Knowledge Discovery and Decision Support Systems Using Natural Language Processing in Applications for Societal Good

---

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Mojtaba Heidarysafa

Mar

2023

Approval Sheet

This Dissertation is submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Systems and Information Engineering)

Mojtaba Heidarysafa

This Dissertation has been read and approved by the Examining Committee:

Laura Barnes, Ph.D. (School of Engineering and Applied Science)

Donald Brown, Ph.D. (School of Engineering and Applied Science)

Michael Porter, Ph.D. (School of Engineering and Applied Science)

Rafael Alvarado, Ph.D.(School of Data Science)

Peter Alonzi , Ph.D. (School of Data Science)

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, Dean, School of Engineering and Applied Science
Feb 2023

Abstract

Artificial Intelligence (AI) is becoming a crucial innovation that significantly impacts our everyday life. Not only have commercial AI applications improved our daily experiences, but AI has also proved to be beneficial in social domains such as healthcare, transportation, and education. As an essential sub-field of AI, Natural Language Processing (NLP) has great potential to benefit such social domains. In particular, many social domains have textual information that different NLP techniques can leverage to help individuals generate insights or aid in quick decision-making. This study presents several different NLP approaches that can provide societal benefits in three problem areas: namely, transportation and safety, security and counter-terrorism, and labor market gaps.

In the first application, we focused on short domain-specific types of texts in the field of transportation and safety. We used accident reports' narratives, a free text field, to identify the causes of accidents. We used two deep-learning architectures, Recurrent Neural Networks and Convolutional Neural Networks, combined with two word embeddings (Word2Vec and GloVe) to train a model capable of identifying any accident's cause based on the accident's narrative. Such a system can be used as a decision-support tool for evaluating accident reports.

In the security and counter-terrorism domain, we used NLP to analyze ISIS's propaganda approach to women and compared it with the approach from a non-violent religious group for women. We collected the relevant texts by using web-scraping and optical character recognition (OCR), and used an unsupervised learning method for analyzing the texts. Furthermore, we used emotion analysis to

check for the emotional aspects of these documents.

Finally, to address the skill gap in data science-related jobs, we collected a large corpus of online job advertisements and used the embedding vector space of the advertised skill terms and phrases in a semi-supervised approach in order to find the hard skills that the jobs required and extrapolate them from these documents. We also presented a complete framework for analyzing skills in the U.S. that allows individuals and organizations to understand the job market.

Acknowledgements

I would like to especially thank the members of my dissertation committee, who have helped this endeavor with their insightful feedback, guidance, time, and support. I would like to express my sincere gratitude to my advisor Prof. Brown for his constant help and support during my study at UVA and through all difficulties, I faced during that period. Finally, I would like to thank my mother for always encouraging and inspiring me through my life despite all the inconveniences for her.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the past decades, innovation in Artificial Intelligence (AI) rapidly changed the world and the way we interact with the world. AI introduced itself as a disruptive force in our world and had massive influences both on the economy and society [5]. Although the majority of AI-driven products have mainly been developed by industries, their potential to help society has not been neglected by both researchers and organizations. This includes the usage of different sub-fields of AI such as computer vision, machine learning, natural language processing, etc., to benefit society in different areas. Society can benefit from AI-powered systems in criminal justice and policing [6], education [7], healthcare [8] and even climate change [9], to name a few.

As previously mentioned, Natural Language Processing (NLP) is one of the AI sub-fields that can also benefit our societies. In fact, the famous "Turing Test", one of the earliest proposals of true artificial intelligence, is an NLP problem [10]. While NLP is one of the most challenging areas of AI due to the complexity of human languages, the abundance of information in many domains makes it a promising field to improve our experience in our future society. This was the driving force to successfully apply NLP techniques to more prominent domains with important public impact, such as healthcare [11], education [12], and social media [13].

Besides these noted domains, other less prominent domains with public benefit received less attention from researchers, who could nonetheless use NLP techniques in those fields to help shape a better society. In reality, a significant portion of the world's knowledge is still stored in plain text, but effectively utilizing it in this format presents a significant challenge. This particularly impacts the public domains that are not very well explored due to the challenges of mapping their problems to an appropriate NLP solution. Furthermore, in many such domains, access to high-quality text is also challenging and adds to the problem's difficulty.

This dissertation presents several use cases of applying NLP methods in such public domains in order to benefit society by providing insights into several different public areas. Various NLP methods have been developed depending on the nature of the problem, and each approach's result is discussed in this work. In terms of machine learning approaches, both supervised and unsupervised approaches have been used throughout this work as well as a semi-supervised approach that is based on self-learning algorithms in NLP. Regarding public domains, we selected three: transportation, security and terrorism, and the labor market. Finally, we present a complete system that uses textual information and provides insights for people about the labor market, and is available as an online platform for users. The rest of this dissertation is organized into five chapters including a background in Chapter 2 where we discuss AI applications that are beneficial for society and explain how NLP impacted important public domains. We also discuss the different abilities of NLP techniques and their constraints, as well as machine learning frameworks related to them.

In Chapter 3, we present a supervised NLP-based model in the transportation domain is presented. In this chapter, railway accident reports were used as the input for multiple deep learning models to predict the cause of accidents. The aim of this work is to investigate the potential of NLP to help identify and label accidents quickly and improve safety measures by providing an AI-assistant model for automated report analysis.

Chapter 4 discusses another application of Natural Language Processing in security and counter-terrorism. We analyzed ISIS publications targeting women and contrasted them with another non-violent religious group's approach to women. To make this comparison, we extracted articles from ISIS magazines related to women and a Catholic women's forum, then used unsupervised Natural Language Processing techniques to extract the topics. Furthermore, we evaluated an algorithm for emotion detection in the audience of discourse. Our findings suggest that such algorithms are able to detect emotional patterns that could be used for automated identification of emotionally rich contexts to significantly improve current counter-terrorism measures.

Chapters 6 and 5 present a system for analyzing labor market information based on online job advertisements. The proposed system collects and processes textual information, then presents it in an online application available for users who are interested in investigating the job market for data science-related jobs in the United States. The system attempts to fill the skill gap between industry and the workforce by clearly presenting the required skills in different fields of data science. By doing so, we can help individuals as well as organizations that are trying to hire applicants with the right skill sets. To do this, we used a semi-supervised approach to extract skills using job advertisements' descriptions. Such a model can also help experts to identify skills automatically using an AI-assisted product while reducing the time needed to build taxonomies for skills. This is another way that such NLP models can help build a better workforce, and as a result, positively impact society.

Finally, in Chapter 7 we bring this dissertation to the end by discussing conclusions and future works and limitations.

Chapter 2

Background

Artificial intelligence altered our world significantly in recent years. It is hard to imagine doing our daily activities without interacting with some AI-powered applications. From virtual assistants on our smartphones to emails that we receive, entertainment recommendations, music or movie suggestions, and search results, we use AI algorithms to improve our lives. The rise of neural networks in the context of deep learning models, combined with high computation powers and the enormous amount of data generated daily, makes AI a promising avenue for many businesses. Moreover, it is expected that 45% of economic gain by 2030 will be generated by advanced AI-powered technologies and the improvement they bring [14]. We briefly introduce the areas where AI has the most growth in businesses with commercial and societal impact.

2.1 Artificial Intelligence's Impact on Society

Many industries have already adopted AI in some capacity to enhance their products and services. Although the list of industries that use AI is extensive, some major areas have been the subject of using AI more extensively than others, which will be briefly discussed here.

- **AI in Finance**: Many AI techniques have been applied to different problems in the finance domain. Examples range from fraud detection to portfolio

management, risk assessment, and even algorithmic trading [15]. However, the effect of most such achievements on society is not direct.

- **AI in Retail and Advertising**: Retail and online retail businesses are another major industry that uses AI. AI is not only used to increase sales and efficiency but also to improve customer experiences. AI can help retailers by giving personalized recommendations and advertisements to customers, as well as enabling customer segmentation so that retailers can best understand how and when various groups are apt to purchase their products or services. Retailers can also benefit from AI-powered customer support when they use chatbots, in addition to gaining other benefits from using AI, such as price optimization and demand forecasting [16]. Such uses might have a direct positive impact on society due to improvement in customer satisfaction.

- **AI in Manufacturing and Logistics**: Today, companies use AI to enhance their manufacturing processes. AI allows companies to improve efficiency by better scheduling and planning, monitoring and maintenance, inventory management, etc. Aside from producing better products, however, these applications generally have minimal impact on society.

Besides these areas, many other AI applications directly impact society. The impact of AI on healthcare has already been very promising. From medical image classification to NLP knowledge discovery and decision-making systems built on large amounts of medical documents, AI has improved the work and performance of healthcare systems. Other areas, such as protecting the environment and public sectors, have also used AI extensively in recent years. It is difficult to describe all the possible use cases of AI that help our society. Still, a recent study by McKinsey presented over 160 use cases of AI over different areas with social benefits as shown in Fig 2.1 [1].

Based on this research, healthcare, environment, and crisis response were among the domains with the highest use cases for a potential AI solution. How-

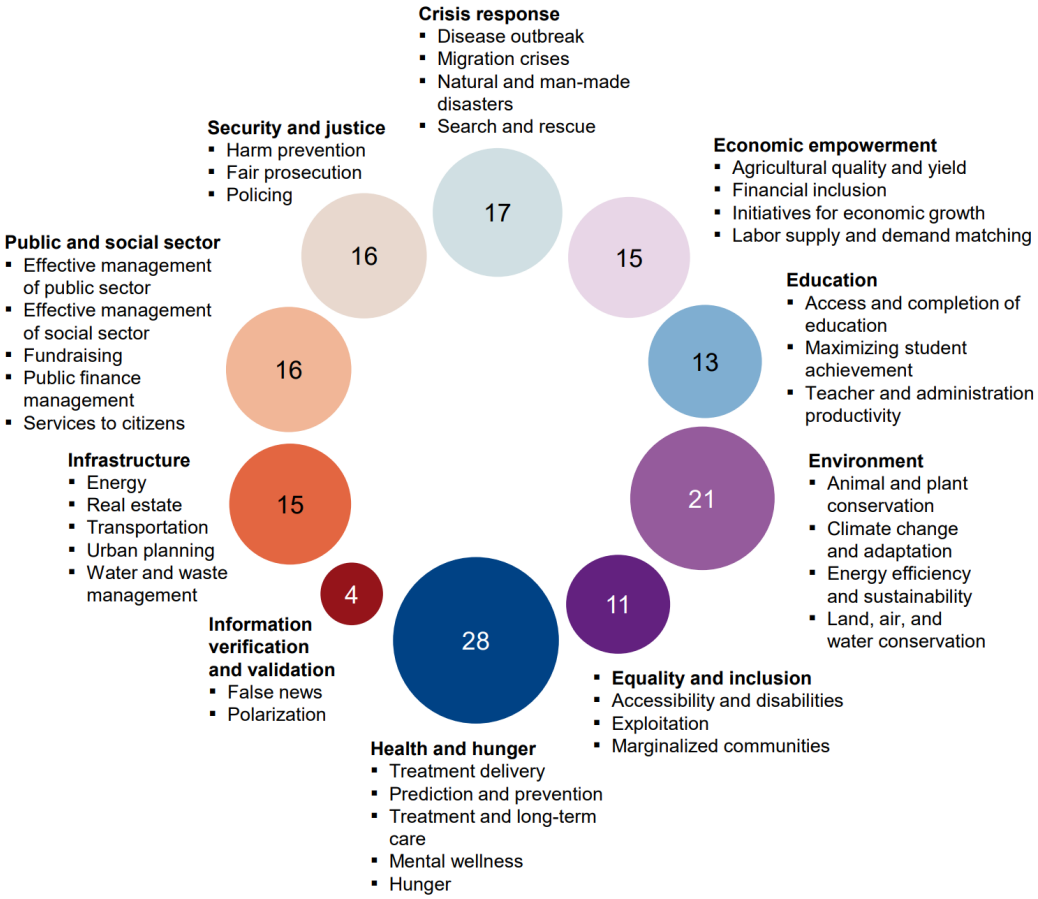

Figure 2.1: Distribution of AI related use cases across different domains [1]

ever, deploying a real AI-powered solution in all these domains is still challenging. For example, despite the potential in areas such as "Security and Justice", "Infrastructure" and "Economic Empowerment", very few AI solutions were actually developed based on this report. This highlights the value of attempts to bring AI-powered solutions to these domains. Moreover, the authors summarized their findings by stating that two capabilities of AI, computer vision and Natural Language Processing, can be applied to a wide range of challenges in social domains. This is due to the complex nature of unstructured data and its automatic processing. Thus, these two capabilities of AI are exciting research areas with high societal impact. We will discuss the capabilities of AI and the importance of NLP in AI in general in the next section.

2.2 Natural Language Processing as a Major AI Capability

AI applications could range from machine learning and predictive modeling to autonomous cars and robotics. Therefore, a large group of applications and capabilities contribute to AI. However, we could organize most of these applications into six major capability categories:

Figure 2.2: Major AI capabilities

- Perception: This is the ability to understand the world that a machine interacts with. Usually, this includes computer vision, where a machine processes and understands images and videos. It could also process information through sensors to understand the world.

- Speech and audio processing: The machine uses audio to understand the world. An essential task in this field is speech recognition, where the machine tries to map the audio it receives into meaningful text for later usage.

- Natural Language Processing: To interact with humans, computers must understand and process human language. The ability to understand and interpret human language and process written texts is another central AI capability, and many of the challenges of true intelligence fall in this category.

- Reasoning: Another essential capability of AI is the ability to perform reasoning and solve complex problems. Solving puzzles and logical conclusions using the high computation power of computers make this ability an exciting feature for AI.

- Knowledge Representation: This ability allows AI to present information better and infer new knowledge based on the information it has. Graph models and ontologies are examples of knowledge representation.

- Planning: Planning refers to mimicking humans' guessing ability and being creative in deciding. AI learns to perform sequences of action to achieve a goal.

Looking at these categories, the importance of Natural Language Processing as a sub-field of AI becomes quite clear. Not only is NLP a direct main category of AI, but also categories such as speech recognition and knowledge representation have a close relationship with language and text. Therefore, understanding NLP's capabilities will be crucial to AI and its applications that benefit society. In the rest of this chapter, we focus on explaining these capabilities and some of their applications in social domains.

2.3 NLP's Capabilities for Knowledge Discovery and Decision Making

Natural Language Processing's toolbox consists of many techniques to use text for a variety of tasks, each with a different purpose. The range of techniques that contribute to the NLP toolbox is vast, and it covers areas as diverse as

grammatical parsing and spell-checking to very advanced tasks such as human dialogue generation and speech recognition. Table 2.1 shows the main tasks that NLP contribute to solving them.

Table 2.1: Main NLP tasks

Fundamental Tasks	**Advanced Tasks**
Tokenization	Information Extraction
Part-of-speech tagging	Paraphrase and text similarity
Named entity recognition	Text Summarization
Coreference resolution	Text-to-speech
Word sense disambiguation	Speech-to-text
Document classification	Dialogue systems
Document Ranking	Question and answering
Sentiment/emotion analysis	Image captioning
Relation extraction	Text generation
Text parsing	Machine translation

While the tasks presented in this table address different problems, not all of them can be mapped to real-world problems in many domains, such as societal ones. However, a group of these techniques has proven beneficial for real-world applications, and we will discuss those techniques here.

2.3.1 Document Classification

The problem of document classification is well-studied and can be applied to many real-world applications. Here, the objective is to put documents in different categories automatically. Such a technique can help a range of applications, from spam email filtering to product review analysis. Proper mapping of a real problem in societal domains using this technique allows for incorporating automation into a helpful system. For example, processing hospital notes on patients could automatically flag patients with dangerous cancer stages among all the reports and help provide quicker actions for those patients. Fig 2.3 shows an overview of the text classification problem framework. After cleaning texts, the first step in document classification is feature extraction. We turn the textual information into a mathematical representation to be used in conjunction with a classification algorithm. This representation usually is in one of the following forms:

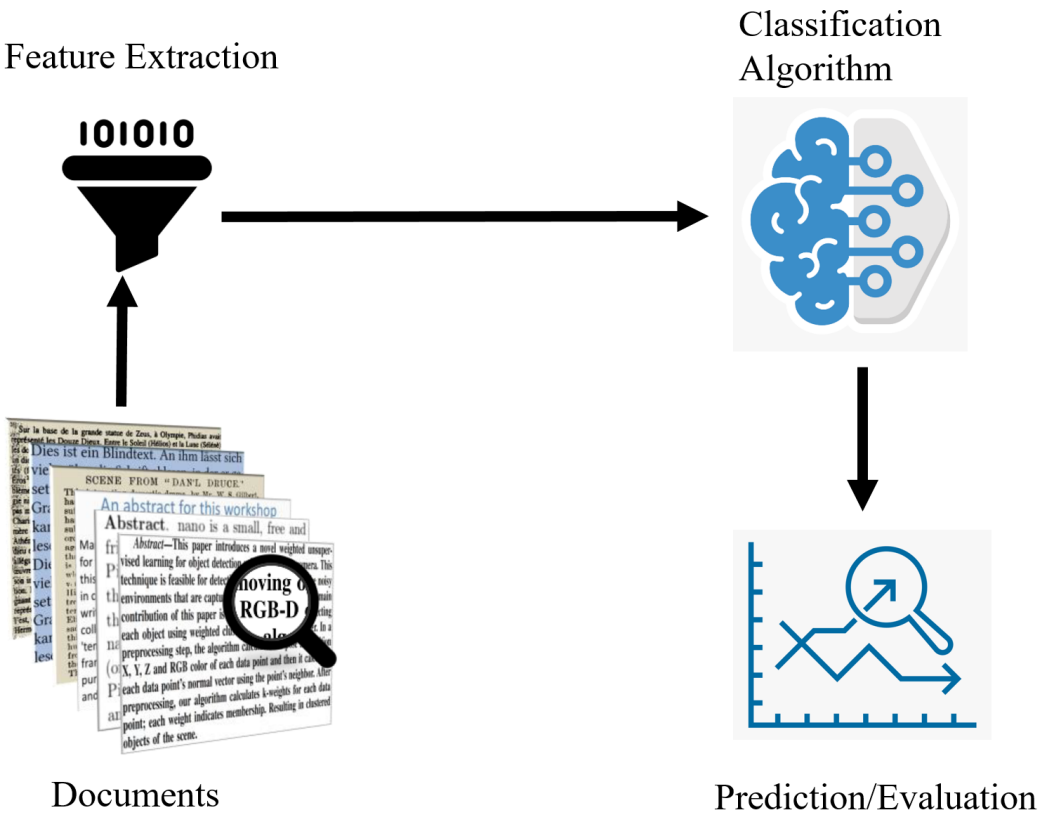

Figure 2.3: Overall Framework for Document Classification

- Term Frequency (tf): It is the frequency count of each word in the documents and presents each document as a vocabulary size vector of numbers.

- Term Frequency-Inverse Document Frequency(tf-idf): Another common document presentation is tf-idf which is very similar to the previous representation but also considers the frequency of the terms in the entire corpus. The idea is that common words that frequently appear in many documents are less important.

- Word Vector Representation (Word2Vec, GloVe): Each word is represented by a fixed-size vector, and thus each document is presented as a matrix. Such representations allow models to incorporate semantics and syntactic aspects of words.

- Contextualize Word Vectors (ELMO, BERT): A more recent advanced vector representation similar to previous ones, which also takes the context into

account and generates vector embedding for words while considering the context. The same word could have multiple embeddings for each meaning/context in these representations.

Next, these representations will be used as input for a machine learning classifier to be trained against the known categories of each document. Common machine learning algorithms for text classification are briefly explained here.

- Naïve Bayes Classifier: Naïve Bayes is the first and one of the simplest classifiers for text classification. Naïve Bayes makes a simplifying (naive) assumption that words are separated features of the document with no interaction between them. Given a document (d) and classes (c_i), Naïve Bayes assigns the highest class using the probabilities as in the following formula:

$$C_{NB} = \underset{c \epsilon C}{\operatorname{argmax}} P(d|c) = \underset{c \epsilon C}{\operatorname{argmax}} P(c) \prod_{i \epsilon positions} P(w_i|c_i) \tag{2.1}$$

  where w_i refers to each word in the document. Naïve Bayes is a generative model, and despite its naive assumption in practice, it often produces surprisingly acceptable results.

- Maxent Classifier(Logistic Regression): Maxnet is a discriminative model commonly used as a baseline for document classification. The usual representation of text for this model is a numeric representation such as term-frequency or tf-idf. The classification function is sigmoid or softmax, and the cost function is the cross-entropy loss function which will be minimized during training.

$$L = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{2.2}$$

  Where N is the number of samples, y_i is the true label for sample i (either 0 or 1), and $\hat{y}_i$ is the predicted probability' To compute this minimization problem gradient descent is used which is an optimization algorithm for iteratively

updating of the weights. At each iteration, the gradient is computed using a learning rate, which determines the step size of the current parameters and gradually converges to the optimal minimum of the cross-entropy loss function finding the best parameters.

- Other traditional ML classifiers: Using other machine learning classifiers for document classification is possible. Support vector machines (SVM) are another type of classifier that tries to find a hyperplane such that it splits classes that maximize the distance of observations to such a plan. In many text classification applications, SVM provides good results and is also considered a good baseline for more complex models. Another famous classifier is random forests, which is a specific ensemble of many decision trees for classification. In random forests, at each split of every tree, only a sub-sample of features will be considered.

- Deep Learning Classifiers: These types of classifiers use neural network architecture to predict the class for a document. For example, a deep neural network with multiple layers can use tf-idf representation of a document to predict its category. Recently, more complex models were developed that can be used for document classification, such as convolutional neural networks (CNN) and Recurrent Neural Networks (RNN). These models can use sequences as the input and therefore take the word order of the document into account. Furthermore, their inputs use vector representations of words that capture more information about words, such as semantics and syntactic information. The cost function usually is the cross-entropy loss, and a variation of gradient descent is commonly used to train these networks. One caveat of RNN models is the problem of the exploding/vanishing gradient. GRUs and LSTM are two variants of RNN where neural network units use gated mechanisms to mitigate the gradient problems.

Since document classification is, in its essence, a supervised learning problem,

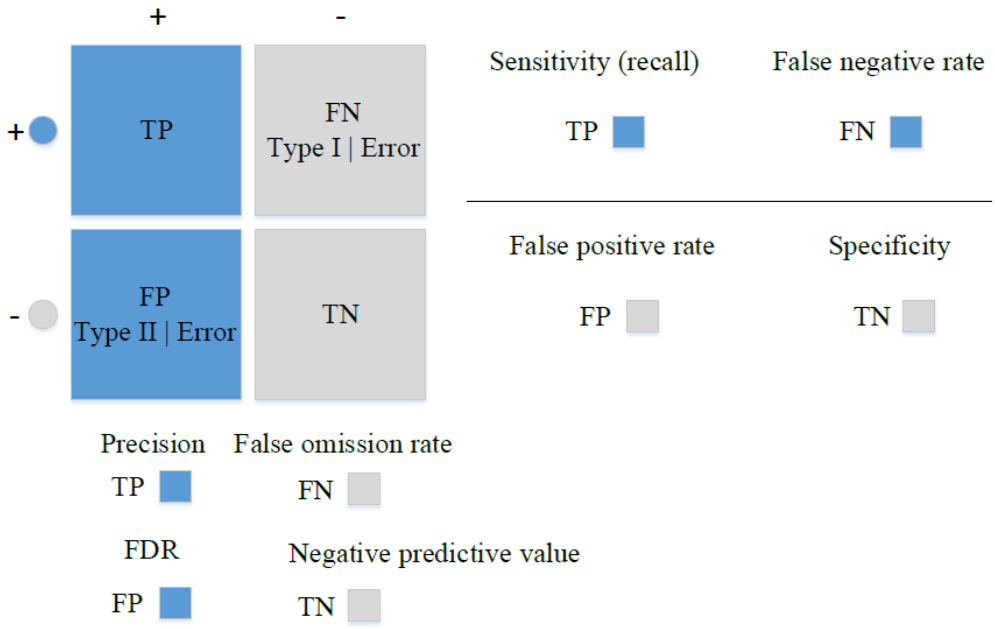

Figure 2.4: Confusion Matrix

we can use the true label of the test set and the model prediction as in any other supervised machine learning approach. The most important evaluations are based on the confusion matrix that shows how many true positive/negative values the model predicted correctly, as shown in fig 2.4.

The common metrics for supervised document classification can be derived from confusion matrix as follows:

$$accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{2.3}$$

$$precision = \frac{\sum_{l=1}^{L} TP_l}{\sum_{l=1}^{L} TP_l + FP_l} \tag{2.4}$$

$$recall = \frac{\sum_{l=1}^{L} TP_l}{\sum_{l=1}^{L} TP_l + FN_l} \tag{2.5}$$

$$F1 - Score = \frac{\sum_{l=1}^{L} 2TP_l}{\sum_{l=1}^{L} 2TP_l + FP_l + FN_l} \tag{2.6}$$

Finally, the classification uses a threshold (usually 0.5), and this can also impact the prediction. Receiver operating characteristics (ROC) curves are valuable graphical tools for evaluating as well, where it shows the effect of changing the

threshold on classification prediction and how confident the model predictions are.

