Large-scale Document Understanding with Knowledge Graphs for Medical Applications

by

Jeremy Costello

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

We introduce the background of the natural language processing field, outlining the benefits and drawbacks of rule-based versus statistical methods. We present knowledge graphs as a way to integrate the explainability of rule-based methods and the power of statistical methods, large language models in particular. The accuracy of natural language processing methods is paramount in sensitive fields such as biomedicine. We aim to create a knowledge graph to help practitioners, caretakers, and patients affected by neurodevelopmental disorders.

We give a background of knowledge graphs, topic modeling, and reinforcement learning. We talk about what knowledge graphs are, the creation process, and natural language processing methods for extracting data from text to populate a knowledge graph. We give a short history of topic modeling, followed by an outline of latent dirichlet allocation, dynamic topic models, topic model evaluation, and recent advances in neural topic modeling. We explain what reinforcement learning is, and outline the different approaches to reinforcement learning.

We develop a pipeline for creating a knowledge graph on neurodevelopmental disorders. We scrape data from both professional academic sources and non-professional webpages, including finances and services for caretakers and patients affected by neurodevelopmental disorders. We take input from practitioners, caretakers, and patients during the knowledge graph creation process in order to generate a knowledge graph that is as useful as possible for non-professionals, in contrast to many existing medical knowledge graphs that only incorporate academic sources.

To improve the topic modeling aspect of our knowledge graph creation pipeline,

we develop a new topic model using reinforcement learning. We make additional improvements to the topic model, including modernizing the neural network architecture, weighting the ELBO loss, and using contextual embeddings. Our unsupervised model outperforms all other unsupervised models and performs on par with or better than most models using supervised labeling. We conduct an ablation study to determine which changes to our model are the most important.

We look to directly extract triples from text using large language models. With the assistance of volunteers, we create two new data sets about FragileX syndrome: one for named-entity recognition and one for relation extraction. We compare a model trained on our FrageileX data set to a model trained on a less specific data set. We find strengths and weaknesses of both models. Our method is likely outdated due to the rapid pace of advancements in large language models.

We give a short concluding statement summarizing what we have done, and provide some brief thoughts on the future of natural language processing for biomedical applications.

Preface

This thesis is an original work by Jeremy James Costello. Professor Marek Z. Reformat has provided guidance for the work presented in this thesis and assisted with the manuscript composition by providing editorial feedback.

Chapter 3 was written in collaboration with Manpreet Kaur, Marek Z. Reformat, and Francois V. Bolduc. A version of this section is accepted in the Journal of Medical Internet Research and is available at https://www.jmir.org/2023/1/e45268

A version of Chapter 4 is accepted in Findings of the Association for Computational Linguistics: ACL 2023. A preprint is available at https://arxiv.org/abs/2305.04843

“ If science is half the man it says it is

then I can build it

the machine that snaps all of time in half ”

- Dan Barrett

To those who've inspired me.

Acknowledgements

Firstly, I would like to thank my family and friends.

I would also like to thank my supervisor, Dr. Marek Z. Reformat, for help and guidance throughout the writing of this thesis and my whole degree. I am thankful to all my collaborators in Dr. Reformat's lab, and to Dr. Francois V. Bolduc and my collaborators in his lab, Manpreet Kaur in particular.

Finally, I would like to thank the creators and maintainers of any and all open-source software I used during the writing of this thesis and my whole degree, especially Daniel R. Aldrich for his University of Alberta thesis template in LaTeX.

Table of Contents

1 Introduction **1**
1.1 Motivation . . . 4
1.2 Thesis Objectives . . . 5
1.3 Thesis Outline . . . 6

2 Background **8**
2.1 Knowledge Graphs . . . 8
2.1.1 Knowledge Graph Creation . . . 8
2.2 Topic Modeling . . . 10
2.2.1 Early Topic Modeling Techniques . . . 10
2.2.2 Latent Dirichlet Allocation . . . 11
2.2.3 Dynamic Topic Models . . . 11
2.2.4 Evaluation of Topic Models . . . 12
2.2.5 Neural Topic Modeling . . . 12
2.3 Reinforcement Learning . . . 12
2.3.1 Markov Decision Processes . . . 13
2.3.2 Environments . . . 13
2.3.3 Policies . . . 14
2.3.4 State Spaces . . . 14
2.3.5 Action Spaces . . . 14
2.3.6 Rewards . . . 15

3 Automated Labeling of Neurodevelopmental Disorders Web Resources 16
3.1 Abstract . . . 16
3.2 Introduction . . . 17
3.3 Related Works . . . 20
3.4 Methodology . . . 21
3.4.1 Data Collection . . . 21
3.4.2 Named-Entity Recognition . . . 23

3.4.3 Topic Modeling . . . 26
3.4.4 Document Classification . . . 27
3.4.5 Location Detection . . . 29
3.5 Results . . . 29
3.5.1 Knowledge Graph Schema . . . 30
3.5.2 Constructed Knowledge Graph: Overview . . . 33
3.5.3 Constructed Knowledge Graph: Utilization . . . 35
3.6 Discussion . . . 41
3.7 Conclusion . . . 44

4 Reinforcement Learning for Topic Models 49
4.1 Abstract . . . 49
4.2 Introduction . . . 49
4.3 Related Work . . . 51
4.4 Background . . . 52
4.4.1 Topic Models – Approaches . . . 52
4.4.2 Topic Models – Evaluation . . . 53
4.4.3 Reinforcement Learning . . . 54
4.4.4 Contextual Embeddings . . . 56
4.5 Methodology . . . 57
4.5.1 Modernizing ProdLDA . . . 57
4.5.2 Document Embeddings . . . 57
4.5.3 Single-step REINFORCE with a Continuous Action Space . . . 57
4.5.4 Weighted Evidence Lower Bound . . . 58
4.5.5 Evaluation Metrics . . . 58
4.5.6 Model Parameter Count . . . 59
4.6 Results . . . 59
4.6.1 Initial Experiments . . . 59
4.6.2 Comparison to Other Topic Models . . . 60
4.6.3 Ablation Study . . . 63
4.7 Data Sets . . . 64
4.7.1 20 Newsgroups . . . 65
4.7.2 New York Times . . . 66
4.7.3 Snippets . . . 66
4.7.4 W2E . . . 67
4.7.5 Wiki20K . . . 67
4.7.6 StackOverflow . . . 67

4.7.7 Google News . . . 67
4.7.8 Tweets2011 . . . 67
4.7.9 IMDb Movie Reviews . . . 67
4.7.10 Wikitext-103 . . . 68
4.8 Discussion . . . 68
4.9 Conclusion . . . 69
4.10 Ethics and Limitations . . . 70
4.10.1 Ethics . . . 70
4.10.2 Limitations . . . 71
4.11 Reproducibility . . . 72
4.11.1 Hyperparameters . . . 72
4.11.2 Ablation Study . . . 75

5 Extracting Knowledge Graph Triples from FragileX Abstracts 81
5.1 Abstract . . . 81
5.2 Introduction . . . 81
5.3 Background . . . 82
5.3.1 Precision, Recall, and F1 Score . . . 84
5.4 Methodology . . . 84
5.5 Results . . . 86
5.6 Discussion . . . 87
5.7 Conclusion . . . 88

6 Conclusion 89

Bibliography 91

List of Tables

3.1 Multi-label transformer model performance results 29
3.2 Examples of text and their annotated categories 37

4.1 Initial Experiment Topic Words 61
4.2 20 Newsgroups Categories 61
4.3 Comparison on no stop words data 62
4.4 Average metrics from best PTHT model (per metric) and our RL model 62
4.5 NPMI coherence comparison between PTHT model and RL model for each number of topics 63
4.6 Hyperparameter search and best results per data set for RL model . . 64
4.7 Comparison to CLNTM 64
4.8 Highlighted results from ablation study 64
4.9 Data Sets - Documents and Vocabularies 65
4.10 Data Sets - Training Document Lengths 66
4.11 Initial Experiments 73
4.12 Ablation Study 74
4.13 BNTM Snippets 74
4.14 BNTM 20 Newsgroups 74
4.15 BNTM W2E-title 75
4.16 BNTM W2E-content 75
4.17 Topic Modeling in Embedding Spaces (*We use $K = 25$ to calculate topic diversity for the final model.) 76
4.18 PTHT Data Set Seeds 76
4.19 Pre-training is a Hot Topic 76
4.20 CLNTM Data Set Seeds 77
4.21 Contrastive Learning for NTM 78
4.22 CLNTM Dropout Sweep 79
4.23 Full Results from Ablation Study 80

5.1 i2b2 2010 Relations 83

5.2 Model Comparison 88

List of Figures

1.1 Bottom-up creation of a general medical KG 7

3.1 Resource website/pages annotations process 22
3.2 Knowledge graph schema: links of the same color represent the same relations; dashed links represent relation *"is-a"* 31
3.3 Example of annotated resources from *raisingchildren.net.au/autism/behaviour/common-concerns/aggressivebehaviour-asd*: (left: a) most relevant annotating nodes; (right: b) n-to-n relations between resources and annotating nodes 34
3.4 Example of *OCCURRED_TOGETHER* and *IS_ASSOCIATED_WITH* connections between nodes. Entities from different sources are represented in different colors: *HPO* (orange), *AIRS* (blue), *UMLS* (green), *ERIC* (purple), *AGE* (grey) 45
3.5 Example of connection strength: (left: a) of relation *CONTAIN* between resource and annotating nodes; (right: b) of relation *OCCURRED_TOGETHER* between annotating nodes 46
3.6 Question interface: obtained entities and unigrams for the query: "aggressive behaviour and kicking and spitting" 46
3.7 Question interface: list of top 10 most relevant resources for the query "aggressive behaviour and kicking and spitting" 47
3.8 Normalized confusion matrix: rows show true labels, columns show predicted labels 48

4.1 Architecture Diagram: gray boxes - processing; white boxes - models/data/information; arrows across boxes - tune-ability 51
4.2 Loss (30 seeds): 20 Newsgroups 60
4.3 Topic Coherence (30 seeds): 20 Newsgroups 60
4.4 Topic Diversity (30 seeds): 20 Newsgroups 60
4.5 Comparison of RL model (ours) to BNTM models 77
4.6 Dropout sweep for 20 Newsgroups 78

5.1 Envisioned KG generation flowchart 85
5.2 Example abstract with entities outlined [197] 86
5.3 Example KG from abstract 87

Chapter 1

Introduction

The International Data Corporation (IDC) tracks and forecasts how much data is created each year, which they call the Global DataSphere [1]. In 2021, the Global DataSphere was 80 zettabytes, and was predicted to grow to 200 zettabytes by 2026 [2]. For reference, 1 zettabyte is 1 million terabytes. Capturing and making sense of this massive amount of data is a huge source of value with a lot of untapped potential. The explosion in popularity of machine learning and deep learning in recent years is largely due to the performance of these algorithms being correlated to how much data they are trained on.

Much of the data being created each year is text data, or can be converted to text data (e.g. by extracting text from video with automatic speech recognition [3]). It is estimated that the current amount of available high-quality text data is between 4.6 trillion and 17 trillion words, and the amount of low-quality text data is between 70 trillion and 70 quadrillion words [4]. The field concerned with making sense of this text data is called Natural Language Processing (NLP). While the amount of text data available for NLP grows each year, it is uncertain whether this growth will be sufficient to keep up with the growing data requirements for current and future data-hungry NLP algorithms.

Historically, there have been two overarching approaches to NLP: using large text corpuses to create statistical models of language, and using rule-based (i.e. symbolic)

methods to define the semantics and syntactics of language [5]. A leading proponent of rule-based methods is Noam Chomsky, who believes that linguistic knowledge is governed by a set of rules (i.e. grammars) inherent to the human brain [6, 7]. This is in opposition to Claude Shannon, who thought of language as a stochastic process that can be probabilistically determined [8], and to B.F. Skinner, whose psychological research was a precursor to computational Reinforcement Learning (RL) [9] and believed that linguistic knowledge was learned and reinforced through social interaction [10].

Statistical models of language have come to dominate in recent years, and the most powerful of these are known as large language models (LLM). The popularity of LLMs was kicked off by the scaling work done by OpenAI [11] (based on earlier work by Baidu [12]) to create the GPT series of models. The most recent GPT model, GPT-3, has 175 billion parameters and can perform many text-based tasks at or near human level [13]. This is evidenced by the explosion in popularity of ChatGPT [14], a model based on GPT-3.5 (a later version of GPT-3) and fine-tuned using reinforcement learning from human feedback (RLHF) [15, 16]. A later LLM, Chinchilla, outperformed GPT-3 despite having only 70 billion parameters [17]. This was because the authors found that the amount of text data required to optimally train these models was much higher than previously thought. OpenAI has since released GPT-4 [18].

LLMs are usually pre-trained in a self-supervised manner, where the model has some way to generate its own optimization targets from unlabeled text (e.g. [19–22]). There are a few self-supervised paradigms for pre-training language models, but the most popular is the generative uni-directional approach [20]. In this approach, models are trained to predict the next word in a sequence based on a context window of previous words in the sequence. Models trained using this approach are referred to as causal language models [23]. Other paradigms include masked language modeling [21] and ELECTRA [22]. Once an LLM is pre-trained, it can be fine-tuned on a variety

of tasks. Fine-tuning is usually done through supervised learning, where input-output pairs for a task are fed to the model and the pre-trained weights are slightly updated to optimize performance on the required task.

LLMs are based on the Transformer architecture [24], which originally found success in NLP but has more recently found success in computer vision [25] and RL [26]. Vision Transformers have been combined with LLMs to translate between these two modalities (e.g. describing an image or video in text). Contrastive Language-Image Pre-training (CLIP) is one such model that learns to match text-image pairs [27].

While statistical language models currently dominate, they would not be where they are without advances made in rule-based NLP methods. The Chomsky Hierarhcy, which defines the classes of formal grammars from Turing machines down to finite-state automata, was an influence on the creation of many programming languages [6]. While not related to NLP, Chomsky's context-free grammars are also used widely in program synthesis. Additionally, ideas such as Cloze deletion [28] and the quote "you shall know a word by the company it keeps" [29] influenced the window- and context-based methods of models such as Word2Vec [30] and BERT [21]. Early chatbots, such as ELIZA [31], also used rule-based methods.

Another advantage of rule-based methods is their explainability. LLMs are prone to halluciations, a phenomenon where the model generates an answer that is false, but usually in a convincing manner [32]. This is a problem in sensitive domains such as the medical field. The greater explainability of rule-based methods means the designer has much more control over the algorithm and can be much more confident in its outputs, but also means they give up the much greater power of statistical methods such as LLMs. Successfully combining these two methods to harness the power of statistical methods and the explainability of rule-based methods should be the goal of anyone looking to apply NLP algorithms to a sensitive field.

Causal Masked Multimodal Model of the Internet (CM3) [33] attempts to understand the current unstandardized nature of HTML pages on the Internet using LLMs

and vision Transformer models. Text, images, hyperlinks, and simplified HTML are extracted from the raw HTML for millions of documents from Common Crawl News and English Wikipedia and a multimodal Transformer model is trained on this data.

As an already-existing alternative to the creation of language models from unstructured Internet data, the Semantic Web [34] proposes the standardization of HTML tagging with something like the Resource Description Framework (RDF) [35]. This standardized data tagging would allow Internet data to be more easily understood with rule-based NLP algorithms. This tagging would require modifying existing websites to become part of the Semantic Web, and wide-spread adoption from web developers to ensure newly-created websites conform to the Semantic Web standard. The standardized tagging of the Semantic Web would allow easier machine understanding of web data.

1.1 Motivation

One approach for combining rule-based and statistical NLP methods is the knowledge graph (KG) [36]. These are graph-based knowledge representations of (subject, relation, object) triples, where subject and object entities are represented by graph nodes and relations between these entities are represented by graph edges. Similar to SQL, these KGs can be efficiently queried using graph query languages such as SPARQL. KGs are a great way to represent large groups of documents, and there are a few ways to go about this. Triples can be directly extracted from text, or relations between documents can be represented as triples. Some methods for creating these triples include named-entity recognition (NER), entity disambiguation, coreference resolution, relation extraction, and topic modeling. These methods can be performed using statistical methods or rule-based methods.

Some people view LLMs as a replacement to KGs [37], but the knowledge contained within a LLM is often incorrect, as can be seen from the hallucination problem, and correcting knowledge in an LLM is much more difficult than correcting knowledge in

a KG. Incorrect nodes and edges in a KG can be directly edited, while knowledge in a LLM can only be updated by changing model weights, the outcome of which is currently impossible to fully verify. Some work has been done on combining LLMs and KGs, such as ERNIE [38], Pretrained Encyclopedia [39], SKILL [40], BertNet [41], and KGLM [42].

One domain where KGs could excel is the medical domain. There exist many LLMs specifically created for medical or general science domains. Examples of these LLMs include BioBERT [43], SciBERT [44], [45], BioMegatron [46], ElectraMed [47], SciFive [48], BioElectra [49], BioM-BERT [50], BioGPT [51], BLOOM [52], Galactica [53], and Med-PaLM [54]. Of these LLMs, those that can be used for text generation still suffer from the hallucination problem. An example of this is the demo for Galactica by Meta (formerly Facebook) being taken down after 3 days because of many examples being shared online of its incorrect text generations [55].

Some examples exist of augmenting medical LLMs with additional knowledge, in a similar vein to how they could be augmented by KGs. Tian *et al.* [56] additionally provide syntactic information to the LLM, which results in superior predictions to the base model. Yuan *et al.* [57] augment a LLM with information from the Unified Medical Language System (UMLS) knowledge base, outperforming other models in most tasks on which they were tested. KGs are a type of knowledge base. Yasunaga *et al.* [58] uses links between documents (e.g. hyperlinks for internet documents) to model connections between documents during LLM pre-training. They pre-train a BioLinkBERT model specifically on biomedical literature. This model was state-of-the-art on various biomedical tasks.

1.2 Thesis Objectives

We aim to combine the power of LLMs and the explainability of KGs for accurate and truthful NLP in the medical domain. Many existing medical KGs are built for general medical knowledge in a top-down manner from academic medical literature.

We envision two main objectives of the work.

- We aim to build a general medical KG in a bottom-up manner by combining domain-specific medical KGs. We believe this will result in more accurate knowledge because medical professionals in a domain can be in the loop for creating the KG for their domain. These KGs can then be combined to create more general medical KGs. We also believe including patients and caretakers in the KG creation process will result in more widely useful KGs containing knowledge useful for those directly affected, not only academic medical knowledge. Figure 1.1 illustrates out vision for the bottom-up creation of a general medical KG. We will outline the first step in our vision by creating a KG for neurodevelopmental disorders (NDD) with input from medical professionals, patients, and caretakers.

- During the process of creating this KG for NDDs, we expect to discover areas where algorithms for creating the KG can potentially be improved. We expect to apply topic modeling to the KG creation process, a method that is not widely used in the creation of KGs. We think there may be room for the development of an improved topic modeling algorithm for creating KGs, or even an improved general topic modeling algorithm. Additionally, recent progress in LLMs has unlocked the potential to directly extract triples from text. We aim to show how this can be done by curating a data set and outlining how this data set can be used to directly extract triples from new documents.

1.3 Thesis Outline

This thesis presents the following subjects:

- Introducing a short history and the current state of the NLP field, the motivation behind our work, and what we aim to complete through this work.

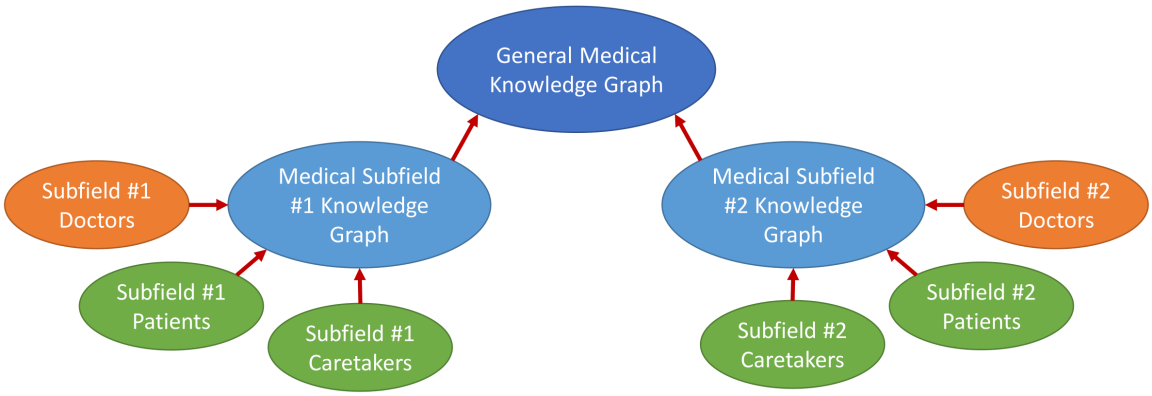


Figure 1.1: Bottom-up creation of a general medical KG

- Giving background knowledge on KGs, topic modeling, and RL.
- Outlining a methodology for creating a KG from web resources and show the KG about NDDs we created using this methodology, along with some applications of this KG.
- Developing a new topic modeling method using RL that is state-of-the-art on many data sets.
- Using human labeling to create a data set of KG triples from FragileX abstracts to illustrate the ability of LLMs to automatically annotate triples in academic medical text.

