data cleaning in phase I, predict the datasets as shown in Table 4.2 by merging the labels i.e. dataset_title, dataset_labels and cleaned_labels by first converting them to lowercase and matched them with the cleaned text.

Table 4.2: Datasets Identified Through Baseline Modelling

Id	PredictionString
2100032a-7c33-4bff-97ef-690822c43466	adni | alzheimer s disease neuroimaging initiative adni
2f392438-e215-4169-bebf-21ac4ff253e1	common core of data | trends in international mathematics and science study | nces common core of data
3f316b38-1a24-45a9-8d8c-4e05a42257c6	noaa storm surge inundation | sea lake and overland surges from hurricanes | slosh model
8e6996b4-ca08-4c0b-bed2-aaf07a4c6a60	rural urban continuum codes

5. Meanwhile datasets are also identified by matching the datasets in **big_gov_datasets** [17] document containing government datasets, openly available on internet. Bold text in Table 4.3 identified the datasets different from baseline modelling which are obtained by using government dataset.

Table 4.3 Datasets Identified by Using Government Datasets

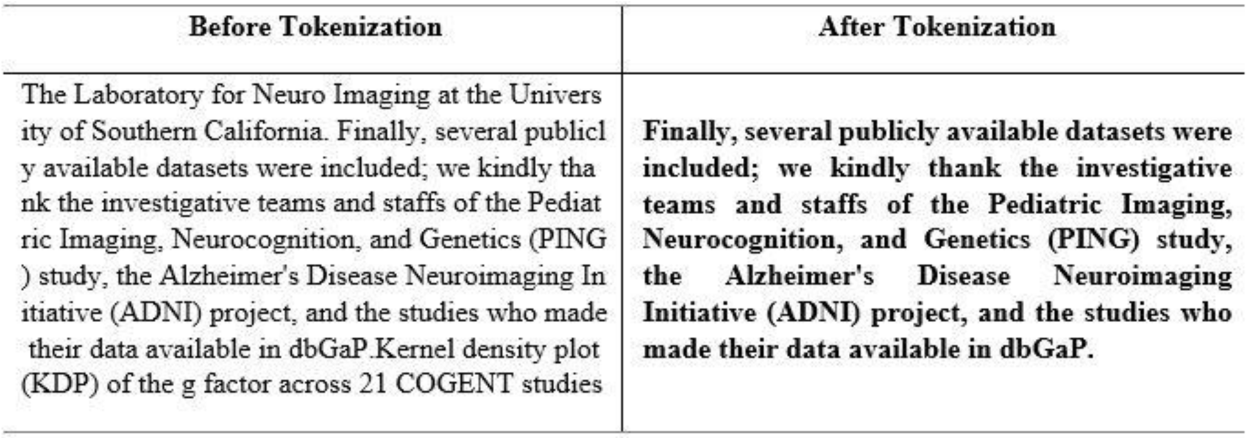
Id	PredictionString
2100032a-7c33-4bff-97ef-690822c43466	adni |alzheimer s disease neuroimaging initiative adni | **pubmed**
2f392438-e215-4169-bebf-21ac4ff253e1	common core of data | nces common core of data | trends in international mathematics and science study | **schools and staffing survey | integrated postsecondary education data system | ipeds | progress in international reading literacy study**
3f316b38-1a24-45a9-8d8c-4e05a42257c6	slosh model | noaa storm surge inundation | sea lake and overland surges from hurricanes
8e6996b4-ca08-4c0b-bed2-aaf07a4c6a60	rural urban continuum codes

Phase II: QA Modelling & SpaCy

1. Phase II start off by tokenizing the sentences having words such as **data**, **datasource**, **datasources**, **dataset**, **datasets, database, databases, sample, samples** etc. to break the long text into small chunks for speedy execution and avoiding the crashing of session to save CPU and GPU memory and also prevent to inclusion of unimportant information or data to predict the datasets. For instance,

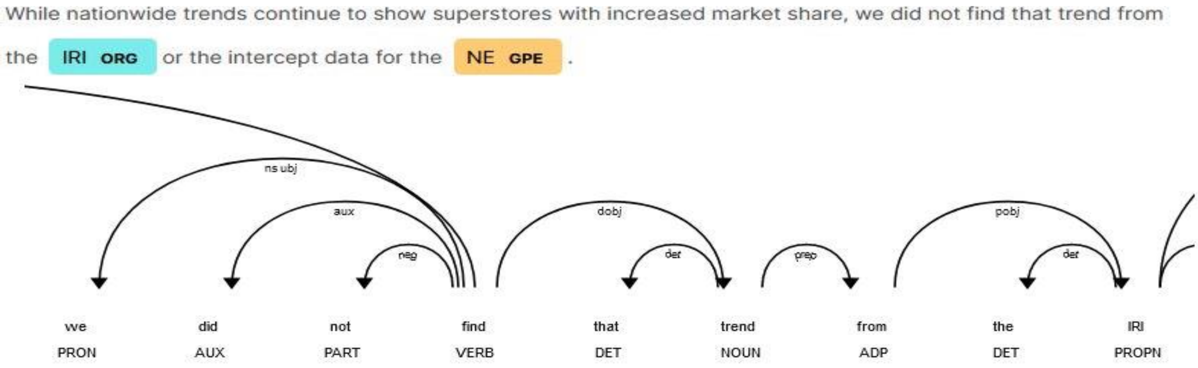
as shown in Table 4.4 the original text before tokenization contains three sentences in paragraph but only one sentence contains the word from bag of words i.e., datasets in second sentence. After tokenization, only second chunked sentence is chosen.

Table 4.4 Tokenized Sentences Having Specific Words



Before Tokenization	After Tokenization
The Laboratory for Neuro Imaging at the University of Southern California. Finally, several publicly available datasets were included; we kindly thank the investigative teams and staffs of the Pediatric Imaging, Neurocognition, and Genetics (PING) study, the Alzheimer's Disease Neuroimaging Initiative (ADNI) project, and the studies who made their data available in dbGaP.Kernel density plot (KDP) of the g factor across 21 COGENT studies	**Finally, several publicly available datasets were included; we kindly thank the investigative teams and staffs of the Pediatric Imaging, Neurocognition, and Genetics (PING) study, the Alzheimer's Disease Neuroimaging Initiative (ADNI) project, and the studies who made their data available in dbGaP.**

2. After tokenizing and chunking the large text, in second step we use NLP library, **SpaCy** for Named Entity Relationship NER tagging of the whole text to make bag of words as shown in figure 4.2.