2.3.2 Summarization and Topic Abstraction

Today, the large amount of data available in organizations makes it difficult to process this information efficiently. A practical solution is to condense the information in documents into a shorter format, such as a summary or collection of the main topics. Text summarization techniques are mainly divided into two categories that will be briefly explained here.

- Extractive Summarization: Here, a subset of sentences considered to have the highest significance will be extracted to serve as the document's summary. This method commonly constructs an intermediate representation and scores sentences based on that representation and finally selects the top N sentences as the summary. The intermediate representation could be frequency-based(tf-idf), or topic modeling ideas such, as Latent Semantic Analysis (LSA) or Latent Dirichlet Allocation (LDA) [17]. Another approach to extract sentences from the text is to use one of the previously explained supervised document classifications with sentences labeled as " summary" or "not-summary" or to use graph-based models inspired by the PageRank [18].

- Abstractive Summarization: This approach aims to generate new sentences from the original text that capture the essence of the original document. This is a new and more advanced problem to solve, but the rise of sequence-to-sequence RNN-based models has improved this task. Furthermore, today's state-of-the-art models based on the transformer concept have shown improvement over the RNN-based model. Two examples of such models are GPT models and T5 [19].

Although the summarization task provides a shorter version of original documents, sometimes it is advantageous to understand what theme or topics the document presents. Describing a document or set of documents based on its topic is another

way of generating insights from documents. We briefly describe some major NLP techniques to extract topics from documents.

Latent Semantic Analysis is one such technique. After pre-processing the documents, it uses a term-document matrix and singular value decomposition (SVD) to generate concepts within the documents using two newly created term-concepts and concept-document matrices. Another variation of this approach has been developed as probabilistic LSA (PLSA) [20].

Another closely related approach, Non-negative Matrix Factorization (NMF), is a multivariate and linear algebraic approach that factorizes the same term-document matrix into feature-term and document-feature matrices. These matrices are constructed such that they are non-negative matrices, and this property makes this method produce more interpretable results.

Finally, another popular unsupervised technique is Latent Dirichlet Allocation (LDA), a generative probabilistic model. LDA assumes that documents are generated from specific topics, and it uses Bayesian inference to identify such underlying topics in the documents. Variations of LDA such as Correlated Topic Modeling (CTM) or Structural Topic Model (STM) also have been proposed [21].

Evaluation of these techniques is more challenging due to their unsupervised nature. The evaluation can usually be divided into intrinsic and extrinsic (where the performance can be measured by how the summary's quality impacts another task, such as text classification). One common intrinsic metric for summarization models is ROUGE-n which compares the generated summary with good candidate summaries using the number of overlapping n-grams to create a score [22].

Similarly, for topic extraction methods from documents, it is possible to measure the performance by how the quality of results affects other tasks. A common intrinsic metric for these models is perplexity, where a lower perplexity value indicates better topics.

Importantly, human-based evaluation, where human judgment is used to evaluate the result, can be a very valuable approach for both problems due to the

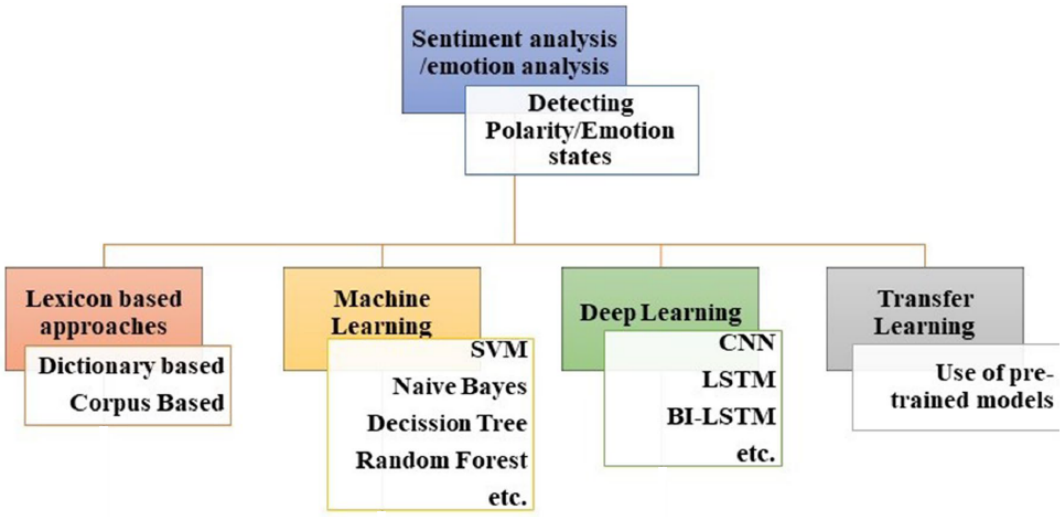

Figure 2.5: General methods for emotion and sentiment analysis

complexity of summarization and topic extraction tasks.

2.3.3 Sentiment and Emotion Analysis

Another NLP task with real-world applications is automatically detecting a text's sentiment and emotions. Many social applications might benefit from automatically detecting emotions.For example, a student who feels unhappy or bored while interacting with tutoring systems could be helped with a better dialogue response using emotion recognition capable of considering his or her emotion. Similarly, a patient's message to his/her primary medical provider that revealed sadness and depressive emotions could automatically be identified and trigger some supportive actions.

No universal framework for defining emotions exists, but most NLP models are based on two main classes. The first group uses a fixed number of emotions as the main emotions, and the second uses three dimensions of valence, arousal, and dominance to describe different emotions. A closely related task to emotion detection that is slightly easier is sentiment analysis, where the sentiment found in text (positive, negative, sometimes neutral) is sufficient for the task. There are many approaches, from lexicon-based models to different supervised models trained with emotion labels to identify emotions, as figure 2.5 shows [23].

2.3.4 Information Extraction, Retrieval and Q&A Systems

: Another important real-world application of NLP is to extract structured data or needed information from unstructured textual formats. Depending on the application, we might be required to extract specific types of words or phrases or even look through a large set of documents and automatically find related ones. We briefly describe these methods in order of complexity.

Name Entity Recognition (NER): One of the most common forms of information extraction is to identify and extract named entities such as people, organizations, locations, and dates. The task has been formulated as unsupervised, semi-supervised, and supervised, with the latter being used most often. Commonly used metrics for evaluating the extraction are precision (in this case, the number of correctly predicted system entities divided by the number that the system predicted) and recall (the number of correctly predicted over human annotated results). The use of deep learning and, specifically, variants of LSTM models proved to be very promising [24]

Information Retrieval (IR): In IR, the objective is to match a query against a set of documents and find the most relevant documents to that query. The basic approach is to present both query and documents as vectors, and using the cosine similarity, find the most similar documents based on the cosine score. The classic approaches use tf-idf and BM25 for generating the vectors. More recent models use dense vectors for these representations based on neural networks. For example, both query and document can be presented using the Bidirectional Encoder Representations from Transformers (BERT) model with dot product [25]. The performance of these models can be measured by precision and recall-like definition similar to above mentioned NER metrics.

Question Answering: Lastly, sometimes, it is required to extract a specific part of text based on a question on hand. This task is referred to as "Question Answering" (QA), which is a challenging task and requires advanced NLP methods. Question answering tasks can be divided into two main categories depending

on the target documents. Knowledge-based QA models refer to QA models that query a structured database. For example, graph-based approaches search over a knowledge graph of factoids to find the answer to a question. Another category of QA is the open domain problem, where given a large set of documents the task is to find the span within the text as the answer to the question. Many models nowadays have been developed to solve this problem. However, transformer-based models have been more successful on standard QA datasets such as SQuAD. Another challenge for these models is the capability of handling missing answers in the text, as in SQuAD2.0 [26].

With the increasing power of complex models that are trained on huge amounts of data, using language models for QA has become another interesting solution. The idea is that we can use the pre-trained parameters of an extremely large model to answer a question in the open-domain scenario. An example of such a language model is openai GPT-3 with over 175 billion parameters [27], which can be used for answering open domain questions [28].

2.3.5 Dialogue Systems and Chat Bots

A machine's ability to seamlessly communicate and understand humans is one of the major artificial intelligence goals and thus, a main area of interest in NLP. Dialogue systems represent such programs that allow communication with a computer system. Most such systems that are developed are task-oriented systems designed to understand human purposes to complete the desired tasks. A more challenging problem is to design chat bots that have the capability for extended interactions with humans while being as human-like as possible.

An illustration of a task-oriented dialogue system architecture is shown in Fig 2.6. A complete system like the one shown uses multiple components such as Automated Speech Recognition (ASR), dialogue state tracker, and text-to-speech component. The dialogue state tracker is the collection of the current state and the historical information of the previous interactions. Dialogue policy decides what

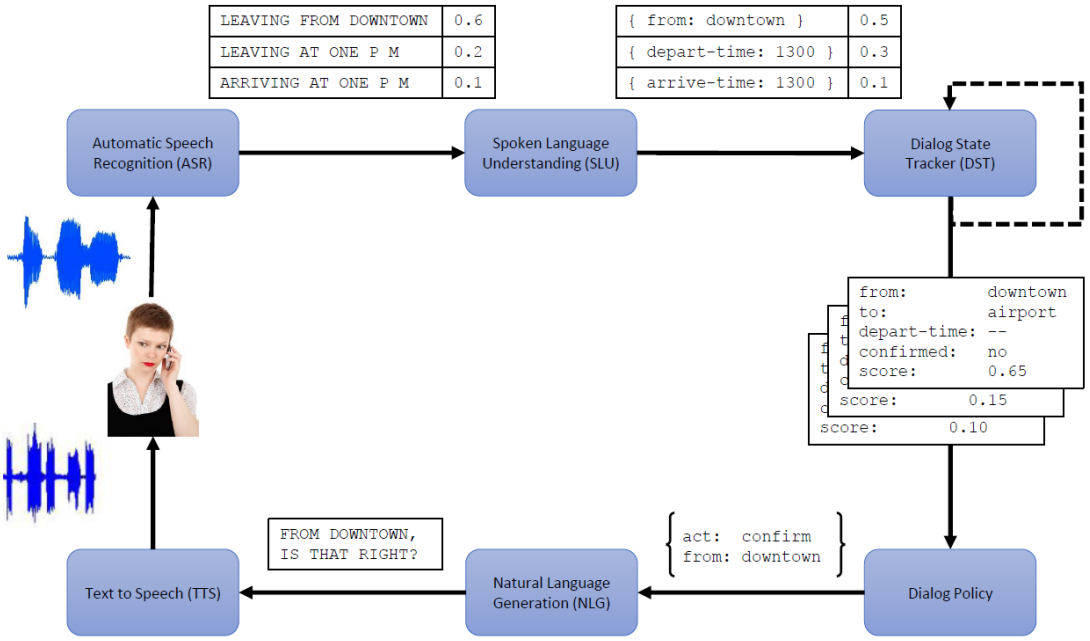

Figure 2.6: Architecture of task-oriented dialogue system [2]

action or utterance should be performed based on the dialogue state's input, then uses a language generation model to generate a coherent response. The two other components of these chat bots, ASR and Text-to-speech, require transforming audio to text or vice versa. These problems have their own challenges, but the common approach uses sequence-to-sequence deep learning models with acceptable results.

Building dialogue systems with the purpose of holding an extended conversation with humans is still challenging. The earliest attempts used rule-based models but recently better chat bots have been proposed. These chat bots use a large corpus to either respond by retrieval or by generation. Mostly, these models use different sequence-to-sequence architectures and specifically transformers [29]. Moreover, a recent chat bot called "ChatGPT", which was created by training a GPT-3.5 type of model and using human feedback through reinforcement learning has proven to work on many of NLP tasks described before [30]. However, the evaluation of Dialogue systems is a challenging task, and often human evaluation is still the preferred method.

2.4 NLP Applications for Social Benefits

As we described in the previous section, certain capabilities of NLP make them suitable solutions for real-world applications which might have benefits for society. Many commercial products use different types of NLP solutions to make life easier for individuals. For example, voice assistant technologies such as Siri and Alexa use ASR systems to make tasks easier for us, while search engines use information retrieval techniques to help us find what we want. Although these applications can be considered beneficial for society, NLP methods can also be used directly in social domains. We will look at some of these applications in this part.

2.4.1 NLP Application in Healthcare

Healthcare is one of the social domains that can use NLP to automate different tasks. The vast amounts of textual information generated by healthcare organizations and hospitals every day, necessitate automated tools to process this information as quickly as possible. A major domain in healthcare is Electronic Health Records (EHR) and its free-text notes. Therefore, the proper processing of EHR can significantly improve the efficiency of many tasks in healthcare, making it much more responsive to patients' needs.

De-identification: The free text notes within EHR often include sensitive information about patients. Not every process in the healthcare system or medical researchers, need to have access to every piece of sensitive data. In order to protect the privacy of patients and comply with government regulations, it is important to remove this data efficiently from the free text. Since this task is similar to name entity recognition, today's system can use advanced deep learning models such as LSTM/GRU to identify and remove this data [31].

Extracting clinical information: EHR's free text contains a lot of medical information on each visit of patients. Medical conditions, drugs, dosages, and biomarkers are just a few useful pieces of information that might exist in these

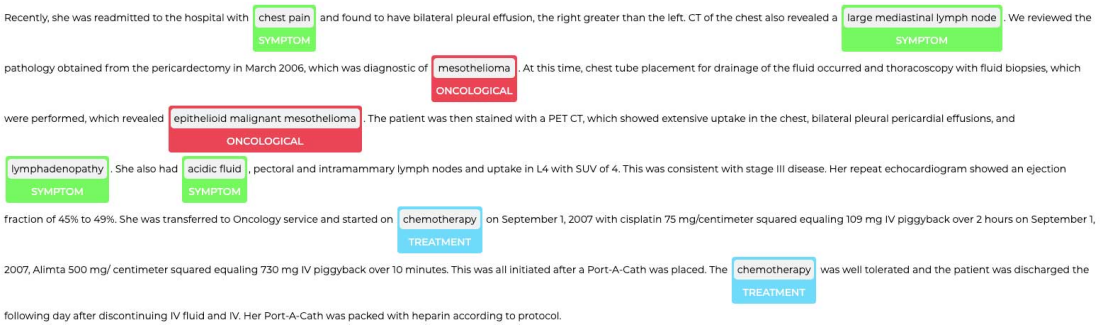

Figure 2.7: Example of medical entity identification by John Snow Lab

documents. Therefore, automatically identifying and extracting these concepts is a valuable task to improve provided services. This task can also be considered an entity recognition task, but the number of different entities in these free-text notes is considerably larger. Until recently, the most popular solutions for clinical information extraction were based on rule-based information extraction approaches. More complex NLP techniques such as deep learning sequence-to-sequence models have begun to be used only recently. An example of medical information extraction using NLP is cTAKES which was developed by the Mayo Clinic and is now used frequently for such extraction by other organizations [32]. Furthermore, the complexity of all types of entities in EHR notes has motivated some companies, such as John Snow Lab, to design specific solutions for this extraction task (Fig 2.7). Recent models like this one also used better contextual embedding, such as bioBERT, to represent the entities [33].

Clinical NLP-based predictive tasks: Document classification can also benefit healthcare organizations by providing automated assistance to clinicians and administrative staff. We briefly mention some of these applications.

- **Patient risk identification**: Automatically identifying high risk patients from the general population of patients can be very useful, allowing medical staff to act quickly or triggering other actions. Here, a text classification model trained with free text can be used to generate patient scores based on the notes.

- **Treatments and diagnosis**: Another area that NPL can be useful in

healthcare is in identifying the diagnosis and possible treatment based on the EHR's free text.The International Classification of Diseases (ICD) includes roughly 155000 diseases, so automatically assigning patients to the right healthcare category could hasten the provision of proper medical services. An example of this could be classifying patients using the free text into different stages of a particular cancer.

- **Administration and triage**:Administrative staff and nurses can benefit by using NLP on EHR notes to identify what type of facilities each patient should be assigned to. Such a task also can be formulated as a text classification problem using different facilities as classes.

Besides the above application, NLP predictive methods can be used in other areas, such as adverse drug reaction identification or medication classification, etc. Furthermore, other applications, such as abstracting documents and speech-to-text are among the other NLP methods that can be beneficial for healthcare organizations.

2.4.2 NLP Applications for Education

Another social domain that can successfully use NLP techniques is the education domain, due to the availability of written texts in this domain. Summarization of articles, books, and other texts in this domain is a direct usage of NLP techniques that we mentioned before. We briefly discuss some of these applications here.

Automated essay grading Automated grading can be very beneficial for teachers but grading essay-like or written short answers is not as simple as grading multiple-choice questions. A common solution is to use a text classification algorithm and train it on graded essays. In this approach, the idea is to build a model that classifies the text to assign the appropriate grade with high accuracy. Another NLP approach that can be used is to measure similarity between the text and one or more golden answers. The similarity can be measured using different metrics, such as Quadratic Weighted Kappa (QWK) and ROUGE [34].

Intelligent tutoring systems: Intelligent Tutoring Systems also use NLP techniques to help students. These systems can use a domain-specific knowledge base to track the concepts that need to be taught and the student's knowledge state. A task-oriented type of chat bot will then be able to guide the student by interacting with him/her. Such computer-assisted tutoring can greatly help online learning and e-learning platforms. Furthermore, in foreign language education, such tutoring systems can improve the process [35].

Plagiarism detection Plagiarism is a critical problem in education as it can seriously impair students' ability to think critically. Identifying plagiarism is a challenging task due to the potential for simply rewording and rephrasing human language without students using any true analysis. Nevertheless, NLP methods have been researched and investigated to tackle this problem. The general idea, in this case, is to find a similarity between the original document and the derived one. Approaches that use exact match or embedding vectors of words and deep learning models have been investigated for this task as well [36].

2.4.3 NLP Applications in the Public Sector

The public sector is another major social domain that can use NLP techniques to address different challenges. For instance, governmental organizations deal with many citizens every day while also handling large amounts of unstructured data in textual format. Mapping the challenges in this domain to the correct NLP solution can help provide better services to citizens as well as improve the efficiency and quality of governmental organizations.

Virtual assistant and customer services: Governmental organizations recognized the importance of helping customers in the era of the internet and electronic services. One of the areas that improves citizens' satisfaction with their state, local, or federal government is the availability of online portals that allow 24/7 support to help citizens by answering their questions or guiding them to the right resources. Such virtual assistants (VA) are built using NLP and task-oriented

chat bots to provide support for citizens. Some examples of these VA technologies are "Ask Jamie" which is designed for Singapore's governmental portal [37] and "Emma" which is designed for the US immigration portal. Furthermore, machine translation and multilingual chat bots can be another NLP application in e-government. HUGO is an example of an e-government platform built for the Latvian e-government portal. It allows translation of different textual data while providing other benefits as well [38].

Analysis of government documents and social media: Besides translation and virtual assistants, other NLP techniques such as document classification can also be used to enhance the performance of government organizations. Classification of sensitive information and identifying features in sensitive documents is one such application. Another example of these techniques is the automatic ability to check for regulatory compliance in the documents. Social media comments and posts can also be analyzed to identify criminal activities automatically with NLP [39]. Other textual sources such as a given population's feedback about different policies can be used with sentiment analysis to evaluate the society's feelings about new policies [40].

2.4.4 NLP Applications in Other Domains

The domains that we described before, are among the most prominent areas of the public sector, and therefore more efforts have been focused on those domains. However, the existence of textual data in many other public fields makes NLP techniques good candidates for automated and efficient decision making or insight generation. In particular, for under-explored public domains, such research could provide additional solutions by automatically processing data or finding patterns. Furthermore, the ability to build systems around processed text to address real-world problems is another area that needs more attention.

In this thesis, we aimed to show the potential of NLP techniques to help public domains that either the usage of textual information is less common or has

less textual information available to work on. We showed potential applications in the three domains of infrastructure and safety, terrorism and security, and labor supply empowerment. In doing so, we also presented how to map different problems into suitable NLP solutions. We utilized supervised, unsupervised, and semi-supervised NLP approaches with text as the input data. The NLP techniques include document classification, topic identification, emotion and affect analysis, and information extraction. On top of that, we presented how we can build a system to use textual information and create different types of insight that can also be made available to the public. We have taken this extra step in order to present the power of textual information analysis as a real-world application that can benefit the general public. The following chapters describe these applications in detail.

Chapter 3

Short Domain Specific Text Classification with Deep Learning

In many social domains, information can be found in textual formats that can be used for building decision-support systems. One approach that uses these texts to build a decision-making support system, is to automatically classify the text into some predefined category. The trained model then can be used to suggest the most probable category for the text to assist in categorizing the documents. In particular, in some social areas, the text could have features that make this task challenging. Short and domain-specific texts are two of the challenges in utilizing this information correctly. An example of such texts is the accident narrative report for railways that are relatively short and uses a specific jargon common to railway accidents. We present this problem and propose an approach to use these reports as a decision-making support tool.

3.1 Introduction

Rail accident reporting in the U.S. has remained relatively unchanged for more than 40 years. The report form has 52 relevant accident fields and many of these fields have sub-fields. Some fields require entry of the value of an accident result or condition, e.g., "Casualties to train passengers" and "Speed". Other fields have

restricted entries to values from a designated set of choices, e.g., "Type of equipment" and "Weather". "Primary cause" is an example of a restricted entry field in the report where the value must be one of 389 coded values. Choosing one of these categories while filling in reports is sometimes challenging and subject to errors due to the wide range of accidents. On the other hand, this field has significant importance for transportation administrations analysis in order to provide better safety regulations.

Field 52 on the report is different from the other fields because it allows the reporter to enter a narrative description of the accident. These accident narratives provide a way for the accident reporter to describe the circumstances and outcomes of the accident in their own words. Among other things, this means providing details not entered in any of the other fields of the report. For example, while the report may show an accident cause of H401 - "Failure to stop a train in the clear," the narrative could provide the reasons and circumstances for this failure. These additional details can be important in improving rail safety by helping in selection of a more accurate cause for the event. As a result, a method that correlates the detailed narratives with causes would be beneficial for both accident reporters and railroad administrators.

Despite the advantages of the narrative field, most safety changes result from fixed field entries since accident descriptions are difficult to automatically process. The advance of methods in text mining and machine learning has given us new capabilities to process and automatically classify textual content. This paper describes the results of a study using the latest methods in deep learning to process accident narratives, classify accident cause, and compare this classification to the causal entry on the form. The results of this study give us insights into the use of machine learning and, more specifically, deep learning to process accident narratives and to find inconsistencies in accident reporting.

This paper investigates how the narrative fields of FRA accident reports could be efficiently used to extract the cause of accidents and establish a relationship

between the narrative and the possible cause. Such relationships could assist the reporters to freely enter the narratives and getting candidate choices for causal field of reports. Our approach uses state-of-the-art deep learning technologies to classify texts based on their causes. The rest of this paper is organized as follows: in Section 4.2, related work in both accident analysis and text classification with deep learning have been presented. Section 4.3 describes in detail the approach that has been used along with evaluation criteria. Section 4.4 provides details of our implementations and section 4.5 reports the results. Finally, Section 3.6 presents the conclusion.

3.2 Related Work

This paper utilizes text mining and a new generation of natural language processing techniques, i.e. deep learning [3, 41, 42] in an effort to discover relationships between accident reports' narratives and their causes. In this section, we describe related work in both railroad accident analysis and text mining with deep learning. Train accident reports have been the subject of considerable research and different approaches have been used to derive meaningful information from these reports to help improve safety. As an example, the relationship between the length of train and accident rate has been investigated in [43]. This paper also emphasizes the importance of proper causal understanding. Other authors [44, 45] have used FRA data to investigate accidents caused by derailments. Recent work has used statistical analysis on FRA data to discover other patterns to investigate freight train derailment rate as an important factor [46]. All of these previous works used only the fixed field entries in the accident reports for their analysis and did not use information in the accident narratives. Some investigators have begun to apply text mining for accident report analysis in an attempt to improve safety. Nayak, et al. [47] provided such an approach on crash report data between 2004 and 2005 in Queensland Australia. They used the Leximancer text mining tool to produce cluster maps and most frequent terms and clusters. Other research [48] introduced

concept of chain queries that utilize text retrieval methods in combination with link-analysis techniques. Recent work by Brown [49] provided a text analysis of narratives in accident reports to the FRA. He specifically used topic modeling of narratives to characterize contributors to the accidents. In this paper, we present a new approach to the analysis of these accident narratives using deep learning techniques. We specifically applied three main deep learning architectures, Convolutional Neural Nets (CNN), Recurrent Neural Nets (RNN), and Deep Neural Nets (DNN), to discover accident causes from the narrative field in FRA reports.