Chapter 2

Background

2.1 Knowledge Graphs

KGs are a method for representing knowledge in a structured and machine-readable way [36, 59]. The form of a KG is as a mathematical graph containing nodes connected by edges [60]. The nodes can be a person, a place, or pretty much any other piece of information. Nodes are connected by directed edges labeled with information on how connected nodes are related. For example, if there are two nodes labeled "Alberta" and "Edmonton", they could be connected by an edge from "Edmonton" to "Alberta" labeled "capital of". These three pieces of information are usually called a triple [61]. The first node, or subject, is connected to a second node, or object, by an edge, or predicate. Triples are of the form (subject, predicate, object); following on the previous example, the triple would be (Edmonton, capital of, Alberta). The subject and object are collectively referred to as entities. The predicate is also often referred to as the relation.

2.1.1 Knowledge Graph Creation

Creating a KG usually consists of three steps [62–64].

1. **Data Integration**. The process of collecting data from one or many sources and modifying this data into a suitable format.

2. **Schema Modeling**. The creation of a desired structure for the KG. Based on the domain of information the KG will handle, a vocabulary describing entities and relations for that domain should be defined. Next, an ontology should be defined to provide a hierarchy to the vocabulary. Finally, possible properties of entities and relations in the KG should be defined.

3. **KG Construction**. Processing the data to extract relevant entities and connect these entities together with relations as defined by the schema.

There are many methods for extracting entities and relations between these entities from data, such as unstructured text, for the KG construction step.

Named-entity Recognition

NER is the process of identifying important entities within unstructured text [65]. Named entities can refer to people, organizations, locations, dates, times, monetary values, or many other things. For example, in the sentence "Charlie Kaufman directed the movie *Synecdoche, New York*." the named entities would be "Charlie Kaufman" and "*Synecdoche, New York*". These entities can be recognized by rule-based methods such as vocabulary matching, or through statistical methods such as a fine-tuned LLM.

Named-entity Disambiguation

Named-entity disambiguation is the process of determining the true entity for a name that could refer to multiple entities [66]. For example, the term "St. John's" could refer to the capital of Newfoundland and Labrador, or to the capital of Antigua and Barbuda. Based on the context surrounding an entity, the true entity should be able to be disambiguated. This disambiguation can be performed using rule-based methods such as another KG, or statistical methods such as a fine-tuned LLM.

Coreference Resolution

Coreference resolution is the process of identifying expressions in a text that refer to the same entity [65]. This technique is most frequently needed for pronouns, abbreviations, and aliases. For example, in the sentences "Benjamin Reichwald, also known as Bladee, is a professional musician. He was born in Stockholm, Sweden." the expressions "Benjamin Reichwald", "Bladee", and "He" all refer to the same entity. This resolution can be performed by rule-based methods such as vocabulary matching, part-of-speech tagging, or dependency parsing. It can also be performed by statistical methods such as a fine-tuned LLM.

Relation Extraction

Relation extraction is the process of identifying and extracting relations between entities in a text [65]. For example, in the sentence "*The Book of the New Sun* is a four-volume science fiction and fantasy novel series written by Gene Wolfe." the relation between "*The Book of the New Sun*" and "Gene Wolfe" could be "written by". This extraction can be performed by rule-based methods such as pattern matching, or through statistical methods such as a fine-tuned LLM.

2.2 Topic Modeling

Topic modeling is a NLP method that tries to find consistent themes across a set of documents [67]. These themes are called topics, and for each topic a set of words is identified that correspond to that topic. Latent Dirichlet Allocation (LDA) [68] has historically been the most popular topic modeling technique, but has been overtaken recently by Neural Topic Modeling (NTM) techniques [69].

2.2.1 Early Topic Modeling Techniques

Two of the precursors to LDA were the mixture of unigrams (MoU) model [70] and the probabilistic latent semantic indexing (pLSI) model [71]. For a document of length

N, the generative process of the MoU model is to choose a topic z and then generate N words from a multinomial conditioned on z. The pLSI model improved on the MoU model by assuming that a document could belong to multiple topics. The main downside of pLSI is that a model created from a training set cannot be easily applied to unseen documents.

2.2.2 Latent Dirichlet Allocation

LDA is a topic modeling technique that represents each document in a corpus as a mixture over latent topics, and each topic as a multinomial distribution over vocabulary words [68]. LDA is a generative model; the generative process is outlined in algorithm 1.

Algorithm 1: Latent Dirichlet Allocation

Input: A corpus of documents, D
Input: A vocabulary of length V
Input: A number of topics, K
Algorithm Parameters:
Dirichlet hyperparameter $\boldsymbol{\alpha} > 0$
A word probability matrix, $\boldsymbol{\beta}$, of size $K \times V$

```
for each document w in D do
    Draw topic distribution θ ~ Dirichlet(α)
    for each word w_n in document w do
        Sample topic z_n ~ Multinomial(θ)
        Sample word token w_n ~ Multinomial(w_n|z_n, β)
    end
end
```

2.2.3 Dynamic Topic Models

Dynamic topic models capture the temporal aspect of a corpus of documents, if it exists. The topics discussed in a corpus of documents can evolve through time, and dynamic topic models are one method to capture this evolution. For example, the paper that introduced dynamic topic models analyzed papers from the journal *Science* from the years 1880 to 2000 [72].

2.2.4 Evaluation of Topic Models

Early topic models were evaluated using perplexity [73]. Perplexity is a metric of how well a language model can predict a text document [74]. This can be how well it predicts the next word(s) in a document, or in the case of topic models how well the model can predict the unordered words in that document. The lower the perplexity of a model, the better.

Research found that a topic model having low perplexity didn't necessarily mean it had interpretable topics [73]. To remedy this, a new topic model performance metric was introduced called coherence [75]. This coherence metric correlated much more strongly with interpretable topics. There are many different ways to calculate the coherence of a topic model, but the method which was found to correlate best with human judgement is normalized pointwise mutual information (NPMI) [76, 77].

Some topic models have high coherence, but have a lot of overlap in words between topics. A way to measure this is topic diversity, which measures the ratio of unique topic words to total topic words across the top-k words of each topic [78].

2.2.5 Neural Topic Modeling

NTMs combines topic models with neural networks (NN). Different types of NN methods have been used for topic modeling, including variational autoencoders (VAE), autoregressive models, generative adversarial networks, graph NNs, and more [69].

2.3 Reinforcement Learning

RL is one of the main paradigms of machine learning, the others being supervised learning and unsupervised learning [79]. In a RL problem, an agent traverses some environment attempting to maximize its cumulative reward. A RL problem is usually represented as a Markov decision process (MDP) [80].

2.3.1 Markov Decision Processes

A discrete-time, infinite-horizon, finite MDP is a tuple $\mathcal{M} \doteq (\mathcal{S}, \mathcal{A}, p, r, \mu_0)$, where $\mathcal{S}$ is a finite (non-empty) set of states, $\mathcal{A}$ is a finite (non-empty) set of actions, $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ gives the probability of transitioning to state s' when action a is taken in state s, written $p(s'|s, a)$, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ gives the *expected* reward when action a is taken in state s, and μ_0 is the initial state distribution [79, 80]. Here we only consider the case where actions are chosen according to a stationary, deterministic policy, a function $\pi : \mathcal{S} \to \mathcal{A}$. Such a policy, together with a MDP, generates a sequence $(S_t, A_t, S_{t+1}, R_{t+1})_{t=0}^{\infty}$ as follows:

1. Sample the initial state s_0 from μ_0.
2. Repeat for $t = 0, 1, 2 \ldots$
   (a) Take action $a_t = \pi(s_t)$.
   (b) Transition to state s_{t+1} sampled $p(\cdot|s_t, a_t)$ and receive reward $r_{t+1} = r(s_t, a_t)$.

2.3.2 Environments

There are two types of environments for RL. The first is the episodic environment, in which the agent traverses the same environment over many episodes. There is some criteria for when an episode ends, and when this happens the agent is reset to some starting state. The other type of environment is the continuing environment. In continuing environments, the agent traverses the environment without reset, so it must learn to act within the environment without the luxury of getting to start over every so often.

RL algorithms can be model-based, in which the agent attempts to create an internal model of the environment and its transition probabilities, or model-free, in which the agent only cares about the optimality of its policy, π.

2.3.3 Policies

RL policies can be based on state-value functions, action-value functions, or policy gradients. State-value functions assign a value to each state in $\mathcal{S}$. From a state, s, the agent's policy, π, chooses an action, a, based on the values of states adjacent to s (e.g. always choosing the highest value adjacent state). Action-value functions assign a value to each state-action pair in $\mathcal{S} \times \mathcal{A}$. From a state s, the agent's policy, π, chooses the best action, a, to take from that state (e.g. always choosing the highest value action).

The policy gradient method differs from the two methods above in that it directly computes the policy, π, rather than basing the policy off of the value of a state or action. The policy is usually parameterized by some function approximator and is optimized by an optimizer such as stochastic gradient ascent.

RL algorithms can be on-policy, where the value function is updated based on the action the agent took, or off-policy, where the value function is updated based on actions other than the action the agent took.

2.3.4 State Spaces

Environments with small $\mathcal{S}$ can have $\mathcal{S}$ stored in a table, such as for tic-tac-toe. As environments grow larger, storing each $s \in \mathcal{S}$ as a unique entry in a table becomes infeasible, such as for chess. When this happens, $\mathcal{S}$ should be represented by a function approximator.

2.3.5 Action Spaces

For a RL agent, $\mathcal{A}$ can be discrete or continuous. Discrete $\mathcal{A}$ are finite, and the agent chooses to perform some subset of $\mathcal{A}$ on each time step. For example, in chess the agent chooses one of the finite number of moves (actions) in a position (state). Continuous $\mathcal{A}$ are infinite, with each $a \in \mathcal{A}$ represented as a probability distribution, such as a Gaussian. $\mathcal{A}$ can contain both discrete and continuous actions, such as for

a video game where the controller buttons are represented as discrete actions and the analog sticks are represented as continuous actions.

2.3.6 Rewards

The cumulative reward obtained by a RL agent is called the return, G. In the return, rewards are usually time-discounted by some discount factor ($\gamma < 1$) so more recent rewards are emphasized over older rewards.

$$G_{t+1} = \gamma * G_t + r_{t+1} \tag{2.1}$$

An alternative to time-discounting is the average reward formulation, where the average reward is optimized rather than the discounted return. Discounting works fine in episodic environments, but it is better to use average reward in continuing environments.

Chapter 3

Automated Labeling of Neurodevelopmental Disorders Web Resources

3.1 Abstract

BACKGROUND. Providing patients and families with trusted information is needed more than ever with the abundance of online information. Several organizations aim to build databases which can be searched based on the needs of target groups. One such group is individuals with neurodevelopmental disorders (NDDs) and their families. NDDs affect up to 18% of the population and have major social and economic impacts. Current limitations in communicating information for individuals with NDDs include the absence of shared terminology and the lack of efficient labeling processes for web resources. These limitations lead to an inability for health professionals, support groups, and families to share, combine, and access resources.

OBJECTIVE. We aim to develop a natural language-based pipeline to label resources by leveraging standard and free-text vocabularies obtained through text analysis, and then represent those resources as a weighted knowledge graph.

METHODS. Using a combination of experts and service/organization databases, we created a dataset of web resources for NDD. Text from these websites was scraped

and collected into a corpus of textual data on neurodevelopmental disorders. This corpus was used to construct a knowledge graph suitable for use by both experts and non-experts. Named entity recognition, topic modelling, document classification, and location detection were used to extract knowledge from the corpus.

RESULTS. We developed a resource annotation pipeline using diverse natural language processing algorithms to annotate web resources and stored them in a structured knowledge graph. The graph contains 78,181 annotations obtained from the combination of standard terminologies and a free-text vocabulary obtained using topic modelling. An application of the constructed knowledge graph is illustrated: a resource search interface using the ordered weighted averaging operator to rank resources based on a user query.

CONCLUSIONS. We have developed an automated labeling pipeline for web resources on NDDs. This work showcases how AI based methods such as natural language processing and knowledge graphs for information representation, can enhance knowledge extraction and mobilization, and could be utilized in other fields of medicine.

3.2 Introduction

Access to curated medical information has become more important than ever due to the growing amount of information available on the internet and many challenges faced with sharing information about medical topics. Neurodevelopmental disorders (NDD) are a range of conditions including autism spectrum disorder, intellectual disability, and attention deficit hyperactivity disorder. These disorders affect up to 18% of the population [81–87] and are affected by the growing amount of online information and misinformation [88, 89]. NDDs have complex medical features and the needs of affected individuals and their families tend to be quite diverse [90–92].

There exists a large amount of information relating to NDDs on the internet, but this information is scattered across many websites, often using different terminology, and containing both reliable information and misinformation. Finding information that is specific, relevant, and trusted is therefore difficult for caregivers of children with NDDs. To remedy this, a knowledge repository containing the available NDD resources annotated with appropriate labels and terms could be constructed. This repository would enable discovery of relevant, trusted resources based on phrases of interest provided by users.

We propose using a knowledge graph (KG) to represent web resources together with terms and phrases annotating them. The utilization of a KG enables web links (i.e., resources), terms, and phrases to be represented as nodes with the relevance between them represented as edges.

A KG indexing web links and information on NDDs would allow experts and non-experts to have a primary repository of NDD knowledge. With this knowledge, doctors could make quicker and more accurate selections of relevant resources/websites, and caregivers of children with NDDs could quickly find appropriate information, services, and financial support. Accurate identification and early help are critical to quality of life (QoL) outcomes for those with NDDs. The proposed graph-based repository could improve many peoples' lives.

The paper describes the methodology of constructing a KG-based repository of NDD resources, called hereafter ***NDD-KG***. It presents:

- an automatic processing of text extracted from websites relating to NDDs and identifying the most accurate terms/phrases describing them based on Named Entity Recognition (NER), topic modeling, location detection, and resource classification.

- a process of determining degrees of relevance between Knowledge Graph entities and resources and storing them as weights of relations in the graph.

- an application of an Ordered Weighted Aggregation (OWA) operator [93] for determining the most relevant resources using the aggregated weights of relations between resources and terms/phrases describing them.

- an example of using the constructed KG-based repository of NDD resources for retrieving a ranking of resources related to a phrase representing the user's interests.

The paper reviews some related works and describes the methodology used for constructing a KG. It also includes an overview of the KG schema, gives an in-depth look at individual techniques used to annotate scraped web resources, and introduces an aggregation process. Finally, a brief overview of the utilization of the constructed graph is presented and an outline of conclusion and possible future work.

The novelty of our constructed KG lies in the domain specificity, the inclusion of patient-focused information from different sources, and application of combining different information extraction methods. Most other medical KGs focus on the entire medical field, and will therefore lose granularity on specific medical topics. We created a KG for NDDs involving input from patients and caretakers affected by NDDs, along with medical professionals who specialize in NDDs. This resulted in a KG containing more extensive knowledge about NDDs than a general medical KG. Input from patients and caretakers allowed us to include resources related to core knowledge, financial help, education, and services. This is in contrast to most other medical KGs, which only focus on extracting medical knowledge from literature.

In addition to the named-entity recognition (NER) pipeline to detect standard terminologies used by medical professionals, we use topic modeling to capture resource specific keywords. Using both the NER and topic modeling allows us to better annotate the resources. Furthermore, document classification is applied to categorize and label the resources into core knowledge, financial help, education, and services. Representing the extracted knowledge along with the resources in KG leads to a cen-

tralized hub that combines resources from different areas of need around NDD to maximize knowledge capture.

3.3 Related Works

KGs have been used for many areas, including medical, cyber security, financial, news, and education. There have been a wide range of KG applications within the medical field. Applications include general KGs across the whole medical domain, and across specific areas such as depression, thyroid disease, and COVID-19 as described below.

Several KGs spanning the entire medical field have been created. For example, Ernst et al. [94] created KnowLife. They used advanced information extraction methods, including NER, pattern mining, and consistency reasoning, to populate entities and relations from scientific literature and online communities, in contrast to many previous works which were manually curated.

Shi et al. [95] developed methods to extract syntactic, semantic, and structural information from conceptual KGs. They used a similar method to KnowLife for creating the KGs, and extended understanding of the resultant KGs by using machine learning methods to prune meaningless relations in the graphs and extract semantic knowledge.

Sheng et al. [96] created DEKGB, a KG of various diseases using prior medical knowledge and electronic medical records (EMR), along with guidance from doctors. Li et al. [97] used quadruplets instead of triples to represent their KG, with the extra information relating to relation strength. Zhang et al. [98] used a clinician-in-the-loop to fine-tune an automated KG construction method.

KGs have also been created for specific medical domains. Huang et al. [99] made a KG solely focused on depression after observing the prohibitive size and high-level nature of general medical KGs. A low-level KG for depression would allow more convenient use by doctors, easier understandability by the public, and higher computational efficiency.

Chai [100] used a KG about thyroid disease as the backbone for an intelligent medical diagnosis system. Vector embeddings were calculated for each entity and relation in the KG. These embeddings were then used to train a bidirectional long short-term memory (LSTM) network as a disease diagnosis model, outperforming other tested machine learning models.

Flocco et al. [101] used tweets related to COVID-19 in the Los Angeles area, along with policy announcements and disease spread statistics, to construct a KG representing the real-world spread of COVID-19 in the Los Angeles area. The sentiment of each tweet was calculated using a rules-based method, and topic modeling was used to extract popular keywords from tweets.

3.4 Methodology

Constructing a KG requires data and methods to represent this data in a format suitable to be a part of the KG. The following sections outline how data has been collected and the methodology used to process it for graph construction purposes. The proposed and applied methodology is illustrated in Figure 3.1. Data processing methods used to analyze texts from websites include NER, topic modeling, document classification, and location detection.

3.4.1 Data Collection

Two sources were used to construct the NDD corpus of text required for the KG. The first source included individuals with lived experience who are part of the family advisory board or have been recruited through advertisement for the project and community support groups focused on NDD: AIDE Canada [102], the Alberta Children's Hospital NDD Care Coordination Project [103]. These individuals/parents were asked to provide links to websites relating to NDDs in such categories as core knowledge, education, services, and funding. The corpus created from these sources will be referred to as the *NDD Caregiver subset*.

The second source of relevant web pages used for scraping was the Inform Alberta website [104]. These resources are referred to as the *Inform Alberta subset*. Finally, the combined list of web pages from both sources and some relevant pages added by the authors was scraped using the Python Scrapy library. For homepages, the entire site was scraped, while for specific/single web pages only those pages were scraped.

As a result, the obtained corpus consisted of 200,000 web pages, with 80,000 pages from the *NDD Caregiver subset* and 120,000 pages from the *Inform Alberta subset*. HTML text was extracted for each page and cleaned by removing boilerplate text using the Python BoilerPy3 library. The collection of cleaned HTML text from the web pages formed the corpus of documents used for the construction of ***NDD-KG***.