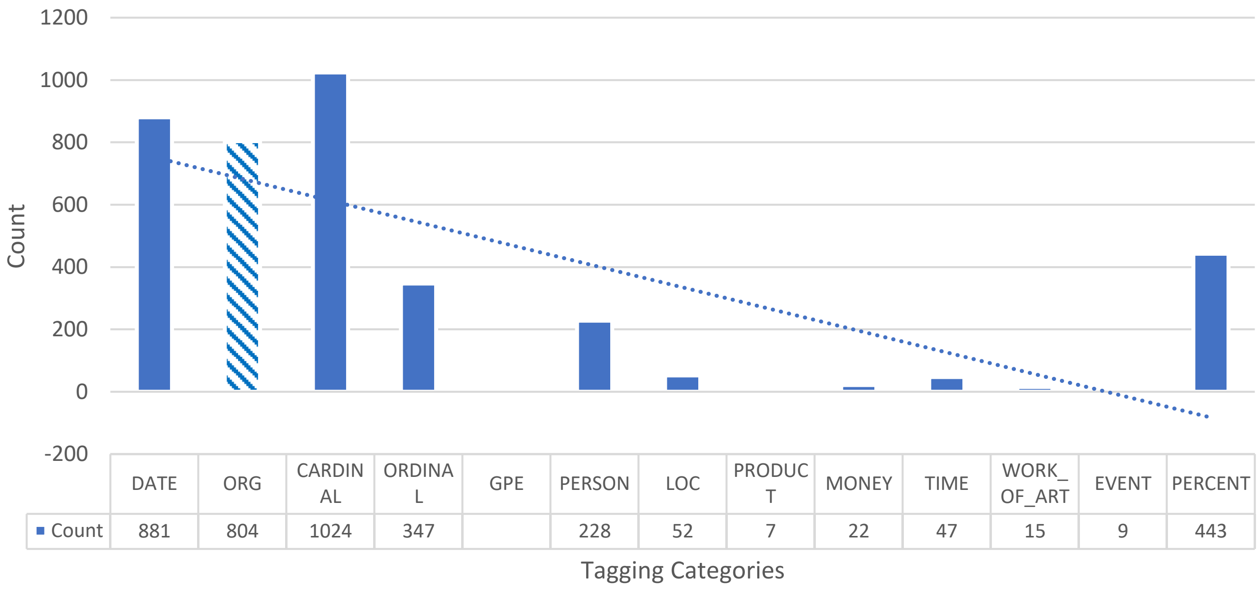
While nationwide trends continue to show superstores with increased market share, we did not find that trend from the IRI ORG or the intercept data for the NE GPE .

Figure 4.2: Excerpt of NER from Sentence

It specifies the connection between headwords and their dependents as shown in the form of directed graph representation. The nodes are words and the edges represent the grammatical connections. We introduced our machine to cultural references and every day names by flagging names of medical codes, quantities, monetary values, percentages. movies, time, important personalities, expressions, locations, organizations etc. that may occur in the document with the help of NER. For instance, consider the following array which extract tags from any sentence or paragraph.

```
[('1997', 'DATE'), [('ADNI', 'ORG'), ('second',
'ORDINAL'), ('Batty et al.', 'PERSON'), ('more than
50%', 'PERCENT'), ('Japan', 'GPE'), ('4', 'CARDINAL')]
```

Figure 4.3 shows the count of tagging labels in our provided text (tagging categories on x-axis and entities count on y-axis).



	DATE	ORG	CARDINAL	ORDINAL	GPE	PERSON	LOC	PRODUCT	MONEY	TIME	WORK_OF_ART	EVENT	PERCENT
■ Count	881	804	1024	347		228	52	7	22	47	15	9	443

Figure 4.3: Graphical Representation of Tagging Entities

From analysis we see that our datasets fall in 'ORG' tagging category, hence with aid of tag of 'ORG', we extracted string by further processing. Trendline also demonstrate that which category contributes more.

3. In third step of phase II, we used one of Question Answering (QA) huggingface BERT models called salti/bert-base-multilingual-cased-finetuned-squad which is to teach our model human sentiment and speech which gives us the best score, to get the prediction string.

4. In model, we passed the questions as query one by one over the sentences which are extracted in step 2, where the answer to each question is a excerpt of text from the matching reading sentence. The questions are like:

   i. Which datasets were included?

   ii. Which datasources are used?

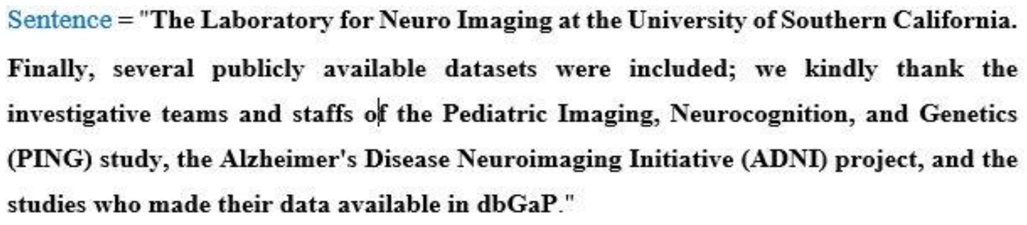
   iii. Which datasets are used?

iv. Which data are used?

v. Which samples are used?

vi. Which data are used in surveys?

For instance, if there is a sentence:

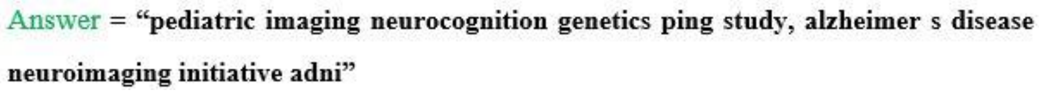


Sentence = "The Laboratory for Neuro Imaging at the University of Southern California. Finally, several publicly available datasets were included; we kindly thank the investigative teams and staffs of the Pediatric Imaging, Neurocognition, and Genetics (PING) study, the Alzheimer's Disease Neuroimaging Initiative (ADNI) project, and the studies who made their data available in dbGaP."

And question is:

Question = **Which datasets are used?**

Then after passing the above sentence over the define query/ queries of our QA model, we get the answer:



Answer = "pediatric imaging neurocognition genetics ping study, alzheimer s disease neuroimaging initiative adni"

5. Following the above step, we get the final list of answers or datasets obtained by combining different list obtained from different questions. Get the prediction string by matching these answers with strings obtained through SpaCy in second step exhibited in Table 4.5.

Table 4.5 Datasets Identified by Using BERT QA Model

Id	PredictionString
2100032a-7c33-4bff-97ef-690822c43466	alzheimer s disease neuroimaging initiative adni | framingham heart study | cardiovascular health study chs | norwegian cognitive neurogenetics ncng | pediatric imaging neurocognition genetics...
2f392438-e215-4169-bebf-21ac4ff253e1	current population survey cps | integrated postsecondary education data system ipeds | nces common core of data ccd | oecd national accounts database | pisa | pirls | progress in international reading literacy study | schools and staffing survey | timss | trends in international mathematics and science study……
3f316b38-1a24-45a9-8d8c-4e05a42257c6	slosh | noaa storm surge inundation | nps | sea lake and overland surges from hurricanes ….
8e6996b4-ca08-4c0b-bed2-aaf07a4c6a60	ces | cnp |efnse ne | fmi | iri | iri cnp | rural urban continuum codes | nielsen homescan survey….