Another study [50] presented an overview of how these methods improved the state-of-the-art machine learning results in many fields such as object detection, speech recognition, drug discovery and many other applications. CNN were first introduced as a solution for problems involving data with multiple array structure such as 2D images. However, the researchers in [51] proposed using a 1D structure to enable CNN applications for text classification. This work was extended by Zhang, et al., who developed character-level CNN for text classification [52]. Other work has provided additional extensions to include use of dynamic k-max pooling for the architecture in modeling sentences [53]. In RNN, the output from a layer of nodes can reenter as input to that layer. This architecture makes these deep learning models particularly suited for applications with sequential data including, text mining. Irsoy et al. [54] showed an implementation of deep RNN structure for sentiment analysis of sentences. The authors of this paper compared their approach to the state-of-the-art conditional random fields baselines and showed that their method outperforms such techniques. Other researchers used different combinations of RNN models with some modifications and showed better performance in document classifications as in [55], [56]. Also, some recent researchers combined CNN and RNN in a hierarchical fashion and showed their overall improved performance for text classification as in [57]. Another hierarchical model for text classification is presented in [41] where they employ stacks of deep learning architectures to provide improved document classification at each level

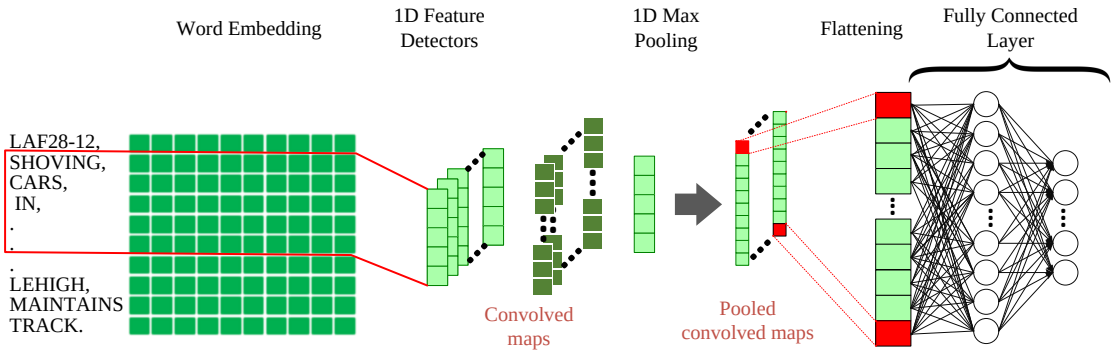

Figure 3.1: Structure of Convolutional Neural Net using multiple 1D feature detectors and 1D max pooling

of the document hierarchy. In our study, we have combined text mining methods and deep learning techniques to investigate the relationship of narrative field with accident cause which has not been explored before using such methods.

3.3 Method

For this analysis, each report is considered as a single short document which consists of a sequence of words or unigrams. These sequences are considered input in our models and the accident cause (general category or specific coded cause) is the target for the deep learning model. We convert the word sequences into vector sequences to provide input to the deep learning models. Different solutions such as"Word Embedding" and tf-idf representation are available to accomplish this goal. This section also provides details on deep learning architectures and evaluation methods used in this study .

3.3.1 Word Embedding and Representation

Different word representations have been proposed to translate words or unigrams into understandable numeric input for machine learning algorithms. One of the basic methods is term-frequency (TF) where each word is mapped on to a number corresponding to the number of occurrences of that word in the whole corpora. Other term frequency functions present word frequency as a Boolean or a loga-

rithmically scaled number. As a result, each document is translated to a vector containing the frequency of the words in that document. Therefore, this vector will be of the same length as the document itself. Although such an approach is intuitive, it suffers from the fact that common words tend to dominate the representation.

3.3.1.1 Term Frequency-Inverse Document Frequency

K. Sparck Jones [58] proposed inverse document frequency (IDF) that can be used in conjunction with term frequency to lessen the effect of common words in the corpus. Therefore, a higher weight will be assigned to the words with both high frequency in a document and low frequency in the whole corpus. The mathematical representation of weight of a term in a document by tf-idf is given in 3.1 .

$$W(d,t) = TF(d,t) * log(\frac{N}{df(t)}) \tag{3.1}$$

Where N is the number of documents and $df(t)$ is the number of documents containing the term t in the corpus. The first part in Equation 3.1 would improve recall and the latter would improve the precision of the word embedding [59]. Although tf-idf tries to overcome the problem of common terms in a document, it still suffers from some other descriptive limitations. Namely, tf-idf cannot account for similarity between words in the document since each word is presented as an index. In recent years, with development of more complex models such as neural networks, new methods have been presented that can incorporate concepts such as similarity of words and part of speech tagging. GloVe is one such word embedding technique that has been used in this work. Another successful word embedding method used in this work is Word2Vec which is described in the next part.

3.3.1.2 Word2Vec

Mikolov, et al. developed the *"word to vector"* representation as a better word embedding approach [60]. Word2vec uses two neural networks to create a high

dimensional vector for each word: Continuous Bag of Words (CBOW) and continuous skip-gram (CSG). CBOW represents the word in context with previous words while CSG represents the word by proximity in the vector space. Overall the word2vec method provides a very powerful relationship discovery approach.

3.3.1.3 Global Vectors for Word Representation (GloVe)

Another powerful word embedding technique is Global Vectors (GloVe) presented in [61]. The approach is very similar to the word2vec method where each word is represented by a high dimension vector, and trained based on the surrounding words over a huge corpus. The pre-trained embeddings for words used in this work are based on 400,000 vocabularies trained over Wikipedia 2014 and Gigaword 5 with 50 dimensions for word representation. GloVe also provides other pre-trained word vectorizations with 100, 200, 300 dimensions which are trained over even bigger corpi as well as over Twitter. Figure 3.2 shows an example of how these embeddings can be used to transfer words to a better representation. As one can see, words such as "Engineer", "Conductor", and "Foreman" are considered close based on these embeddings. Similarly, words such as "inspection" and "investigation" are considered very similar.

3.3.2 Text Classification with Deep Learning

Three deep learning architectures used in this paper to analyze accident narratives, are Convelutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Deep Neural Networks (DNN) [3, 41, 42]. The building blocks of these classifiers are described in greater detail in this section.

3.3.2.1 Deep Neural Networks (DNN)

DNN's structure is designed to learn by multiple connections between layers where each layer only receives connections from previous layer and provides connections only to the next layer [3, 42]. The input is a vectorized representation of docu-

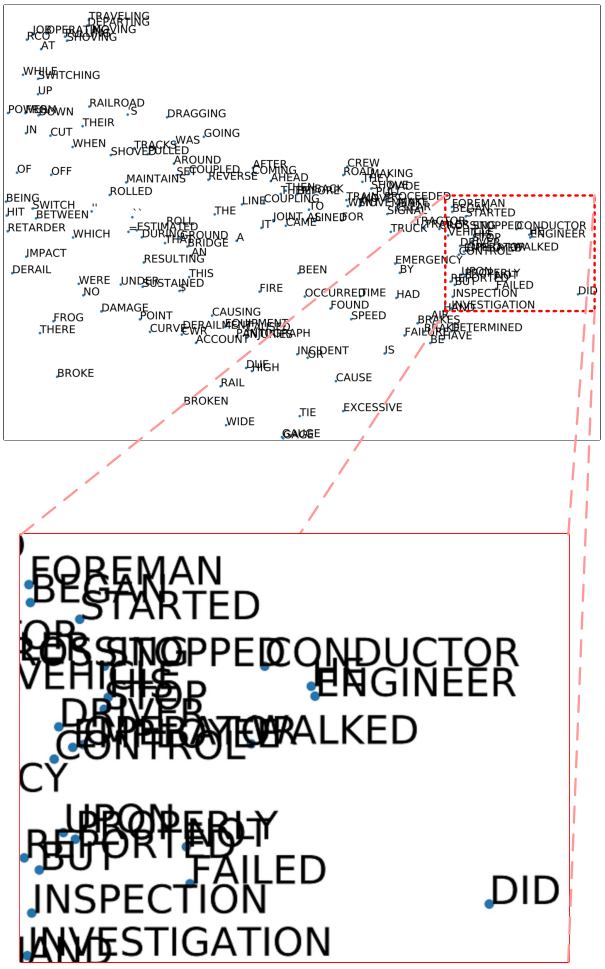

Figure 3.2: T-sne visualization of Word2vec 300 most common words

ments, which connects to the first layer. The output layer is number of classes for multi-class classification and only one output for binary classification. The implementation of Deep Neural Networks (DNN) is discriminative trained model that uses standard back-propagation algorithm. Different activation functions for nodes exist such as sigmoid or tanh but we noticed ReLU [62] (Equation 3.2) provides better results. The output layer for multi-class classification, should use $Softmax$ as shown in Equation 3.3.

$$f(x) = \max(0, x) \tag{3.2}$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{3.3}$$

$$\forall\ j \in \{1, \ldots, K\}$$

Given a set of example pairs $(x, y), x \in X, y \in Y$, the goal is to learn from these input and target spaces using hidden layers. In our text classification, the input is a string which is generated by vectorization of text using tf-idf word weighting.

3.3.2.2 Convolutional Neural Nets

Convolutional neural networks (CNN) were introduced by Yann Lecun [63] to recognize handwritten digits in images. The proposed design, though powerful, did not catch the attention of the computer-vision and machine learning communities until almost a decade later when higher computation technologies such as Graphics Processing Units (GPU) became available [50]. Although CNNs have been designed with the intention of being used in the image processing domain, they have also been used in text classification using word embedding [3, 42, 64].

In CNN, a convolutional layer contains connections to only a subset of the input. These subsets of neurons are *receptive fields* and the distance between receptive fields is called *stride*. The value at any neuron in the receptive field is given by the output from an activation function applied to the weighted sum of all inputs to the receptive field. Common choices for activation functions are sigmoid,

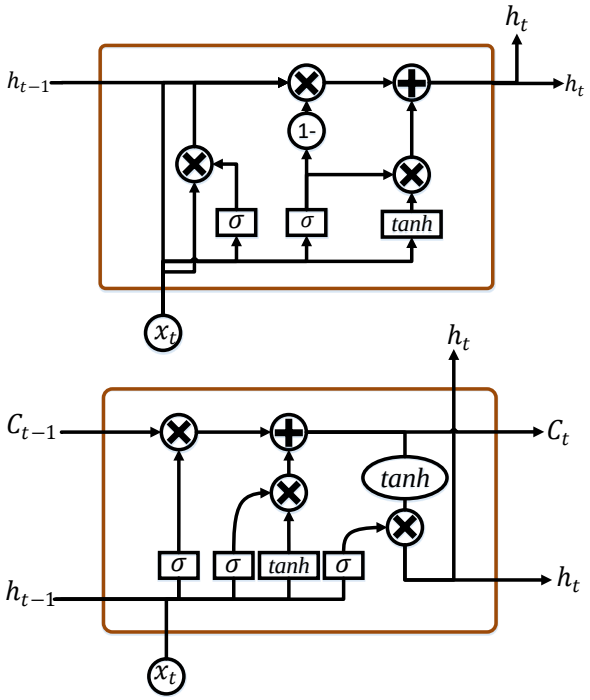

Figure 3.3: Top Fig: A cell of GRU, Bottom Fig: A cell of LSTM [3]

hyperbolic tangent, and rectified linear. As with most CNN architectures, in this study we stack multiple convolutional layers on top of each other.

The next structure in the CNN architecture is a pooling layer. The neurons in this layer again sample a small set of inputs to produce their output value. However, in this case they simply return the minimum, average or maximum of the input values. Pooling reduces computation complexity, and memory use. Additionally, it can improve performance on translated and rotated inputs [65]. Pooling can be repeated multiple times depending on the size of input and the complexity of the model.

The final layer is traditional fully connected layers taking a flattened output from the last pooling layer as its input. The output from this fully connected network is run through a softmax function for multinomial (i.e., multiple labels) problems, such as classifying cause from accident narratives.

Figure 3.1 shows the structure of an example CNN with one convolutional and max pooling layer for text analysis.

3.3.2.3 Recurrent Neural Networks (RNN)

RNN are a more recent category of deep learning architectures where outputs are fed backward as inputs. Such a structure allows the model to keep a memory of the relationship between words in nodes. As such, it provides a good approach for text analysis by keeping sequences in memory [66].

The general RNN structure is formulated as in Equation 3.4 where x_t denotes the state at time t and $\boldsymbol{u_t}$ refers to the input at step t.

$$x_t = F(x_{t-1}, \boldsymbol{u_t}, \theta) \tag{3.4}$$

Equation 3.4 can be expanded using proper weights as shown in Equation 3.5.

$$x_t = \mathbf{W_{rec}}\sigma(x_{t-1}) + \mathbf{W_{in}}\mathbf{u_t} + \mathbf{b}. \tag{3.5}$$

In this equation $\mathbf{W_{rec}}$ is the recurrent matrix weight, $\mathbf{W_{in}}$ are the input weights, $\mathbf{b}$ is the bias, and σ is an element-wise function.

The general RNN architecture has problems with vanishing and, less frequently, exploding gradients. This happens when the gradient goes through the recursions and gets progressively smaller or larger in vanishing or exploding states respectively. [67]. To deal with these problems, long short-term memory (LSTM), a special type of RNN that preserves long-term dependencies was introduced which shows to be particularly effective at mitigating the vanishing gradient problem [68].

Figure 3.3 shows the basic cell of an LSTM model. Although LSTM has a chain-like structure similar to RNN, LSTM uses multiple gates to regulate the amount of information allowed into each node state [3, 41, 42].

Gated Recurrent Unit (GRU) The Gated Recurrent Unit (GRU) [69] is a more recent and simpler gating mechanism than LTSM. GRU contains two gates, does not possess internal memory (the C_{t-1} in Figure 3.3), and unlike LSTM, a

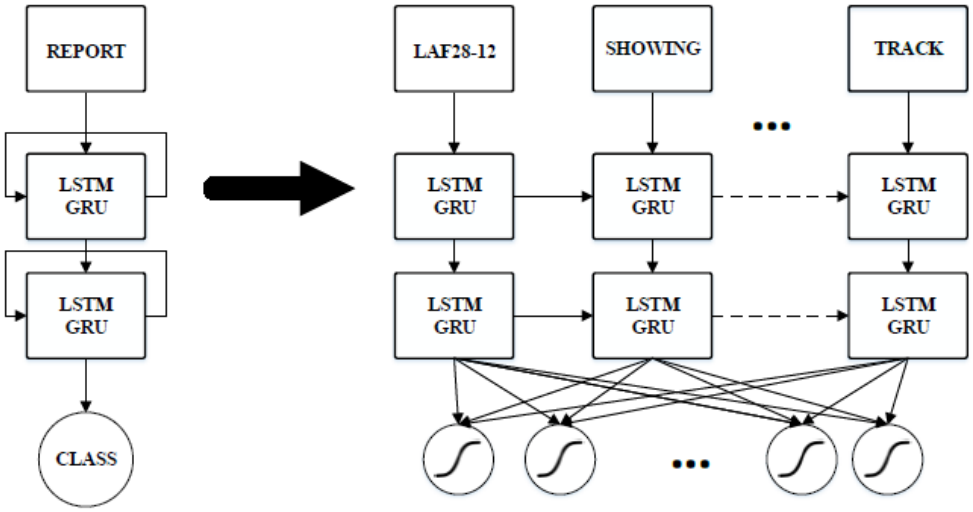

Figure 3.4: Structure of Recurrent Neural Net for report analysis using two LSTM/GRU layers

second non-linearity is not applied (tanh in Figure 3.3). We used GRU as our main RNN building block. A more detailed explanation of a GRU cell is given in following:

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1}] + b_z), \tag{3.6}$$

Where z_t refers to update gate vector of t, x_t stands for input vector, W, U and b are parameter matrices and vector, σ_g is the activation function, which could be sigmoid or ReLU.

$$\tilde{r_t} = \sigma_g(W_r x_t + U_r h_{t-1}] + b_r), \tag{3.7}$$

Where r_t stands for the reset gate vector of t.

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{3.8}$$

Where h_t is output vector of t, r_t stands for reset gate vector of t, z_t is update gate vector of t, σ_h indicates the hyperbolic tangent function.

Figure 3.4 shows the RNN architectures used in this study by employing either LSTM or GRU nodes.

Table 3.1: Distribution of data point and specified categories according to FRA

Total reports	'H306-7'	'T110'	'H702'	'T220-207'	'T314'	'M405'	'H704'	'H503'
11982	2613	2448	2171	1716	1053	753	652	576

3.3.3 Evaluation

In order to understand how well our model performs, we need to use appropriate evaluation methods to overcome problems such as unbalanced classes. This section describes our evaluation approach.

3.3.3.1 F1 measurement

With unbalanced classes, as with accident reports, simply reporting the overall accuracy would not reflect the reality of a model's performance. For instance, because some of these classes have considerably more observations than others, a classifier that chooses these labels over all others will obtain high accuracy, while misclassifying the smaller classes. Hence, the analysis in this paper requires a more comprehensive metric. One such metric is F1- score and its two main implementations: Macro-averaging and Micro-averaging. The macro averaging formulation is given in Equations 3.12, using the definition of precision (π) and recall (ρ) in Equation 3.9,3.10.

$$\pi_i = \frac{TP_i}{TP_i + FP_i} \tag{3.9}$$

$$\rho_i = \frac{TP_i}{TP_i + FN_i} \tag{3.10}$$

$$F_i = \frac{2\pi_i \rho_i}{\pi_i + \rho_i} \tag{3.11}$$

$$F_{1-macro} = \frac{\sum_{i=1}^{N} F_i}{N} \tag{3.12}$$

Here TP_i, FP_i, TN_i represent true positive, false positive and true negative, respectively, for class i and N classes.

Our analysis uses macro averaging which tends to be biased toward less populated classes [70]. As a result, we provide a more conservative evaluation since

deep learning methods tend to perform worse with smaller data sets. Another performance measure used in this study, is confusion matrix. A confusion matrix compares true values with predicted values and therefore, provides information on which classes are mostly misclassified to what other classes .

3.4 Experiments

In this section, we describe the embeddings that are used for our analysis as well as the structure of each deep learning model and the hardware that has been used to perform this work. To create word2vec presentation, we used gensim library to construct a 100 dimension vector for each word using a window of size 5. Similarly, we used a 100 dimension representation of Glove trained over 400K vocabulary corpus. The input documents have been padded to be of the same size of 500 words for all narratives. Our experiments showed that higher dimensions would not have a significant effect on the results.

Our DNN implementation consists of five hidden layers, where in each hidden layer, we have 1000 units with ReLu activation function followed by a dropout layer.

Our CNN implementation consists of three 1D convolutional layers, each of them followed by both a max pool and dropout layer. Kernel size for convolution and max pooling layers was both 5. At the final layer our fully connected layer has been made from 32 nodes and used a dropout layer as well.

RNN implementation is made of two GRU layers with 64 units in each followed by dropout after them. Final layer is a fully connected layer with 64-128 nodes at the end. This layer also includes a dropout similar to previous layers. The dropout rate is between 0.1 to 0.5 depending on the task and model which helps to reduce the chance of overfitting for our models.

The processing was done on a $Xeon\ E5-2640\ (2.6GHz)$ with 32 cores and $64GB$ memory. We used Keras package [71] with Tensorflow as its backend for our implementation.

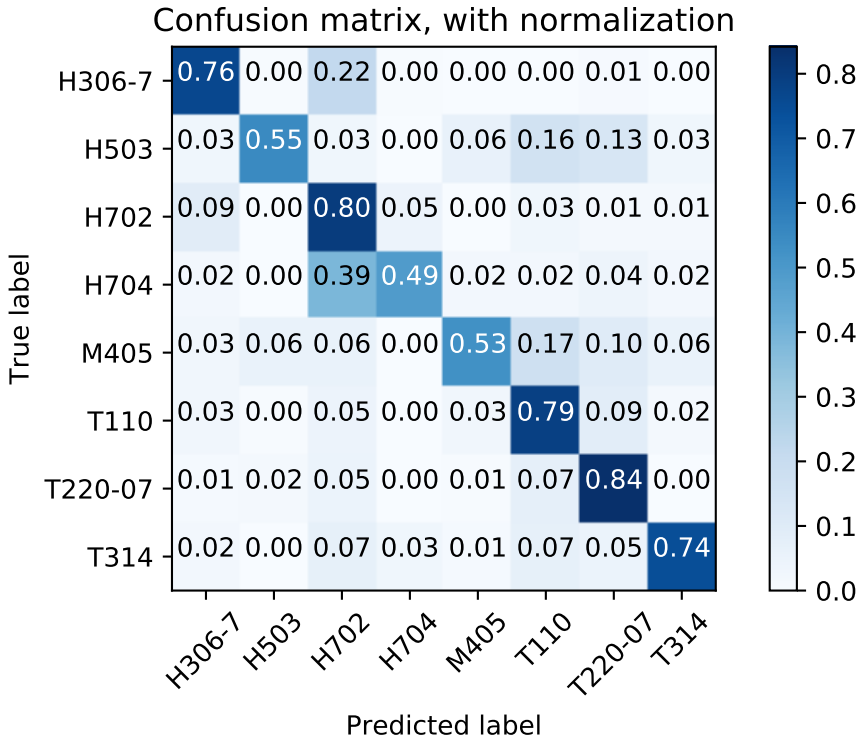

Figure 3.5: Confusion matrix for the best classifier

3.5 Results

This work has been performed using Federal Railroad Administration (FRA) reports collected during 17 consecutive years (2001-2017) [72]. FRA provides a narrative for each accident with the corresponding cause reported on that accident. The results are in two sections. In the first section, we show the performance in labeling the general cause for each accident based on its narrative and in the second section, we focus on the specific accident cause, on most common type of accidents according to reported detailed cause. In both of these analyses, we also compare our performance with some of traditional machine learning algorithms such as Support Vector Machines (SVM), Naive Bayes Classifier (NBC) and Random Forest as our baselines. Finally, we look at our misclassified results using confusion matrix and analyze errors made by our models.

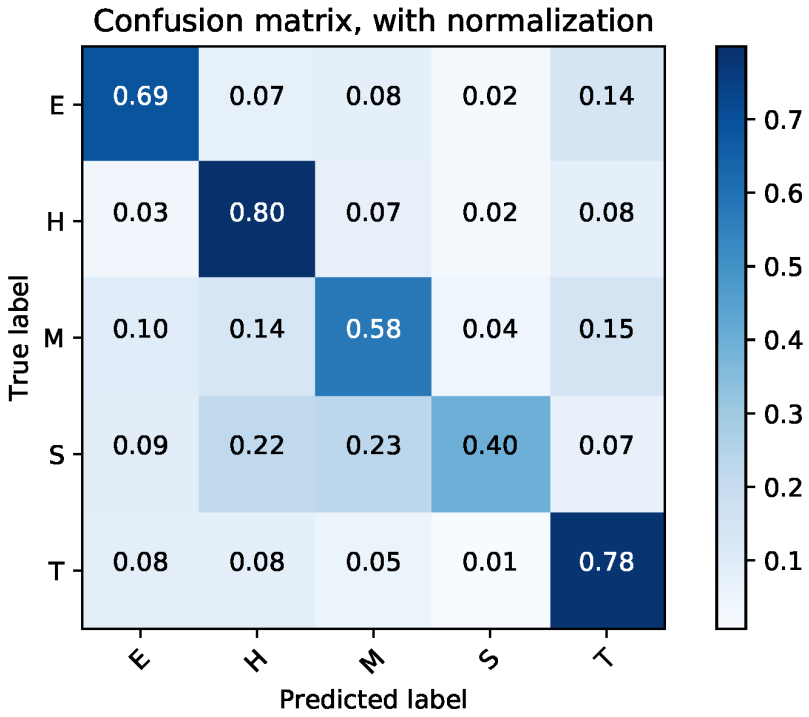

Figure 3.6: Confusion matrix for the best classifier

Table 3.2: Distribution of data points and general labels (E: Electrical failure, H: Human Factor, M: Miscellaneous, S: Signal communication, and T: Track)

Total reports	E	H	M	S	T
40164	5118	15152	5762	786	13256

3.5.1 General cause analysis

The general accident cause is in the reported cause field of accident reports. This analysis considers 40,164 reports with five labels as general causes. Table 3.2 shows the five causal labels and their distribution.

To classify the reports, both RNN and CNN along with two word embeddings, Word2Vec and Glove, and DNN with tf-idf are used. Table 3.3 shows the performance of our techniques and compare it with our baselines. Generally, Word2Vec embedding produces better F1 scores over the test set. Also, the differences between RNN and CNN results are not significant.

Figure 3.6 shows the confusion matrix for the best classifier. This confusion matrix shows that deep learning models in conjunction with vector representations of words can provide good accuracy especially on categories with more data points.