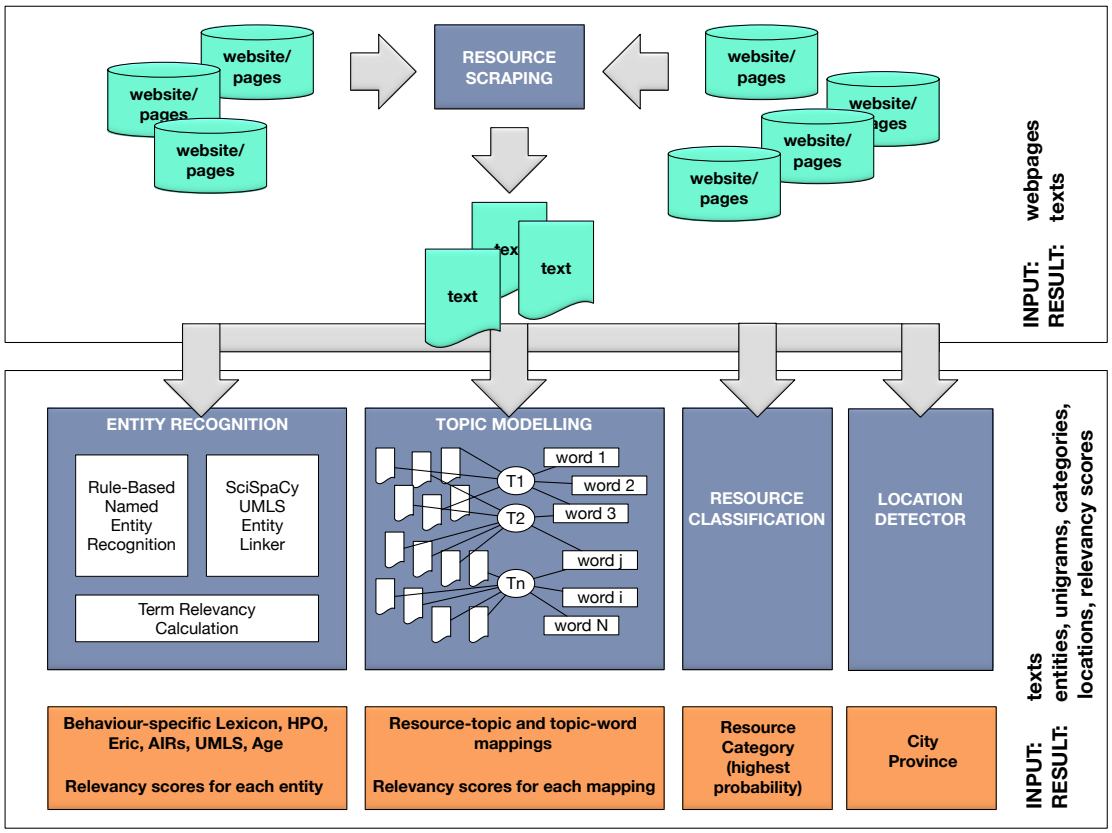


Figure 3.1: Resource website/pages annotations process

3.4.2 Named-Entity Recognition

The list of NDD resources contains a mixture of website, homepage, and web page urls. To perform web page level indexing, when a given url refers to a homepage/website, the Scrapy framework was used to scrape all the web pages of that website. Repetitive urls were removed from the final list of all the web pages. Many web pages contained the same HTML boilerplate, such as headers, navigation bars, and footers. The Boilerpy3 python library was used to remove this boilerplate HTML.

Entity Vocabulary

The dataset contains various web pages related to services, education, financial help, and core health knowledge within the NDD field. Different standard terminologies, including Unified Medical Language System (UMLS), Human Phenotype Ontology (HPO), Education Resources Information Center (ERIC) thesaurus, and a taxonomy by the Alliance of Information and Referral Systems (AIRS), were extracted from the pages. In the constructed graph, they were used to annotate the web page URLs.

UMLS is a collection containing over 4 million concepts from over 100 controlled vocabularies including but not limited to ICD10, MeSH, and SNOMED CT [105]. It covers all the medical and related entities. HPO provides a standardized vocabulary of phenotypic abnormalities encountered in human diseases. It currently contains over 13,000 terms [106]. ERIC thesaurus is a list of topics in education and comprises about 11,818 terms, including 4,552 unique terms called descriptors and 7,133 synonyms of descriptors [107]. AIRS taxonomy is the North American standard for indexing and accessing human service resource databases [108]. The taxonomy is a hierarchical system containing more than 9,000 terms covering the complete range of human services.

As some web pages were more specific to a particular age as well as location, a list of age terms and all Canadian cities and provinces was used to index web pages. As behavioral issues are common in individuals with NDD, with expert's feedback, 10

categories of challenging behavior were considered: sleep issues, sensory issues, hyperactivity, inattention, repetitive behavior, speech and language development, adaptive behavior, cognitive development, social skills, and behavioral concerns. For each category, we collected commonly used phrases or synonyms with the help of the parent advisory group, as well as a manual search on the UMLS interface [109].

Named-Entity Recognition Process

NER is a sub-task of Natural Language Understanding used to detect named entities that refer to specific objects. The named entities we used were domain-specific terms such as medical terms, educational terms, services, challenging behaviors, age, and location. All controlled vocabulary terms were given an entity label the same as their source vocabularies (i.e., HPO, ERIC, AIRS, age, and location). Similarly, All challenging behavior phrases or vocabulary were labelled with their respective categories. They were lemmatized using the NLTK library [110]. A single pattern file was created as an input into SpaCy's rule-based entity recognition component called EntityRuler. A pattern file is a dictionary with two keys: a 'label' specifying the label to be assigned to the entity if the pattern is matched, and a 'pattern' indicating the phrase to be matched. Webpage text was preprocessed by removing stop words and lemmatizing the text and passed to EntityRuler to annotate the text. The UMLS entity Entity Linker from an open-source framework SciSpaCy [111] was used to extract UMLS entities from text and only the respective canonical concepts of UMLS entities were considered for further analysis.

Entity Relevance Calculation

Indexing the web pages with the existence or non-existence of an entity does not provide information if a document is more relevant to a given entity. A document that mentions a given entity more often than the other documents could be considered more relevant to this entity. Depending upon the number of occurrences of an entity,

a weight is assigned to each entity – it is called an entity relevance weight. The weight provides information on how relevant an entity is to a document. However, using the term (entity) frequency alone will favor common words as well as long documents [112].

It is essential to normalize the term (entity) frequency to incorporate such factors as high term frequency and document length. This is especially so in the case of HTML documents because of keyword stuffing, a process where website owners deliberately add specific keywords to their site in order to improve its search engine ranking. We use logarithmic term frequency as a way to de-emphasize high-frequency terms and adjust within-document term frequency.

For normalization, the *pivoted unique normalization* method was used, which considers the document length as a factor. The principle of the pivoted normalization is as follows: the higher the value of the normalization factor for a document, the lower the chances of its retrieval. Therefore, to boost the chances of retrieving documents of a certain length, the normalization factor for those documents should be lowered. Singal et al. [112, 113] suggested considering the average document length in a corpus as a reference point, called the *pivot*, and using a parameter called the *slope* to penalize longer documents and give higher weight to shorter documents. Normalized term relevancy weight is defined as:

$$relevance = \frac{1 + \log(tf)}{(1 - slope) * pivot + (slope * d_l)} \tag{3.1}$$

where tf is the term frequency in the document, and *slope* is set to 0.2 as suggested in Singal's work. The value of *pivot* is set to the average number of distinct named entities per document in the entire collection, and d_l is the length of the documents referred to by a unique number of entities in a document. Documents with the length $d_l = pivot$ are not penalized as the normalization factor is equal to the pivot. For $d_l > pivot$, documents are penalized and have lower chances of retrieval, while for $d_l < pivot$, documents are rewarded with a smaller normalization factor.

3.4.3 Topic Modeling

Topic modeling using latent Dirchlet allocation (LDA) was used to extract similar topics across the corpus for inclusion in the KG. A novel form of topic modeling, referred to as hierarchical topic modeling (HTM), was used to extract more specific topics from the corpus. Topic modeling was performed separately on the *NDD Caregiver subset* and on the *Inform Alberta subset* of the corpus due to computational constraints. Unigram topics were extracted.

Data Preparation

Each web page (document) in a subset of the corpus was pre-processed before being transformed into a count vector for modeling with LDA. The first step was to remove all punctuation from the document, followed by changing all words to lowercase. Next, the document was tokenized and lemmatized. Finally, a stop list was used to remove unwanted words from the document. The stop list used here was the default English stop list from NLTK augmented with some words added by the authors through iterative testing and analysis of the topic modeling outputs. Finally, pre-processed documents were transformed into a count vector for LDA.

Topic Modeling Process

The HTM algorithm initially performed LDA on the corpus subset and re-performed LDA on topics containing several documents greater than a chosen threshold. Then, the process was repeated until each topic included less than the threshold number of documents, or no more progress was made. It resulted in more specific topic words than running LDA once over the whole corpus, as found by a subjective analysis comparing the outputs of both methods. The LDA algorithm from the Python scikit-learn library was used with the following hyperparameters: max iterations of 10, the online learning algorithm, learning decay of 0.7, batch size of 128, and max features of 50,000.

For the *NDD Caregiver subset* of the corpus, the initial LDA was set to have 200 topics, and for the *Inform Alberta subset* to have 300. These numbers were chosen to be in proportion to the number of documents in each corpus subset. The threshold for hierarchy termination was set to 300 documents for both corpus subsets. Only the lowest level of the topic hierarchy was used for the KG construction, as these topics seemed to be the most relevant following a subjective analysis.

Topic Relevance Calculation

It is essential to have information about the 'strength' of connections between identified topics, documents (web pages), and unigrams, i.e., words identified by LDA as describing each topic and indirectly representing documents associated with a given topic. In the case of LDA, such information was extracted from the LDA algorithm.

3.4.4 Document Classification

There were five categories of web pages in the corpus: 'financial help', 'education', 'services', 'core knowledge/health' and 'other'. To automatically label each web page a few classification models were investigated. To construct models, a subset of the corpus was hand-labeled as belonging to one or more of the five categories. This is a multi-label classification task, as documents (web pages) can belong to more than one category.

The hand-labeled data consisted of 2158 documents, with 116 labeled as 'financial help', 420 as 'education', 1419 as 'services', 1024 as 'core knowledge/health' and 143 as 'other'. This data set was highly unbalanced. The data set was split into train, validation, and test sets, with 80% of the data used for training, 10% for validation, and 10% for testing. The data was split equally along categories where possible.

We tested three groups of models for classifying these documents: (1) multi-label k-nearest-neighbors, (2) five single-label transformers, and (3) a multi-label transformer. Among these models, we ultimately chose the multi-label transform-

ers, due to it achieving the highest macro f1 score on a held-out test set. The multi-label transformer was the 6-layer version of MiniLM-v2 fine-tuned on the prepared training data set. The pre-trained model found on the HuggingFace website named *nreimers/MiniLM-L6-H384-uncased* was used; it is the same model as the *all-MiniLM-L6-v2* from *Sentence-BERT* [114]. A dropout layer, with dropout probability of 0.3, and a final sigmoid activation layer with 5 outputs were added to the base model as a multi-label classification head.

Training hyperparameters for this model were as follows. The loss function used for training was the binary cross entropy loss that was optimized using the AdamW optimizer. The optimizer learning rate was 3×10^{-4}, with a linear warmup to this value and cosine decay to one-tenth of this value during training. The other optimizer hyperparameters were $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1 \times 10^{-8}$, and a weight decay of 0.01. The batch size was 64, and all gradients were clipped to a norm of 1.0 to mitigate gradient explosion.

The model was fine-tuned for 20 epochs. This is higher than the 2-3 epochs used in the original BERT paper [21], but we mitigated possible overfitting by increasing the dropout probability and evaluating model performance on a held-out validation set after each epoch. The model with the best performance on the validation set was chosen as the final model. The model outputs five probabilities between 0.0 and 1.0. A threshold value was chosen where values above this threshold were considered members of the corresponding class. Finally, a more fine-tuned macro f1 score was calculated on the validation data set for threshold values from 0.0 to 1.0 in intervals of 0.1.

The best version of the multi-label transformer model, determined based on the validation set, achieved a macro f1 score of 0.504, and an accuracy of 84.1% on the test set. For reference, the training set macro f1 score was 0.804 with an accuracy of 93.8%. Details of the selected model's performance can be found in Table 3.1.

Label		Financial Help	Education	Services	Other	Core Knowledge / Health
Train	Cases	89	331	1131	111	815
	Precision	0.745	0.769	0.920	0.756	0.860
	Recall	0.787	0.764	0.984	0.532	0.977
	F1	0.765	0.767	0.951	0.624	0.914
Test	Cases	12	40	141	14	101
	Precision	0.143	0.333	0.840	0.400	0.743
	Recall	0.083	0.500	0.965	0.286	0.832
	F1	0.105	0.400	0.898	0.333	0.785

Table 3.1: Multi-label transformer model performance results

3.4.5 Location Detection

Using regular expressions, link text was matched to scrape specific pages such as "contact us", "our locations", and "locate us". Then Canadian/US postal codes were matched using regular expressions and queried using the Google Maps API to get the city and province for a given postal code. Named entities were detected, along with cities and provinces. To get the final annotations, results from both modules were combined. As it was challenging to remove false positive location entities due to the manual annotation requirements, each city/province was given a weight equal to the proportion of entities that refer to a city/province. This way, for a given city/province, resources could be ranked based upon the score.

3.5 Results

The presented methodology of processing resources (i.e. web pages) provides a collection of items (i.e., entities, unigrams, age ranges, locations, web page categories) and challenging behaviours used to annotate the web pages. The integration of this information is done using a knowledge graph – ***NDD-KG***. The web pages and items

mentioned above are nodes, while the relevance between them is represented as edges labeled with a relevancy strength.

3.5.1 Knowledge Graph Schema

NDD-KG, as any KG, is a network of entities connected through relations. Each piece of knowledge in a KG is represented as a triple, with two entities connected through a relation. These triples are in the form of $\langle$*subject, relation, object*$\rangle$. For example, to represent the piece of information that Edmonton is the capital of Alberta in a graph, the following triple is used: $\langle$*Edmonton, capital of, Alberta*$\rangle$.

To effectively utilize a KG, names representing types of KG nodes and relations between the nodes must be established. This set – called the *vocabulary* – is one of the essential aspects of constructing a KG. The vocabulary is often called the KG schema.

The ***NDD-KG*** schema is shown in Figure 3.2. Names of node types are represented by circles, differing in colour by node type. Some of them are labeled with extra information, shown in the text inside the node. Links between the nodes represent relations between entities. Some relations are labeled with extra information, shown in text on the relation arrow. Each node type and relation type are outlined in the following sections. All collected data was represented as triples and fed into Neo4j to construct the graph automatically.

Entities as Nodes

There are a total of 11 node types in the ***NDD-KG***. The primary node type is *resource*. It represents all the documents (web page URLs) in the corpus. Each of these resources is labeled with their associated URL from the corpus, the resource source (*NDD Caregiver* or *InformAlberta*), and the resource type (web page/video/pdf). The document text was not saved in the KG for size reasons. Instead, an external file was kept with processed document text for each URL, and each URL could be

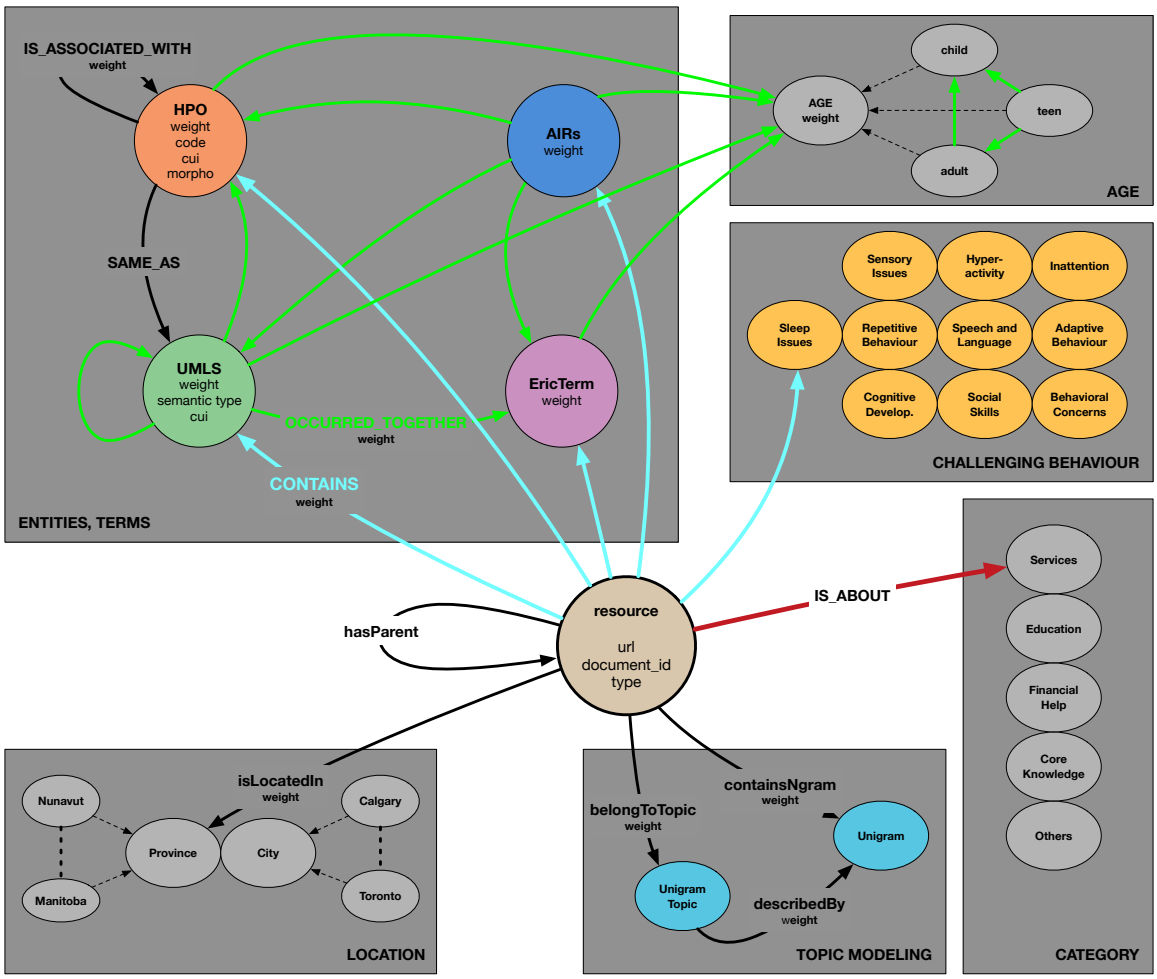


Figure 3.2: Knowledge graph schema: links of the same color represent the same relations; dashed links represent relation *"is-a"*

accessed through the Internet, assuming the web page is still active.

The other node types are linked by edges, either directly or indirectly, to a *resource* node. *HPO*, *UMLS*, *EricTerm*, *AIRs* and challenging behaviour nodes were extracted using Named-Entity Recognition methods. *EricTerm* nodes are from the ERIC database and are labeled by the canonical term for the recognized entity. *UMLS* nodes are from the UMLS database and are marked by the canonical term for the recognized entity, its semantic type, and concept unique identifiers (CUI). *HPO* nodes are from the HPO-DDD database and are labeled by the canonical term for the recognized entity and unique HP and CUI identifiers. *AIRs* nodes are from the AIRs database and are labeled by the canonical term for the recognized entity.

Two types of nodes represent terms extracted from the topic modeling method. First, unique topic nodes were placed into the ***NDD-KG***, and then each resource (web page) was linked with the unigram topic. Further, each unique topic node was connected with the corresponding unigram terms that were represented as the ***NDD-KG*** nodes.

The remaining four node types are *Province*, *City*, *Age*, *Category*, and *Challengingbehaviour*. The *Province* and *City* for each resource were extracted using methods outlined in the location detection method. The age associated with each resource was also extracted using similar methods. Possible subtypes for *Age* are *Child*, *Teen*, and *Adult*. The node type *Category* has five subtypes: *Services*, *Education*, *Financial Help*, *Core Knowledge*, and *Others*. Resources were linked to one of the subtypes after the classification method outlined in document classification was executed.

Relations and Weights

Nodes are connected by edges to one of eight types of relations. Web pages scraped from a parent website, and both represented as resource nodes, are connected to the relation *hasParent*. Location nodes, for both cities and provinces, are connected to the relation *isLocatedIn*. City nodes are connected to their corresponding province nodes with the relation *inProvince*. NER-related nodes, *Age*, and *ChallengingBehaviour* nodes are connected to corresponding resource nodes with the relation *CONTAINS*. NER-related nodes are also connected to identically named entities from different databases with the *IS_ASSOCIATED_WITH* relation. NER-related nodes and *Age* are connected to each other with a relation *OCCURRED_TOGETHER*.

Topic nodes are connected to corresponding resource nodes with the relation *belongsToTopic*. The relation *describedBy* was used to connect topic nodes to their contained topic word – unigram nodes. Finally, resources are directly connected to relevant unigrams with the *containsNgram* relation.