Phase III: Acronyms & Their Abbreviations Extraction

1. In phase III we get datasets by extracting all abbreviations and their acronyms used in large cleaned text and
2. Matched them with abbreviations and their acronyms extracted in phase II as answers to form prediction strings. In this way, as shown in Table 4.6, we get datasets having their names both as acronyms and their respective abbreviations.

Table 4.6 Datasets Identified by Matching Acronyms & Abbreviations

Id	PredictionString
2100032a-7c33-4bff-97ef-690822c43466	alzheimer s disease neuroimaging initiative adni | adni | cardiovascular health study chs | chs | framingham heart study fhs | fhs | hbcs | helsinki birth cohort study hbcs | ncng |norwegian cognitive neurogenetics ncng | ping | pediatric imaging neurocognition genetics...
2f392438-e215-4169-bebf-21ac4ff253e1	ccd | common core of data ccd | cps |current population survey cps | integrated postsecondary education data system ipeds | ipeds | oecd | organization for economic cooperation and development oecd| pisa| pisa | program for international student assessment pisa | progress in international reading literacy study pirls | pirls |schools and staffing survey | timss | trends in international mathematics and science study timss……
3f316b38-1a24-45a9-8d8c-4e05a42257c6	national oceanic atmospheric administration noaa | national park service nps | noaa | slosh | noaa storm surge inundation | nps | sea lake and overland surges from hurricanes slosh | rise risk management study slrrms | sltms….
8e6996b4-ca08-4c0b-bed2-aaf07a4c6a60	ces | consumer expenditure survey ces | consumer network panel cnp | economic research service ers | ers | fmi | food marketing institute fmi | information resource incorporated iri | iri | rural urban continuum codes | nielsen homescan survey | nhs….

Aggregate Prediction Strings

After getting prediction strings as datasets from phase I, II and III respectively, now aggregate them by merging them into one dataframe, remove the duplicate datasets or values if exists and sort them alphabetically separated by pipe '| 'character.

Let us summarize the above proposed solution. This framework consists of NLP techniques and models. DEFNLP consists of three phases. In **Phase I** after data cleaning, we move one step forward to baseline modelling in which we predict the datasets by merging labels, dataset_title and dataset_label with the cleaned_label and match it with cleaned text. Meanwhile, we also match strings in big government dataset with cleaned text. In **Phase II**, we tokenize the large cleaned obtained in Phase I by chunking large text into sentences having the words of our interest such as data, dataset, datasets, sample, samples, survey, surveys etc. Extract acronyms and abbreviations used in sentences. Meanwhile, we extract majority of datasets by using salti/bert QA model. Phase III consists of different extraction techniques. We extract datasets by matching acronyms and their abbreviations of large cleaned text obtained in Phase I with the acronyms and their abbreviations obtained in Phase II. And finally, following the concept of transfer learning of machine learning we get the desired datasets by merging all the datasets obtained in three phases as prediction string in a dataframe. In this way we proposed the automated NLP solution to search desired information from large datasets from numerous research areas.

CHAPTER 5: EXPERIMENTS & RESULTS

5.1. EXPERIMENTATION:

In this chapter, we conduct ablation experiments on a variety of features of our DEFNLP framework to understand better their relative importance. The model is implemented in Kaggle repository utilizing Keras with TensorFlow on Central Processing Unit (CPU) backend having 13 GB RAM and 25 GB Disk Space maximum. The code was written in Python 3.9 with many packages including Numpy, pandas, csv, JSON, nltk, SpaCy, WordCloud, STOPWORDS, AbbrX etc.

Based on the evaluation of three phases, as shown in figure 5.1, we came across that in phase I, only 14 datasets are identified, the number is very less as compare to next phases. In phase II and III, 30 and 65 datasets are identified respectively. In phase III, our framework reached to the peak of extracting almost all the datasets used in publications, where on x-axis, phases and on y-axis, the number of datasets identified in each phase is shown.

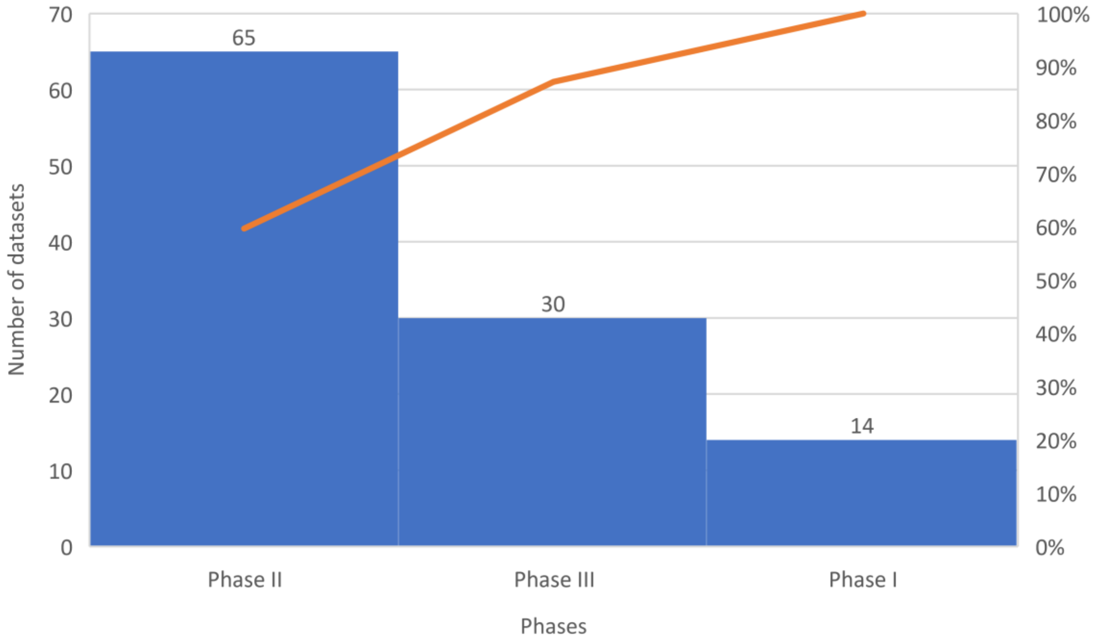


Figure 5.1: Datasets Stats Identified in Phase I, II & III

5.1.1. K-Means Visualization of Prediction Strings of All Phases

Let us proof and verified our results (shown in Figure 5.1) through K-means for further visualization and experimentation. I have always been a fan of utilizing graphs and charts to illustrate subjects, and it's generally helped me acquire a deeper understanding of what's going on with different algorithms. So, let's examine what K-means looks like after each iteration.