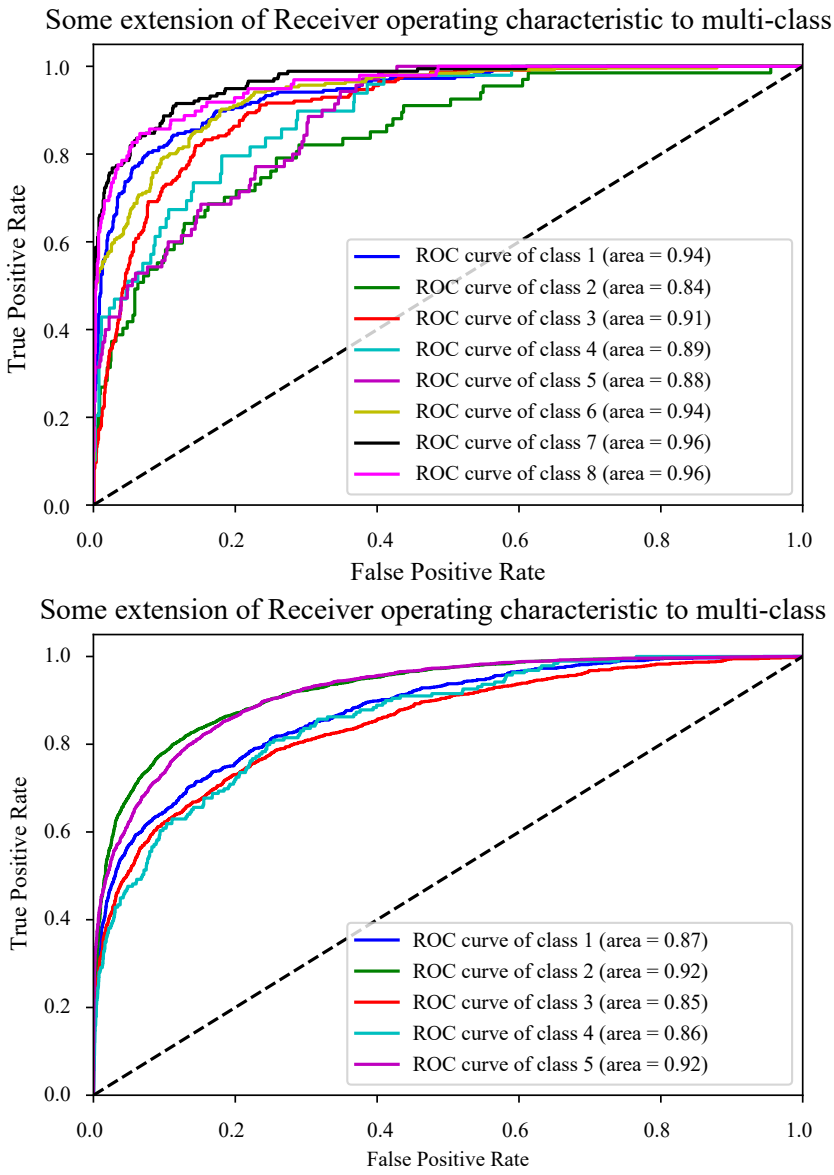

Figure 3.7: ROC curves for classifier of general and specific causes

3.5.2 Specific cause analysis

Our analysis also considers more specific accident causes in FRA reports (one of 389 code categories). An obvious issue with more detailed causal labels is that there are some cause categories with very few reports. Therefore, over the period studied, the top ten most common causes (combined into 8 categories since H307 and H306 have the same description and the description of T220 and T207 is very similar) have been selected for analysis. Table 3.1 shows the distribution of reports on these categories. Figure 3.5 shows the confusion matrix for the best classifier

Table 3.3: Classification F1 score of combined techniques

Feature Extraction		Technique	F1 measure (Macro)
Word Embedding	Word2Vec	CNN	0.65
	Word2Vec	RNN	0.64
	GloVe	CNN	0.63
	Glove	RNN	0.59
Word Weighting	tf-idf	DNN	0.61
	tf-idf	SVM	0.57
	tf-idf	NBC	0.61
	tf-idf	Random Forest	0.57

for the top 8 categories of causes.

We also investigate classifier performance using ROC curves as in Figure 3.7 for both general and specific causes.

Table 3.4 shows the results for specific causes along with a comparison with our baselines' performances. Similar to our previous results, models using Word2Vec embedding perform better than the ones using GloVe both in CNN and RNN architecture.

3.5.3 Error analysis

To better understand model performance, we investigated the errors made by our classifiers. The confusion matrices, clearly show that the number of instances in the classes plays a major role in classification performance.

As an example, reports labeled with Signal as the main cause are the smallest group and not surprisingly, the model does poorly on these reports due to the small number of training data points.

There is, however, another factor at work in model performance which comes from rare cases where the description seems uncorrelated to the cause. As an example of such cases, our model predicted the following narrative "DURING HUMPING OPERATIONS THE HOKX112078 DERAILED IN THE MASTER DUE TO EXCESSIVE RETARDER FORCES." in mechanical category while the original category reported is cause by Signal. This seems not consistent with the

Table 3.4: Classification F1 score of combined techniques for specific causes

Feature Extraction		Technique	F1 measure (Macro)
Word Embedding	Word2Vec	RNN	0.71
	Word2Vec	CNN	0.66
	GloVe	RNN	0.64
	GloVe	CNN	0.62
Word Weighting	tf-idf	DNN	0.64
	tf-idf	SVM	0.61
	tf-idf	NBC	0.33
	tf-idf	Random Forest	0.62

report narrative.

Identifying such inconsistencies in reports' narratives is important because both policy changes and changes to operation result from aggregate analysis of accident reports.

3.6 Conclusion and Future Work

This paper presents deep learning methods that use the narrative fields of FRA reports to discover the cause of each accident. These textual fields are written using specific terminologies, which makes the interpretation of the event cumbersome for non-expert readers. However, our analysis shows that when using proper deep learning models and word embeddings such as GloVe and especially Word2Vec, the relationship between these texts and the cause of the accident could be extracted with acceptable accuracy. The results of testing for the five major accident categories and top 10 specific causes (according to FRA database coding) show the deep learning methods we applied were able to correctly classify the cause of a reported accident with overall 75 % accuracy. Also, the results indicate that applying recent deep learning methods for text analysis can help exploit accident narratives for information useful to safety engineers. This can be done by providing an automated assistant that could help identify the most probable cause of an accident based on the event narrative. Also, these results suggest that in some rare cases, narrative description seems inconsistent with the

suggested cause in the report. Hence, these methods may have promise for identifying inconsistencies in the accident reporting and thus could potentially impact safety regulations. Moreover, the classification accuracy is higher in more frequent accident categories. This suggests that as the number of reports increases, the accuracy of deep learning models improves and these models become more helpful in interpreting such domain specific texts.

Chapter 4

Unsupervised Natural Language Processing for Data-Poor Social Domains

In the previous chapter, we presented a problem in the social domain that can use NLP to build a decision support tool using the texts. Unfortunately, some areas in social domains have small amounts of text that might have valuable information. Furthermore, in the absence of labels, a supervised NLP approach can not be used. Nevertheless, the automatic processing of these resources can provide a quick way of analyzing these documents and knowledge discovery from them. Using proper unsupervised methods can allow us to get such insights and even compare the documents with similar collections. An example of such a niche domain is the analysis of ISIS documents in counter-terrorism. The scope of the original documents is limited to a few published magazines and analysis of these documents to understand how ISIS targets women becomes even more challenging since not all their publication address women. Here, we discuss the application of unsupervised NLP methods in such a data-poor domain.

4.1 Introduction

Since its rise in 2013, the Islamic State of Iraq and Syria (ISIS) has utilized the Internet to spread their ideology, radicalize individuals, and recruit them to their cause. In comparison to other Islamic extremist groups, ISIS's use of technology was more sophisticated, voluminous, and targeted. For example, during ISIS advance toward Mosul, ISIS' related accounts tweeted 40000 tweets in one day [73].However, this high engagement forced social media platforms to institute policies to prevent unchecked dissemination of terrorist propaganda to their users, forcing the ISIS to adapt other means to reach their target audience.

One such approach was the publication of online magazines in different languages including English. Although discontinued now, these online resources provide a window into ISIS ideology, recruitment, and how they would like the world to perceive them. For example, after initially predominantly recruiting men, ISIS begun to include articles in their magazines that specifically addressed women. ISIS encouraged women to join the group by either traveling to the caliphate or carrying out domestic attacks on behalf of ISIS in their countries. This tactical change concerned both practitioners and researchers in the counter-terrorism community. New advancements in data science can shed light on exactly how the targeting of women on extremist propaganda works and whether it differs significantly from mainstream religious rhetoric.

To accomplish this, we utilizing natural language processing methods to answer three questions:

- What are the main topics in women related articles in ISIS online magazines?
- What of similarities or differences do these topics have with non-violent non-Islamic religious material addressed specifically to women?
- What kind of emotions do these articles evoke in their readers and are there similarities in the emotions evoked in readers of both ISIS and non-violent

religious materials?

As these questions suggest, to understand what, if anything, makes extremist appeals distinctive, we need a point of comparison in terms of the outreach efforts to women from a mainstream, non-violent religious group . For this purpose, we rely on an online catholic women's forum. Comparison between the catholic material and the ISIS online magazines allows for novel insight into the distinctiveness of exxtremist rhetoric when targeted toward female population. To accomplish this task, we employ topic modeling and an unsupervised emotion detection method.

The rest of the paper is organized as follows: in Section 4.2, we review related works on ISIS propaganda and applications of natural language methods. Section 4.3 describes in detail the approach we used and the data collection. section 4.4 reports the results, and finally, Section 4.5 presents the conclusion.

4.2 Related Work

Soon after ISIS emerged and declared its caliphate, counter-terrorism practitioners and political science researchers started to turn their attention to understanding how the group operated. Researchers investigated their origin of ISIS, leadership, funding, and how they rose to become globally dominant [74]. This interest in the organization's distinctiveness immediately led to inquiries into ISIS rhetoric, and particularly the use of social media and online resources in recruitment and ideological dissemination. For example, Al-Tamimi examines how ISIS differentiated itself from other jihadist movements by using social media with unprecedented efficiency to improve its image with locals [75]. One of ISIS's most impressive applications of its online prowess was in the recruitment process. The organization has used a variety of materials, especially videos, to recruit both foreign fighters and locals. Research shows that ISIS propaganda is designed to portray the organization as a provider of justice, governance, and development in a fashion that resonates with young westerners [76]. This propaganda machine has become a significant area for research, with scholars such as Winter identifying key themes

in it such as brutality, mercy, victim-hood, war, belonging and utopianism. [77]. There has, however, been insufficient attention to how these approaches have particularly targeted and impacted women. This is significant given that scholars have identified the distinctiveness of this population when it comes to nearly all facets of terrorism.

ISIS used different types of media to carry its message such as videos ,images, texts, and even music. Twitter was particularly effective and the Arabic twitter app allowed ISIS to tweet extensively without triggering spam-detection mechanisms twitter uses [73]. Scholars followed the resulting trove of data and this became the preeminant way in which to assess ISIS messages. For example, in [78] they use both lexical analysis of tweets as well as social network analysis to examine ISIS support or opposition in twitter. Other researchers used data mining techniques to detect pro-ISIS user divergence behaviour points in time [79]. By looking at these works, the impact of using text mining and lexical analysis to address important questions becomes obvious. Proper usage of these tools allows the research community to analyze big chunks of unstructured data. This approach, however, became less productive as the social media networks began cracking down and ISIS recruiters moved off of them.

With their ability to operate freely on social media curtained, ISIS recruiters and propagandists increased their attentiveness to another longstanding tool– English language online magazines targeting western audiences. Al Hayat, the media wing of ISIS, published multiple online magazines in different languages including English. The English online magazine of ISIS was named Dabiq and first appeared in Dark web at July 2014 and continued publishing for 15 issues. This publication was follower by Rumiyah which produced 13 English language issues through September 2017. The content of these magazines provides a valuable but underutilized resource for understanding ISIS strategies to appeal to recruits, specifically English-speaking audiences. They also provide a way to compare ISIS approach with other radical groups. Ingram compared Dabiq contents with In-

spire(Al Qaeda publication) and suggested that Al Qaeda heavily emphasized identity-choice, while ISIS messages were more balanced between identity-choice and rational-choice [80]. In another research, Wignell et al. [81] compared Dabiq and Rumiah by examining their style and what both magazine messages emphasized. Despite the volume of research on these magazines, few researcher used lexical analysis and mostly relied on experts' opinions. [82]is one exception to this approach where they used word frequency on 11 issues of Dabiq publications and compared attributes such as anger, anxiety, power, and motive etc.

This paper seeks to establish how ISIS specifically tailored propaganda targeting western women, who became a particualr target for the organization as the "caliphate" expanded. Although the number of recruits is not known, in 2015 it was estimated that around 10 percent of all western recruits were female [83]. Some researchers have attempted to understand how ISIS propaganda targets women. Kneip, for example, analyzed women's desire to join as a form of emancipation [84]. We extend that line of inquiry by leveraging technology to answer key outstanding questions about the targeting of women in ISIS propaganda.

To further assess how ISIS propaganda might affect women, we used emotion detection methods on these texts. Emotion detection techniques are mostly divided into lexicon-base or machine learning-base methods. Lexicon-base methods relay on several lexicons while machine learning(ML) methods use algorithm to detect relation of texts as inputs and emotions as the target, usually trained on a large corpus. Unsupervised methods usually use Non-negative matrix factorization (NMF) and Latent Semantic Analysis (LSA) [85] approaches. An important distinction that should be made when using text for emotion detection is that emotion detected in the text and the emotion evoked in the reader of that text might be different. In case of propaganda, it is more desirable to detect possible emotions that will be evoked in a hypothetical reader. In the next section, we describe methods to analyze content and technique to find awoken emotions in a potential reader using available natural language processing tools.

4.3 Method

In this section, we describe our data sources and methods used for comparing topics and evoked emotions in both ISIS and non-violent religious materials.

4.3.1 Data collection

ISIS online magazines are valuable resources for understanding how the organization attempted to appeal to western audiences, particularly women. Looking through both Dabiq and Rumiyah, many issues of the magazines contain articles specific addressing women usually with " to our sisters " incorporated into the title. Seven issues of Dabiq and all thirteen issues of Rumiyah contain such articles, clearly suggesting increasing attention to women over time.

We converted all the ISIS magazines to texts using pdf readers and 20 articles were selected for our analysis. To facilitate comparison with a mainstream, non-violent religious group, we collected articles from catholicwomensforum.org, an online resource catering to catholic women. We scrapped 132 articles from this domain. While this number is larger, the articles themselves are much shorter that those published by ISIS.

4.3.2 Content Analysis

The key task in comparing ISIS material with that of a non-violent group involves analyzing the content of these two corpora to identify the topics. For our analysis, we considered a simple uni-gram model where each word is considered as a single unit. Understanding what words appear most frequently provides a simple metric for comparison. To do so we normalized the count of words with the number of words in each corpora to account for the size of each corpus. It should be noted, however, that a drawback of word frequencies is that there might be some dominant words that will overcome all the other contents without conveying much information.

Topic modeling methods are the more powerful technique for understanding the contents of a corpus. These methods try to discover abstract topics in a corpus and reveal hidden semantic structures in a collection of documents. Most popular topic modeling methods use probabilistic approaches such as probabilistic latent semantic analysis (PLSA) and latent Dirichlet allocation (LDA). LDA is a generalization of pLSA where documents are considered as a mixture of topics and the distribution of topics is governed by a Dirichlet prior (α) figure 4.1 shows plate notation of general LDA structure [86]. Since LDA is among the most widely utilized algorithms for topic modeling, we applied it to our data. However, the coherence of the topics produced by LDA is poorer than expected.

To address this lack of coherence, we applied non-negative matrix factorization (NMF). This method decomposes the term-document matrix into two non-negative matrices as shown in figure 4.2. The resulting non-negative matrices are such that their product closely approximate the original data. Mathematically speaking, given an input matrix of document-terms V, NMF finds two matrices by solving the following equation [87]:

$$\min_{W,H} \|V - WH\|_F \quad s.t \quad H \geq 0, W \geq 0.$$

Where W is topic-word matrix and H represents topic-document matrix.

NMF appears to provide more coherent topic on specific corpora. O'callaghan et al. compared LDA with NMF and conclude that NMF perform better in corpora with specific and non-mainstream areas [88]. Our findings align with this assessment and thus our comparison of topics is based on NMF.

4.3.3 Emotion detection

Propaganda's effectiveness is hinges on the emotions that it elicits. But detecting emotion in text requires that two essential challenges be overcome.

First, emotions are generally complex and emotional representation models are correspondingly contested. Despite this, some models proposed by psychologists

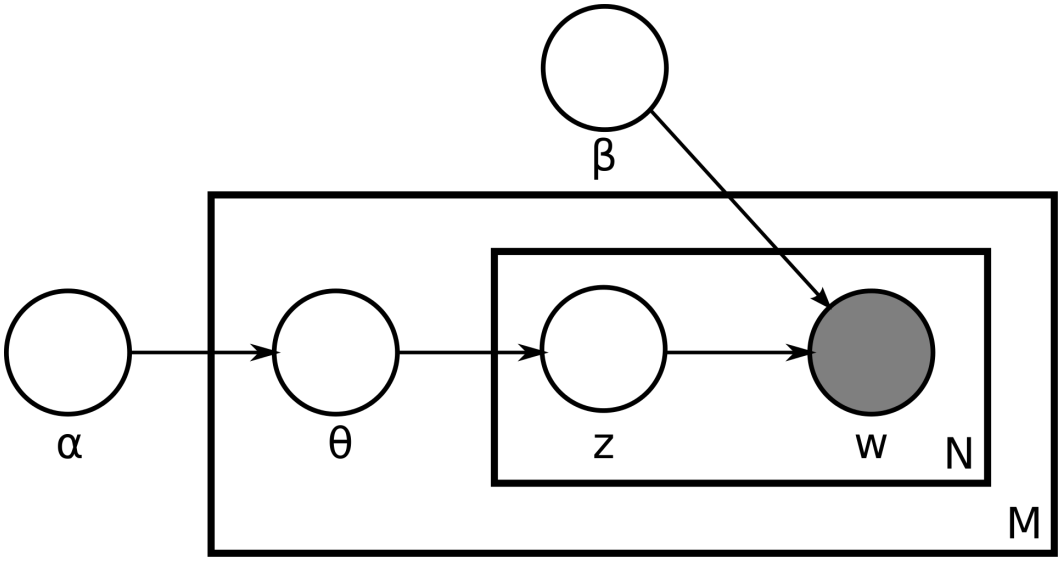

Figure 4.1: Plate notation of LDA model

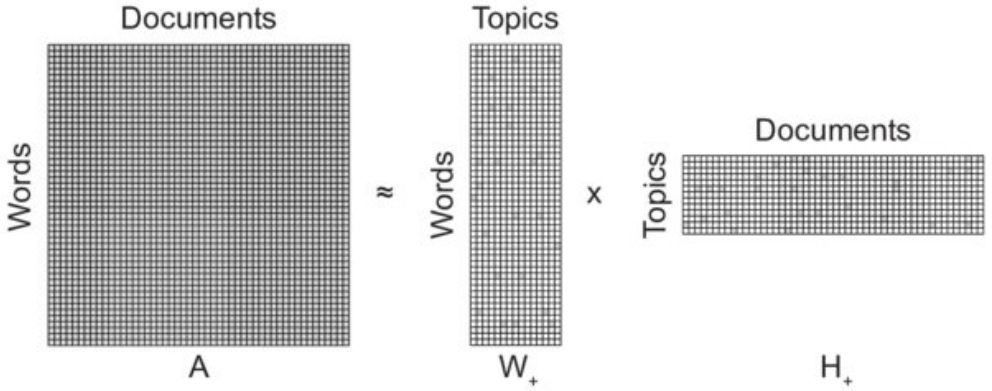

Figure 4.2: NMF decomposition of document-term matrix [4]

have gained wide-spread usage that extends to text-emotion analysis. Robert Plutchik presented a model that arrange emotions from basic to complex in a circumplex as shown in figure 4.3. The model categorize emotion into 8 main subsets and with addition of intensity and interactions it will classify emotions into 24 class [89]. Other models have been developed to capture all emotions by defining a 3 dimensional model of pleasure, arousal, and dominance.

The second challenge lies in using text for detecting emotion evoked in a potential reader. Common approaches use either lexicon-base methods (such as keyword-based or ontology-based model) or machine learning-base models (usually using large corpus with labeled emotions) [85]. These methods are suited to

addressing the emotion that exist in the text, but in the case of propaganda we are more interested in emotions that are elicited in the reader of such materials. The closest analogy to this problem can be found in work that seeks to model feelings of people after reading a news article. One solution for this type of problems is to use an approach called Depechemood .

Depechemood is a lexicon-based emotion detection method gathered from crowd-annotated news [90]. Drawing on approximately 23.5K documents with average of 500 words per document from rappler.com, researchers asked subjects to report their emotions after reading each article. They then multiplied document-emotion matrix and word-document matrix to derive emotion-word matrix for these words. Due to limitation of their experiment setup, the emotion categories that they present does not exactly match the emotions from the Plutchik wheel categories. However, they still provide a good sense of the general feeling of an individual after reading an article. The emotion categories of depechemood are: AFRAID, AMUSED, ANGRY, ANNOYED, DON'T CARE, HAPPY, INSPIRED, SAD. Depechemood simply create dictionaries of words where each word has scores between 0 and 1 for all of these 8 emotion categories. We present our finding using this approach in the result section.

4.4 Results

In this section, we present the results of our analysis based on the contents of ISIS propaganda materials as compared to articles from the catholic women forum. We then present the results of emotion analysis conducted on both corpora.

4.4.0.1 Content Analysis

After pre-processing the text, both corpora were analyzed for word frequencies. These word frequencies has been normalized by number of words in each corpus. figure 4.4 and 4.5 shows the most common words in each of these corpora.