Relations were assigned a weight using various methods if applicable. The *in-*

Province, *hasParent*, and *describedBy* have no weights. The relations *CONTAINS*, *IS_ASSOCIATED_WITH*, and *isLocatedIn* are weighted using term relevancy as outlined in the named entity recognition method. The relation *OCCURRED_TOGETHER* is labeled with several co-occurrences of connected entities (nodes).

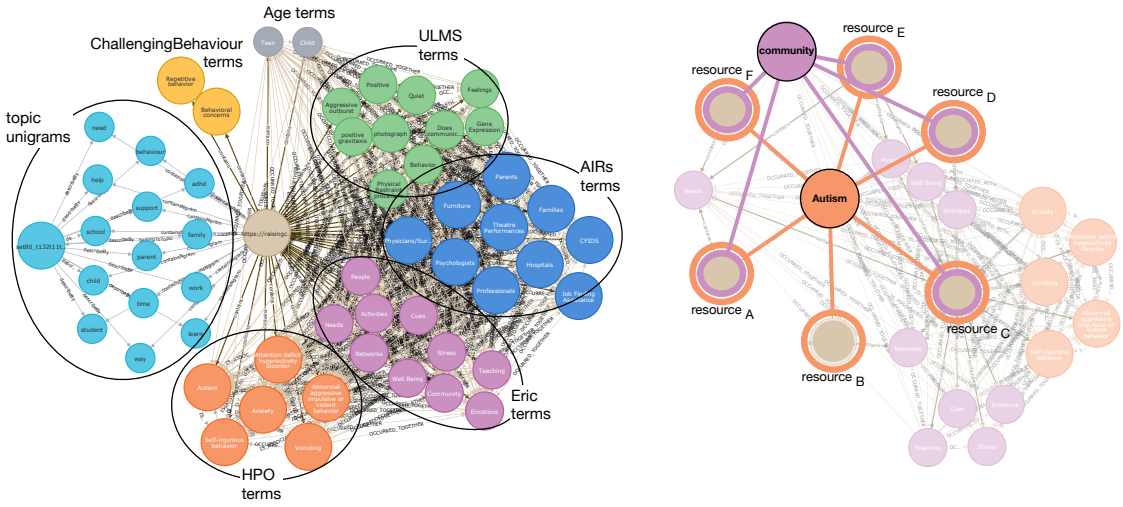
The relations *belongsToTopic*, *containsNgram*, and topic-related *describedBy* have weights calculated as the output of the LDA process. Weights for the *belongsToTopic* represent a degree of how strongly a resource belongs to each topic. Weights for the topic-related *describedBy* relations indicate how strongly each word in the topic vocabulary belongs to each topic. The *containsNgram* weights were obtained by matrix multiplication of the *belongsToTopic* and topic-related *describedBy* weight matrices.

3.5.2 Constructed Knowledge Graph: Overview

The constructed ***NDD-KG*** contains 264,167 nodes. There are 185,986 resource nodes. For NER-related nodes, there are 2,448 *AIRs* nodes, 11,617 *EricTerm* nodes, 4,181 *HPO* nodes, and 41,599 *UMLS* nodes. For topic modeling, there are 14,373 unigram nodes and 2045 unigram topic nodes. In addition, there are 3 Age nodes, 5 Document Category nodes, 10 Challenging behaviour nodes, 1832 City nodes, and 68 Province/State nodes. The graph contains a total of 22,621,522 relations.

To illustrate interesting features of the graph, a single resource is extracted from ***NDD-KG*** together with several annotated nodes on the left (a) of Figure 3.3. The resource, a light brown circle in the middle, is linked with a group of unigrams (blue circles on the left), two types of *Challengingbehaviour* nodes (yellow circles), *Age* (gray), *UMLS* (green), *AIRs* (blue), *EricTerm* (violet), and *HPO* (orange) nodes.

These terms and unigrams define/describe the resource. The relations between resources and the annotating nodes are n-to-n. This means that a single annotating node is also linked with multiple resources. Such a scenario is illustrated on the right (b) of Figure 3.3. Two terms – *HPO*'s autism and *EricTerm*'s community – are



(a) (b)

Figure 3.3: Example of annotated resources from *raisingchildren.net.au/autism/behaviour/common-concerns/aggressivebehaviour-asd*: (left: a) most relevant annotating nodes; (right: b) n-to-n relations between resources and annotating nodes

connected to multiple resources.

Besides the relations resource-annotating node, the graph contains multiple relations between annotating nodes. These are two types of relations *OCCURRED_TOGETHER* and *IS_ASSOCIATED_WITH*. A fragment of the graph illustrating these relations is shown in Figure 3.4.

The existence of these many relations creates a highly interconnected representation of resources and their annotated nodes. However, it introduces an issue if a user tries to identify the most relevant resources, as there would be no difference in relevance between nodes. Therefore, a degree of relevance is added to denote the most appropriate resources. The relevance represents the connection strength between a resource and an annotating node (entity).

There are two types of relevance weights used in ***NDD-KG***. One weight is linked with the relation *CONTAIN*. Its value is determined using the procedure presented in the entity relevance calculation method. The other weight is linked with the relation *OCCURRED_TOGETHER*. This weight is a measure of the co-occurrence of different

annotating nodes.

The first type of weight is illustrated on the left (a) of Figure 3.5. Although all connected nodes contribute to the description of a resource, their contributions are of different strengths. The second type of weight is illustrated on the right (b) of Figure 3.5. A snippet of the graph shows connections between annotating nodes: *HPO*, *AIRs*, and *EricTerm*. The weights are represented as integer numbers that indicate how often both terms co-occurred in the extracted texts (i.e., degrees to which given nodes are 'related to each other').

3.5.3 Constructed Knowledge Graph: Utilization

The ***NDD-KG*** can be perceived as a source of valuable information about connections between resources and annotating nodes. The graph can be used to identify the most relevant resources when a user provides a textual phrase expressing her interests.

The richness of connections of the graph and a need to provide a user with the best possible match to the entered phrase has led to the application of an aggregation operator that combines the weights (i.e. the values of relevance between resources and annotating nodes) in a 'controlled' way. It means that if a user wants to find the most relevant resource satisfying multiple nodes, an aggregation function is invoked and combines all suitable relevance values.

A simple use case showing the abilities of ***NDD-KG*** to provide the user with relevant NDD resources is included.

User Interface for Resource Extraction

NDD-KG can be used to identify the most relevant resources when a user provides some text input. A simple web-based interface has been developed to enable users to use ***NDD-KG*** when they want to obtain a list of relevant resources. The interface allows the user to enter a text query containing several phrases representing their interest. Considering that end users can search with verbose queries, such as "my child

hits other children at school," to infer one of the 10 challenging behaviour categories, we have also trained a text classification model that will classify the intent of the entered user text (section 3.5.3).

Additionally, the richness of connections of the graph and the need to provide users with the best possible match to their text query led to the need for an aggregation operator that combined the weights (i.e. the values of relevance between resources and annotating nodes) in a controlled way. This meant that if a user wanted to find the most relevant resource for a text query that satisfied multiple nodes, an OWA aggregation function was invoked. OWA combines all suitable weights for these nodes to rank the resources based upon their relevance (section 3.5.3).

The entered text is processed with the following steps:

1. Extract unigrams and entities (HPO, UMLS, ERIC, AIRS, and challenging behaviour) using the developed natural language processing pipeline.

2. Classify user text as one of the 10 challenging behaviour categories using the transfer learning–based text classification model. Then, add the detected category to the entities list obtained in step 1.

3. Query the KG-based repository of NDD resources to retrieve all resources that are connected to nodes representing entities and unigrams obtained in the above steps 1 and 2. For each retrieved resource, all annotating nodes are extracted together with the weights of the relations.

4. The weights are aggregated using OWA to determine the relevance of each retrieved resource.

5. A list of "sorted by relevance" resources is displayed to the user.

The text query, along with extracted entities and unigrams, is shown for a simple example in Figure 3.6. As can be seen, *HPO* and *UMLS* entities have been identified ("abnormal, aggressive, impulsive, or violent behavior" and "spitting," respectively).

Additionally, behavioural concerns as a category of challenging behaviour has been recognized. Two unigrams ("aggressive" and "behaviour") are extracted from the text. The resulting list of the most relevant sites has been determined and is shown in Figure 3.7. The list contains web pages and a video, all from the category *core knowledge*.

Challenging Behaviour Detection

We fine-tune a BERT model [21] to detect mentions of challenging behaviour within text queries. The training data set included 1,219 natural language descriptions of challenging behaviours, each annotated by the recruited parents group into one of the ten available categories. These ten categories are: Cognitive development, Sleep issues, Speech and language development, Sensory issues, Social skills, Hyperactivity, behavioural concerns, Inattention, Adaptive behaviour, and Repetitive behaviour. A few examples of the text and their annotated categories can be seen in Table 3.2.

Text	Category
He just can't deal with the work he needs to do in order to have help	Adaptive Behaviour
Tendency to be over-sensitive to noise	Sensory Issues
He isn't particularly interested in our weekday routines and mostly sleeps through them	Sleep Issues

Table 3.2: Examples of text and their annotated categories

A pre-trained BERT language model with a fully-connected classification head was used to classify the natural language text into one of the ten annotated categories. The smaller version of the pre-trained BERT model, called the base model, was used to have faster inference time while still resulting in acceptable accuracy. The BERT-base model has almost 110 million parameters. The output of the pre-trained BERT model was passed through a dropout layer with a probability of 0.1 to avoid overfitting to the training data [115]. The fully-connected layer has an output of ten features in

order to be able to classify between the different classes.

The data set was split randomly with a seed of 66 to training and validation sections with an 80/20 ratio. The pre-trained model used was BERT-base. The input text was truncated, and the maximum input size set to 128 tokens. The classifier model was trained using the AdamW optimizer [116] with a learning rate of 2×10^{-5} and weight decay of 0.01 for ten epochs. The code was written in PyTorch. Fine-tuning was done using an NVIDIA Titan RTX GPU.

The fine-tuned model resulted in an accuracy of 85.7% on the validation dataset. Among the different classes, the model was the most confused when text regarding repetitive behaviour was given, and it incorrectly predicted behavioural concerns. The full confusion matrix can be found in Figure 3.8.

The final trained model has a disk space size of 433 MB. The model is hosted on a remote server and is served using an API. Remote hosting of the classification model results in leaner code for the chatbot. This modular architecture also results in an easy and independent update of the classification model without adversely affecting the rest of the chatbot architecture. The model is currently hosted on HuggingFace server 1 and, on average, has an inference time of less than 200 ms, which makes it a good fit for time-sensitive applications, such as our case of chatbots.

OWA

One of the most interesting and commonly used aggregation operators is the Ordered Weighted Averaging (OWA) operator [93]. In the simplest possible statement, this operator is a weighted sum of ordered pieces of information. In a formal representation, the OWA operator, defined on the unit interval I and having dimension n, is a mapping $OWA : I^n \longrightarrow I$ such that

$$OWA(a_1, a_2, ..., a_n) = \sum_{j=1}^{n} w_j * b_j \tag{3.2}$$

where b_j is the j^{th} largest of the a_i's. $W = \{w_1, w_2, ..., w_n\}$ is a weighting vector such that $0 \leq w_j \leq 1$ and $\sum_{j=1}^{n} w_j = 1$.

To obtain a weighting vector W associated with an OWA, a family of RIM quantifiers Q has been introduced. A fuzzy subset Q represents a RIM quantifier if:

1. $Q(0) = 0$;

2. $Q(1) = 1$;

3. $Q(r_1) > Q(r_2)$ for all $r_1 > r_2$

Assuming a RIM quantifier Q, the weighting vector W can be determined such that for $j = 1$ to n:

$$w_j = Q(\frac{j}{n}) - Q(\frac{j-1}{n}). \tag{3.3}$$

A function Q can be of different form and be associated with different linguistic quantifiers, such as *for all*, *mean*, *most*, or *as many as possible* [117]. In the paper, the quantifier *most* is used, which leads to the following form of Q:

$$Q(r) = \begin{cases} 0 & \text{if } 0 \leq r \leq \alpha \\ \frac{r-\alpha}{\beta-\alpha} & \text{if } \alpha \leq r \leq \beta \\ 1 & \text{if } \beta \leq r \leq 1 \end{cases} \tag{3.4}$$

where $\alpha = 0.3$ and $\beta = 0.8$.

Resource Ranking Process

Identifying a list of most relevant resources is represented as an Algorithm 1. The algorithm takes ***NDD-KG*** and the user's phrase as its input. The phrase is processed, and sets of entities and unigrams are obtained, lines 11 and 12. Based on both sets, a *Neo4j* query is created, and a set of resources is obtained, line 14.

The weights associated with the relations *CONTAIN* are used to determine degrees of relevancy. In the beginning, the weights of all relations *CONTAIN* are extracted, lines 16 to 19. Similarly, the weights of links connecting the resource and unigrams are retrieved.

The retrieved weights – $\mathbf{Weight}_i^E$ and $\mathbf{Weight}_i^{UG}$ – for a resource $\mathbf{Res}_i$ are aggregated individually using OWA, and then the results are multiplied, line 24. The obtained value – $\mathbf{significance}_i$ – is used to determined a ranking of all resources that satisfy the user's phase.

Algorithm 2: Resource Ranking Algorithm

```
1  Input:
2    NDD-KG
3    UInput                                          // User_Input_Phrase
4  Output:
5    RRL                                           // Ranked_Resource_List

6  Initialization:
7    set: UEntity = {}                 // Entities extracted from UInput
8    set: UUniGram = {}                // Unigrams extracted from UInput
9    set: SRes = {}                                  // Selected Resources
10   list: RRL = []

11 UEntity ← entityExtraction(UInput)
12 UUniGram ← unigramExtraction(UInput)
13 Neo4j_query ← queryConstruction(UEntity ∪ UUniGram)
14 SRes ← execute(Neo4j_query)

15 for each Res_i from SRes do
16     for each entity_j from UEntity do
17         weight^E_{i,j} ← getWeight(Res_i − contains − entity_j)
18         Weight^E_i ← weight^E_{i,j}
19     end
20     for each unigram_j from UUniGram do
21         weight^{UG}_{i,j} ← getWeight(Res_i − contains − unigram_j)
22         Weight^{UG}_i ← weight^{UG}_{i,j}
23     end
24     significance_i = OWA(Weight^E_i) * OWA(Weight^{UG}_i)
25     ResList_significance ← ⟨Res_1, significance_i⟩
26 end

27 RRL ← ranking(ResList_significance)
28 return
```

3.6 Discussion

This paper describes the methodology for processing texts extracted from web pages to generate a set of entities and terms used for annotating these web pages. Both web pages and entities/terms are the basis for constructing a knowledge base of resources. This base — built as a graph called ***NDD-KG*** — is a highly interconnected network linking resources with annotating terms and entities. Furthermore, edges in ***NDD-KG*** have weights representing relevance between resources and annotating terms/entities. Edge weights, aggregated using a specialized aggregation operator, are used to rank resources. The constructed ***NDD-KG*** is a repository of resources about NDD that can be queried using textual phrases, with relevant results shown to the user using an interface.

Most of the prior work in building medical knowledge graph uses scientific literature such as PubMed and electronic medical records and only specific types of entities such as diseases, chemicals, and genes are considered [97, 118, 119]. Ernst et. al. [120] uses the patient-oriented online health portals to build KG indicating the importance of medical information spread across different sources. Shi et al. [95] represented heterogeneous textual medical knowledge as one KG to utilize it further for semantic reasoning. Yu et. al. constructed a KG for traditional chinese medicine to integrate terms, documents, and databases in one base to facilitate sharing and utilization of TCM health care knowledge [121].

To our knowledge, this is the first method which integrates credible online information from different areas of need around NDDs (i.e. financial help, services, education, and core knowledge) into a single location. Our developed NLP pipeline can be used to annotate resources from the above-mentioned areas. Representing the extracted knowledge into a KG would allow finding connections among different resources on a scale that would be impossible for a single human. Many ontologies and scattered information sources, both on a professional and layperson level, exist on the internet.

Our KG compiles all this information into a single place. Connecting all this information will open up many areas of improvements for the NDD field. These could include new research directions, new treatment opportunities, and the possibility of collaboration between services.

The methodology used to construct this KG is scalable and could be expanded to other medical domains besides NDDs. In creating more of these domain-specific medical KGs with the guidance of medical professionals, patients, and caretakers, we can provide information in a similar way to patients afflicted with other conditions. These specific KGs could even be connected on a higher level to slowly create a field-wide medical KG, which would be of great benefit in complex medical conditions where individuals present with multiple hyper-specialized domains. The bottom-up nature of the creation of this may result in a better product than the top-down field-wide medical KGs that currently exist.

While having more documents means more information is available, there is also a trade-off between number of documents and KG query speed [122]. Some ways to overcome this include indexing the KG [123], and pruning irrelevant nodes and edges on the KG [124]. Another limitation of KG construction is that pivoted unique normalized logarithmic term frequency is used to calculate weights for edges labeled *CONTAINS*, which affects the performance of the resource retrieval method when the size of the document is significantly greater than the average document length in the corpus. Pivoted unique normalization over-penalizes longer documents as shown in the original paper [125]. When the length of a document is much larger than the average document length in the corpus, a higher normalization factor could yield almost zero relevance score for that document's entities [125]. This limitation can be overcome by implementing a term relevance method which not only considers the term frequency but also co-occurring terms (represented with the OCCURRED_TOGETHER relationship in ***NDD-KG***) [126].

As future work, ***NDD-KG*** based semantic search methods will be further studied

to address the exact keyword matching issue in the resource retrieval system. Potential solutions include using query expansion techniques [127–129]. The application of OWA in identifying the most relevant list of resources opens another possibility of enhancing a user query interface. OWA is known for its ability to include linguistic quantifiers such as *SOME*, *MOST*, *ALL*, and *AT LEAST* n in the aggregation process. So far, we only use *MOST* to aggregate query results, yet a user can control to what degree documents should satisfy different criteria using different quantifiers.

To further improve the user's experience with the ***NDD-KG*** based resource retrieval process, we aim to build a transparent interface that will enable path-based explanations in ***NDD-KG*** to provide relevant background knowledge in a human-understandable format [130, 131] using interpretable machine learning approaches. Explainable Artificial Intelligence is an emerging research field which not only focuses on the performance of the models but also the interpretability of what factors led the model to make a particular decision. This promotes credibility and trust in the suggested results [132, 133].

Although patients and caretakers were included in some of the vital steps of creating this KG such as collecting resources and challenging behavior vocabulary, user feedback is an important step in the process of validating our created KG. We will design the evaluation strategy to validate the ***NDD-KG*** based document retrieval system by collecting the gold standard relevance assessment from the human judges. Douze et. al. [134] found that the relevance assessments provided by the human judges also depend upon their subjective needs. Therefore, we will collaborate with a group of parents of children with NDDs to create a gold standard test collection to evaluate the model to check if the ***NDD-KG*** based document retrieval satisfies their needs.

3.7 Conclusion

The need for helping families with NDDs could leverage the potential that online information has to offer (e.g. to supplement gaps in the health/social support system). This need became more important than ever during the COVID-19 pandemic and will continue gaining importance into the future. Building an efficient repository of trusted web resources has proven challenging due to the lack of uniformly labeled resources. This challenge is not unique to NDD and is seen across other medical fields as well. Such repositories of online resources should provide users with an intelligently generated ranking of resources based on a simple text query entered by the users. Experts and non-experts can use NDD-KG to improve the QoL of people with NDDs. Future work includes enhancements of user interface for resource retrieval as well as mechanisms for continuous modifications of NDD-KG when new information is discovered, or old information is found to be outdated.