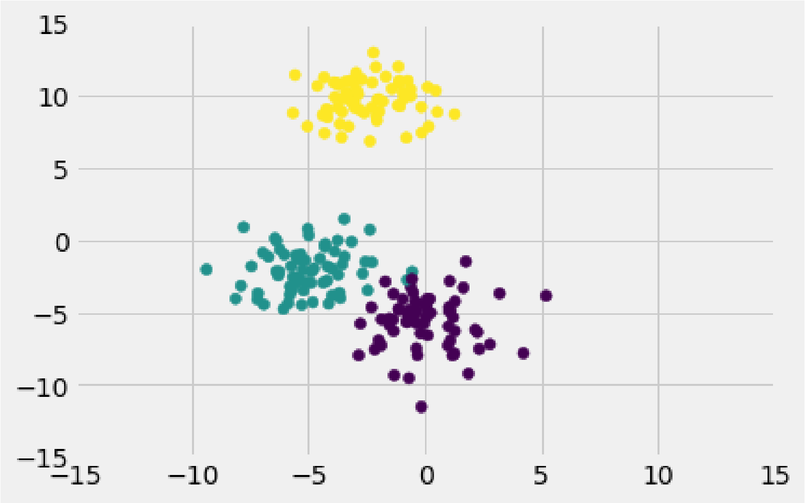


Figure 5.2: K-Means Algorithm

The figure: 5.2 above depicts K-means in action. We have set k = 3 to allocate data to one of three clusters at each iteration. First, relates to the centroids being initialized randomly. Second, allocate the data points to the cluster that is closest to them, and finally, assign new centroids based on the average of the data in each cluster. This will continue till we reach our halting point (minimize our cost function J or for a predefined number of iterations). Hopefully, the description above, together with the visualization, has provided a clear grasp of what K-means is doing in our experiment.

a) Elbow Method for Optimal Number of Clusters

When utilizing K-means, one of the most important things to remember is to select the appropriate number of clusters. If we use too little, we risk grouping data that have significant differences. If we have too many clusters, we will overfit the data and our conclusions will not apply efficiently. We will utilize the ***elbow method***, which is a typical strategy for this task, to answer this issue. We select a number where adding more clusters just significantly improves the score. When graphed as shown in the figure 5.3, the result clearly resembles an elbow (or upside-down elbow in this case). The optimum choice for the number of clusters is where the elbow forms, which is three in our instance. We could potentially play with four clusters, but for this implementation, we'll stick with three.

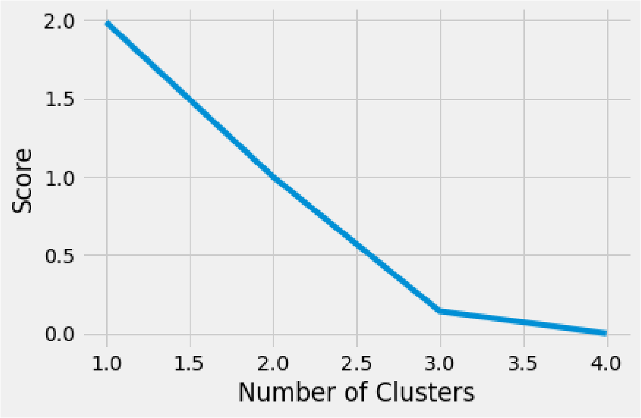
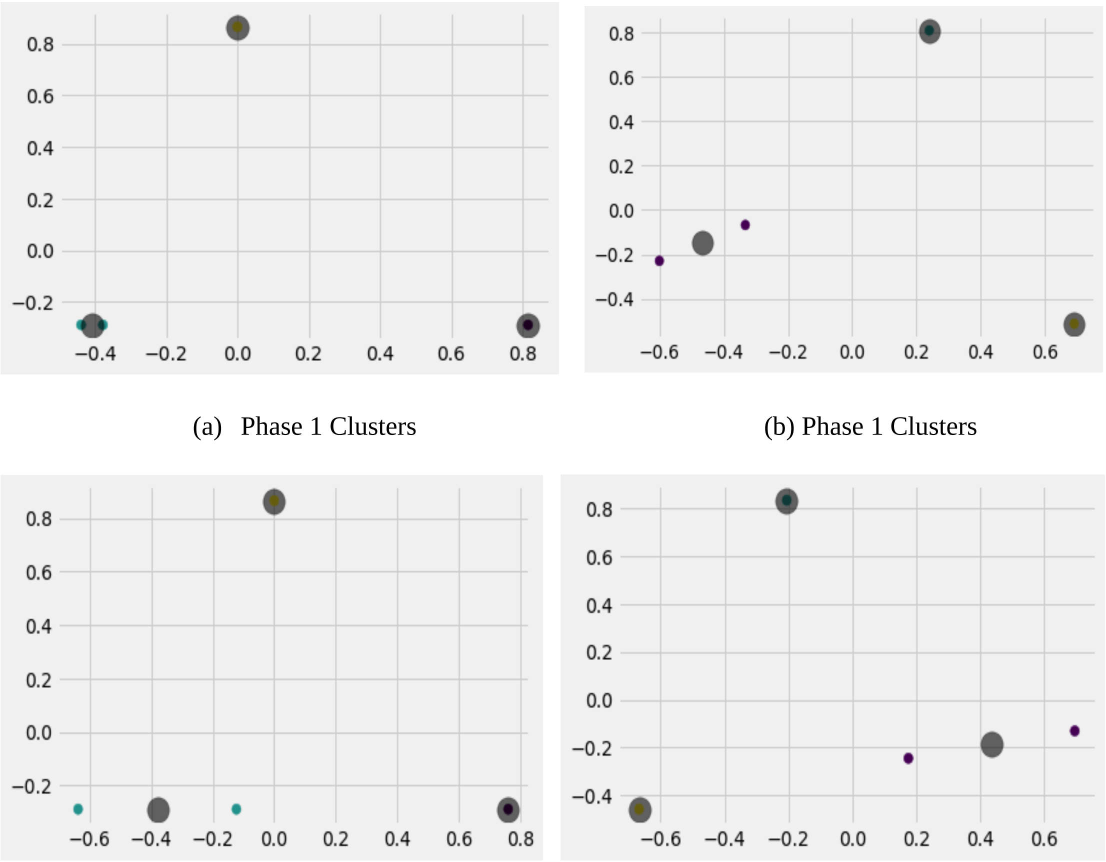


Figure 5.3: Elbow Method for Optimal Number of Clusters

b) PCA sklearn Clustering Results

To visualize our clusters graphically, we will use PCA to decrease the dimensionality of our feature matrix so that it can be shown in two dimensions. With that in mind, we select two components and convert our tf_idf_array using the PCA class's fit_transform () method. Then we construct an instance of our K-means class, selecting three clusters based on the results of our previous research. It's now only a matter of using the fit_kmeans () and predict () methods to sort our data into clusters. Because we projected our array into a 2-d dimension, we can easily see it together with the cluster centers using a scatter plot.