A comparison of common words suggests that those related to marital rela-

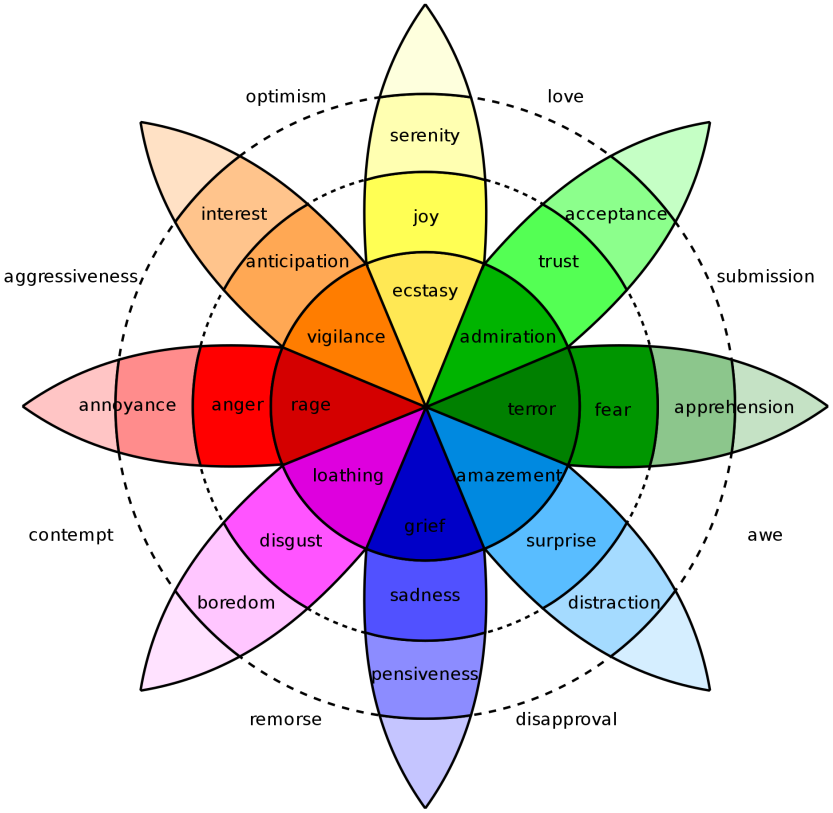

Figure 4.3: 2D representation of Plutchik wheel of emotions

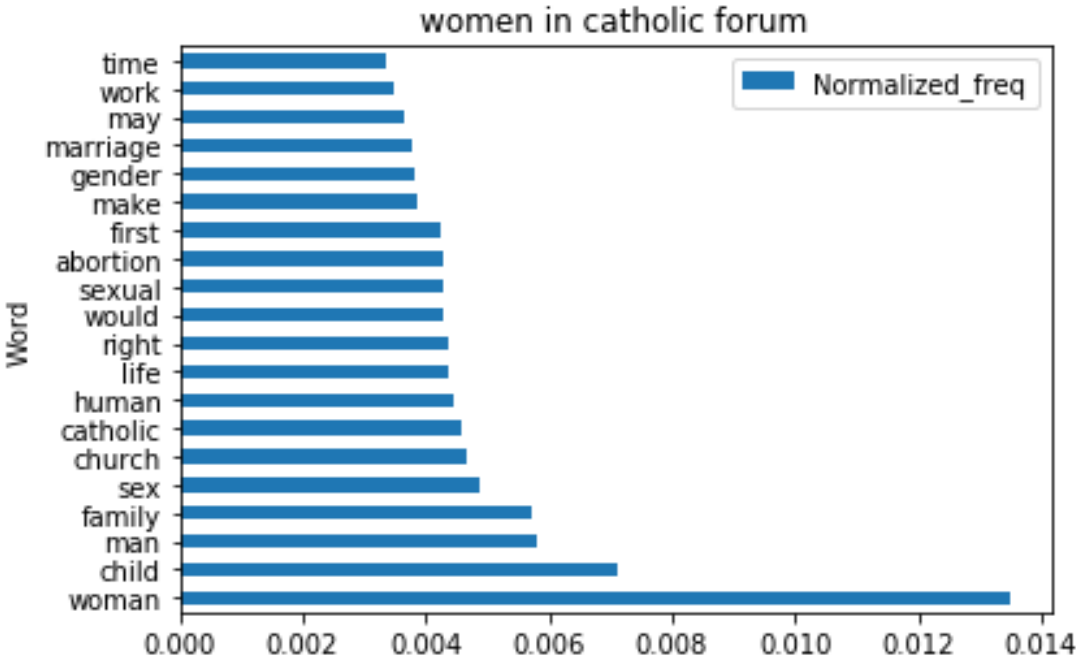

Figure 4.4: word frequencies of catholic material

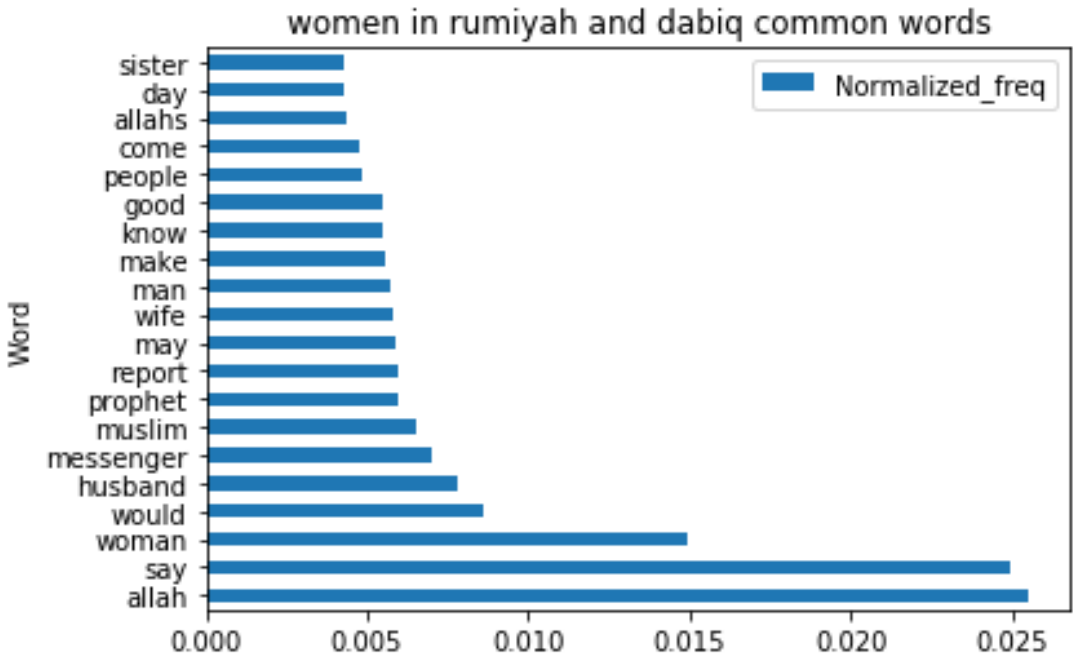

Figure 4.5: word frequencies of ISIS material

tionships (husband, wife,etc) appear in both corpus, but the religious theme of ISIS material appears to be stronger. A stronger comparison can be made using topic modeling techniques to discover main topics of these documents. Although we used LDA our result by using NMF outperform LDA topics due to the nature of these corpora. Also few number of ISIS documents might contribute to the comparatively worse performance. Therefore, we present only NMF results. Base on their coherence, we selected 10 topics for analyzing both corpora. Table 4.1 and 4.2 show the most important words in each topic with a general label that we assigned to the topic manually. Based on NMF output, ISIS articles that address women include topics mainly about Islam, women role in early Islam, and hijrah (moving to another land), spousal relations, marriage, and motherhood.

The topics generated from the catholic women forum are clearly quite different. Some, however, exist in both contexts. More specifically, marriage/divorce, motherhood, and to some extent spousal relations appeared in both generated topics. This suggests that when addressing women in a religious context, these may be very broadly effective and appeal to the audience. More importantly, suitable topic modeling methods will be able to pick these similarities no matter the

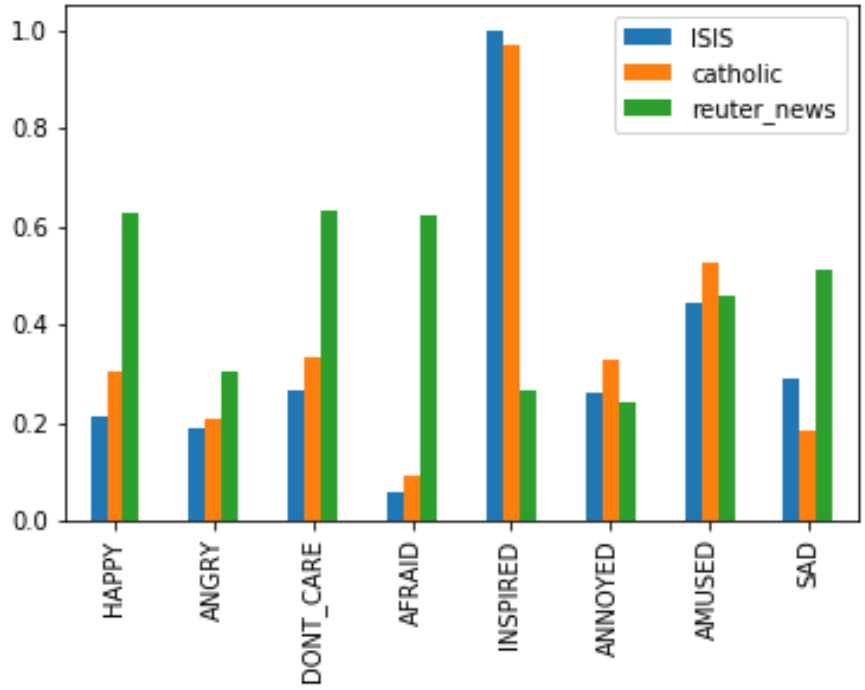

Figure 4.6: Comparison of emotions of our both corpora along with Reuters news

size of corpus that we have in hands. Although, finding the similarities/differences between topics in these two groups of articles might provide some new insights, we turn to emotional analysis to also compare the emotions evoked in the audience.

4.4.1 Emotion Analysis

We rely on Depechemood dictionaries to analyze emotions in both corpora. These dictionaries are freely available and come in multiple arrangements. We used a version that includes words with their part of speech (POS) tags. Only words that exist in the depechemood dictionary with the same POS tag are considered for our analysis. We aggregated the score for each word and normalized each article by emotions. To better compare the result, we added a baseline of 100 random articles from Reuters news dataset as a non-religious general resource which is available in NLTK python library. figure 4.6 shows the aggregated score for different feeling in our corpora. Both catholic and ISIS related materials score the highest in "inspired" category. Furthermore, in both cases, being afraid has the lowest score. However, this is not the case for random news material such as

early islam women	Islam / khilafah	marriage	islamic praying	women's life	hijrah	islamic	divorce	motherhood	spousal relationship
Topic 0	**Topic 1**	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**	**Topic 6**	**Topic 7**	**Topic 8**	**Topic 9**
said	wa	husband	masjid	worldly	islamic	wala	mourning	said	spouses
khadijah	hijrah	woman	prayer	spend	state	said	iddah	ibn	said
munafiqin	sallam	sharah	said	regards	khilafah	bara	widow	jihad	husband
iman	alayhi	said	masajid	wearing	abu	sake	home	reported	backbiting
abu	sallallhu	wives	home	mat	brothers	enmity	husband	children	listening
ibn	khilfah	wife	going	menstruation	arrived	kab	ibn	charity	wife
dunya	ab	marry	woman	prophet	knew	salam	said	child	abu
mother	sisters	ibn	ibn	life	mujahidin	remained	perfume	abu	divorce
people	radiyallhu	married	prophet	clothing	previous	prayer	woman	wealth	word
asma	ibn	jihd	hadith	small	soon	shariah	away	flock	problems
husband	qurn	duny	reported	lived	children	muslims	night	muslim	instead
bakr	islamic	say	prevent	time	later	affection	wear	cause	backbite
believers	praise	permitted	men	family	informed	islam	widows	albukhari	relationship
hearts	state	lawful	default	aishah	hijrah	anger	house	prophet	tongue
killed	said	prohibited	muslim	garment	shariah	aqidah	sleep	policy	woman
know	slavegirl	lord	leaving	despite	dua	good	married	shepherd	person
steadfast	land	islam	abu	follow	journey	religion	clothing	soul	brother
sumayyah	firm	fear	stay	concern	returned	return	ab	waging	home
sisters	people	man	wives	living	umm	state	pregnant	lord	hind
spreading	lands	sister	shariah	food	leave	relatives	husbands	mother	narrated

Table 4.1: NMF Topics of women in ISIS

	feminism	law	gender identity	marrage/ divorce	church	motherhood	birth control	life	sexuality	parenting
	Topic 0	**Topic 1**	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**	**Topic 6**	**Topic 7**	**Topic 8**	**Topic 9**
	women	abortion	gender	marriage	church	mom	humanae	human	sex	parents
	abortion	court	lgbt	divorce	catholic	god	pill	social	sexual	school
	pro	constitutional	identity	family	bishops	mary	vitae	ecology	men	children
	men	circuit	transgender	children	pope	christ	contraception	work	regnerus	schools
	life	federal	ideology	marital	synod	mother	control	revolution	cheap	federalist
	feminist	decision	trans	marriages	priests	jesus	birth	people	consent	education
	march	abortions	sex	divorced	francis	like	women	family	women	students
	feminists	state	sexual	spouses	women	baby	contraceptives	moral	desire	transgender
	woman	kansas	male	annulment	rome	love	health	political	porn	child
0	feminism	supreme	reality	families	letter	motherhood	effects	politics	weinstein	reading
1	equality	law	female	spouse	christ	child	mandate	man	market	kids
2	female	medicaid	activists	married	mccarrick	world	fertility	dignity	marriage	youth
3	choice	roe	catholic	abandoned	vatican	life	iud	person	mating	girl
4	male	constitution	biological	love	bishop	charlie	catholic	ecological	power	family
5	movement	judge	dysphoria	catholics	holy	children	contraceptive	society	pornography	gender
6	vulnerable	case	person	church	god	light	sanger	nature	fly	confused
7	sex	planned	agenda	fidelity	faithful	son	medical	world	good	continue
8	today	parenthood	people	older	faith	little	prevent	liberty	like	boy
9	time	right	orientation	situations	priesthood	got	sexual	self	risk	public
0	human	government	report	husband	authority	day	free	middle	kind	news

Table 4.2: NMF Topics of women in catholic forum

Table 4.3: Words with highest inspiring score

[width=/10+2, height=0.8cm] Words Group	**Catholic**	**ISIS**
$Word_1$	avarice	uprightness
$Word_2$	perceptive	memorization
$Word_3$	educationally	merciful
$Word_4$	stereotypically	affliction
$Word_5$	distrustful	gentleness
$Word_6$	reverence	masjid
$Word_7$	unbounded	verily
$Word_8$	antichrist	sublimity
$Word_9$	loneliness	recompense
$Word_{10}$	feelings	fierceness

Reuters corpus, which are not that inspiring and, according to this method, seems to cause more fear in their audience. We investigate these result further by looking at the most inspiring words detected in these two corpora. Table 4.3 presents 10 words that are among the most inspiring in both corpora. The comparison of the two lists indicates that the method picks very different words in each corpus to reach to the same conclusion. Also, we looked at separate articles in each issue of ISIS material addressing women. Figure 4.7 shows emotion scores in each of the 20 issues of ISIS propaganda. As it can be seen, in every separate article, this method gives the highest score to evoking inspirations in the reader. Also, in most of these issues the method scored "being afraid" as the lowest score in each issue.

4.5 Conclusion and Future Work

In this paper, we have applied natural language processing methods to ISIS propaganda materials in an attempt to understand these materials using available technologies. We also compared these texts with a non-violent religious group (both focusing on women related articles) to examine possible similarities or differences in their approaches. To compare the contents, we used word frequency and topic modeling with NMF. Also, our results showed that NMF outperform LDA due to the niche domain and relatively small number of documents.

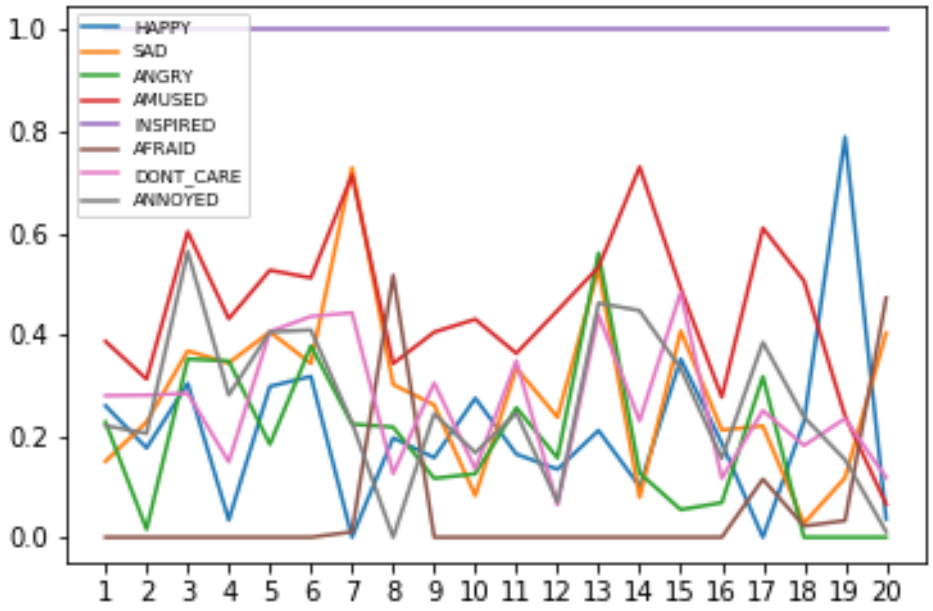

Figure 4.7: Feeling detected in each issue of ISIS material (first 7 from Dabiq last 13 from Rumiyah)

The results suggest that certain topics play a particulary important roles in ISIS propaganda targeting women. These relate to the role of women in early Islam, Islamic ideology, marriage/divorce, motherhood and spousal relationships and Hijrah (moving to a new land).

Comparing these topics with those that appeared on a Catholic women forum, it seems that both ISIS and non-violent groups use topics about motherhood, spousal relationship, and marriage/divorce while they address women. Moreover, we used depechemood methods to analyze the emotions that these materials are likely to elicit in readers. The result of our emotion analysis suggest that both corpus used words that aim to inspire readers while avoiding fear. However, the actual words that lead to these effects are very different in the two contexts. Overall, our findings indicate that, using proper methods, automated analysis of large bodies of textual data can provide novel insight insight into extremist propaganda that can assist counter-terrorism community.

Chapter 5

Semi-Supervised Natural Language Processing in Social Domains

In previous chapters, we describe a supervised NLP method based on deep learning as a decision support tool in the transportation domain and an unsupervised approach for data-poor domains to create valuable insights in the counter-terrorism domain. In other problems of public domains, we may have a large amount of text, but acquiring labels would be either very costly or not feasible. In these types of problems, one can consider applying semi-supervised NLP-based approaches to use on these texts. An example of these types of problems is extracting skills from job descriptions to decrease the skill gap in the labor market, which is presented in this chapter.

5.1 Introduction

In today's job market, identifying the required skills for a position is crucial. Employers would like to recruit talent that would fulfill most of the requirements for the position in hand. Similarly, individuals who are looking for jobs would like to prepare themselves by acquiring the most relevant skills for the job they seek. One

solution for identifying these skills is by using available job advertisements for specific types of jobs. The idea is to develop an information extraction methodology from such unstructured data resources.

The problem becomes more important when the skill set required for a job can vary a lot, and new technologies appear every day in that field. Data science's job market is one such area where many different skills might be required depending on the job specification. Furthermore, it is an evolving area with constant breakthroughs that add new skills to the data scientist's required skill pool. This work focuses on data science as the main domain and tries to identify the hard skills required based on online job advertisement postings as the main source of information. We collected over 100K unique job descriptions from online job postings that are related to data science (data analyst, data scientist, and machine learning engineer). We used this data as our corpus to build and investigate models to identify the required skills of a data scientist.

Designing an automated skill extraction solution is a challenging task. There are two main reasons for this difficulty:

- There is no clear guideline on what can be considered as a hard skill for a candidate in data science. A skill could be a single word or a combination of words.

- Information Extraction (IE) [91] Specifically, expertise retrieval [92] is inherently considered a challenging problem. In text mining and Natural Language Processing, IE focuses on specific Name Entity Recognition, relationship extraction, and coreference resolution tasks. Unfortunately, skills extraction would not fit into any of these categories. Furthermore, most supervised and unsupervised machine learning methods for text can not be used directly to extract skills.

In this work, we proposed an approach using Natural Language Processing tools by combining vector embedding with part of speech tagging to identify skills

related to data science. This method relies on the similarity of words or phrases in the vector space. This approach uses a handful of skills as the seed and outputs words/phrases as candidate skills in this domain. The vector space is generated by training two main word embedding algorithms (GloVe and Word2Vec).

The idea is that these algorithms work in such a way that the embedding would capture the semantic and syntactic similarities. This is proven to be useful as the words that appear near each other or can be replaced in the context would have similar representations. In the context of job descriptions, skills usually appear in similar places, and therefore, these vector representations tend to capture this reality. To evaluate our approach, we used a sample of job descriptions that were labeled using Amazon Mechanical Turk as the ground truth. The method uses multiple hyperparameters at different stages, and we evaluated their effect using the ground truth.

A benefit of using such an approach is automating this task without the need for human intervention. This means that new sets of skills will be automatically extracted using the same seeds. A significant drawback is that the model needs to be periodically retrained on the latest data to extract emerging skills. However, the process of training itself would be relatively easy. We will explain the details of this approach in the following sections.

5.2 Related Work

As previously mentioned, identifying skills in an area has been an important step in helping understand the labor market. Understanding these skills will help to close the gap between academic training and job market needs. As a result, both recruitment industries and researchers attempt to address this need by proposing different solutions. These solutions range from expert annotating skills to using machine learning algorithms to identify and extract skills.

The importance of understanding job market requirements and skills has resulted in the development of generalized frameworks to identify requirements and

skills supported by governmental institutes. An example of these frameworks is O*Net [93]. O*Net is sponsored by the U.S. Department of Labor and lists categories of jobs, along with the technologies and tools associated with them. A similar framework, ESCO, uses a similar approach, supports 27 languages, and was developed for the European labor market. Although generalized, these databases use human resource experts to list the skills required for each job, thus demanding rather expensive resources to build and maintain them over time. This drawback and the availability of job market data on the web, encouraged researchers to propose alternative solutions.

Researchers have employed other approaches to identifying the necessary skills in online job markets by using standard lists of skills. Such skill set lists (a.k.a skill bases) can be used by other researchers to create insights into the different aspects of labor markets. Papoutsoglou et al. [94] used the list of tags in StackOverflow to identify hard skills in online job advertisements. They use these skill lists and exploratory factor analysis to find the correlation between required skills and identify patterns. In another research on the IT security job market, Brooks et al. used a pre-defined list of skills to identify the dominance and popularity ranking of job advertisements in that domain [95].

Building skill bases usually requires significant efforts, including using web crawling and other extra steps to refine the lists of required skills. Hoang et al. collected 60 million resumes and 1.6 million job postings, then focused on the technical skills mentioned in those resumes and compared them with the requirement sections of job postings [96]. They refined their input by removing noise and utilizing Wikipedia API to improve the final list of skills. A common method to build skill bases is to use external resources such as Wikipedia and DBpedia. DBpedia is an open data project available on the web that extracts structured information from Wikipedia [97]. Malherbe & Aufaure used linked data from DBpedia and tags from StackOverflow and a Q & A website to build knowledge graphs [98]. They further break these knowledge graphs into sub-graphs to extract

skill bases from candidate profiles. Similarly, Zhao et al. [99] used Wikipedia API on candidate profiles to build a skill base for hard and soft skills. Despite the efforts mentioned, most of these research projects did not rely mainly on machine learning and Natural Language Processing (NLP).

Recent breakthroughs in artificial intelligence and deep learning methods have resulted in considerable success in several areas of NLP for developing unstructured text solutions. The problem of skill extraction can be mapped into both supervised and unsupervised machine learning problems to tackle it differently. Mainly, skill extraction can be done using two different concepts of supervised machine learning and NLP. The first approach frame the problem as a multi-label text classification where labels are the skills and train a model to predict the existence of skills in a job post. Sayfullina et al. used this approach by using LSTM and Convolutional Neural Networks (CNN) on job postings and a collection of resumes [100]. The emergence of new state-of-the-art models such as Bidirectional Encoder Representations from Transformers (BERT) [101] encouraged using this model for other domains such as skill extraction. Tamburri et al. [102], and Bhola et.al [103] used this model for skill extraction as a multi-label text classification.

Another approach is to frame the problem into a Name Entity Recognition (NER) problem in the text. NER is a known problem in NLP and has been a hot topic for research. However, skills in the format they are presented in the job advertisements will not fit into known categories of NER. Nevertheless, a deep learning model can be trained to look for these words as named entities, then used to extract the skills later on. Jia et al. [104] used this approach with an LSTM model for Chinese skill extraction.

Despite being promising, supervised methods require a large set of high-quality labels that should be labeled by domain-expert annotators. Because of this, Unsupervised methods have also been researched for skill extraction problems. A common method for addressing the problem of extracting insights from unstructured text is topic modeling. Debortoli et al. [105] used Latent Semantic Analysis (LSA)

to identify and compare the most sought skills in job advertisements. De Mauro et al. [106] used Latent Dirichlet Allocation (LDA) to identify the most popular keywords referring to skills. One drawback of using such an unsupervised method is the difficulty of evaluating the result, and usually requires human judgment.

Recent advances in NLP bring new hope for creating alternative unsupervised methods that can be used in different domains. One such breakthrough was the introduction of word embeddings. The idea is to leverage word embeddings of skills smartly. Subsequent studies have used skill embeddings to create clusters of skills that appear together [96]. In another research, Smith et al. [107] used the Word2Vec algorithm with synthetic headwords to identify skills in a job-matching platform. The immense potential of these skill embedding experiments motivated us to use the semi-supervised approach based on skill embeddings, and to investigate the effects of using different parameters in the performance of such methods. The details of this approach are given in the following section.

5.3 Method

We describe the details of our method in this section. Our method is constructed of mainly three parts. First, we start with pre-processing the raw input, then describe different embeddings, and finally, explain our approach to find similarities between the embeddings and skills identification.

5.3.1 Pre-processing

It is a common practice to perform pre-processing steps on the text before working with it in a downstream task. These steps usually include removing punctuation and stop words, stemming, and lemmatization. In the context of skill extraction, stemming and lemmatization are not preferred, as they will change the words, therefore affecting skills. Similarly, removing stop words reduces the context around skills and as a result, affects embedding vector representations. Despite not requiring these pre-processing steps, designing a skill extraction solu-

tion requires overcoming other challenges. One main challenge is that skills often are not a single word but a combination of words. Most NLP tokenizers in their default version treat each word like a single token and would create complications later on in the process. One alternative is to generate multiple n-grams and construct tokens based on these n-grams.

The n-gram approach, however, introduces many tokens that are not clearly mapped to a meaningful entity, and thus it requires other steps to reduce the amount of noise these n-grams would generate. We utilize Part of Speech tagging with some rules to overcome this problem. The intuition here is that skills can be considered nouns or noun phrases that refer to an entity. First, we tag all the words using a Part of Speech tagger. Next, we build a set of grammar rules to combine the words according to those rules and treat the new phrases as new words. This process is also known as chunking, and by using this process, we reconstruct entities such that tokens can be multiple words such as "Natural Language Processing" for the next step of the algorithm.