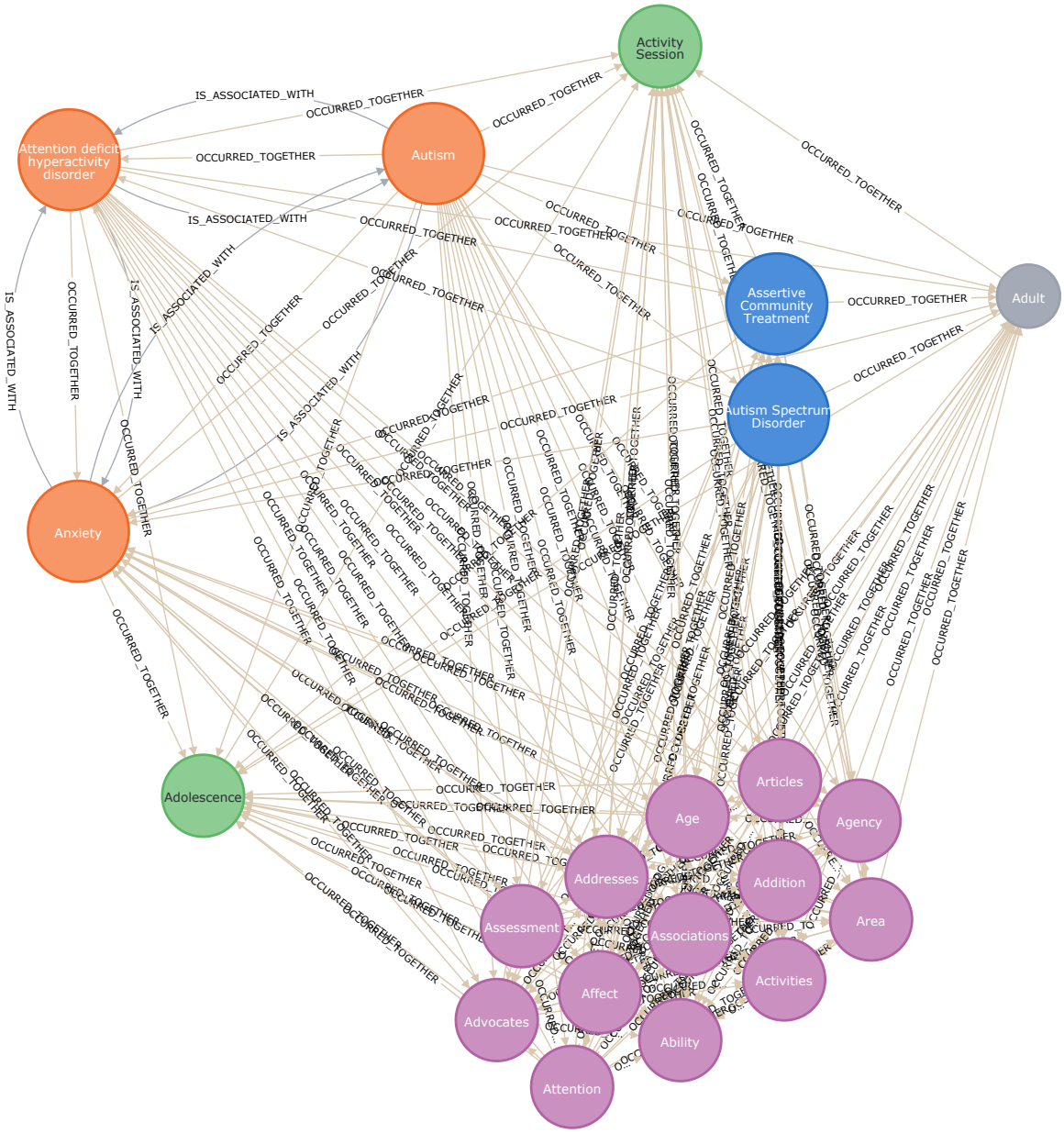
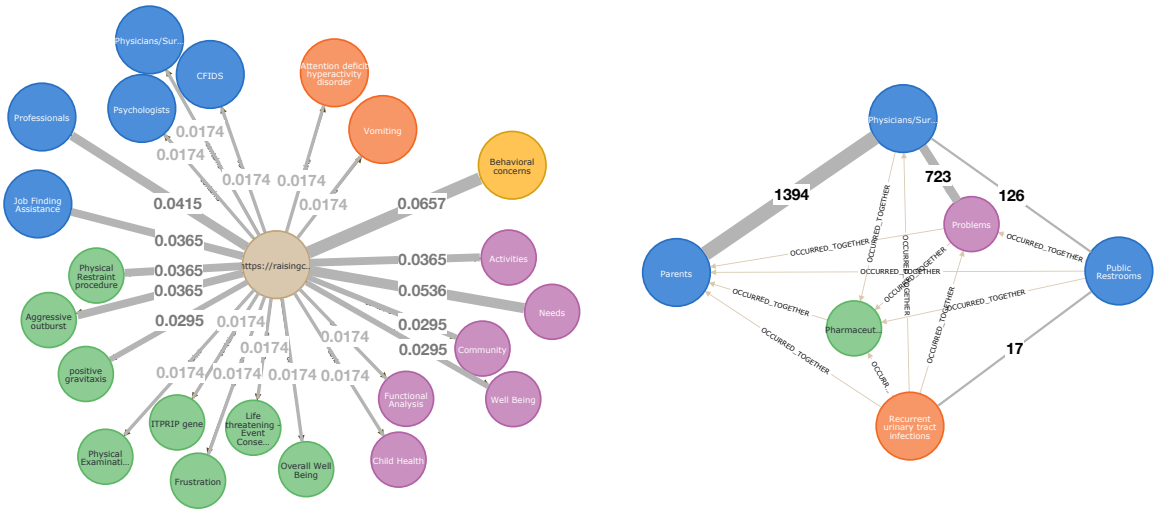


Figure 3.4: Example of *OCCURRED_TOGETHER* and *IS_ASSOCIATED_WITH* connections between nodes. Entities from different sources are represented in different colors: *HPO* (orange), *AIRS* (blue), *UMLS* (green), *ERIC* (purple), *AGE* (grey)



(a) (b)

Figure 3.5: Example of connection strength: (left: a) of relation *CONTAIN* between resource and annotating nodes; (right: b) of relation *OCCURRED_TOGETHER* between annotating nodes

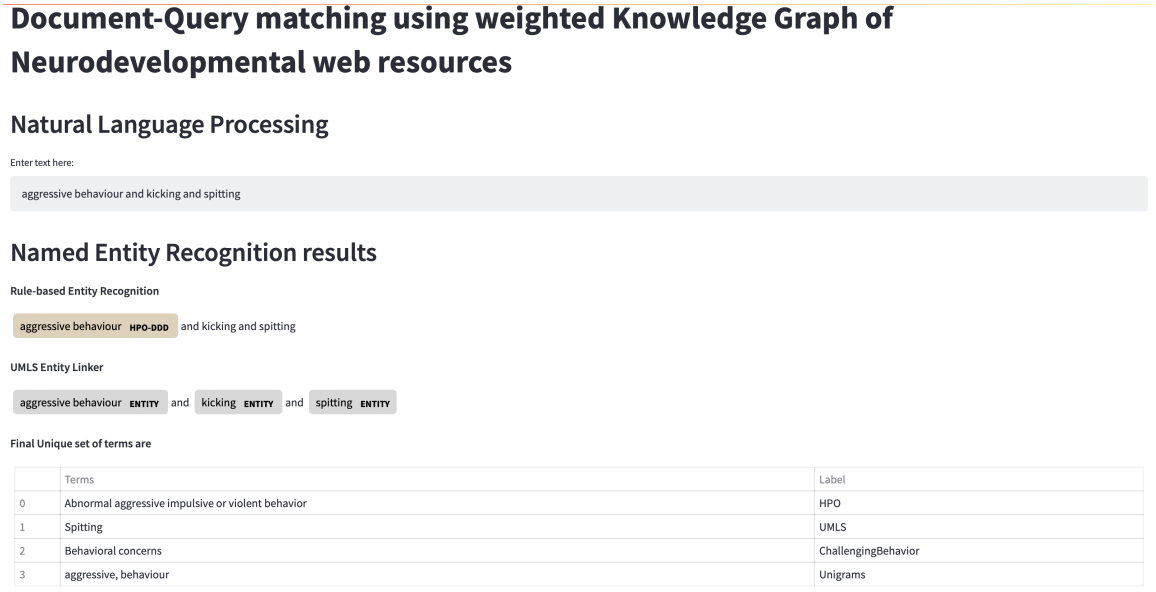


Figure 3.6: Question interface: obtained entities and unigrams for the query: "aggressive behaviour and kicking and spitting"

Top 10 resources based upon OWA aggregation

	title	type	category
0	Disability And Safety: Aggressive Behavior And Violence | Cdc	webpage	core knowledge
1	How To Help With Your Autistic Child'S Behaviour - Nhs	webpage	core knowledge
2	(Pdf) Interventions For Challenging Behaviour In Intellectual Disability	webpage	core knowledge
3	Aboutkidshealth	webpage	core knowledge
4	https://www.youtube.com/watch?v=_V0NLsvauCE	video	core knowledge
5	Sleep Patterns Predictive Of Daytime Challenging Behavior In Individuals With Low-Functioning Autism	webpage	core knowledge
6	Kids Health Information : Challenging Behaviour – Toddlers And Young Children	webpage	core knowledge
7	Kids Health Information : Challenging Behaviour – School-Aged Children	webpage	core knowledge
8	https://www.youtube.com/watch?v=KmrokQdsjTA	video	core knowledge
9	What Is Challenging Behaviour? - Medication Pathway	webpage	core knowledge

Figure 3.7: Question interface: list of top 10 most relevant resources for the query "aggressive behaviour and kicking and spitting"

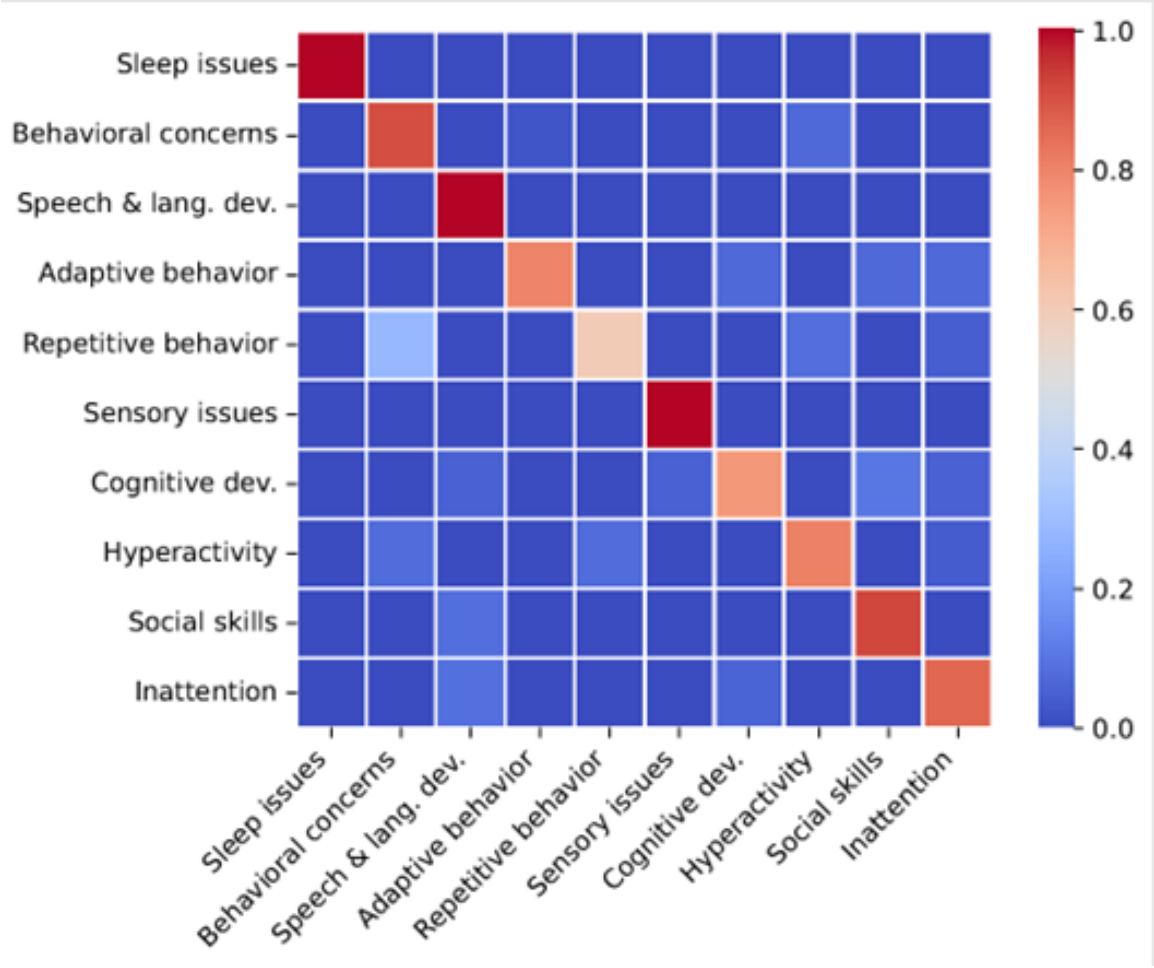


Figure 3.8: Normalized confusion matrix: rows show true labels, columns show predicted labels

Chapter 4

Reinforcement Learning for Topic Models

4.1 Abstract

We apply reinforcement learning techniques to topic modeling by replacing the variational autoencoder in ProdLDA with a continuous action space reinforcement learning policy. We train the system with a policy gradient algorithm REINFORCE. Additionally, we introduced several modifications: modernize the neural network architecture, weight the ELBO loss, use contextual embeddings, and monitor the learning process via computing topic diversity and coherence for each training step. Experiments are performed on 11 data sets. Our unsupervised model outperforms all other unsupervised models and performs on par with or better than most models using supervised labeling. Our model is outperformed on certain data sets by a model using supervised labeling and contrastive learning. We have also conducted an ablation study to provide empirical evidence of performance improvements from changes we made to ProdLDA and found that the reinforcement learning formulation boosts performance.

4.2 Introduction

The internet contains large collections of unlabeled textual data. Topic modeling is a method to extract information from this text by grouping documents into topics

and linking these topics with words describing them. Classical techniques for topic modeling, the most popular being Latent Dirichlet Approximation (LDA) [68], have recently begun to be overtaken by Neural Topic Models (NTM) [69].

ProdLDA [135] is a NTM using a product of experts in place of the mixture model used in classical LDA. ProdLDA uses a variational autoencoder (VAE) [136] to learn distributions over topics and words. ProdLDA improved on NVDM [137] by explicitly approximating the Dirichlet prior from LDA with a Gaussian distribution and using the Adam optimizer [138] with a higher momentum and learning rate.

Perceiving Reinforcement Learning (RL) as probabilistic inference has brought practices of such an inference into the RL field [139] [140]. New algorithms using these techniques include MPO [141] and VIREL [142]. MPO optimizes the evidence lower bound (ELBO), which is the same optimization objective used in VAEs.

Inspired by the adoption of probabilistic inference techniques in RL, we look to apply RL techniques to probabilistic inference in the realm of topic models. We use REINFORCE, the simplest policy gradient (PG) algorithm, to train a model which parameterizes a continuous action space, corresponding to the distribution of topics for each document in the topic model. We keep the product of experts from ProdLDA to compute the distribution of words for each document in the topic model.

We additionally improve our topic model by using Sentence-BERT (SBERT) embeddings [114] rather than bag-of-word (BoW) embeddings, modernizing the neural network (NN) architecture, adding a weighting term to the ELBO, and tracking topic diversity and coherence metrics throughout training. The model architecture is shown in Figure 4.1. Our method outperforms most other topic models. It is beaten only on some data sets by advanced methods using document labels for supervised learning, while our procedure is fully unsupervised.

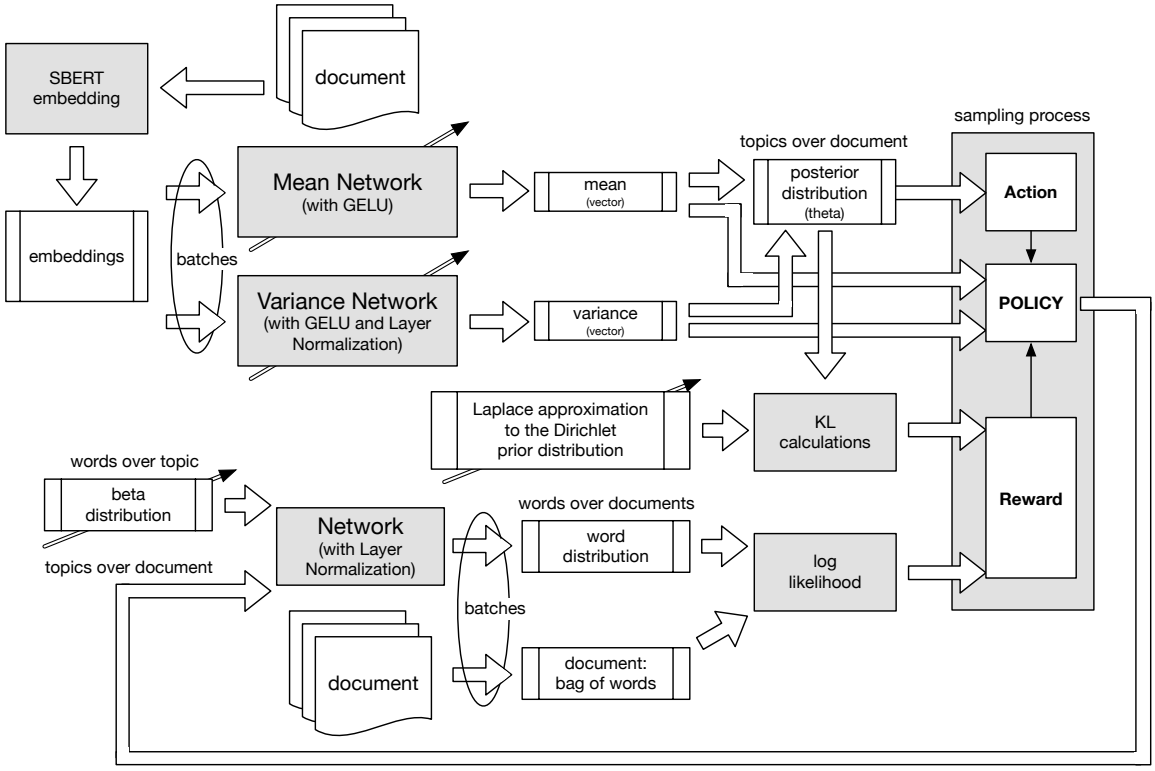


Figure 4.1: Architecture Diagram: gray boxes - processing; white boxes - models/data/information; arrows across boxes - tune-ability

4.3 Related Work

Zhao *et al.* [69] provide a survey of NTMs. Variations of VAEs are presented which use different distributions, correlated and structured topics, pre-trained language models, incorporate meta-data, or model on short texts rather than documents. Methods other than VAEs are also used for NTMing, including autoregressive models, generative adversarial networks, and graph NNs.

Doan and Hoang [143] compare ProdLDA and NVDM, along with six other NTMs and three classical topic models, in terms of held-out document and word perplexity, downstream classification, and coherence. Scholar [144], an extension of ProdLDA taking document metadata and labels into account where possible, performed best in terms of coherence. NVDM and NVCTM [145], an extension of NVDM which additionally models the correlation between documents, performed best in terms of

perplexity and downstream classification. The other NTMs were GSM [146], NVLDA [135], NSMDM [147], and NSMTM [147]. The classical topic models were non-negative matrix factorization (NMF) [148], online LDA [149], and Gibbs sampling LDA [150].

BERTopic [151] and Top2Vec [152] use dimensionality reduction and clustering to group document embeddings from pre-trained language models into meaningful clusters. Contextualized Topic Models (CTM) [153] augments the BoW embeddings used in ProdLDA with SBERT [114] embeddings, resulting in an improved topic model.

Dieng *et al.* [78] develop the embedded topic model (ETM) by using word embeddings to augment a variational inference algorithm for topic modeling. Their method outperforms other topic models, especially on corpora with large vocabularies containing common and very rare words. Nguyen and Luu [154] augment Scholar [144] with contrastive learning [155] and outperform all topic models compared against.

Gui *et al.* [156] use RL to filter words from documents, with reward as a combination of the resulting topic model's coherence and diversity, or how few words overlap between topics. Kumar *et al.* [157] use REINFORCE [158], a PG RL algorithm, to augment ProdLDA. Their model slightly outperforms ProdLDA in terms of topic coherence.

4.4 Background

We briefly outline topic models, RL process, KL divergence, and contextual embeddings.

4.4.1 Topic Models – Approaches

Latent Dirichlet Allocation (LDA) [68] is a three-level hierarchical Bayesian model: documents → topics → words. Each document is a mixture over latent topics, where the topic distribution θ is randomly sampled from a Dirichlet distribution. Each

topic is a multinomial distribution over vocabulary words.