We can observe three different clusters here, with notably significant divergence between all three clusters, indicating a significant variation in the content of the datasets. As shown in figure 5.4, the bulk of the data is included inside the blue cluster in (a), however there is a significant separation for the purple cluster in (b), as well as for the blue cluster in (c) and (d) after merging prediction strings from all phases.



(a) Phase 1 Clusters

(b) Phase 1 Clusters

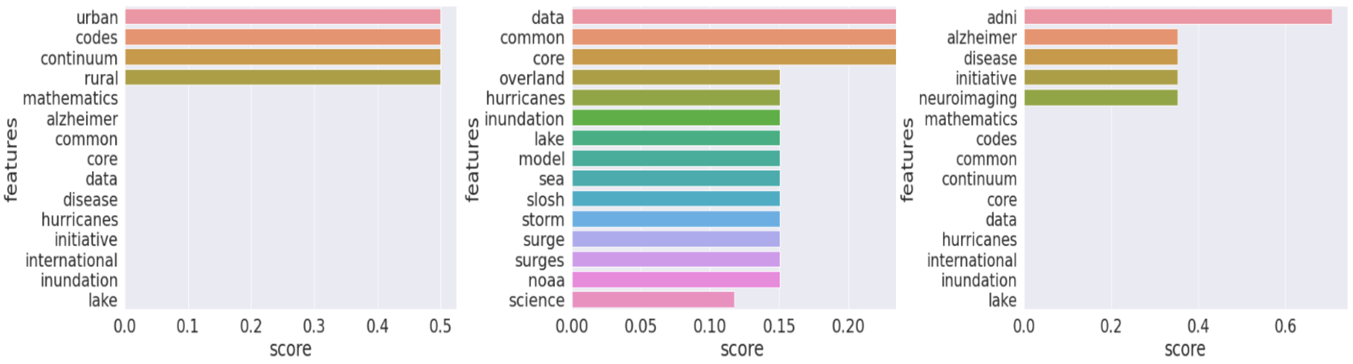
(c) Phase 3 Clusters

(d) Aggregate Predictions Clusters

Figure 5.4: PCA sklearn Clustering Results

c) Extracting Top Features from Each Cluster

The three graphs below correspond to the top 15 words in each cluster in each phase, ordered by TF-IDF relative significance. Ok so, what are figures 5.5 (a), (b), and (c) attempting to tell us? Is there anything interesting features that stands out? In generally; (a) ***Cluster 0*** appears to have quite a few words of datasets or prediction strings which could be quite important. Immediately, we could start to look at datasets having words urban, codes, continuum and rural and see if there is any interesting content. (b) ***Cluster 1*** seems to generally be about more of datasets with features like data, hurricanes and science. Again, this may be quite helpful in narrowing down which types of datasets we want to investigate further in publications.



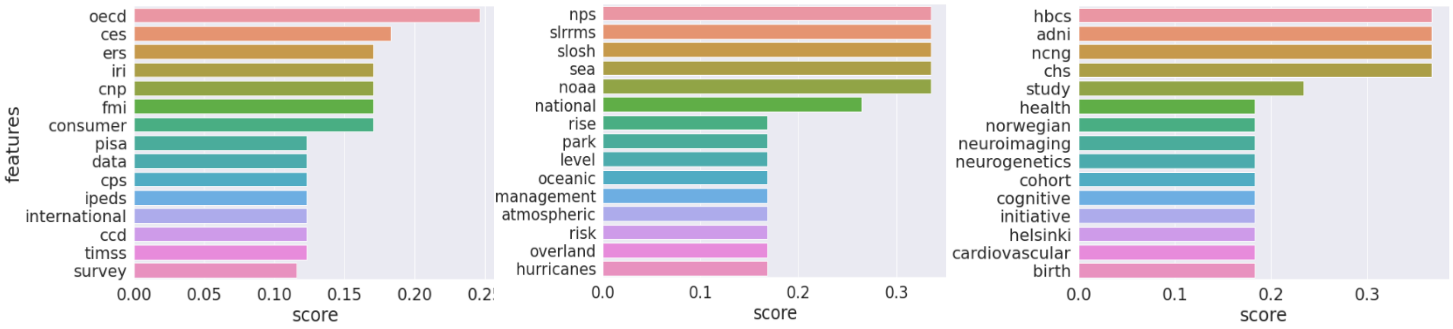
(a) Cluster 0

(b) Cluster 1

(c) Cluster 2

Figure 5.5: Phase I Clustering Results

Cluster 2 seems to have a lot of words suggesting the prediction strings were from diseases like adni, neuroimaging etc. Although this does not appear to be immediately interesting, it may be worth further examination. Below is the full visualization based on experimentation of clustering of rest of phases.