5.3.2 Static Embedding Representation

Embedding representation became the most powerful type of word representation of the past decade. The idea is to create vector representations of words based on their context. It relies on the hypothesis that words appearing in the same context tend to have similar meanings. In NLP, this is also called "Representation Learning", which is commonly a self-supervised task. Furthermore, due to the formulation of this approach, it can also capture the syntactic properties of words. Therefore, these representations are good candidates for skill extraction tasks as the context is similar, and skills tend to appear next to each other in most job advertisements. Here we briefly explain Word2Vec and GloVe algorithms as the two primary embedding representations that are used in this work.

Word2Vec: Mikolov presented an algorithm to learn context-based presentation for words as semantic vectors. The approach used two different neural

network self-learning models: Continuous Bag of Words, where given the context words, the algorithm predicts the missing word in the context; and Skip-Gram, where given the word, the model tries to predict the surrounding words in a given window around it. Figure 5.1 shows an illustration of this approach that can be used for building proper representation for skill phrases with a window size of 2. The size of the window that moves over the context significantly affects the generated representation for tokens. Large sizes (closer to 10) would produce a more generalized topical representation. Therefore, it is preferred to keep the window reasonably small to capture the semantic and syntactic properties of the tokens. During the training process, the target word is fed as one-hot encoding to the network, and using the first embedding matrix, the values for the hidden layer are computed. Next, these values go through another embedding matrix in the neural network that tries to maximize the probability of one-hot encoding of the context, as shown in figure 5.2. After training, the algorithm generates two sets of embedding (W1, W2) while using cross entropy as the objective function with output softmax. The usual approach is to average over these two matrices to generate embeddings for tokens. A faster alternative optimization for the same problem uses negative sampling, where a classifier predicts correct words. **GloVe(Global Vector for Word Representation)**: GloVe is another powerful static word embedding developed at Stanford University. In a similar approach to Word2Vec, it generates vectors that capture more global representations. The model uses the co-occurrence of words in the context window, and combines the intuition of Word2Vec with count-based models like Positive Pointwise Mutual

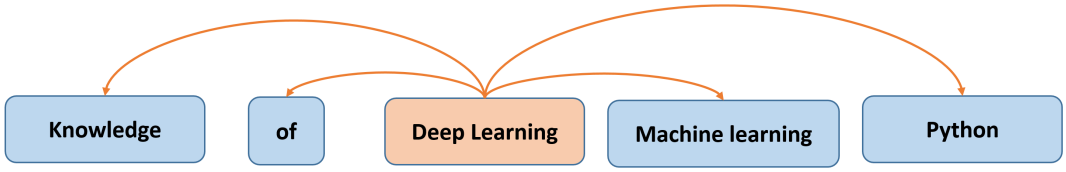

Figure 5.1: Skip gram for skill representation learning with a window size of 2

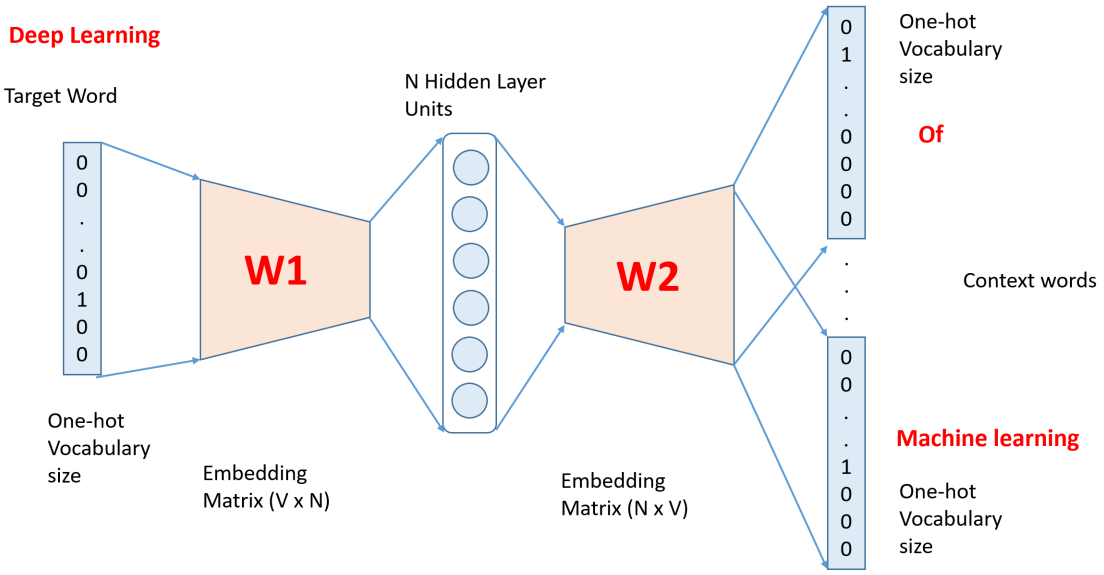

Figure 5.2: Visualization of skip gram neural network and the embedding matrices (W1,W2)

Information (PPMI). Therefore, the objective function to optimize is based on the co-occurrence values as shown in the following formula:

$$J(\theta) = \frac{1}{2}\Sigma_{i,j=1}^{W} f(P_i j)(v_i^T u_j - log(P_i j))^2 \tag{5.1}$$

where $f(x)$ is a linear function that maximizes to 1 for larger values and $P_i j$ is the value of the co-occurrence of words i and j in our co-occurrence matrix. Other embedding vector representations also exist that try to generate contextual word embedding, such as BERT embedding, but those embeddings are learned based on a huge corpus, and thus fine-tuning them for a task like this would be problematic, and therefore we have focused on these two embeddings for our method.

5.3.3 Similarly-Based Skill Selection

In this work, the semi-supervised skill extraction method uses the similarity of word embeddings for a few known skills and extracts words or phrases with the most similarity. A common method to measure similarities in vector space is cosine similarity. Due to the high dimension of embedding vectors, cosine similarity is

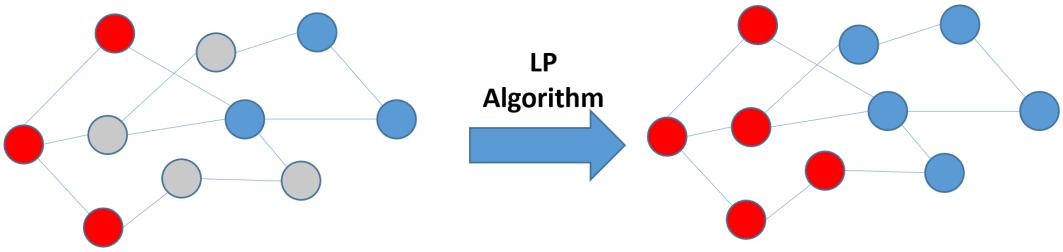

Figure 5.3: Illustration of label propagation algorithm

preferred, and in practice, works well. Another approach that we examined is to use the graph-based semi-supervised algorithm to label the unknown data points.

Graph Approach with Label Propagation: Label Propagation uses a subset of positive and negative labeled data. It propagates these labels through the network using a semi-supervised algorithm and in the process, assigns new labels to unlabeled data. Figure 5.3 illustrates the label propagation process. In the skill extraction context, we can utilize label propagation by defining a few known skills as the positive subset and some non-skills words as the negative subset. Next, we use the embedding vectors as the representation of words in each node in the network and label the unknown words as either a skill or not a skill.

Threshold-based selection on cosine similarity: Alternatively, we can use a threshold for cosine similarity of our few known skills with the rest of the tokens and select tokens with the highest similarities as our new skills. The following algorithm combines all the pieces and uses the threshold selection to identify the new skills. The above method has multiple parameters that might influence the performance. Besides the threshold, the embedding vector's size and the embedding type might affect the result. We will present these effects in the result section.

5.3.4 Evaluation

In order to evaluate the performance of such a skill extraction method, a validation set of job descriptions with their corresponding skills is needed. It is common to use crowd-sourced labeling to produce a gold standard for evaluation. One platform

Algorithm 1 Threshold based skill extraction

Input: $Skills = list\ of\ known\ skills$
Step 1: Run POS on all texts
Step 2: Use chunking to reconstruct tokens including phrases
Train an embedding algorithm on tokens and generate V_t
for S in Skills **do**
  for T in tokens **do**
    if $cos(V_S, V_T) \geq thr$ **then**
      append T to Skills
    end if
  end for
end for
return Skills

that allows crowd-sourced labeling is the AWS ground truth platform. It allows us to either use Mechanical Turk to ask the general population to label data or assign a group of annotators to label the data. Identifying skills is not easy; using mechanical turk with ordinary people would not provide accurate labels. For this task, an expert familiar with such skills is required. AWS allows performing skill labeling through its text multi-label labeling task. After labeling, AWS generates a JSON file containing the span of labels, and thus the labels (skills) can be extracted from that.

Computing the performance of such a skill extraction requires a slightly different definition. In this context, accuracy (the number of correct skills and correct non-skills) would be misleading due to the large number of non-skills in the input tokens. Therefore, we define metrics that focus only on skills identified using precision and recall defined as follows:

$$Precision = \frac{\#correctly\ identified\ skills}{\#total\ number\ of\ true\ skills}$$

$$Recall = \frac{\#correctly\ identified\ skills}{\#total\ number\ of\ extracted\ skills}$$

In the next section, we will use these metrics to present the result and investigate

the effect of hyperparameters on performance.

5.4 Empirical Results

This section describes the empirical results of our method. To build the corpus, we used approximately 100K job descriptions that were collected in a span of a year from the U.S. job market. This corpus went through all the different processes explained in the previous section to generate candidate skills. Furthermore, a random 100 job descriptions were labeled in AWS ground truth multi-labeling task as the gold standard.

Semi-supervised Label Propagation: Our effort in using label propagation proved to be insufficient as the algorithm implementation failed to converge on our embedding vectors as its inputs. We suspect this is due to the very few positive samples (known skills) and the high dimension of vector embedding space.

Threshold-based skill extraction: The threshold-based method is another semi-supervised approach that we used for skill extraction. We used a few (six known skills) as the starting skills for the algorithm to extract other skills based on their similarity. Here, we also limited the extracted skills to the most frequent five hundred extracted skills in our corpus. The performance of the skill extraction method based on the similarity threshold can be affected by multiple parameters, from the threshold to the embedding used to generate the embedding. We will compare the performance metrics mentioned on our gold standard validation set.

1. Effect of Threshold: An important parameter of the method described in this work is the choice of the threshold for the cosine similarity of word embedding with known seeds. Our analysis showed that the best threshold would be in the 0.5-0.75 range, and increasing the threshold increases the precision (correctly picking skills) while decreasing the recall (better overall skill suggestion). Figure 5.4 shows this effect for the Word2Vec model.

2. Effect of Embedding Model: As mentioned, we used both Word2Vec and GloVe as the static word embedding models to generate the word vectors for

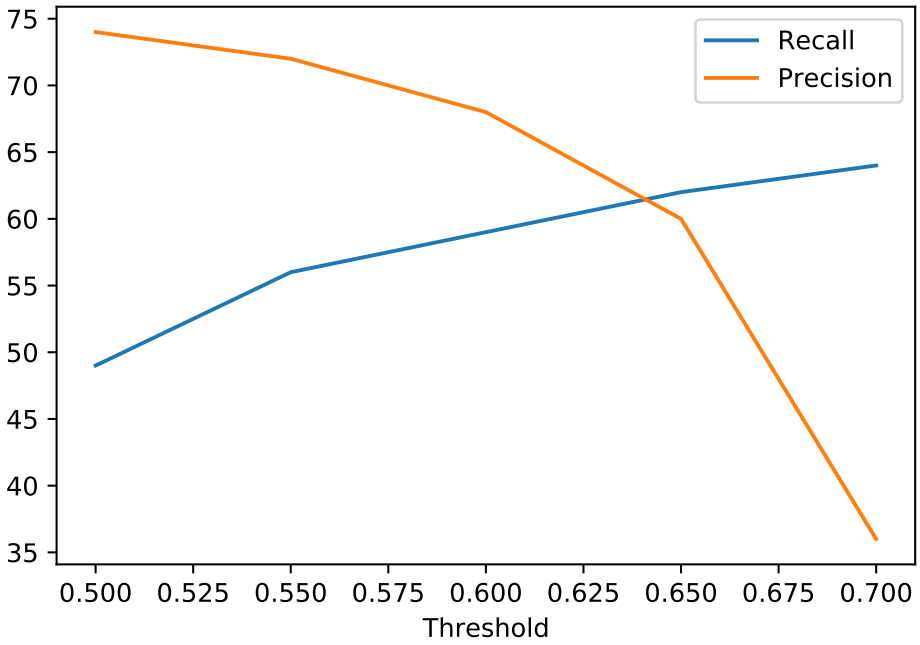

Figure 5.4: The effect of increasing threshold for Word2Vec model trained on full 100k data (by percentage)

later comparison. The result of our experiments shows that in general Word2Vec provides better results in terms of both precision and recall. Table 5.1 shows the result of the GloVe embedding vectors with different thresholds.

Table 5.1: Comparison of GloVe Embedding with Best Word2Vec performance

Model	Threshold	Precision	Recall
GloVe	0.8	37%(12%)	75%(12%)
GloVe	0.7	48%(13%)	47%(9%)
GloVe	0.6	63%(12%)	40%(7%)
Word2Vec	**0.5**	**74%(10%)**	**49%(9%)**

This is not surprising as the Word2Vec, in theory, performs better for local and syntactic similarities, which skill extraction tries to use, and global representation might reduce such similarities. Therefore, we used Word2Vec embedding to compare other parameters for the rest of our analysis.

3. Effect of Embedding Size: Another parameter that might affect the performance of the skill extraction is the size of the embedding vector. Our result shows although it has an effect on the result, the effect is not monotonic. Table 5.2

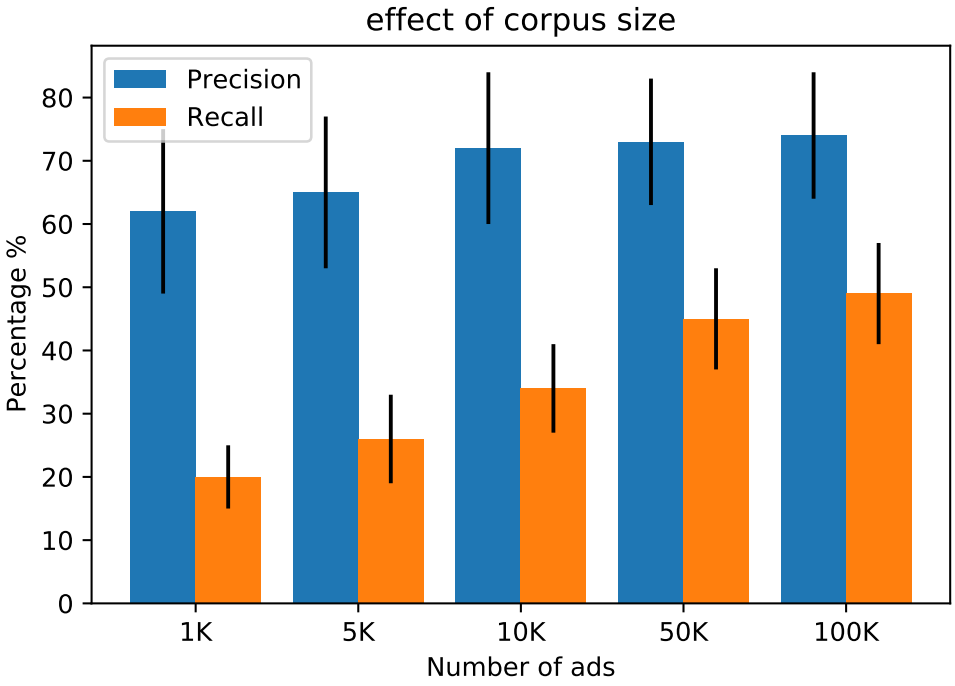

Figure 5.5: The effect of corpus size on precision and recall percentage

shows the result of using different embedding sizes.

Table 5.2: Effect of embedding size of Word2Vec on precision & recall

Embedding Size	Precision	Recall
300D	64%(11%)	59%(10%)
100D	74%(10%)	49%(9%)
60D	73%(11%)	46%(7%)

4. Effect of Corpus Size: Another interesting parameter to investigate is the size of our data set for training the embedding vectors. We investigated this effect by randomly selecting different subsets of the original data and training embeddings using these subsets. Figure 5.5 shows the effect of corpus size on the precision and recall of our method.

As can be seen by increasing the size of the corpus, both precision and recall have been improved. However, the precision stays relatively similar between 10K-100K ads, while the recall keeps improving. This suggests that by increasing the size of the training corpus, the number of correctly predicted skills from the total would not improve after a while. On the other hand, the quality of skills generated

gets better, and less noise would be introduced to the extracted skills.

5.5 Conclusion and Future Work

Skill extraction is inherently challenging, as there is no solid definition of what can be considered a skill. In this work, a semi-supervised skill extraction method based on embedding has been presented, and its performance has been evaluated. The method uses the cosine similarity of a few known skills with the other words or phrases and then selects new skills based on a threshold. The examination of two embedding methods showed the Word2Vec model performs better than GloVe embedding.

The effect of threshold criteria, embedding size, and the size of the training corpus were also investigated. The result shows that a larger sample size leads to better performance in general, although increasing the sample size alone will not improve the results. Furthermore, it is concluded that although the label propagation method seemed promising, as in the present format, it is not useful for this task due to the large vocabulary and vector space dimension. The result of this work shows that such models can capture many skills and can be used as an assistant system to extract skills automatically. This will allow a fast improvement over doing the skill extraction using human judgment.

Chapter 6

NLP-based Application

6.1 Introduction

In 2012, Harvard business review published an article "Data Scientist: The Sexiest Job of the 21st Century" and in it, authors predicted that certain sectors will face a shortage of data scientists in the near future [108]. Soon after, universities were scrambling to design programs for data science to appropriately fill the gap. However, the term "Data Science" is confusing and without a proper understanding, it will fade away into a new buzz word. Although data science is claimed to be driven from statistics as it uses statistical methods (exploratory analysis, machine learning, reproducibility, etc), the two fields are not the same and in a sense "Data Science" encapsulates a broader scope [109]. Data science is more intertwined with other important fields such as big data, and artificial intelligence and often deals with heterogeneous and unstructured data such as text, image and video [110], [111].

Another interesting aspect of this new field is related to the growth rate of data science-related jobs. According to linkedin economic graphic, machine learning engineering, and data science were the top emerging jobs between $2012 - 2017$ with a growth rate of $9.8X$ and $6.5X$ respectively [112]. According to Glassdoor it was the number one best job in the United States in 2019 and the third one in 2020. The expectation is that the need will still increase but the exact growth

percentage is not known. These reports indicate that data science is particularly a fast pace growing field with new technologies being introduced to it every day. Therefore, it is important to observe the job market to stay agile in such a fast pace field.

Other motivation to introduce a framework such as the one presented here includes figuring out different branches of data science as a broad term in job search engines and different skills associated with each of these categories. Mostly,individuals can not have fluency in all of these skills and thus prioritizing the skills needed for an individual based on the path they want to pursue as a data scientist would be crucial.

This paper presents a framework to understand data science based on its job market available positions. The proposed framework allows to both look into temporal and spatial changes of data science job market in the United States and the scope of the field and related skills needed for job placements. Moreover, using this approach, the hope is to get a skill base definition of data science through the industrial requirement lens. The rest of this paper is organized as follow: Section 6.2 presents the related works, Section 6.2.3 discusses our method and Section 6.3 presents the result. Finally, Section 6.4 discusses future works.

6.2 Related Works

In this paper, we combined multiple data science tools to build a system capable of understand spatial and temporal distributions of data science skills in the US. Before delving into our approach, we investigated solutions to related problems from different perspectives. In the first section, we describe the work on market analysis and more specifically data science market analysis. Next, text mining methods for job advertisements will be discussed.

6.2.1 Job Market Analysis

During the past decades, the internet has gradually become the first place to look for jobs and as a result, more and more job advertisements appear on the web. These collections of job ads themselves have become valuable resources in order to investigate what requirements different jobs might have and how industries generally perceive those positions. As an example, Daneva et al. used online job market advertisements in the Netherlands to understand what industry means by "requirements engineers" [113]. Although they had a small corpus and did not utilize text mining methods, they were able to present skills, competencies, and preferred background for these positions. Other researchers used job advertisements to investigate trends in the qualifications and responsibilities of Electronic Resources Librarians in a period between 2000–2012 [114]. At the same time, the importance of this information-rich source for the industry itself has not been ignored. Multiple companies and start-ups shaped around gathering and creating insights from the job market data. The work described in this paper aims to provide a platform for both finding a skill based definition and understanding the trends in the data science job market.

6.2.2 Data Science Market Analysis

Data science has a unique position among the IT jobs because of particular characteristics of the field. First, data scientists are required to have a very wide range of skills, and usually come from different backgrounds and are recruited by different sectors. On top of that, the field is a fast-moving field with new tools and technologies introduced to it each day. Therefore, it is reasonable for researchers to investigate this domain further and the result could help in understanding the field.

In an article published in 2017 by IBM researcher *Steven Miller*, the authors addressed how disruptive data science is in the labor market. They analyzed job market advertisements in all categories of Data Science jobs in 2015 for both

skillsets and recruiting sectors [115]. They also made an analysis of the growth rate of skills and projected the number of jobs for the coming years. As previously mentioned, this information can be used to guide educational institutes to build a better curriculum for their data science degrees. *Manieri et al.* took this approach in identifying the needed skills based on a corpus of 2500 job ads on data science in 2015. They used principal component analysis on their data for dimension reduction and their result identified programming skills, big data skills, database knowledge ,and machine learning as the main components that emerged from data science job markets [116]. Moreover, researchers combined courses offered and job advertisements for data science in an effort to develop semi-automatic service to analyze and detect the gaps in demand and supply of skills [117]. Although these researches tackled different problems in the data science domain, the work presented here is different from them due to its emphasis on the temporal and spatial aspect of this market and skills and providing a visual quantitative guide for individuals .

6.2.3 Text Mining Methods for Job Ads

Since job posting provides a rich body of information for market analysis, researchers have used it for understanding job markets around the world. A fruitful analysis of these postings however requires applying some sort of text mining methods to these collections. In particular, matching jobs and skills were of interest to these researches. Such skill requirements matching has been researched in different domains such as big data [118], information technology [119], and librarians [120] to name a few. The text mining methods in these research vary widely from simple frequency counts of terms to using external resources in conjunction with the job postings. For example, [119] used term frequency of terms and after cleaning the corpus from spare words presented the top 30 words in both job titles as well as job descriptions. In [121], the author used a demand skill index which is derived by normalization of words' term frequencies. In another research, they used term

frequency - inverse document frequency to rank the top terms [122]. A problem with this approach is that terms are not necessarily equivalent to the skills and there would be a need for post-processing of the result to remove other noises. Another popular method for this task is using occupational taxonomies that are created by organizations such as ESCO, ISCO or O*NET [123]. As an example, Karakatsanis et al. used O*NET occupation descriptions and job postings while applying Latent Semantic Indexing (LSI) to both and using cosine similarity between the results [124]. other researchers used techniques such as LDA or word2vec to classify the job postings into ISCO or O*NET classes [125, 126]. The problem with these taxonomies is the constant need for updates by experts [127]. However, depending on the quality of the generated lexicon, this approaches will yield acceptable results. Other researchers used algorithmic approaches such the work of Mirjana Pejic-Bacha et al. where they use phrase clustering with Jaccard coefficient for distances on industry 4.0 job advertisements [128]. Other researchers used innovative methods such as ontology-based or graph-based models(using hyperlink Wikipedia graph) to extract skills [129, 130]. For our implementation on this framework, we build an external lexicon specific to data science skills using

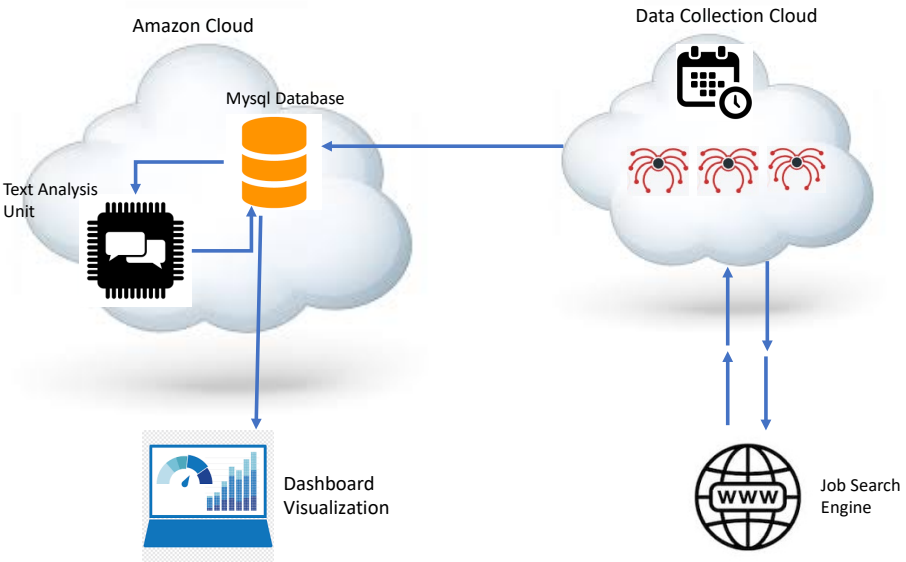

Figure 6.1: Overview of data science skill tracking system

information on the web that will cover most of skills and concepts of this field for further analysis. In the next section the details of this approach will be presented.

sectionMethod The framework presented in this work consists of multiple subsystems. Mainly, there are three modules that should interact with each other seamlessly by passing relevant data. The first module is responsible for data collection, the second module processes the collected data, in this case job descriptions from the web and finally, the last module will provide an interface to the insights gathered and provides visualizations for users to examine temporal and spatial distribution of data science related jobs and skills. Figure 6.1 shows an overview of the proposed system. Each of these modules will be described in more detail in the following. As it can be seen in Figure 6.1 , this system heavily relies on cloud computing structures that provide stable facilities for continuous tasks with scheduling to collect, analyze, and present its findings dynamically.