Autoencoding Variational Inference for Topic Models (AVITM) [135] is a neural topic model using a VAE to learn a Gaussian distribution over topics. VAEs use a reparameterization trick (RT) to randomly sample from the posterior distribution to remain fully differentiable. At the time, there was no known RT for Dirichlet distributions, so AVITM used a Gaussian distribution and a Laplace approximation of the Dirichlet prior.

AVITM contains two models: NVLDA and ProdLDA. NVLDA uses the mixture model from LDA to infer a distribution over vocabulary words, while ProdLDA uses a product of experts.

Evidence Lower Bound (ELBO) is the optimization objective for AVITM. ELBO optimization [159] simultaneously tries to maximize the log-likelihood of the topic model and minimize the forward Kullback–Leibler (KL) divergence [160] between the posterior P and prior Q topic distributions.

$$\text{ELBO} = D_{KL}(P||Q) - \textit{log-likelihood} \tag{4.1}$$

4.4.2 Topic Models – Evaluation

Topic Coherence is a metric for evaluating topic models. It uses co-occurence in a reference corpus to measure semantic similarity between the top-K words in a topic. Topic model coherence is the average of each topic's coherence.

Normalized pointwise mutual information (NPMI) [76] was the coherence measure found to correlate best with human judgment [77]. When computing NPMI, a window size of 20 for co-occurrence counts is used in Srivastava and Sutton [135], while Dieng *et al.* [78] uses full document co-occurrence.

NPMI coherence is calculated for each of the top-K words in a topic and averaged to obtain the coherence for that topic. The overall *topic-coherence* is the average

of the coherence for each topic. For a word i, the NPMI coherence is calculated according to Equation 4.2.

$$\text{NPMI}(w_i) = \sum_{j}^{K-1} \frac{\log \frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{-\log P(w_i, w_j)} \tag{4.2}$$

where $P(w_i)$ is the probability of word i occurring in a document in the corpus, and $P(w_i, w_j)$ is the probability of words i and j co-occurring in a document in the corpus.

Topic Diversity is another metric for evaluating topic models. It measures the uniqueness of the top-K words across all topics. Dieng *et al.* [78] use $K = 25$ for reporting topic diversity.

$$\textit{topic-diversity} = \frac{\textit{number-of-unique-words}}{K * \textit{number-of-topics}} \tag{4.3}$$

Topic Quality is a topic modeling metric introduced by Dieng *et al.* [78].

$$\textit{topic-quality} = \textit{topic-coherence} * \textit{topic-diversity} \tag{4.4}$$

4.4.3 Reinforcement Learning

RL is a sequential decision-making framework focused on finding the best sequence of actions executed by an agent [79]. An agent takes actions $a \in \mathbf{A}$ to traverse between states $s \in \mathbf{S}$ in an environment, receiving a reward r on each transition. The goal of an RL task is to find the best set of actions —referred to as the policy —which maximizes the reward. RL problems can be episodic, where the agent completes the environment and is reset, or continuing, where the agent continuously traverses the environment without reset. Through traversing the environment, the agent learns a policy π of which actions in each state will maximize return. Return is the cumulative reward received by the agent in an episode or its lifetime. It is usually discounted by a factor γ to favor near-term reward over long-term reward. An alternative to discounting is the average reward formulation.

Policy Gradient (PG) Algorithms Many RL algorithms learn a value function – representing values associated with selecting specific actions – and a corresponding policy that chooses the action or subsequent state with maximum value. PG algorithms [161] provide an alternative approach directly learning a parameterized policy. The parameters of the policy function are optimized through stochastic gradient ascent.

REINFORCE is a Monte Carlo PG algorithm for episodic problems [158]. See algorithm 3, where $\boldsymbol{\rho}$ is a vector of optimized parameters.

Algorithm 3: REINFORCE

Input: A differentiable parameterized policy function $\pi(a|s, \boldsymbol{\rho})$
Algorithm Parameters:
step size $\alpha > 0$,
discount factor $\gamma < 1$

Initialize $\boldsymbol{\rho}$ (e.g. $\boldsymbol{\rho} \sim N(0, 0.02)$)
for *each episode* **do**
  Generate an episode
  s_0, a_0, r_1, ..., s_{T-1}, a_{T-1}, r_T
  following policy π
  for *each step in the episode (t from* 0 *to* $T-1$*)* **do**
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k$
    $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} + \alpha \gamma^t G \nabla \ln \pi(a_t|s_t, \boldsymbol{\rho})$
  end
end

Continuous Action Spaces are one advantage of PG algorithms [79]. Parameterized policies allow action spaces that are parameterized by a probability distribution, such as a Gaussian. For Gaussian action spaces, the mean μ and standard deviation σ are given by function approximators parameterized by $\boldsymbol{\rho}$. For a state s, an action a is sampled from the distribution and the policy is updated according to Equation 4.5.

$$\pi(a|s, \boldsymbol{\rho}) \doteq \frac{1}{\sigma(s, \boldsymbol{\rho})\sqrt{2\pi}} \exp\left(-\frac{(a - \mu(s, \boldsymbol{\rho}))^2}{2\sigma(s, \boldsymbol{\rho}^2)}\right) \tag{4.5}$$

Kullback-Leibler (KL) Divergence [160] measures the similarity between two probability distributions P and Q. It is used in AVITM [135] to force the posterior distribution parameterized by the VAE to be the Laplace approximation of the Dirichlet prior. The KL divergence calculation for N topics is shown in Equation 6.

$$D_{KL}(P||Q) = \frac{1}{2}\sum_{1}^{N}\left(\frac{(\mu_P - \mu_Q)^2}{\sigma_Q^2} + \frac{\sigma_P^2}{\sigma_Q^2} - \log\frac{\sigma_P^2}{\sigma_Q^2} - 1\right) \tag{4.6}$$

KL divergence has recently become popular in continuous action space RL algorithms. One application is to prevent policy updates from making large changes to the policy that could result in poorer performance. Two algorithms using KL divergence for this are TRPO [162] and MPO [141]. Another application is for optimistic RL [163] [164]. Vieillard *et al.* [165] investigate the usage of KL divergence as regularization in RL. KL divergence has also been used in optimal control [166], which is closely related to RL.

4.4.4 Contextual Embeddings

Contextual embeddings dominate NLP tasks, replacing earlier methods, including Word2Vec [30], GloVe [167], and BoW. Words and sequences of words are encoded into vector embeddings by large Transformer models [24].

The BoW document representation used in ProdLDA is augmented with contextual embeddings from SBERT Bianchi *et al.* [153]. They test three models: one with BoW, one with contextual embeddings, and one with both. They find that using both embeddings produces the best results, and the other two methods perform almost as well. One advantage of using solely contextual embeddings is that multilingual language models can encode documents from different languages into the same embedding space, enabling easy creation of multilingual topic models [168].

Sentence-BERT is an extension of BERT using a Siamese network to extract semantically meaningful sentence embeddings [114]. In contrast to BERT, this allows

SBERT embeddings to be compared using dot product or cosine similarity, making SBERT more suitable for tasks such as semantic similarity search and clustering.

4.5 Methodology

4.5.1 Modernizing ProdLDA

Following Liu *et al.* [169], we contemporize the architecture of the inference network within ProdLDA. We replace the SoftPlus activation function [170] with a GELU activation function [171], replace batch normalization [172] with layer normalization [173], and replace all Xavier initialization [174] with $\boldsymbol{\rho} \sim N(0, 0.02)$.

For the inference network, we increase the number of units in each layer from 100 to 128, add weight decay of 0.01 to each layer, and place dropout layers [115] after each fully connected layer.

We replace the softmax activation after the topic distribution with an RL policy formulation (Equation 4.5). We use a training batch size of 1024. We clip all gradients to a maximum norm of 1.0 to prevent gradient explosion [175]. Following Bianchi *et al.* [153], we set both distributional priors as trainable parameters. We lower the learning rate from 2×10^{-3} to 3×10^{-4} and momentum from 0.99 to 0.9.

4.5.2 Document Embeddings

Following Bianchi *et al.* [153], we replace the BoW used by ProdLDA with contextualized embeddings from SBERT. We use the "all-MiniLM-L6-v2" model for encoding unpreprocessed documents as embedding vectors. BoW embeddings, used to calculate the log-likelihood of the topic model, are created using preprocessed documents.

4.5.3 Single-step REINFORCE with a Continuous Action Space

We adopt the view of RL as a statistical inference method [140]. The modernized inference network from ProdLDA is used to parameterize a continuous action space

from which an action is sampled, and the policy is computed according to Equation 4.5. The topic model distribution over vocabulary words uses the product of experts from ProdLDA. We use REINFORCE to train the network, with a weighted version of ELBO as the reward. Each document embedding is a state in the environment, and each episode terminates after a single step (i.e., action). Each action is a sample from the topic distribution.

4.5.4 Weighted Evidence Lower Bound

Following Higgins *et al.* [176], we allow modifiable relative entropy between the prior and posterior by weighting the KL divergence term in the ELBO. We define a hyperparameter λ as a multiplier on the KL divergence term.

$$\text{ELBO}_{weighted} = \lambda D_{KL}(P||Q) - log\text{-}likelihood \tag{4.7}$$

4.5.5 Evaluation Metrics

We track topic diversity, coherence, perplexity, and loss for the training and test sets if applicable. Topic diversity and coherence are calculated based on the top-K words in each topic, with K noted for each experiment. We use NPMI coherence with co-occurence based on full document windows.

Most previous NTMs have only reported the coherence of the final model, presumably because coherence is not tracked during training for computational reasons. To enable tracking of coherence during training, we modify a vectorized implementation of UMass coherence[1] to calculate NPMI coherence and add caching for further speed-up. We also implement a GPU-optimized algorithm to calculate topic diversity during training.

Tracking these metrics during training provides two main benefits. The first benefit is that if training is going poorly, it can be terminated. Poor training could be

[1] https://github.com/maifeng/Examples_UMass-Coherence

caused by component collapse (low topic diversity), or if the model is unable to fit to coherent topics (low coherence). The second benefit is enabling deeper performance comparisons between models and between training runs for a single model. Most existing NTMs only track loss and perplexity during training, so additionally tracking topic diversity and coherence could provide additional insights on model performance.

4.5.6 Model Parameter Count

The number of parameters (P) in the model differs based on the total number of parameters across all inference layers (L), the number of topics (N), and the vocabulary size (V). Trainable parameters are the inference layers, the prior distribution of topics (N x 1), and the distribution of words over topics (V * N). Total parameters can be calculated with Equation 4.8.

$$P = L + N + V * N \tag{4.8}$$

The largest model we use is for the Wikitext-103 data set with 200 topics. This model has 4,001,224 parameters.

4.6 Results

4.6.1 Initial Experiments

We initially evaluate our topic model on the 20 Newsgroups data set with 20 topics. Results averaged over 30 random seeds are shown: loss in Figure 4.2, topic coherence in Figure 4.3, and topic diversity in Figure 4.4. Mean and 90% confidence intervals are plotted. Topic diversity and coherence are calculated with $K = 10$. Documents are preprocessed following Bianchi *et al.* [153] with the additional step of removing all words with less than three letters. Models are trained for 1000 epochs with the AdamW optimizer ($\alpha = 3e - 4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$). We use $\lambda = 5$, inference network dropout of 0.2, and no dropout after the RL policy (policy dropout). All

other experiments use these same settings unless otherwise noted.

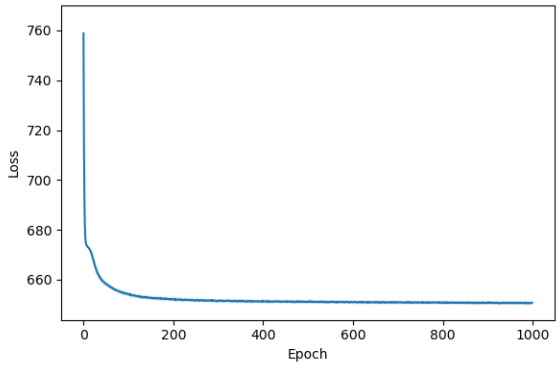


Figure 4.2: Loss (30 seeds): 20 Newsgroups

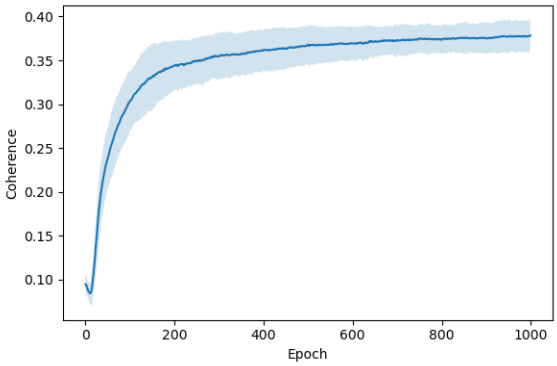


Figure 4.3: Topic Coherence (30 seeds): 20 Newsgroups

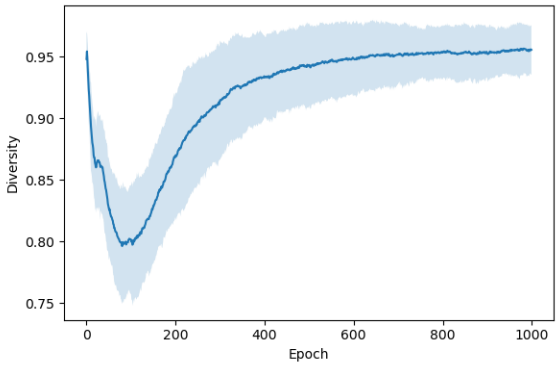


Figure 4.4: Topic Diversity (30 seeds): 20 Newsgroups

Topic Words from Initial Experiments We choose one example of the top 10 words for all 20 topics from the initial experiments on the 20 Newsgroups data set. We choose the seed with the 15th highest coherence (out of 30 seeds). Topic words are shown in Table 4.1. Each document in the Twenty Newsgroups data set is labeled as belonging to one of 20 categories. These 20 categories are shown in Table 4.2.

4.6.2 Comparison to Other Topic Models

We compare our method to recent topic models found in the literature.

Topic Words
max giz bhj chz pts buf air det pit bos
morality objective cramer moral livesey optilink keith homosexual clayton gay
window xterm widget lib windows font usr mouse motif application
gun guns militia firearms weapons cops weapon amendment semi arms
team players hockey game season nhl games play teams leafs
max giz bhj sale chz shipping offer monitor copies condition
jesus god bible christ christians faith church christian heaven lord
geb banks msg patients gordon pitt disease pain doctor medical
fbi batf koresh compound atf waco sandvik udel fire kent
car insurance cars dealer oil saturn honda engine bmw miles
jpeg image bits display gif file program files format color
clipper encryption key chip escrow keys privacy crypto secure nsa
wire ground circuit connected cable atheism electrical universe keyboard output
israel israeli arab jews arabs peace palestinian attacks bony villages
turkish armenian armenians armenia turks serdar argic turkey genocide soviet
pub ftp anonymous tar graphics privacy mailing archive motif faq
moon space lunar orbit nasa spacecraft henry launch shuttle solar
dog bike dod riding ride motorcycle rider bmw went cops
scsi ide drive controller drives bus disk floppy bios isa
stephanopoulos president jobs myers russia russian administration package launch clinton

Table 4.1: Initial Experiment Topic Words

Category
alt.atheism
comp.graphics
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware
comp.windows.x
misc.forsale
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey
sci.crypt
sci.electronics
sci.med
sci.space
soc.religion.christian
talk.politics.guns
talk.politics.mideast
talk.politics.misc
talk.religion.misc

Table 4.2: 20 Newsgroups Categories

Benchmarking Neural Topic Models (BNTM)

In the beginning, our approach is compared with all models evaluated by Doan and Hoang [143]. We use their preprocessed documents and replicate their results using $K = 10$ to calculate topic coherence. Following the authors, we sweep from 0.5*N topics to 3*N topics in intervals of 0.5*N (N being the "correct" number of topics for each data set). Next, we do a hyperparameter sweep over λ of 1, 3, 5, and 10. Results are averaged over ten random seeds and shown in Figure 4.5.

Topic Modeling in Embedding Spaces

Next, the comparison is done with Dieng *et al.* [78] on the New York Times data set with 300 topics and without using stop words. Results are shown in Table 4.3.

We increase batch size to 32768 and only train for 20 epochs on one random seed. Additionally, we increase the number of units in each layer of the inference network to 512, increase dropout in the inference network to 0.5, and decrease λ to 1. Topic diversity is calculated using $K = 25$.

Model	Coherence	Diversity	Quality
ETM	0.18	0.22	0.0405
RL model (ours)	**0.24**	**0.32**	**0.0778**

Table 4.3: Comparison on no stop words data

Pre-training is a Hot Topic (PTHT)

We also compare our model, using all metrics, with the best model as evaluated by Bianchi *et al.* [153]. Results are shown in Table 4.4. Metrics are averaged over 25, 50, 75, 100, and 150 topics: 30 seeds for each number of topics. We use the same preprocessing as the authors. We use $\lambda = 1$.

Data Set	Paper	NPMI	Word2Vec	Inverse RBO
Wiki20K	PTHT best	0.1823	0.2110	**0.9950**
	RL model (ours)	**0.2509**	**0.2368**	0.9799
StackOverflow	PTHT best	0.0280	0.1598	**0.9914**
	RL model (ours)	**0.1249**	**0.1617**	0.9860
Google News	PTHT best	0.1207	0.1325	**0.9965**
	RL model (ours)	**0.3563**	**0.1485**	0.9934
Tweets2011	PTHT best	0.1008	**0.1493**	0.9956
	RL model (ours)	**0.3559**	0.1417	**0.9962**
20 Newsgroups	PTHT best	0.1300	**0.2539**	0.9931
	RL model (ours)	**0.2696**	0.1798	**0.9932**

Table 4.4: Average metrics from best PTHT model (per metric) and our RL model

Data Set	Paper	NPMI Coherence (N = number of topics)				
		N = 25	N = 50	N = 75	N = 100	N = 150
Wiki20K	PTHT	0.17	0.19	0.18	0.19	0.17
	RL model (ours)	**0.33**	**0.30**	**0.25**	**0.22**	**0.19**
StackOverflow	PTHT	0.05	0.03	0.02	0.02	0.02
	RL model (ours)	**0.17**	**0.14**	**0.12**	**0.11**	**0.10**
Google News	PTHT	0.03	0.10	0.15	0.18	0.19
	RL model (ours)	**0.38**	**0.41**	**0.38**	**0.34**	**0.30**
Tweets2011	PTHT	0.05	0.10	0.11	0.12	0.12
	RL model (ours)	**0.36**	**0.39**	**0.38**	**0.35**	**0.31**
20 Newsgroups	PTHT	0.13	0.13	0.13	0.13	0.12
	RL model (ours)	**0.35**	**0.30**	**0.27**	**0.25**	**0.22**

Table 4.5: NPMI coherence comparison between PTHT model and RL model for each number of topics

Contrastive Learning for NTM (CLNTM)

We compare results with the contrastive Scholar model from Nguyen and Luu [154]. For each data set we perform a hyperparameter search with 50 topics. Search ranges and best results for each data set are shown in Table 4.6. We use the best hyperparameters from this search for final training runs with 50 and 200 topics. We train for 2000 epochs. Results are averaged over 30 random seeds and shown in Table 4.7.

To show the tradeoff between topic diversity and coherence, we perform a sweep over policy dropout from 0 to 0.9 at intervals of 0.1 using the 20 Newsgroups data set with 50 topics. Other hyperparameters are kept the same. We train for 2000 epochs. Results are averaged over 30 random seeds and shown in Figure 4.6.

4.6.3 Ablation Study

To provide empirical evidence that performance improvements come from the RL policy formulation, we do a study ablating relevant changes from the final RL model

Experiment	Layer Size	Inference Dropout	Policy Dropout	λ
Hyperparameter Search	{128, 512}	{0.2, 0.5}	{0.0, 0.25, 0.5}	{1, 5}
20 Newsgroups	128	0.5	0.5	1
IMDb Movie Reviews	512	0.5	0.25	1
Wikitext-103	512	0.5	0.25	5

Table 4.6: Hyperparameter search and best results per data set for RL model

Model	20 Newsgroups		IMDb Movie Reviews		Wikitext-103	
	50 Topics	200 Topics	50 Topics	200 Topics	50 Topics	200 Topics
Contrastive Scholar	0.334	0.280	0.197	**0.188**	**0.497**	**0.478**
RL model (ours)	**0.449**	**0.308**	**0.199**	0.139	0.432	0.268

Table 4.7: Comparison to CLNTM

down to the original ProdLDA model. All comparisons are performed on the 20 Newsgroups data set with 20 topics and use the same settings as subsection 4.6.1. Results are averaged over 30 random seeds and shown in Table 4.8.