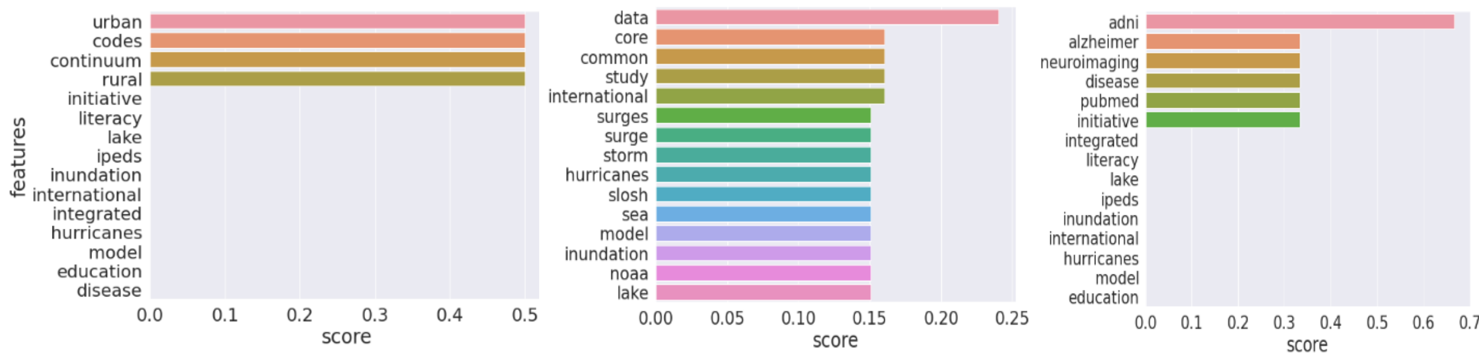
(a) Cluster 0

(b) Cluster 1

(c) Cluster 2

Figure 5.6: Phase II Clustering Results

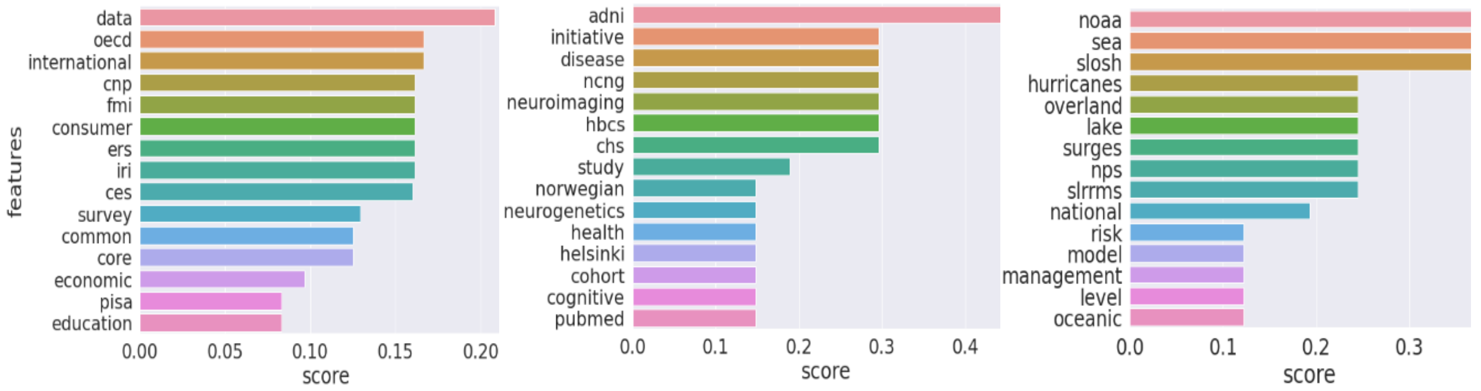
In phase II clustering experimentation, results are much better as shown in figure 5.6 (a) ***Cluster 0*** appears to have words of datasets or prediction strings like international, education, consumer and survey. (b) ***Cluster 1*** seems to have content like sea, atmospheric and hurricanes and (c) ***Cluster 2*** seems to contain content of diseases and medical.



(a) Cluster 0 (b) Cluster 1 (c) Cluster 2

Figure 5.7: Phase III Clustering Results

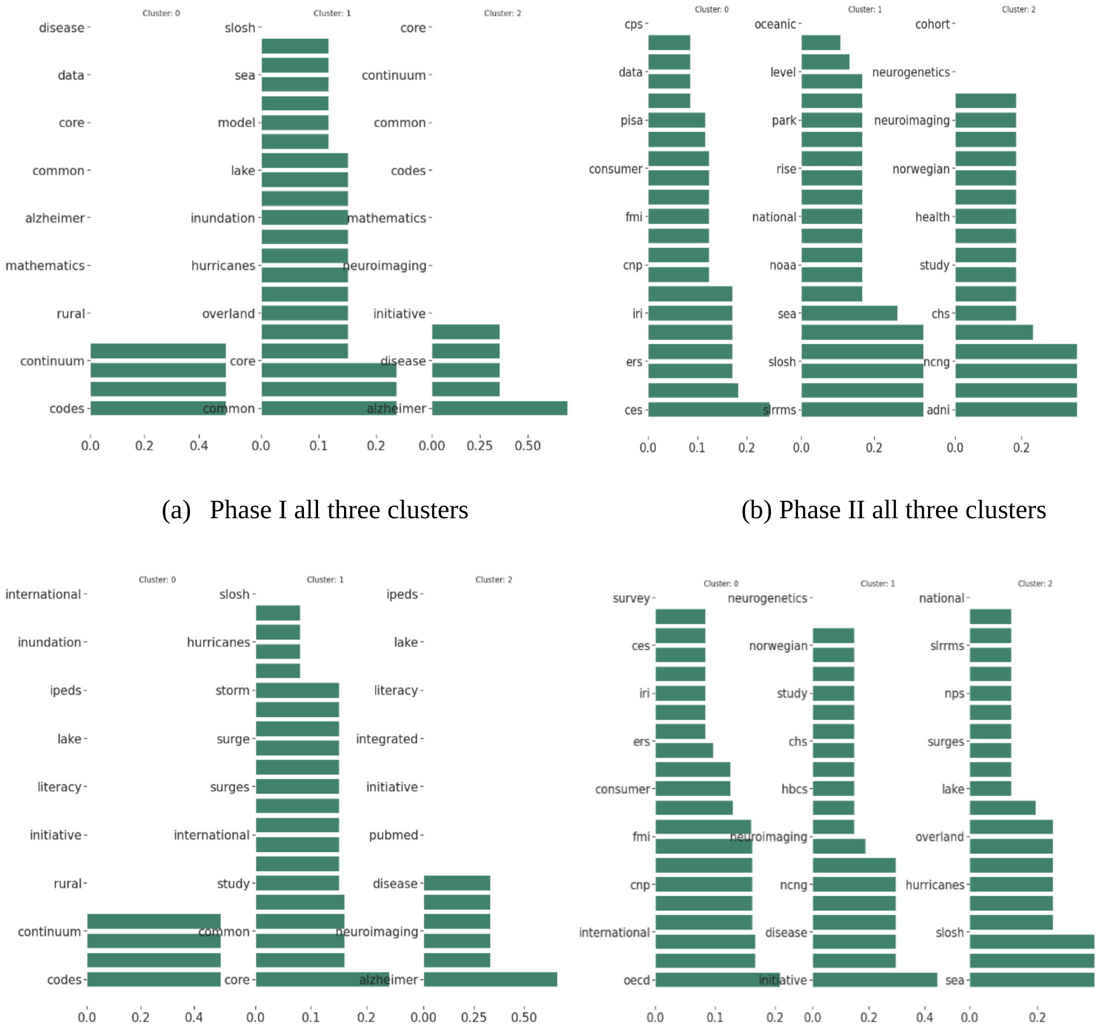
As shown in Figure 5.7; (a) ***Cluster 0*** appears to have words of datasets or prediction strings like rural, urban. (b) ***Cluster 1*** seems to have content like sea, atmospheric and hurricanes and (c) ***Cluster 2*** seems to contain content of diseases and medical**.** Phase III clustering experimentation, results are better in comparison to phase I but not much satisfactory as phase III results are. The reason is that it contains the datasets consists of acronyms and abbreviations only while in phase III, we extracted proper logical answers to our questions.