6.2.4 Data Collection Module

The first module of this framework handles the data collection for the system. The purpose of this unit is to continuously collect a consistent sample of job advertisements from the web. To collect the job posts, the module uses Scrapy which is a very powerful python Framework able to handle different aspects of data collection by its object-oriented implementation using spiders. Spiders are web crawlers that follow the links and can parse the result and extract the requested part at the same time.

These spiders return the results of queries for the following terms: “data scientist”, “data analyst”, and “machine learning engineer” in job search engine. These three queries were selected based on the main tracks that individual want to pursue in this field. Nevertheless, the returned results covers a very broad spectrum of job positions including more specific jobs such as AI, Computer Vision, or NLP specialist to name a few. In addition, the results not only include the job descriptions which are used for skill extraction but also includes metadata from

job advertisement posts allowing temporal and spatial mapping of the result of queries for final visualization. Table 6.1 shows samples of the returned fields for queries on data analyst. For a framework like the one presented, it is not feasible to use a local machine for data collection as it requires to work continuously for a long period of time with no interruptions (such as internet disconnection or power outage, etc.). Such a machine and maintaining its connectivity would cause a lot of hassle and scrapy cloud services solve this problem by understanding the data collection framework. It allows to create periodic jobs to continuously collect the data using this platform. This module runs its data collection spiders every week and samples job postings on web periodically. The results of these queries are then delivered into a mysql database on Amazon web services (AWS) for further processing that will be explained in the next section.

6.2.5 Skill Extraction Module

The second module of this framework is the skill extraction module that process the data collected by data collection module. Similar to the previous step, this task should also be continuous and thus resides on the cloud. This work uses AWS architecture as a solution for continuous storage of data and text processing for extracting skills. Amazon Web Services is a safe and well-developed solution for cloud computing and storage. By connecting the output of scrapy cloud service to the amazon databases, we collect data and store them as tables in an AWS mysql database. The Text Analysis Unit is using AWS Lambda service in combination with cloud watch service. Clould watch allows specifying a periodical

Table 6.1: data collection sample

job_title	company	description	posted_date	state	city	term
Analytics Analyst II	Horizon Blue Cross	job summary:...	4/25/2020	NJ	Hopewell	data analyst
Sr. Customer Data Analyst	Bottomline Technologies	bottomline is...	4/25/2020	NH	Portsmouth	data analyst
Master Data Project Analyst	BaronHR Staffing	hiring experience...	4/25/2020	TX	Plano	data analyst
Business Analyst	AltaSource Group	consultant-business analyst ...	4/25/2020	WA	Seattle	data analyst
Business Analyst	FHLB Office of Finance	position: business analyst....	4/25/2020	VA	Reston	data analyst

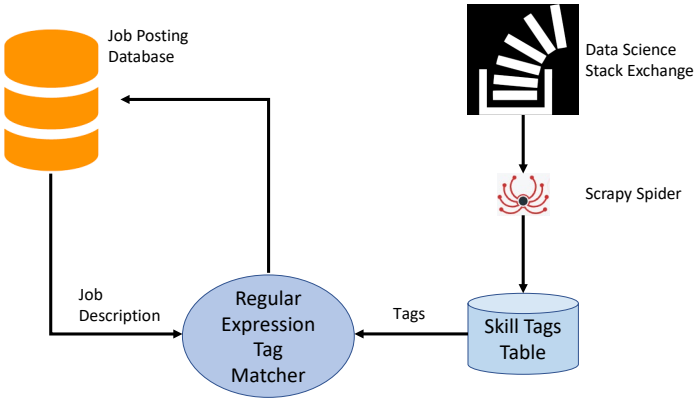

Figure 6.2: Overview of skill extraction mechanism

task by defining a cron expression. As a result, the python script will be executed periodically based on the schedule given in the form of cron expression. The module perform skill extraction on the collected data and add the result to the database for later usage by the visualization module. Given that job posts often include required skills, skill extraction of them would give us valuable insights as to what skills the job market mainly looks for. To extract the skills, an external lexicon has been developed by using datascience.stackexchange.com. The website is a platform for asking questions related to topics within the data science community along with proper tags. Such tags would cover a large set of concepts and skills needed for data science tasks. Figure 6.2 shows the text analysis unit architecture. First, a spider crawled the website and all labels of the website have

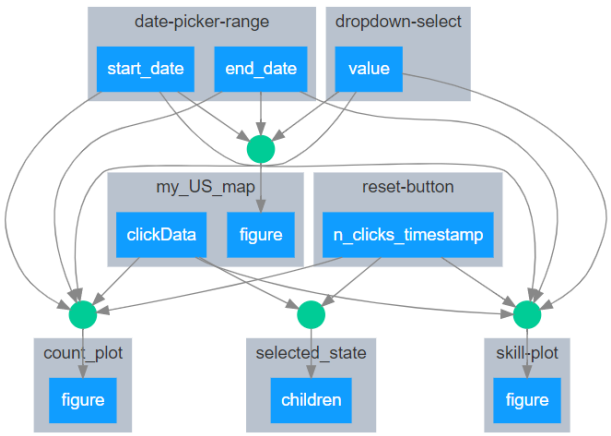

Figure 6.3: Interaction graph of visualization components

been collected. The lookup table of skills generated then will be fed into a regular expression pattern-matcher to find matches in the job descriptions. The result of this match will be added to each description in the database table. Doing so will identify a majority of the main skills mentioned in the job market given the large number of words in this lexicon. Next, we will discuss the visualization module that is using the data processed by this module.

6.2.6 Web-based Visualization

A Framework such as the one presented here, requires a way to present insights by visualization. However due to complexity of the interactions between fields a simple static presentation of results would not provide significant abilities to investigate temporal and spatial components of the job market in the United State. Therefore, a dynamic visualization that provides a mean to interact with different components (e.g. time range and states) would be preferred. Common solutions to provide a dynamic visualization includes R Shiny, and plotly dash for python. Plotly dash is a powerful framework which is build on top of flask framework and allows interaction with the created dashboard for users. It is also compatible with python and thus was used for the app with dynamic visualization in our work. The app has 6 main components:

- Job title selection: allowing user to select any of the three main branches of data science or all of them as the input
- Data range: allows selecting a period by user.
- US map: present the result of the input over the US map and allows selecting a specific state for further investigation.
- Skills visualization: presents top skills given the other fields.
- Count visualization: number of jobs samples during each week given the period and the state

- Reset state: resent state selection and returns the states to the United State as a whole.

The effect and interactions between these components are complex. Figure 6.3 shows these interactions in our dash model. As it can be seen the spatial and temporal aspect of skills could be investigated by using these components. Other components would be added as the work continues.

6.3 Results

In this section we present our results based on a 12 month period of April 2020 to April 2021. The framework collected approximately 244K job postings. This includes 129K, 71K, 43K job posting for data analyst, machine learning engineering, and data scientist respectively. This shows the job postings under the query term

Table 6.2: Top 20 results based on no. job by states and company

(a) Top 20 companies based on no. job posting

Company	Total Ads
Amazon.com Services LLC	7,251
JPMorgan Chase Bank, N.A.	3,873
Deloitte	3,461
Wells Fargo	2,030
Amazon Web Services, Inc.	1,850
Microsoft	1,661
Accenture	1,474
Pearson	1,388
Booz Allen Hamilton	1,266
Apple	1,243
Capital One - US	1,166
US Department of the Air Force	1,103
CACI	1,084
Amazon Dev Center U.S., Inc.	1,082
Facebook	1,069
Thermo Fisher Scientific	1,037
US Department of the Navy	999
US Department of the Army	969
UnitedHealth Group	967
Capital One	950

(b) Top 20 states based on no. job posting

State	No. Job ads
California	35.4K
Texas	17.3K
Virginia	16K
New York	15.7K
Washington	15K
Illinois	11K
Massachusetts	10.5K
Florida	8.8K
pennsylvania	8.5K
Maryland	8.2K
North Carolina	8.1K
Georgia	7.4K
New Jersey	7.3K
Ohio	5.9K
Colorado	5.7K
Minnesota	4.8K
Michigan	4.5K
Arizona	4.5K
Missouri	3.7K
Tennessee	3.2K

Table 6.3: Top skills for the three main tracks

(a) Data Scientist

Python	25.2K
Machine learning	22.8K
Statistics	21.2K
SQL	18.2K
Progeramming	17.0K
R	16.6K
Mathematics	1.05K
Algorithms	13.1K
Data analysis	10.9K
Visualization	9.9K
Cloud	9.7K
AI	8.1K
SAS	8.1K
Databases	7.9K
Tableau	7.7K
Spark	7.3K
AWS	7.3K
Deep Learning	7.3K
Java	7.0K
Hadoop	5.8K

(b) Machine Learning Engineer

Machine learning	60.7K
Python	36.0K
Cloud	30.0K
Progeramming	27.4K
Java	24.5K
AWS	22.0K
Software Development	21.1K
Algorithms	19.5K
SQL	17.4K
AI	15.6K
C++	13.5K
Spark	11.8K
Databases	11.3K
Statistics	9.8K
Mathematics	9.5K
Linux	9.5K
Optimization	9.5K
Javascript	9.0K
Deep Learning	8.8K
Hadoop	8.5K

(c) Data Analyst

Excel	49.7K
SQL	43.3K
Data analysis	28.2K
Tableau	22.0K
Databases	21.0K
Statistics	20.0K
Dashboards	19.0K
Visualization	17.0K
Programming	16.9K
Python	16.2K
R	12.3K
Mathematics	11.6K
Cloud	9.8K
SAS	9.5K
Forcasting	8.4K
ETL	7.7K
Data mining	7.1K
SAP	6.0K
Machine learning	5.0K
Classification	5.2K

" data scientist" are significantly less than the other two categories. Similarly, to build or lexicon, 612 labels were collected covering a broad spectrum of skills and concepts related to data science field using the method described in section 6.2.3.

6.3.1 Aggregated Results

First, the spatial aspect of job ads in the united states has been investigated. Looking at the aggregated results, the top states with the most job postings in this period can be extracted. Table 6.2b shows the top 20 states according to the number of job posted in this period. The result given in this table is aggregated for all job categories but the framework interface allows to investigate the results by job title and period of time.

Furthermore, by aggregating the skills in job postings and looking at the most repeated skills, we could identify the top mentioned skills for each category and the differences between them. Table 6.3 shows the top 20 skills in each job posting category. The result in this table clearly shows the difference between these main tracks of data science. While machine learning engineer job postings mostly shows a domination of skills such as programming, machine learning, cloud, and big data technologies, a data analyst mostly needs skills for data retrieving (e.g. SQL,

Excel,database), and Visualization (e.g Tableau, Power BI). A candidate for data scientist jobs, on the other hand would need skills from both of the previous categories. Other important aspect that this result show points to the importance of deep learning, cloud, and big data knowledge for a data scientist. Another interesting observation here is that a common skill for all of those categories is python which appears more than R suggesting that it is a more popular language for data related jobs in industry by comparison.

Lastly, We investigated the top companies that posted jobs related to data science in this period. Table 6.2a shows the top 20 companies during this period. Despite the expectation that tech sector dominate the field, it appears that other sectors contribute to this market significantly. Namely, consultant and advisory section (e.g. Booz Allen, Deloitte)and financial institutes(e.g JPMorgan, Wells Fargo) and government organization contribute significantly to this market along with the tech sector.

6.3.2 Temporal Insights

The framework allows us to investigate the interested skills along the time. Certain comparisons is being presented as a result of this system to provide extra insights that this system can provide. One important insight is the contribution and importance of programming languages required for data scientist. Figure 6.4 shows this comparison between top programming languages mentioned during this period.As it can be seen, the top language is python following by R and Java. This indicated not only python is the most dominant language but the trend and the gap shows it appears to be in demand in future in comparison to others. Similarly, a comparison between deep learning framework is presented in Figure 6.5. The result shows that tensorflow followed by pytorch are the most mentioned deep learning frameworks. Also H2O (an R based package) is mentioned much less which indicates the overwhelming attention to python for deep learning. As it can be seen, the framework allows temporal insights to this job market and based on their gen-

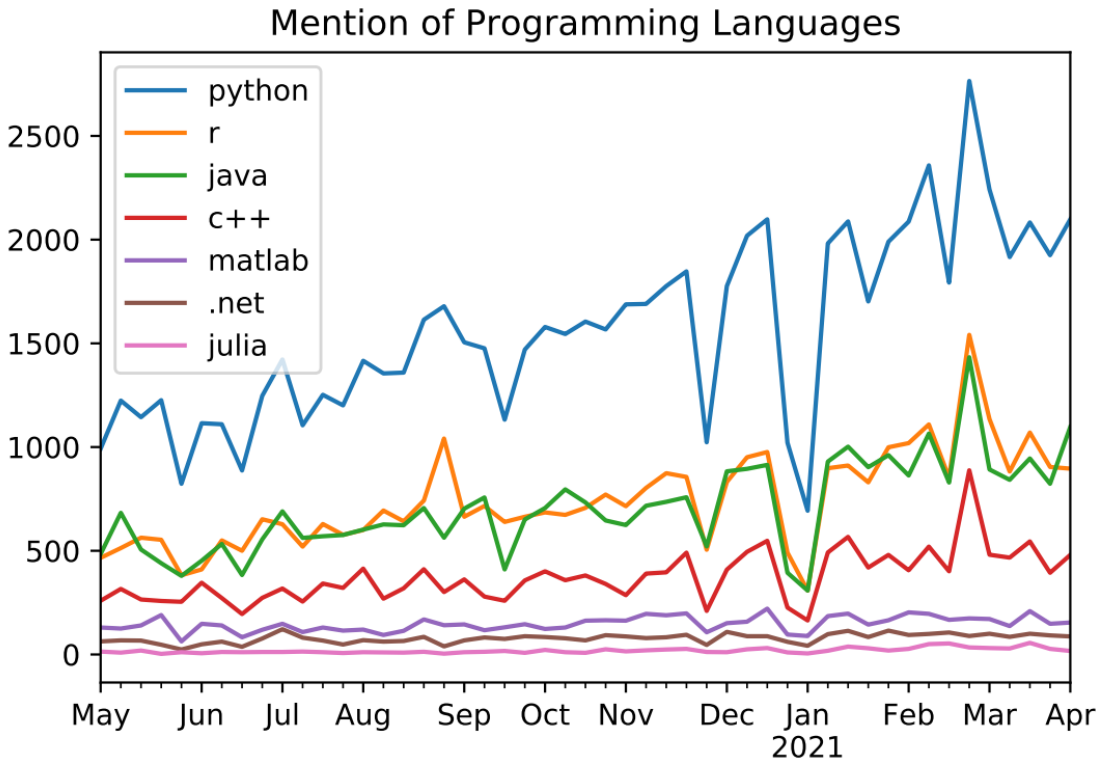

Figure 6.4: weekly comparison of programming languages

eral importance, such insights can be added to the interface designed for users.

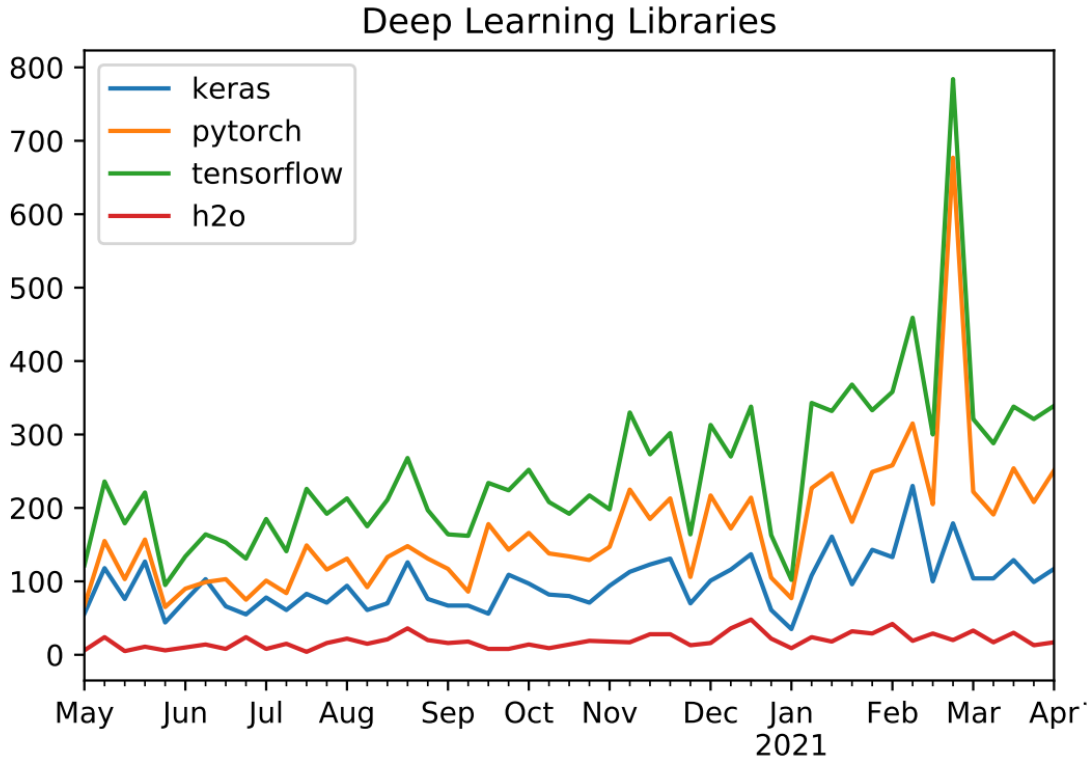

Figure 6.5: weekly comparison of deep learning frameworks

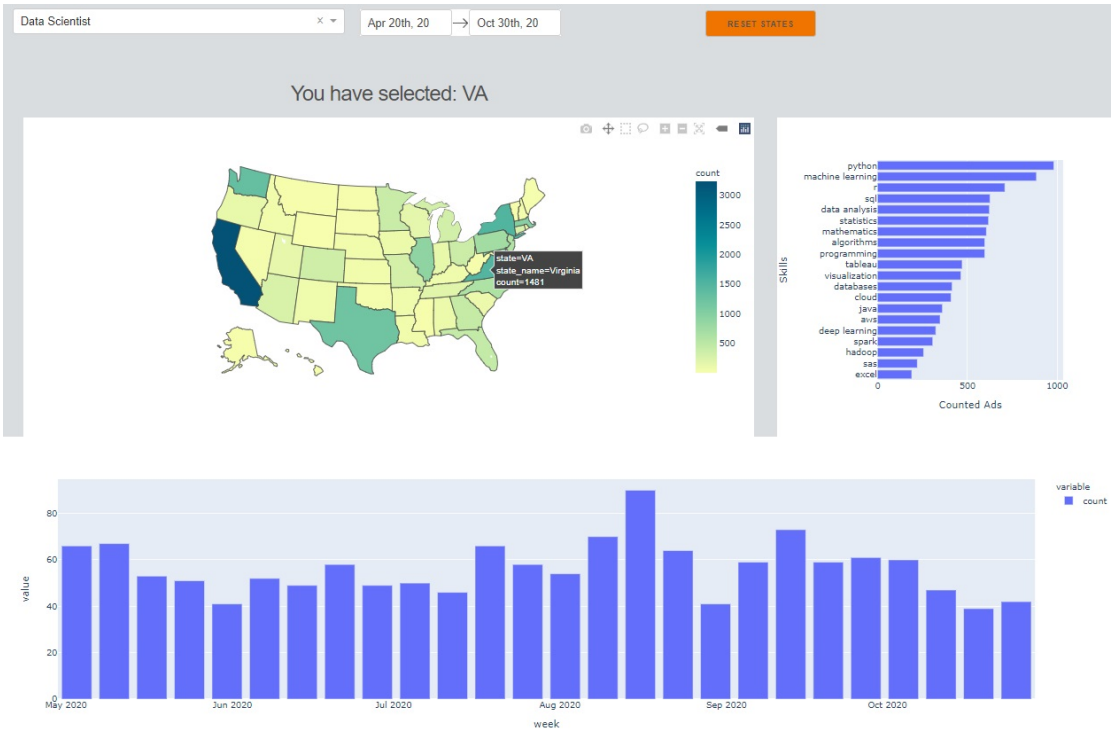

Figure 6.6: Web-based interface of job market monitoring app

Finally, the framework provides an interface for individuals to interact with. The interface deployed on heroku as host for this application using 1 dyno at the moment with the latest update in April (available at dsi-usa2.herokuapp.com). The interface connects to AWS database and retrieves relevant pieces of information for the visualization. Figure 6.6 shows the interface as a standalone application. Users can select any query field in a given period of time and the skillset (counts) and weekly number of advertise will be updated accordingly. Also, selecting any state will update these charts accordingly. To return back to all of the United States, "reset states" button should be used. The weekly count chart allows us to track the number of job postings during a period.

6.4 Conclusion and Future Work

This paper introduces a framework capable of collecting and analysing of spatial and temporal aspects of data science's job market using available tools within data science toolbox. The framework provides a free front-end interface developed

using plotly dash for this market. The result provides a quantitative guide for individuals and organizations to recognize the most important skills and concepts in this domain based on industry perception. Possible extension to the work here includes sector analysis of job posting and improving on skill extraction technique. Other temporal insights will be added to the interface based on the importance of the information they can add for individuals using this interface.

Chapter 7

Conclusions & Future Directions

7.1 Summary

NLP, as a major sub-field of Artificial Intelligence, can be applied in a wide range of domains with direct impacts for societal good. Proper usage of NLP techniques to build decision support tools or generate valuable insights in different domains is an important step to achieving this objective. This dissertation addresses the challenges and benefits of using textual resources to benefit society in different domains. We described applications of NLP in three under-explored social domains, including transportation and safety, counter-terrorism and security efforts, and labor market supply and demand, and how the result can be used to help different areas of society directly. Furthermore, by applying NLP techniques for societal good, we identified the challenges of using NLP in these problems. More specifically, while supervised NLP methods have well-defined metrics for performance evaluation in real-world applications, semi-supervised and unsupervised NLP techniques evaluation require extra steps to evaluate properly, and this can be quite challenging. Nevertheless, these methods can generate insights and help our society by improving certain aspects of above mention social domains. Another application of NLP for societal good comes from building NLP-powered applications that can benefit individuals and organizations. This dissertation presents a proof of concept for such an application to help understand the job market and

the skills required in order to educate potential future candidates better. These contributions will be explained in more detail in the following sections.

7.1.1 Empowering Society with Effective NLP Techniques

In numerous areas with societal impact, organizations continue to rely on expert opinions and expertise to foster societal improvements. Our work demonstrates the potential advantages of employing various NLP approaches as alternatives to solely depending on expert opinions, thereby benefiting society across multiple domains. The techniques showcased here can serve as decision support tools, aiding experts in making more informed and expedited decisions or generating valuable insights from domain-specific content. We present our contributions across three distinct domains.

Enhancing railway safety measures through NLP analysis of accident reports: Utilizing NLP as a supplementary decision-making tool for cause detection enhances the precision of prompt and dependable reporting, subsequently improving safety measures. We employed two deep learning architectures with distinct word embeddings on accident reports for accident cause detection. Our results indicate 75% accuracy, with performance expected to improve as the number of railway accident reports increases since deep learning models work better with larger datasets and most of the misclassifications in our experiments happened in small classes. Once trained, the model can suggest the most likely cause of the accident to experts, streamlining their workload.

Improve Counter-terrorism efforts by topic extraction and analysis of emotional impact: NLP techniques can be employed to lessen the burden of analyzing highly sensitive documents by deriving insights through topic models. We explored ISIS's gender-specific recruitment strategies by examining sections of their online publications aimed at women. Utilizing a topic modeling approach, we sought to comprehend the content and contrast it with a non-violent religious group. The results revealed specific themes, such as "hijrah" (relocating to a new

land), emerging from the topic modeling of ISIS publications, which aligns with expert perspectives. Moreover, the emotional impact analysis indicated that these materials are inspiring, and can be leveraged in counter-terrorism efforts to detect highly suspicious activities.