RL Policy	Embedding	λ	θ Softmax	θ / Policy Dropout	Coherence	Diversity
✓	SBERT	5	×	0.0	**0.3848**	**0.9530**
×	SBERT	5	×	0.0	0.2795	0.453
✓	BoW	5	×	0.0	0.3379	0.9403
✓	SBERT	1	×	0.0	0.3414	0.9070
✓	SBERT	5	✓	0.0	0.1932	0.6927
✓	SBERT	5	×	0.2	0.3769	0.7315
×	BoW	1	✓	0.2	0.2650	0.7390

Table 4.8: Highlighted results from ablation study

4.7 Data Sets

We evaluate models on the test set where available, and on the training set if there is no test set. Coherence and diversity for the training and test set are the same, as they are evaluated on the word distribution over topics which doesn't change per

document. In the code, training coherence and diversity are computed after each batch, while test coherence and diversity are computed after each epoch. Number of training/test documents and vocabulary sizes are shown in Table 4.9. Average original and preprocessed training document lengths are shown in Table 4.10.

Data Set	Comparison Paper	Training Docs	Test Docs	Vocab Size
20 Newsgroups	This one [153] [154]	11,314	7,532	2,000
	[143]	15,465	N/A	4,134
New York Times	[78]	1,864,470	N/A	10,283
Snippets	[143]	12,295	N/A	4,666
W2E-title	[143]	105,457	N/A	3,703
W2E-content	[143]	83,548	N/A	10,508
Wiki20K	[153]	20,000	N/A	2,000
StackOverflow	[153]	16,407	N/A	2,236
Google News	[153]	11,108	N/A	8,099
Tweets2011	[153]	2,472	N/A	5,097
IMDb Movie Reviews	[154]	25,000	25,000	5,000
Wikitext-103	[154]	28,472	60	20,000

Table 4.9: Data Sets - Documents and Vocabularies

4.7.1 20 Newsgroups

The 20 Newsgroups data set [177] consists of around 19,000 newsgroup posts from 20 topics. We perform experiments on this data set with three different preprocessing methods. For our initial experiments, we follow the preprocessing in Bianchi *et al.* [153] and additionally remove all words with less than 3 letters. For the comparisons with Bianchi *et al.* [153] and Nguyen and Luu [154], we follow the preprocessing in Bianchi *et al.* [153]. For the comparison with Doan and Hoang [143], we use their already preprocessed data set.

Data Set	Comparison Paper	Average Training Document Length	
		Original	Preprocessed
20 Newsgroups	This one	287.5	95.9
	[153] [154]	287.5	107.6
	[143]	N/A	73.5
New York Times	[78]	558.1	484.5
Snippets	[143]	N/A	14.4
W2E-title	[143]	N/A	6.8
W2E-content	[143]	N/A	209.1
Wiki20K	[153]	49.8	17.5
StackOverflow	[153]	N/A	4.9
Google News	[153]	N/A	6.2
Tweets2011	[153]	N/A	8.6
IMDb Movie Reviews	[154]	233.8	101.7
Wikitext-103	[154]	295.8	133.2

Table 4.10: Data Sets - Training Document Lengths

4.7.2 New York Times

The New York Times data set [178] consists of over 1.8 million articles written by the New York Times between 1987 and 2007. We follow the preprocessing from Bianchi *et al.* [153], but do not remove stopwords.

4.7.3 Snippets

The Web Snippets data set [179] consists of around 12,000 snippets of text from websites linked on "yahoo.com". The snippets are grouped into 8 domains. We use the already preprocessed data set from Doan and Hoang [143].

4.7.4 W2E

The W2E data set [180] consists of news articles from media channels around the world. The W2E-title subset is the titles from the news articles, while the W2E-content subset is the text content of the articles. The articles are grouped into 30 topics. We use the already preprocessed data set from Doan and Hoang [143].

4.7.5 Wiki20K

The Wiki20K data set [168] consists of 20,000 English Wikipedia abstracts randomly sampled from DBpedia. We follow the preprocessing from Bianchi *et al.* [153].

4.7.6 StackOverflow

The StackOverflow data set [181] consists of around 16,000 question titles randomly sampled from 20 different tags in a larger data set crawled from the website "stackoverflow.com" between July and August 2012. We use the already preprocessed data set from Qiang *et al.* [181].

4.7.7 Google News

The Google News data set [181] consists of around 11,000 titles and short samples from Google News articles clustered into 152 groups. We use the already preprocessed data set from Qiang *et al.* [181].

4.7.8 Tweets2011

The Tweets2011 data set [181] consists of around 2,500 tweets in 89 clusters sampled from the larger Tweets2011 corpus [182] crawled from Twitter between January and February 2011. We use the already preprocessed data set from Qiang *et al.* [181].

4.7.9 IMDb Movie Reviews

The IMDb Movie Reviews data set [183] consists of 50,000 movie reviews, each with an associated sentiment label, from the website "imdb.com". We follow the preprocessing

from Bianchi *et al.* [153].

4.7.10 Wikitext-103

The Wikitext-103 data set [184] consists of around 28,500 Wikipedia articles classified as either Featured articles or Good articles by Wikipedia editors. We follow the preprocessing from Bianchi *et al.* [153].

4.8 Discussion

For the initial experiments on the 20 Newsgroups data set, the average loss (Figure 4.2) reaches a near plateau around the 200th epoch. Past this epoch, coherence (Figure 4.3) continues to increase slowly, and topic diversity (Figure 4.4) increases substantially until around the 400th epoch, past which it also continues to increase slowly. It shows that training beyond a plateau in loss can still improve NTM performance.

Compared to Doan and Hoang [143], the RL model performs on par with or better than other models across all four data sets, while the performance of other models varies greatly between data sets. On the Snippets, 20 Newsgroups, and W2E-content data sets, the RL model with lower values of λ usually performs better as the number of topics increases. However, it reverses on the W2E-title data set where $\lambda = 10$ outperforms $\lambda = 1$ on the two highest number of topics.

The RL model outperforms the Labeled ETM model from Dieng *et al.* [78] in topic diversity, coherence, and quality. Furthermore, this comparison had no pruning of stop words, showing the RL model can deal with vocabularies containing many common words.

Compared to Bianchi *et al.* [153], the RL model significantly outperforms all other models on all data sets evaluated in terms of NPMI coherence. Furthermore, the RL model performs similarly to the best of the other models in terms of inverse RBO. We state the topic diversity used by Dieng *et al.* [78] is a more useful metric than

inverse RBO, as it usually has a higher variance in values and is more intuitive to understand. For Word2Vec coherence, the RL model performs on par with the best of the other models, except when compared to ETM [78] on the 20 Newsgroups data set.

If we consider models from Nguyen and Luu [154], our RL model performs similarly on 50 topics but worse on 200 topics. The RL model's performance on larger topic sizes and vocabularies could be improved by adding supervised labels, applying contrastive learning, scaling up inference layer sizes, or performing a hyperparameter sweep with 200 topics.

Topic diversity and coherence values should be provided when reporting topic model performance. In Figure 4.6, the highest topic quality is achieved when there is no policy dropout. Topic diversity can be sacrificed for some gain in coherence. Applications of topic models may want to maximize topic diversity, coherence, or both. The description of topic model performance should reflect this.

In the ablation study, removing the RL policy formulation causes the model to perform worse than the original one. It confirms RL policy augments the improvements from other changes to the model. Performance suffers the most when the softmax distribution is re-added to the topic distribution during training. To recapture the softmax distribution of topics, it can be applied to the topic distribution during inference. Adding policy dropout significantly reduces topic diversity and leads to a slight coherence reduction. Performance improves with SBERT embeddings, and the model can still reconstruct the BoW within the ELBO without direct access. Increasing λ to 5 improves performance, but as seen from other experiments, this is only sometimes the case.

4.9 Conclusion

Inspired by the introduction of probabilistic inference techniques to RL, we take the approach to develop a NTM augmented with RL. Our model builds on the ProdLDA

model, which uses a product of experts instead of the mixture model used in classical LDA. We improve ProdLDA by adding SBERT embeddings, an RL policy formulation, a weighted ELBO loss, and the improved NN architecture. In addition, we track topic diversity and coherence during a training process rather than only evaluating these metrics for the final model. Our fully unsupervised RL model outperforms most other topic models. It is only topped by contrastive Scholar —a method using supervised labels during training —in a few select cases.

We have identified some possible paths for future work. The SBERT embeddings could be fine-tuned during training rather than calculating them during pre-processing and freezing them during training. The RL formulation of our model could be extended to dynamic topic models [72]. More complex PG RL algorithms could be used rather than REINFORCE, or a baseline could be added to REINFORCE. Exploration techniques from RL could be applied. The influence of hyperparameters (e.g. inference network layer sizes) on varied corpora (e.g. those with large vocabularies) could be explored. The Laplace approximation of the Dirichlet prior could be replaced by a true Dirichlet prior, making use of the Dirichlet RT [185] and a Dirichlet RL policy [186]. Finally, λ and the policy dropout could be scheduled during training to provide an automated tradeoff between topic diversity and coherence.

4.10 Ethics and Limitations

4.10.1 Ethics

All data sets used in this paper are cited. The New York Times data set[2] is licensed under "The New York Times Annotated Corpus Agreement"[3]. The Tweets2011 corpus[4] is available under the "TREC 2011 Microblog Dataset Usage Agreement"[5] which

[2]https://catalog.ldc.upenn.edu/LDC2008T19
[3]https://catalog.ldc.upenn.edu/license/the-new-york-times-annotated-corpus-ldc2008t19.pdf
[4]https://trec.nist.gov/data/tweets/
[5]https://trec.nist.gov/data/tweets/tweets2011-agreement.pdf

additionally requires following the "Twitter terms of service"[6]. All other data sets are obtained from the recent literature. No sensitive information is used or inferred in this paper. The risk of harm from our model is low. Any artifacts in this paper are used following their intended use cases.

4.10.2 Limitations

The main limitation identified for our RL model is decreased performance as the vocabulary size increases. Our RL model also has a higher variance than some other topic models to which we compared. While our RL model performed well on all the data sets tested, this performance may not generalize to different data sets. The insights from the policy dropout sweep conducted may not apply to other topic models. The performance difference for NPMI coherence compared with Bianchi *et al.* [153] may be overstated since the model in that paper used a deprecated SBERT model that produces sentence embeddings of low quality[7]. For the comparison to Nguyen and Luu [154], we used slightly different preprocessing than the authors. While the model can work on any languages with associated embedding models, all data sets used in this paper were in English. Our model has additional hyperparameters compared to some other models. So, it may require more tuning and, therefore, more GPU computing. The initial model was developed on a system with 8GB of RAM and a Nvidia GTX 1060 with 3GB of VRAM for a total of approximately 100 GPU hours. A single run of the model for 1000 epochs on this GPU requires less than an hour. Experiments using the New York Times data set were run on a system with 256GB of RAM and a Nvidia RTX 3090 for approximately 100 GPU hours. All other experiments were run on a system with 128GB of RAM and a Nvidia TITAN RTX for approximately 600 GPU hours.

[6]https://twitter.com/en/tos
[7]https://huggingface.co/sentence-transformers/stsb-roberta-large

4.11 Reproducibility

4.11.1 Hyperparameters

We show the hyperparameters for each experiment we performed. Experiment seeds are generated with a meta-seed for reproducibility. The meta-seed is randomly chosen from integers between 0 and 2^{32}. Values in {curly brackets} indicate a search over multiple parameters. Values in [square brackets] indicate NN layer sizes (e.g. [128, 128] represents two layers of size 128).

Initial Experiments and Ablation Study

We use the same meta-seed for the ablation study as we did for the initial experiments. Hyperparameters for the initial experiments can be found in Table 4.11. Further tables for all experiments will only show hyperparameters that differ from this table. Hyperparameters for the ablation study can be found in Table 4.12.

Benchmarking Neural Topic Models

We show hyperparameters for the comparison with Doan and Hoang [143]. Hyperparameters for Snippets can be found in Table 4.13. 20 Newsgroups in Table 4.14. W2E-title in Table 4.15. W2E-content in Table 4.16.

Topic Modeling in Embedding Spaces

Hyperparameters for the comparison with Dieng *et al.* [78] can be found in Table 4.17.

Pre-training is a Hot Topic

We show hyperparameters for the comparison with Bianchi *et al.* [153]. Data set and seed information can be found in Table 4.18. All other hyperparameters are the same for each data set; these can be found in Table 4.19.

Hyperparameter	Value(s)
Meta-seed	4174224060
Num. Seeds	30
Num. Epochs	1000
Data Set	20 Newsgroups
Vocab Size	2000
Embedding	SBERT
Num. Topics (N)	20
Inference Dropout	0.2
Policy Dropout	0.0
Inference Layers	[128, 128]
Activation	GELU
Initialization	$\boldsymbol{\rho} \sim N(0, 0.02)$
Normalization	Layer
λ	5
Topic Words (K)	10
RL policy	✓
θ Softmax	×
Learning Rate (α)	3e-4
Adam β_1, β_2	0.9, 0.999
Weight Decay	0.01
Batch Size	1024
Gradient Clipping	1.0

Table 4.11: Initial Experiments

Contrastive Learning for NTM

We show hyperparameters for the comparison with Nguyen and Luu [154]. Some hyperparameters are already shown in Table 4.6 and won't be shown again here. Data set and seed information can be found in Table 4.20. Other hyperparameters

Hyperparameter	Value(s)
Meta-seed	4174224060
Num. Seeds	30
Data Set	20 Newsgroups
Embedding	{BoW, SBERT}
θ / Policy Dropout	{0.0, 0.2}
λ	{1, 5}
RL policy	{✓, ×}
θ Softmax	{✓, ×}

Table 4.12: Ablation Study

Hyperparameter	Value(s)
Meta-seed	193270011
Num. Seeds	10
Data Set	Snippets
Vocab Size	4666
Num. Topics (N)	{4, 8, 12, 16, 20, 24}
λ	{1, 3, 5, 10}

Table 4.13: BNTM Snippets

Hyperparameter	Value(s)
Meta-seed	1216545997
Num. Seeds	10
Data Set	20 Newsgroups
Vocab Size	4157
Num. Topics (N)	{10, 20, 30, 40, 50, 60}
λ	{1, 3, 5, 10}

Table 4.14: BNTM 20 Newsgroups

Hyperparameter	Value(s)
Meta-seed	4014169843
Num. Seeds	10
Data Set	W2E-title
Vocab Size	3703
Num. Topics (N)	{15, 30, 45, 60, 75, 90}
λ	{1, 3, 5, 10}

Table 4.15: BNTM W2E-title

Hyperparameter	Value(s)
Meta-seed	1359128464
Num. Seeds	10
Data Set	W2E-content
Vocab Size	10508
Num. Topics (N)	{15, 30, 45, 60, 75, 90}
λ	{1, 3, 5, 10}

Table 4.16: BNTM W2E-content

are the same for each data set; these can be found in Table 4.21. Hyperparameters for the policy dropout sweep can be found in Table 4.22.

4.11.2 Ablation Study

We show full results from the ablation study in Table 4.23.

Hyperparameter	Value(s)
Meta-seed	2337766308
Num. Seeds	1
Num. Epochs	20
Data Set	New York Times
Vocab Size	10283
Num. Topics (N)	300
Inference Dropout	0.5
Inference Layers	[512, 512]
λ	1
Topic Words (K)	10*
Batch Size	32768

Table 4.17: Topic Modeling in Embedding Spaces (*We use $K = 25$ to calculate topic diversity for the final model.)

Data Set	Vocab Size	Meta-seed	Num. Seeds
Wiki20K	2000	359491602	30
StackOverflow	2236	1459046441	30
Google News	8099	925040003	30
Tweets2011	5097	1321150024	30
20 Newsgroups	2000	3277797161	30

Table 4.18: PTHT Data Set Seeds

Hyperparameter	Value(s)
Num. Topics (N)	{25, 50, 75, 100, 150}
λ	1

Table 4.19: Pre-training is a Hot Topic

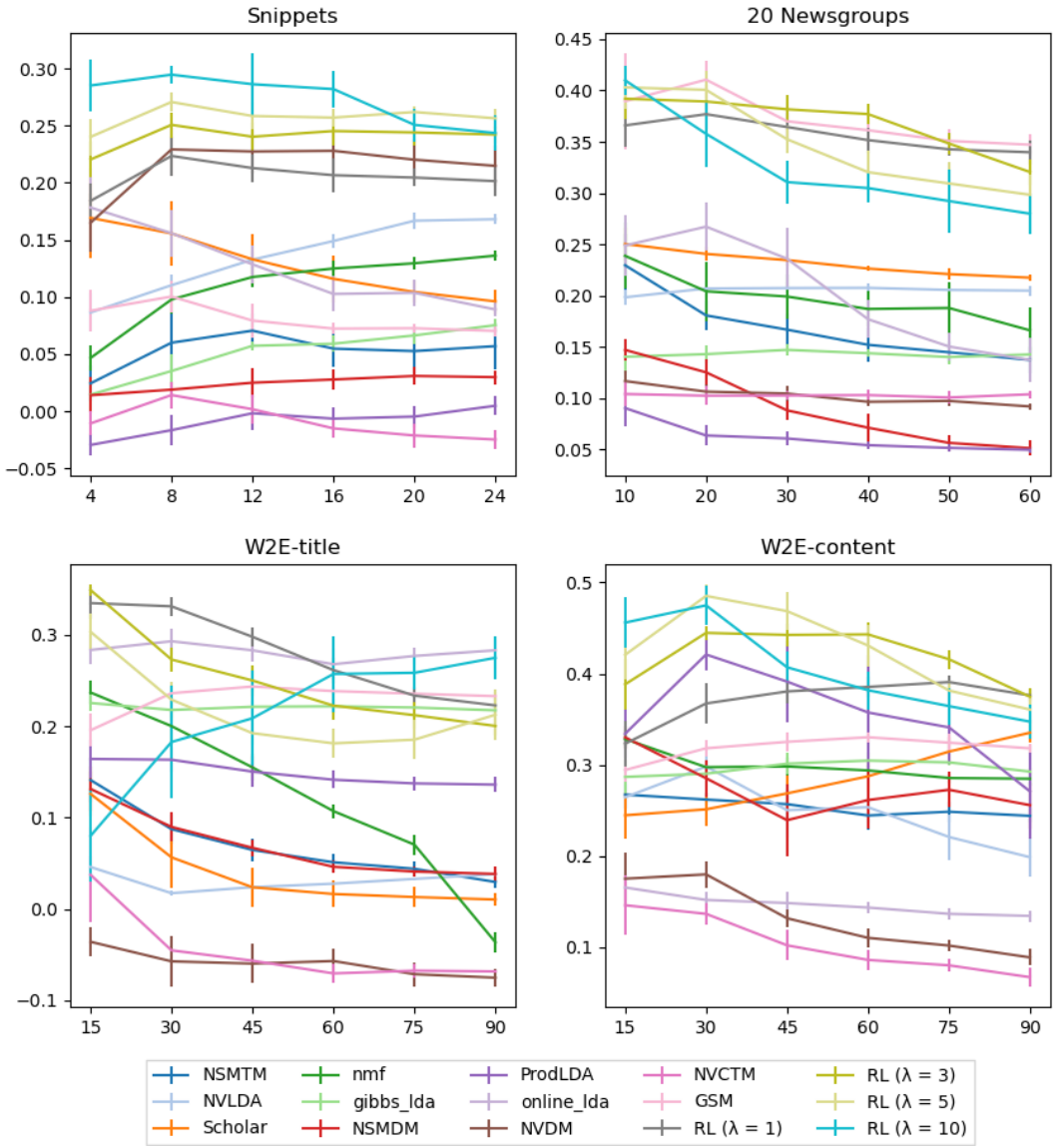


Figure 4.5: Comparison of RL model (ours) to BNTM models

Data Set	Vocab Size	Meta-seed	Num. Seeds
20 Newsgroups	2000	1553571489	30
IMDb Movie Reviews	5000	3747305026	30
Wikitext-103	20000	2672751736	30