(a) Cluster 0 (b) Cluster 1 (c) Cluster 2

Figure 5.8: Datasets Aggregation Clustering Results

Aggregating clustering results outperforms from all phases clustering as shown in figure 5.8; (a) ***Cluster 0*** appears to have words of datasets or prediction strings like international, education, consumer and survey. (b) ***Cluster 1*** seems to have content like diseases and medical study and (c) ***Cluster 2*** seems to contain content of sea, atmospheric and hurricanes**.**



(a) Phase I all three clusters

(b) Phase II all three clusters

(c) Phase III all three clusters

(d) Aggregate clustering of all Phases

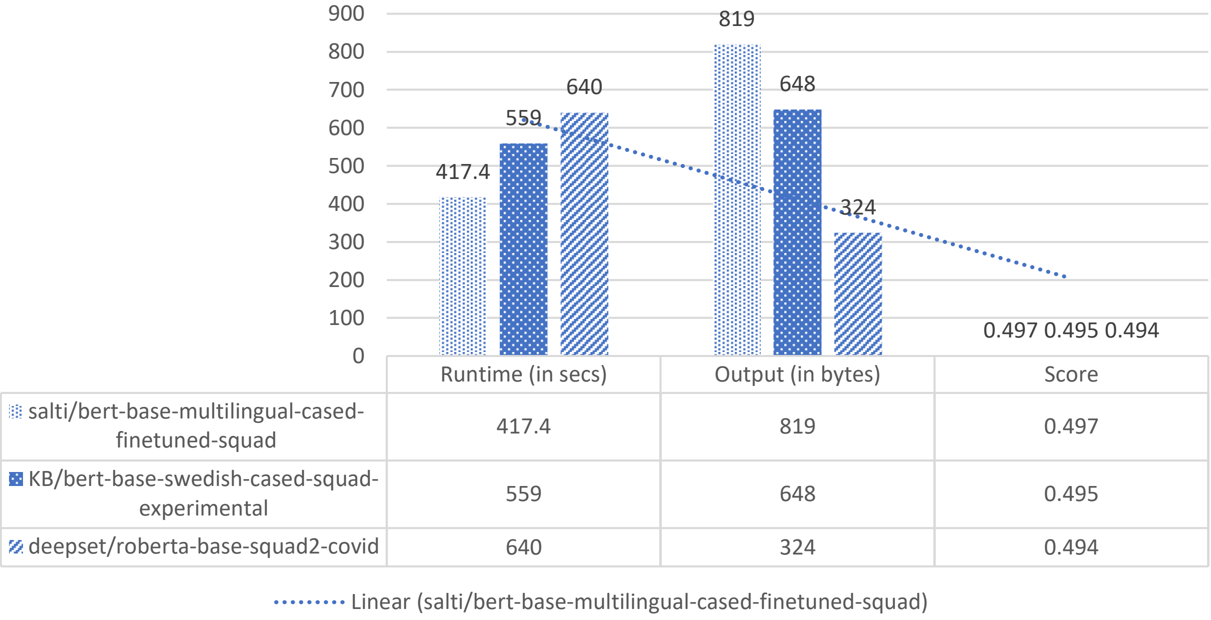
Figure 5.9: Aggregation Clustering Results

Figure 5.9 compares the clusters within each phase to figure out the features of the publications. So, the turnout of this experiment and visualization with our prediction strings is that the four publications in our test data having a dataset which constitutes of content of atmospheric & education (shown in (a) and (d)), atmospheric, consumption, medical and education (shown in (b) and (d)) and atmospheric (shown in (c) and (d)).

5.2. Individual Performance of BERT QA Models

After testing almost all the huggingface QA models, we came across that BERT models which perform comparatively better to our problem and dataset as compare to others. Ablation analysis also shows that various models having greater maximum sequence length and batch size does not always imply greater performance on cross-domain benchmark tasks.

If we compare BERT models within its other models, salti/bert-base-multilingual-cased-finetuned-squad achieve quite better performance as shown in Figure 5.10 with the trending line along with runtime of 417.4 secs and output of 819B. Whereas other two QA models (i.e., KB/bert-base-swedish-cased-squad-experimental and deepset/roberta-base-squad2-covid) takes more execution or run time i.e., 559s and 640s respectively with the output size of 648B and 324B respectively.



	Runtime (in secs)	Output (in bytes)	Score
░ salti/bert-base-multilingual-cased-finetuned-squad	417.4	819	0.497
▦ KB/bert-base-swedish-cased-squad-experimental	559	648	0.495
▨ deepset/roberta-base-squad2-covid	640	324	0.494

Figure 5.10: Individual Performance of BERT QA Models

5.3. Performance Metrics of BERT Models Under DEFNLP

For evaluation of performance of different QA models or architectural variants, we use different QA bert huggingface models but **salti/bert-base-multilingual-cased-finetuned-squad** outperforms from all under DEFNLP as shown in Table 5.1.

Table 5.1 Ablation Analysis of Different BERT QA Models Under DEFNLP

BERT Models	**Baseline Model**	**QA Model**	**Govt Dataset**	**LB Score**
salti/bert-base-multilingual-cased-finetuned-squad	**0.022**	**0.497**	**0.135**	**0.654**
KB/bert-base-swedish-cased-squad-experimental	0.022	0.495	0.135	0.652
deepset/roberta-base-squad2-covid	0.022	0.494	0.135	0.651

The score of salti bert QA model is 0.497, highest than other QA models which definitely raise the leaderboard score i.e., 0.654. Its computation time to answer each query on CPU is 0.0696 s and suitable hyperparameters for our dataset as maximum answer length is 64, greater batch size as well as learning rate which is again far better than other QA models.

CHAPTER 6: CONCLUSION & FUTURE WORK

Takeaway is that **DEFNLP** help, support evidence in government data to make wiser, more transparent public investments. Automated NLP approaches enable government agencies and researchers to quickly find the information they need, no matter how large data is, with high speed of execution in seconds. The proposed approach could be used to develop data usage scorecards to better enable agencies to show how their data are used and bring down a critical barrier to the access and use of public data. This paper also introduced the new dataset "Coleridge Initiative- Show Us the Data," which contains 14316 examples of research publications or papers will foster new research in search or QA. The limitation is that **DEFNLP** is used for English language. We could also use this framework for other languages which means it can be extended to support other human languages with no code changes by using different QA linguistic models but it might affect the accuracy of searching results because in our case dataset consist of English language. If the query is poorly worded or confusing, the system may be unable to provide the appropriate answer. Hence, we are working on Complex Question Language in order to achieve the best, exact and proper results without an ambiguity. Lacks user interface and capabilities that allow users to engage with the system further. Hence, also engage in making DEFNLP more user friendly and attractive, so that in one click, user could achieve the desire results in seconds. Proposed methodology is quite flexible which demonstrates that it provides long-term maintenance in the future to address issues and fulfil new requests as compared to other benchmark transfer learning models used to extract desired information.