Improving skill identification to benefit job market search: The comprehension of the necessary skills for job positions is crucial in delivering enhanced education to prospective applicants. This endeavor calls for specialists to consistently update and manage skill databases and ontologies. To showcase NLP's potential, we integrated vector space embeddings, an unsupervised NLP technique, with limited supervision to develop a semi-supervised NLP model for hard skill identification. The outcome revealed that our method could pinpoint the required skills with 74% accuracy, relying only the text in job postings, and could be employed without necessitating expert supervision and maintenance.

7.1.2 NLP-Powered Applications

Another contribution of this dissertation is to demonstrate the usefulness of NLP techniques as part of a data-driven system. In order to do so, we presented a system that used textual data and the meta data related to them to provide insights into a specific section of the U.S. job market. The system is built of three major components allowing text collection, text processing, and visualization. This conceptual framework has also been deployed on the web to allow access for interested individuals. This framework shows an example of NLP-powered applications that can transfer the available information on the web to insights in a social domain (in this case, labor market demand). This dissertation shows a successful attempt in building such systems given access to the relevant text in social domains.

7.2 Challenges of Using NLP in Social Domains

In this section, we described some of the challenges that an NLP solution might face in under-explored domains. We will mention two major obstacles of NLP

approaches in social domains.

1. **Access to textual data**: One challenge of using NLP in different domains is the access to high-quality texts. This is especially a problem in social domains where access is limited to a particular group of people, as in the case of healthcare data, or identifying threats based on sensitive communications. We encountered this problem when trying to process women-related ISIS documents automatically. The only available resources were in pdf format which required Optical Character Recognition (OCR) to turn them into textual format. Another issue is the amount of textual information that is accessible. In the previous example, few NLP techniques can be used to generate meaningful insights due to lack of a big corpus.

2. **Evaluation of NLP solutions**: Another difficulty in using some of NLP techniques is proper evaluation metrics. In particular, measuring the performance of unsupervised methods is still challenging, and often an extrinsic evaluation approach, such as the performance of an upstream supervised task, is the best evaluation, next to human judgment. In contrast, supervised NLP problems have well-known metrics such as accuracy, F-1, or AUC. As a result, in some cases a very common approach is to define similar metrics for these unsupervised or semi-supervised methods. We leveraged a similar concept in the skill extraction problem by defining precision and recall slightly differently.

7.3 Future Works

In this work, we presented multiple NLP solutions within different social domains that can be helpful to the public. We illustrated the usage of supervised, unsupervised, and semi-supervised NLP approaches in three domains and demonstrated that by utilizing the right resources, such NLP methods can be applied to other public sectors for knowledge discovery and decision-making support. Apart from

extension to other fields, several other expansions to work presented here can be considered as future directions that will be discussed briefly here:

- **Extensions of current approaches**: There are several ways that the NLP work presented here can be useful in similar cases. For example, improving safety by using NLP to categorize the causes of railway accidents could also be used in evaluating other accident reports in the form of an AI-assisted system. . If integrated into the reporting platform, such AI assistance can help the reporting task by suggesting the correct label to categorize incidents more quickly and accurately.

  Similarly, identifying readers' emotions can be used by other governmental organizations to quickly identify certain documents of interest in online posts or other sensitive propaganda efforts. For example, organizations can use this approach instead of the common sentiment analysis when more is needed for the task.

  Furthermore, the analysis of online job advertisements can also be improved by adding other information to the proposed framework. One such improvement is to include industrial sector analysis of the job market data and present that as part of the framework. In this case, we can map this problem to a document classifier where documents could be automatically categorized into one of the main branches of industry. Also, keeping such a platform to continuously working for a long period of time would allow us to identify the trends and even potentially predict the importance of skills in near future. To do so, we need to consider the number of the appearance of skills over time as a time series forecasting problem. Recent advances in time series analysis provide good candidates, such as LSTM-based models or Facebook's prophet model.

  Finally, there are several possible advantages to using NLP on skill extraction tasks. Using a different label propagation (LP) implementation that allows incorporating cosine similarity could improve this approach. Such

an LP algorithm should be separately designed and tested to measure its performance. A better alternative is to turn the problem into a supervised multi-label problem. However, this approach requires many accurately labeled job advertisements and resources to be more feasible.

- **Integration and MLOPS practice**: We have mentioned that the proposed NLP methods can be used in the form of AI-assistant solutions in real world systems and have presented an example of how such an NLP-powered framework can improve understanding in the job market. However, building such systems requires special machine learning pipeline best practices. As a result, integration of such models must be carefully planned. One example of the challenges involved in creating such a pipeline is the necessity of creating a monitoring tool to ensure the pipeline is working properly. The framework we presented also requires automated monitoring for its data collection step. This was done manually, and it has proven to be very time-consuming. Lastly, the design should consider that not any part of the system be allowed to become a bottleneck that can cause future problems.

References

[1] M. Chui, M. Harryson, J. Manyika, R. Roberts, R. Chung, A. van Heteren, and P. Nel, "Notes from the ai frontier: Applying ai for social good," *McKinsey Global Institute*, pp. 5–8, 2018.

[2] J. D. Williams, A. Raux, and M. Henderson, "The dialog state tracking challenge series: A review," *Dialogue & Discourse*, vol. 7, no. 3, pp. 4–33, 2016.

[3] M. Heidarysafa, K. Kowsari, D. E. Brown, K. Jafari Meimandi, and L. E. Barnes, "An improvement of data classification using random multimodel deep learning (rmdl)," vol. 8, no. 4, pp. 298–310, 2018.

[4] D. Kuang, P. J. Brantingham, and A. L. Bertozzi, "Crime topic modeling," *Crime Science*, vol. 6, no. 1, p. 12, 2017.

[5] I. M. Cockburn, R. Henderson, and S. Stern, "The impact of artificial intelligence on innovation: An exploratory analysis," in *The economics of artificial intelligence: An agenda*, pp. 115–146, University of Chicago Press, 2018.

[6] A. Završnik, "Criminal justice, artificial intelligence systems, and human rights," in *ERA Forum*, vol. 20, pp. 567–583, Springer, 2020.

[7] L. Chen, P. Chen, and Z. Lin, "Artificial intelligence in education: A review," *Ieee Access*, vol. 8, pp. 75264–75278, 2020.

[8] K.-H. Yu, A. L. Beam, and I. S. Kohane, "Artificial intelligence in healthcare," *Nature biomedical engineering*, vol. 2, no. 10, pp. 719–731, 2018.

[9] C. Kadow, D. M. Hall, and U. Ulbrich, "Artificial intelligence reconstructs missing climate information," *Nature Geoscience*, vol. 13, no. 6, pp. 408–413, 2020.

[10] R. M. French, "The turing test: the first 50 years," *Trends in cognitive sciences*, vol. 4, no. 3, pp. 115–122, 2000.

[11] O. G. Iroju and J. O. Olaleke, "A systematic review of natural language processing in healthcare," *International Journal of Information Technology and Computer Science*, vol. 8, pp. 44–50, 2015.

[12] K. M. Alhawiti, "Natural language processing and its use in education," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 12, 2014.

[13] A. Farzindar and D. Inkpen, *Natural language processing for social media*. Morgan & Claypool Publishers, 2017.

[14] A. S. Rao and G. Verweij, "Sizing the prize: What's the real value of ai for your business and how can you capitalise," *PwC Publication, PwC*, pp. 1–30, 2017.

[15] L. Cao, "Ai in finance: challenges, techniques, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–38, 2022.

[16] R. Pillarisetty and P. Mishra, "A review of ai (artificial intelligence) tools and customer experience in online fashion retail," *International Journal of E-Business Research (IJEBR)*, vol. 18, no. 2, pp. 1–12, 2022.

[17] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text summarization techniques: a brief survey," *arXiv preprint arXiv:1707.02268*, 2017.

[18] A. Nenkova and K. McKeown, "A survey of text summarization techniques," *Mining text data*, pp. 43–76, 2012.

[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[20] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57, 1999.

[21] M. E. Roberts, B. M. Stewart, D. Tingley, C. Lucas, J. Leder-Luis, S. K. Gadarian, B. Albertson, and D. G. Rand, "Structural topic models for open-ended survey responses," *American journal of political science*, vol. 58, no. 4, pp. 1064–1082, 2014.

[22] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, pp. 1–66, 2017.

[23] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Social Network Analysis and Mining*, vol. 11, no. 1, p. 81, 2021.

[24] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," *arXiv preprint arXiv:1910.11470*, 2019.

[25] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," *arXiv preprint arXiv:2004.04906*, 2020.

[26] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *arXiv preprint arXiv:1806.03822*, 2018.

[27] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models

are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[28] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for gpt-3?," *arXiv preprint arXiv:2101.06804*, 2021.

[29] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith, *et al.*, "Recipes for building an open-domain chatbot," *arXiv preprint arXiv:2004.13637*, 2020.

[30] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, "Is chatgpt a general-purpose natural language processing task solver?," *arXiv preprint arXiv:2302.06476*, 2023.

[31] T. Ahmed, M. M. A. Aziz, and N. Mohammed, "De-identification of electronic health record using neural network," *Scientific reports*, vol. 10, no. 1, pp. 1–11, 2020.

[32] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute, "Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications," *Journal of the American Medical Informatics Association*, vol. 17, no. 5, pp. 507–513, 2010.

[33] J. A. Fries, E. Steinberg, S. Khattar, S. L. Fleming, J. Posada, A. Callahan, and N. H. Shah, "Ontology-driven weak supervision for clinical entity classification in electronic health records," *Nature communications*, vol. 12, no. 1, p. 2017, 2021.

[34] J. G. Borade and L. D. Netak, "Automated grading of essays: a review," in *Intelligent Human Computer Interaction: 12th International Conference, IHCI 2020, Daegu, South Korea, November 24–26, 2020, Proceedings, Part I 12*, pp. 238–249, Springer, 2021.

[35] D. Tafazoli, E. G. María, and C. A. H. Abril, "Intelligent language tutoring system: Integrating intelligent computer-assisted language learning into language education," *International Journal of Information and Communication Technology Education (IJICTE)*, vol. 15, no. 3, pp. 60–74, 2019.

[36] M. Ilyas, N. Malik, A. Bilal, S. Razzaq, F. Maqbool, and Q. Abbas, "Plagiarism detection using natural language processing techniques," *Technical Journal*, vol. 26, no. 01, pp. 90–101, 2021.

[37] R. K. H. Chua, "Faq chatbot web framework for response comparisons and performance analysis," 2020.

[38] A. Utka *et al.*, "Language technology platform for public administration," in *Human Language Technologies–The Baltic Perspective: Proceedings of the Ninth International Conference Baltic HLT 2020*, vol. 328, p. 182, IOS Press, 2020.

[39] W. D. Eggers, N. Malik, and M. Gracie, "Using ai to unleash the power of unstructured government data," *Deloitte Insights*, 2019.

[40] E. W. Ngai and P. T. Y. Lee, "A review of the literature on applications of text mining in policy making," 2016.

[41] K. Kowsari, D. E. Brown, M. Heidarysafa, K. Jafari Meimandi, , M. S. Gerber, and L. E. Barnes, "Hdltex: Hierarchical deep learning for text classification," in *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*, pp. 364–371, IEEE, 2017.

[42] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, and L. E. Barnes, "Rmdl: Random multimodel deep learning for classification," in *Proceedings of the 2nd International Conference on Information System and Data Mining*, pp. 19–28, ACM, 2018.

[43] D. Schafer and C. Barkan, "Relationship between train length and accident causes and rates," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2043, pp. 73–82, 2008.

[44] X. Liu, C. Barkan, and M. Saat, "Analysis of derailments by accident cause: evaluating railroad track upgrades to reduce transportation risk," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2261, pp. 178–185, 2011.

[45] X. Liu, M. Saat, and C. Barkan, "Analysis of causes of major train derailment and their effect on accident rates," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2289, pp. 154–163, 2012.

[46] X. Liu, "Statistical temporal analysis of freight train derailment rates in the united states: 2000 to 2012," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2476, pp. 119–125, 2015.

[47] R. Nayak, N. Piyatrapoomi, and J. Weligamage, "Application of text mining in analysing road crashes for road asset management," in *Engineering Asset Lifecycle Management*, pp. 49–58, Springer, 2010.

[48] W. Jin, R. K. Srihari, H. H. Ho, and X. Wu, "Improving knowledge discovery in document collections through combining text retrieval and link analysis techniques," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pp. 193–202, IEEE, 2007.

[49] D. E. Brown, "Text mining the contributors to rail accidents," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 346–355, 2016.

[50] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[51] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *arXiv preprint arXiv:1412.1058*, 2014.

[52] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, pp. 649–657, 2015.

[53] P. Blunsom, E. Grefenstette, and N. Kalchbrenner, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.

[54] O. Irsoy and C. Cardie, "Opinion mining with deep recurrent neural networks.," in *EMNLP*, pp. 720–728, 2014.

[55] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification.," in *EMNLP*, pp. 1422–1432, 2015.

[56] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification.," in *AAAI*, vol. 333, pp. 2267–2273, 2015.

[57] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, "Hierarchical attention networks for document classification.," in *HLT-NAACL*, pp. 1480–1489, 2016.

[58] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[59] T. Tokunaga and I. Makoto, "Text categorization based on weighted inverse document frequency," in *Special Interest Groups and Information Process Society of Japan (SIG-IPSJ*, Citeseer, 1994.

[60] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[61] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[62] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[63] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[64] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[65] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," *Artificial Neural Networks–ICANN 2010*, pp. 92–101, 2010.

[66] A. Karpathy, "The unreasonable effectiveness of recurrent neural networks," *Andrej Karpathy blog*, 2015.

[67] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[68] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks.," *ICML (3)*, vol. 28, pp. 1310–1318, 2013.

[69] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[70] A. Özgür, L. Özgür, and T. Güngör, "Text categorization with class-based and corpus-based keyword selection," *Computer and Information Sciences-ISCIS 2005*, pp. 606–615, 2005.

[71] F. Chollet *et al.*, "Keras: Deep learning library for theano and tensorflow.(2015)," 2015.

[72] "Federal railroads administration reports." http://safetydata.fra.dot.gov/OfficeofSafety/default.aspx.

[73] J. P. Farwell, "The media strategy of isis," *Survival*, vol. 56, no. 6, pp. 49–55, 2014.

[74] Z. Laub and J. Masters, "Islamic state in iraq and greater syria," *The Council on Foreign Relations. June*, vol. 12, 2014.

[75] A. J. Al-Tamimi, "The dawn of the islamic state of iraq and ash-sham," *Current Trends in Islamist Ideology*, vol. 16, p. 5, 2014.

[76] S. Gates and S. Podder, "Social media, recruitment, allegiance and the islamic state," *Perspectives on Terrorism*, vol. 9, no. 4, 2015.

[77] C. Winter, *The Virtual'Caliphate': Understanding Islamic State's Propaganda Strategy*, vol. 25. Quilliam London, 2015.

[78] E. Bodine-Baron, T. C. Helmus, M. Magnuson, and Z. Winkelman, "Examining isis support and opposition networks on twitter," tech. rep., RAND Corporation Santa Monica United States, 2016.

[79] M. Rowe and H. Saif, "Mining pro-isis radicalisation signals from social media users," in *Tenth International AAAI Conference on Web and Social Media*, 2016.

[80] H. J. Ingram, "An analysis of inspire and dabiq: Lessons from aqap and islamic state's propaganda war," *Studies in Conflict & Terrorism*, vol. 40, no. 5, pp. 357–375, 2017.

[81] P. Wignell, S. Tan, K. O'Halloran, and R. Lange, "A mixed methods empirical examination of changes in emphasis and style in the extremist magazines dabiq and rumiyah," *Perspectives on Terrorism*, vol. 11, no. 2, pp. 2–20, 2017.

[82] M. Vergani and A.-M. Bliuc, "The evolution of the isis'language: a quantitative analysis of the language of the first year of dabiq magazine," *Sicurezza, Terrorismo e Società= Security, Terrorism and Society*, vol. 2, no. 2, pp. 7–20, 2015.

[83] A. Perešin, "Fatal attraction: Western muslimas and isis," *Perspectives on Terrorism*, vol. 9, no. 3, 2015.

[84] K. Kneip, "Female jihad–women in the isis.," *politikon*, vol. 29, 2016.

[85] L. Canales and P. Martínez-Barco, "Emotion detection from text: A survey," in *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*, pp. 37–43, 2014.

[86] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[87] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[88] D. O'callaghan, D. Greene, J. Carthy, and P. Cunningham, "An analysis of the coherence of descriptors in topic modeling," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5645–5657, 2015.

[89] R. Plutchik, "The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice," *American scientist*, vol. 89, no. 4, pp. 344–350, 2001.

[90] J. Staiano and M. Guerini, "Depechemood: a lexicon for emotion analysis from crowd-annotated news," *arXiv preprint arXiv:1405.1605*, 2014.

[91] R. Grishman, "Information extraction," *IEEE Intelligent Systems*, vol. 30, no. 5, pp. 8–15, 2015.

[92] K. Balog, Y. Fang, M. De Rijke, P. Serdyukov, L. Si, *et al.*, "Expertise retrieval," *Foundations and Trends® in Information Retrieval*, vol. 6, no. 2–3, pp. 127–256, 2012.

[93] N. R. Council *et al.*, "A database for a changing economy: Review of the occupational information network (o* net)," 2010.

[94] M. Papoutsoglou, N. Mittas, and L. Angelis, "Mining people analytics from stackoverflow job advertisements," in *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 108–115, IEEE, 2017.

[95] N. G. Brooks, T. H. Greer, and S. A. Morris, "Information systems security job advertisement analysis: Skills review and implications for information systems curriculum," *Journal of Education for Business*, vol. 93, no. 5, pp. 213–221, 2018.

[96] F. Javed, P. Hoang, T. Mahoney, and M. McNair, "Large-scale occupational skills normalization for online recruitment," in *Twenty-Ninth IAAI Conference*, 2017.

[97] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*, pp. 722–735, Springer, 2007.

[98] E. Malherbe and M.-A. Aufaure, "Bridge the terminology gap between recruiters and candidates: A multilingual skills base built from social media and linked data," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 583–590, IEEE, 2016.

[99] M. Zhao, F. Javed, F. Jacob, and M. McNair, "Skill: A system for skill identification and normalization," in *Twenty-Seventh IAAI Conference*, 2015.

[100] L. Sayfullina, E. Malmi, and J. Kannala, "Learning representations for soft skill matching," in *International conference on analysis of images, social networks and texts*, pp. 141–152, Springer, 2018.

[101] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[102] D. A. Tamburri, W.-J. Van Den Heuvel, and M. Garriga, "Dataops for societal intelligence: A data pipeline for labor market skills extraction and matching," in *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pp. 391–394, IEEE, 2020.

[103] A. Bhola, K. Halder, A. Prasad, and M.-Y. Kan, "Retrieving skills from job descriptions: A language model based extreme multi-label classification framework," in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 5832–5842, 2020.

[104] S. Jia, X. Liu, P. Zhao, C. Liu, L. Sun, and T. Peng, "Representation of job-skill in artificial intelligence with knowledge graph analysis," in *2018 IEEE symposium on product compliance engineering-asia (ISPCE-CN)*, pp. 1–6, IEEE, 2018.

[105] S. Debortoli, O. Müller, and J. vom Brocke, "Comparing business intelligence and big data skills," *Business & Information Systems Engineering*, vol. 6, no. 5, pp. 289–300, 2014.

[106] A. De Mauro, M. Greco, M. Grimaldi, and P. Ritala, "Human resources for big data professions: A systematic classification of job roles and required skill sets," *Information Processing & Management*, vol. 54, no. 5, pp. 807–817, 2018.

[107] E. Smith, A. Weiler, and M. Braschler, "Skill extraction for domain-specific text retrieval in a job-matching platform," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 116–128, Springer, 2021.

[108] T. H. Davenport and D. Patil, "Data scientist," *Harvard business review*, vol. 90, no. 5, pp. 70–76, 2012.

[109] I. Carmichael and J. Marron, "Data science vs. statistics: two cultures?," *Japanese Journal of Statistics and Data Science*, vol. 1, no. 1, pp. 117–138, 2018.

[110] V. Dhar, "Data science and prediction," *Communications of the ACM*, vol. 56, no. 12, pp. 64–73, 2013.

[111] F. Provost and T. Fawcett, "Data science and its relationship to big data and data-driven decision making," *Big data*, vol. 1, no. 1, pp. 51–59, 2013.

[112] "Linkedin's 2017 u.s. emerging jobs report." https://economicgraph.linkedin.com/research/LinkedIns-2017-US-Emerging-Jobs-Report. Accessed: 2020-03-29.

[113] M. Daneva, C. Wang, and P. Hoener, "What the job market wants from requirements engineers? an empirical analysis of online job ads from the netherlands," in *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 448–453, IEEE, 2017.

[114] E. Hartnett, "Nasig's core competencies for electronic resources librarians revisited: An analysis of job advertisement trends, 2000–2012," *The journal of academic librarianship*, vol. 40, no. 3-4, pp. 247–258, 2014.

[115] S. Miller and D. Hughes, "The quant crunch: How the demand for data science skills is disrupting the job market," *Burning Glass Technologies*, 2017.

[116] A. Manieri, F. S. Nucci, M. Femminella, and G. Reali, "Teaching domain-driven data science: public-private co-creation of market-driven certificate," in *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 569–574, IEEE, 2015.

[117] A. S. Belloum, S. Koulouzis, T. Wiktorski, and A. Manieri, "Bridging the demand and the offer in data science," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 17, p. e5200, 2019.

[118] A. Gardiner, C. Aasheim, P. Rutner, and S. Williams, "Skill requirements in big data: A content analysis of job advertisements," *Journal of Computer Information Systems*, vol. 58, no. 4, pp. 374–384, 2018.

[119] I. A. Wowczko, "Skills and vacancy analysis with data mining techniques," in *Informatics*, vol. 2, pp. 31–49, Multidisciplinary Digital Publishing Institute, 2015.

[120] Q. Yang, X. Zhang, X. Du, A. Bielefield, and Y. Q. Liu, "Current market demand for core competencies of librarianship—a text mining study of american library association's advertisements from 2009 through 2014," *Applied Sciences*, vol. 6, no. 2, p. 48, 2016.

[121] H. Darabi, F. Karim, S. Harford, E. Douzali, and P. Nelson, "Detecting current job market skills and requirements through text min-ing," in *2018 ASEE Annual Conference and Exposition, Conference Proceedings*, 2018.

[122] M. M. Maer-Matei, C. Mocanu, A.-M. Zamfir, and T. M. Georgescu, "Skill needs for early career researchers—a text mining approach," *Sustainability*, vol. 11, no. 10, p. 2789, 2019.

[123] J. Burrus, T. Jackson, N. Xi, and J. Steinberg, "Identifying the most important 21st century workforce competencies: An analysis of the occupational information network (o* net)," *ETS Research Report Series*, vol. 2013, no. 2, pp. i–55, 2013.

[124] I. Karakatsanis, W. AlKhader, F. MacCrory, A. Alibasic, M. A. Omar, Z. Aung, and W. L. Woon, "Data mining approach to monitoring the requirements of the job market: A case study," *Information Systems*, vol. 65, pp. 1–6, 2017.

[125] F. Colace, M. De Santo, M. Lombardi, F. Mercorio, M. Mezzanzanica, and F. Pascale, "Towards labour market intelligence through topic modelling," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[126] E. Colombo, F. Mercorio, and M. Mezzanzanica, "Applying machine learning tools on web vacancies for labour market and skill analysis," *Terminator or the Jetsons? The Economics and Policy Implications of Artificial Intelligence*, 2018.

[127] J. Djumalieva and C. Sleeman, "An open and data-driven taxonomy of skills extracted from online job adverts," *Developing Skills in a Changing World of Work: Concepts, Measurement and Data Applied in Regional and Local Labour Market Monitoring Across Europe*, vol. 425, 2018.

[128] M. Pejic-Bach, T. Bertoncel, M. Meško, and Ž. Krstić, "Text mining of industry 4.0 job advertisements," *International Journal of Information Management*, vol. 50, pp. 416–431, 2020.

[129] E. M. Sibarani, S. Scerri, C. Morales, S. Auer, and D. Collarana, "Ontology-guided job market demand analysis: a cross-sectional study for the data science field," in *Proceedings of the 13th International Conference on Semantic Systems*, pp. 25–32, 2017.

[130] I. Kivimäki, A. Panchenko, A. Dessy, D. Verdegem, P. Francq, H. Bersini, and M. Saerens, "A graph-based approach to skill extraction from text," in *Proceedings of TextGraphs-8 graph-based methods for natural language processing*, pp. 79–87, 2013.