The question comes up in mind that why we divided this process of extracting information into three phases? The reason is that, by using simple string-matching method, we googled the existing dataset, and cherry-picked the datasets which we think are likely to be contained in test data. Although the amount was small (about 2kB). We can make a huge lift-up in public leaderboard using this external dataset but again at the end we are not able to search each and every dataset because most of datasets are not listed in external dataset. So, we decided to include string matching method in phase I as a part of proposed framework. So in order to get the possible predictions, and not to miss any information which we need, we split up the process in three phases. Speed of execution of **DEFNLP** is better as compare to other models. In terms of timing and performance, each epoch took around 5 minutes on average on a computer with an output size of 3.27kB. Our framework also intuits that a larger model size does not always imply greater performance on a cross-domain benchmark task.

REFERENCES

1. Kaggle – Coleridge Innitiative: Show US The Data. URL: https://www.kaggle.com/c/coleridgeinitiative-show-us-the-data (accessed on May 16, 2021)
2. Bea – Bureau of Economic Analysis. https://www.bea.gov/evidence (accessed on May 16, 2021)
3. Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon, "Evaluation of Bert and ALBERT Sentence Embedding Performance on Downstream NLP Tasks", 2020 25th International Conference on Pattern Recognition (ICPR), arXiv:2101.10642v1 [cs.CL] 26 Jan 2021.
4. M Zaheer et al., "Big Bird: Transformers for Longer Sequences", arXiv preprint arXiv:2007.14062, 2020.
5. Iz Beltagy, Matthew E. Peters, Arman Cohan., "Longformer: The Long-Document Transformer", arXiv:2004.05150, 2020.
6. K. Masuda and T. Matsuzaki, "Semantic Search based on the Online Integration of NLP Techniques," vol. 27, no. Pacling, pp. 281–290, 2011.
7. Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection. In The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)
8. I. Annamoradnejad, M. Fazli, and J. Habibi, "Predicting Subjective Features from Questions on QA Websites using BERT," in 2020 6th International Conference on Web Research, ICWR 2020, 2020, doi: 10.1109/icwr49608.2020.9122318.
9. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392.
10. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert:Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
11. Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In ICLR.

12. Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In ICLR.

13. Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno and Julian Martin Eisenschlos. 2020. TAPAS: Weakly Supervised Table Parsing via Pre-training. arXiv preprint arXiv:2004.02349 (2020).

14. Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled
version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108, 2019.

15. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

16. Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and
Wen-tau Yih. Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906, 2020. URL https://arxiv.org/abs/2004.04906.

17. Kaggle.URL:https://www.kaggle.com/mlconsult/bigger-govt-dataset-list (accessed on May 16, 2021)

18. Chorus – Advancing Open Access to Open Research. URL: https://www.chorusaccess.org/ (accessed on May 16, 2021)

19. CIO.GOV URL: https://www.cio.gov/policies-and-priorities/evidence-based-policymaking/ (accessed on May 16, 2021)

20. THE WHITE HOUSE URL: https://www.whitehouse.gov/briefing-room/presidential-actions/2021/01/27/memorandum-on-restoring-trust-in-government-through-scientific-integrity-and-evidence-based-policymaking/ (accessed on May 16, 2021)

21. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683, 2019.

22. Daniel Keysers, Nathanael Sch ̈arli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In International Conference on Learning Representations.

23. Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K ̈uttler, Mike Lewis, Wen-tau Yih, Tim Rockt ̈aschel, Sebastian Riedel, and Kiela Douwe.Retrieval-augmented generation for knowledge-intensive nlp tasks. arXiv preprint arXiv:2005.11401,2020.

24. Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language

understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf, 2018.

25. Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading wikipedia to answer open domain questions. arXiv preprint arXiv:1704.00051.

26. Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In Empirical Methods in Natural Language Processing (EMNLP). pages 1400–1409.

27. Bordes, Antoine, Usunier, Nicolas, Chopra, Sumit, and Weston, Jason. Large-scale simple question answering with memory networks. arXiv preprint arXiv:1506.02075, 2015.

28. Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015a. Towards AI-complete question answering: A set of prerequisite toy tasks. arXiv:1502.05698.

29. Tao Lei, Yu Zhang, Sida I. Wang, Hui Dai, and Yoav Artzi. Simple recurrent units for highly

parallelizable recurrence. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMzhou 2018, pages 4470–4481.

Association for Computational Linguistics, sep 2018. ISBN 9781948087841. doi: 10.18653/v1/d18-1477. URL http://arxiv.org/abs/1709.02755.

30. Hoo-Chang Shin, Yang Zhang, Evelina Bakhturina, Raul Puri, Mostofa Patwary, Mohammad Shoeybi, and Raghav Mani. 2020. BioMegatron: Larger biomedical domain language model. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4700–4706, Online. Association for Computational Linguistics.
31. Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. OpenNRE: An open and extensible toolkit for neural relation extraction In Proceedings of EMNLP-IJCNLP, pages 169–174.
32. Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. arXiv preprint arXiv:1905.03197, 2019.
33. Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. arXiv preprint arXiv:2002.08909, 2020.
34. Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking the world state with recurrent entity networks. In ICLR, 2017.
35. Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. arXiv preprint arXiv:1503.08895.
36. Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. In Proceedings of the 8th International Conference on Learning Representations (ICLR 2020).
37. Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. In arXiv:1901.08746.

38. Stefan Schweter and Alan Akbik. 2020. Flert: Document-level features for named entity recognition. arXiv preprint arXiv:2011.06993

39. L. Weber, M. S¨anger, J. M¨unchmeyer, M. Habibi, U. Leser, Hunflair: An easy-to-use tool for state-of-the-art biomedical named entity recognition, arXiv preprint arXiv:2008.07347 (2020).

40. Guillaume Lample, Miguel Ballesteros, Sandeep Sub- ramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In NAACL-HLT.

41. Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In ACL.

42. Guillaume Lample, Miguel Ballesteros, Sandeep Sub ramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In NAACL-HLT.

43. Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs- CRF. In ACL.

44. Ritter, A., M. Sam Clark, and O. Etzioni. 2011. Named entity recognition in tweets: An experimental study. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP ’11, Association for Computational Linguistics, Stroudsburg, PA, pp. 1524–1534.

45. Kocaman, V., Talby, D.: Biomedical named entity recognition at scale. arXiv preprint arXiv:2011.06315 (2020).

46. Zhixiu Ye and Zhen-Hua Ling. 2018. Hybrid semi- markov crf for neural sequence labeling. In Proceedings of ACL, pages 235–240.

47. Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019a. A unified mrc framework for named entity recognition. arXiv preprint arXiv:1910.11476. .

48. A. Lê, “A deep neural network model for the task of named entity recognition,” International Journal of Machine Learning and Computing, vol. 9, 02 2019.

49. Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2660–2670, 2017.

50. Juntao Yu, Bernd Bohnet, and MassimoPoesio. 2020. Named entity recognition as dependency parsing. arXiv preprint arXiv:2005.07150.