Table 4.20: CLNTM Data Set Seeds

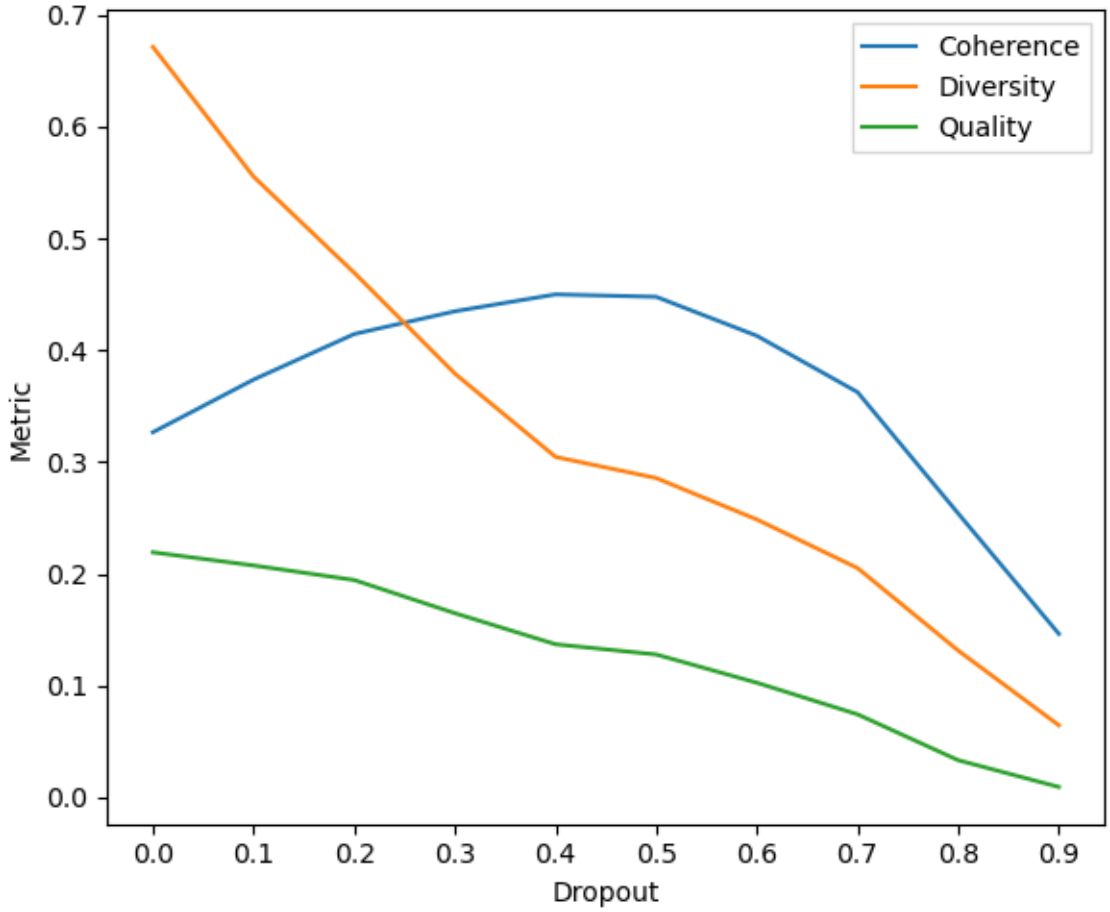


Figure 4.6: Dropout sweep for 20 Newsgroups

Hyperparameter	Value(s)
Num. Epochs	2000
Num. Topics (N)	{50, 200}

Table 4.21: Contrastive Learning for NTM

Hyperparameter	Value(s)
Meta-seed	3432645033
Num. Seeds	30
Data Set	20 Newsgroups
Num. Epochs	2000
Num. Topics (N)	50
Inference Dropout	0.5
Policy Dropout	{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
Inference Layers	[128, 128]
λ	1

Table 4.22: CLNTM Dropout Sweep

RL Policy	Embedding	λ	θ Softmax	θ / Policy Dropout	Coherence	Diversity
×	BoW	1	✓	0.0	0.2906	0.8457
×	BoW	1	×	0.0	0.2373	0.6943
✓	BoW	1	✓	0.0	0.2748	0.8905
✓	BoW	1	×	0.0	0.2738	0.8707
×	BoW	5	✓	0.0	0.2526	0.6598
×	BoW	5	×	0.0	0.2619	0.6928
✓	BoW	5	✓	0.0	0.2032	0.5965
✓	BoW	5	×	0.0	0.3379	0.9403
×	BoW	1	✓	0.2	0.2650	0.7390
×	BoW	1	×	0.2	0.2193	0.5195
✓	BoW	1	✓	0.2	0.2082	0.5692
✓	BoW	1	×	0.2	0.2798	0.7740
×	BoW	5	✓	0.2	0.2526	0.6222
×	BoW	5	×	0.2	0.2257	0.5768
✓	BoW	5	✓	0.2	0.1222	0.314
✓	BoW	5	×	0.2	0.3284	0.8092
×	SBERT	1	✓	0.0	0.2845	0.6207
×	SBERT	1	×	0.0	0.2948	0.5995
✓	SBERT	1	✓	0.0	0.2158	0.8080
✓	SBERT	1	×	0.0	0.3414	0.9070
×	SBERT	5	✓	0.0	0.2726	0.4458
×	SBERT	5	×	0.0	0.2795	0.4530
✓	SBERT	5	✓	0.0	0.1932	0.6927
✓	SBERT	5	×	0.0	**0.3848**	**0.9530**
×	SBERT	1	✓	0.2	0.2532	0.6063
×	SBERT	1	×	0.2	0.2554	0.5430
✓	SBERT	1	✓	0.2	0.1133	0.5520
✓	SBERT	1	×	0.2	0.3649	0.7663
×	SBERT	5	✓	0.2	0.2435	0.4478
×	SBERT	5	×	0.2	0.2080	0.3698
✓	SBERT	5	✓	0.2	0.0967	0.9227
✓	SBERT	5	×	0.2	0.3769	0.7315

Table 4.23: Full Results from Ablation Study

Chapter 5

Extracting Knowledge Graph Triples from FragileX Abstracts

5.1 Abstract

We collect two volunteer-labeled data sets from FragileX syndrome PubMed abstracts: one for named-entity recognition and one for relation extraction. We fine-tune a biomedical large language model on our FragileX relation extraction data set. We compare our model to a baseline model fine-tuned on the i2b2 2010 data set. Our model has lower precision than the baseline, but much higher recall. Our methodology is slightly outdated due to rapid advancements in language modeling, so we present some alternatives.

5.2 Introduction

Fragile X syndrome (FXS) is a disorder associated with intellectual disabilities such as autism spectrum disorder (ASD) and attention deficit-hyperactivity disorder (ADHD) [187–189]. Large amounts of academic literature exists on FXS, but extracting succinct and actionable information from this text corpus is difficult. Representing this information in a compact and readable format, such as a knowledge graph (KG), would allow trends across FXS literature to be readily accessed and understood.

KGs represent data in the form of triples: a subject entity, an object entity, and the relation between these entities [61]. Natural language processing (NLP) methods

have become increasingly popular in recent years for understanding large text corpora. Two NLP techniques in particular can be combined to discover triples in free text: named-entity recognition (NER) and relation extraction (RE) [65].

This paper outlines a methodology for collecting a corpus of FXS text, obtaining human labels of entities and relations within the text, and using this labeled corpus to train models for predicting entities and relations in unseen FXS text.

5.3 Background

Triples consist of two entities connected by a relation [61]. Extracting triples from free text requires two steps; relevant entities in the text must be identified, and each combination of entities must be labeled with a suitable relation, or as unrelated if no suitable relation exists. NER is a NLP method for identifying entities within free text. NER involves a model classifying words in a text as entities or non-entities based on an exact match or sufficient similarity to words from some database or training corpus [65].

RE is a NLP method for predicting relations between entities [65]. There are three popular RE data sets for medical relation extraction: drug-drug interactions (DDI) [190], chemical-protein interactions (CHEMPROT) [191], and the informatics for integrating biology and the bedside (i2b2) 2010 relations challenge data set [192, 193]. The i2b2 2010 data set, consisting of labeled patient discharge summaries, is the most similar to this paper's application and will be used as a starting point for labeling the new FXS data set. Since its creation, the i2b2 data set has been renamed as the national NLP clinical challenges (n2c2) data set but will still be referred to as the i2b2 data set within this paper [194]. The i2b2 2010 data set consists of nine labeled relations, as outlined in Table 5.1.

An accurate method for RE is fine-tuning a masked language model (MLM) on some RE data set. MLMs are transformer-based neural network models trained to predict "masked" tokens in sentences from large text corpora, and in the process

Table 5.1: i2b2 2010 Relations

Relation	Abbreviation
Medical problem indicates medical problem	PIP
Test conducted to investigate medical problem	TeCP
Test reveals medical problem	TeRP
Treatment is administered for medical problem	TrAP
Treatment causes medical problem	TrCP
Treatment improves medical problem	TrIP
Treatment is not administered because of medical problem	TrNAP
Treatment worsens medical problem	TrWP
Does not fit into one of the above defined relationships	false

learn an approximation of the syntactic and semantic structure of language. Within sentences from the corpora, certain tokens (i.e. full or partial words) are "masked" (i.e. hidden) from the model and the model must predict the correct token. This process is referred to as pre-training, and a pre-trained MLM can be fine-tuned to perform some specific task.

The original transformer-based MLM, bidirectional encoder representations from transformers (BERT), is a popular MLM pre-trained on large text corpora [21]. While BERT would learn some medical knowledge from sources within its pre-training corpora such as English Wikipedia and BooksCorpus, a larger corpus of medical literature would allow a similar model to perform better on medical tasks. To facilitate this, a few MLMs were pre-trained on the BERT corpora with an additional corpus of PubMed abstracts.

One such MLM is ouBioBERT, which was pre-trained on English Wikipedia (2.20 billion words), BooksCorpus (0.85 billion words), and PubMed abstracts (3.11 billion words) [45]. At the time, ouBioBERT was state-of-the-art on nine of ten tasks in the BLUE benchmark, including the DDI, CHEMPROT, and i2b2 2010 RE tasks,

and was therefore chosen as the most appropriate model to fine-tune for this paper's applications.

5.3.1 Precision, Recall, and F1 Score

A common metric for evaluating classification models is the F1 score, which is based on two other metrics called precision and recall [195, 196]. The equations for each are shown below.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{5.1}$$

$$precision = \frac{TP}{TP + FP} \tag{5.2}$$

$$recall = \frac{TP}{TP + FN} \tag{5.3}$$

The abbreviations in these equations are as follows. TP is true positives, or positive examples labeled correctly by the model. FP is false positives, or positive examples labeled incorrectly by the model. FN is false negatives, or negative examples labeled incorrectly by the model. There also exist true negatives (TN), or negative examples labeled correctly by the model, but this is not used in the above equations.

Precision measures how many of the examples labeled positive by the model were ground truth positives. Recall measures how many of the ground truth positives were labeled correctly by the model.

5.4 Methodology

The FXS corpus is created from 78 abstracts relating to FXS. These abstracts were obtained through a PubMed search. Each abstract is split into its constituent sentences, and for each sentence the medical named entities are identified using the scispaCy python package. For each combination of two entities within the sentence,

a relation is predicted from the i2b2 2010 relations (Table 5.1). This prediction was performed by a version of ouBioBERT fine-tuned on the i2b2 2010 data set by the authors. A flowchart summarizing this process is shown in Figure 5.1.

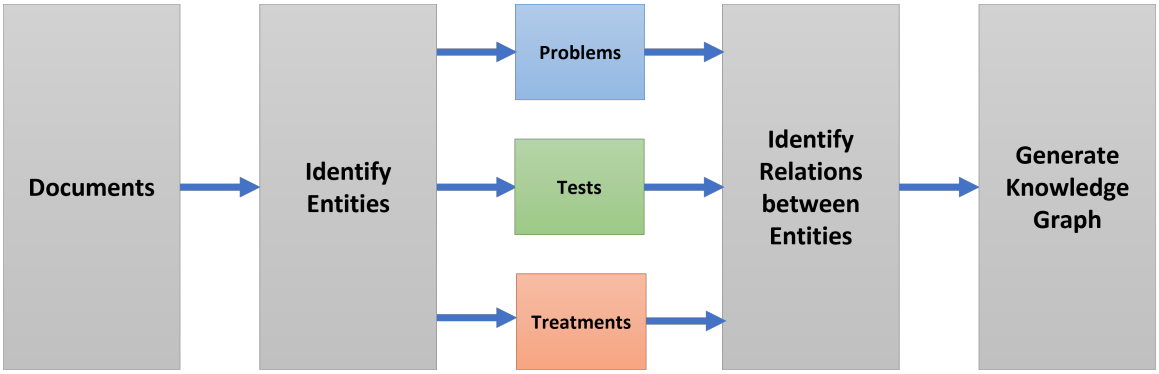


Figure 5.1: Envisioned KG generation flowchart

Labeling the corpus of FXS abstracts was performed by twelve volunteers. Abstracts were split up so each abstract would be labeled by three volunteers. For each subset of abstracts, a volunteer labeled each identified entity as a "medical problem", "test", or "treatment", as these are the three possible entities for relations in the i2b2 2010 data set (see Table 5.1). Entities could also be labeled as none of the above. For each 2-entity combination from the remaining entities, ouBioBERT predicted the relation (from a cache of prediction for each possible entity type combination) and this prediction was presented to the volunteer, who could either agree with the label or suggest a new label from the i2b2 2010 relations (Table 5.1). Only relations within the same sentence were considered. An example abstract with entities outlined is shown in Figure 5.2, and the resulting KG from this abstract labeling is shown in Figure 5.3.

Two new data sets are created from this volunteer labeling, one for NER and one for RE. Each data set is split into training, validation, and test subsets. Abstracts are randomly placed into one of the subsets; 48 abstracts are placed in the training subset, 10 in the validation subset, and 20 in the test subset. Evaluation of a baseline model is performed on the new data sets by fine-tuning ouBioBERT on the training

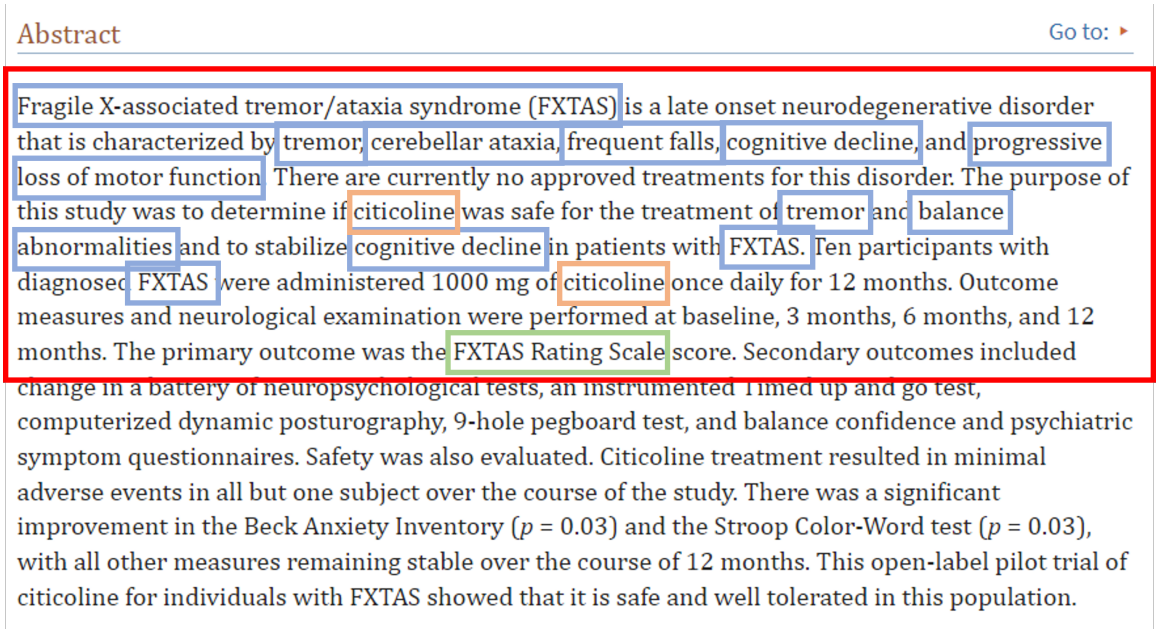


Figure 5.2: Example abstract with entities outlined [197]

subset and reporting performance on the test subset. The validation subset is used for comparing model checkpoints during training. For the NER data set, ouBioBERT is fine-tuned to predict each word as a "medical problem", "test", "treatment", or none of the above. For the RE data set, ouBioBERT is fine-tuned to predict each relation as one of the i2b2 2010 relations. Performance on the RE data test subset is compared between the model fine-tuned on the RE data training subset and the model fine-tuned on the i2b2 2010 data set.

5.5 Results

We perform experiments on the RE data set created from the volunteer labeling. We fine-tune an ouBioBERT model on the training subset of the FXS data set, with early stopping using the validation subset of the same data set. As a baseline, we use ouBioBERT fine-tuned on the training subset of the original i2b2 2010 data set, with early stopping based on the validation subset of the same data set. We compare the results of both models on the test subset of the FXS data set. Results are show in Table 5.2.

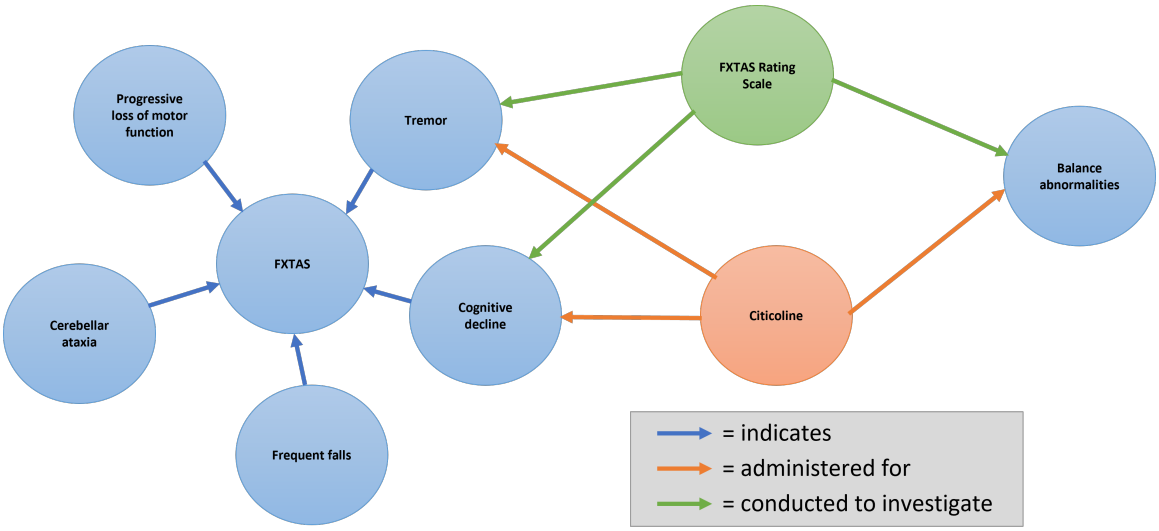


Figure 5.3: Example KG from abstract

5.6 Discussion

Both models perform fairly well on our FXS data set. The FXS model has slightly lower precision than the i2b2 model (0.757 vs. 0.847), but has much higher recall (0.730 vs. 0.528). This shows that our model has much fewer false negatives, but also more false positives. This shows that there are benefits to both models, and the larger i2b2 2010 data set teaches the model some things that were not learned from the smaller FXS data set. Training on both data sets may improve performance further.

The model we used, ouBioBERT [45], was state-of-the-art in many biomedical machine learning tasks at the time of doing this research. Since then, many improvements have been made in language modeling and biomedical machine learning. The current state-of-the-art as of writing this is probably GPT-4 [18], which could be prompted with a few examples to extract triples from medical text, requiring much less labeling than was required for our experiments. For reference, GPT-4 got an 84% (zero-shot) on the United States Medical Licensing Exam [198].

GPT-4 is closed-source, so if one was looking for an open-source alternative there are a few options. BioGPT was specifically trained for biomedical tasks [51]. Cerebras-

Label	Support	Precision		Recall		F1 Score		Accuracy	
		FXS	i2b2	FXS	i2b2	FXS	i2b2	FXS	i2b2
PIP	304	0.799	0.975	0.717	0.391	0.756	0.559	0.843	0.791
TeCP	254	0.746	0.870	0.776	0.528	0.761	0.657	0.862	0.844
TeRP	58	0.351	0.556	0.224	0.690	0.274	0.615	0.923	0.944
TrAP	136	0.797	0.882	0.926	0.770	0.857	0.823	0.953	0.950
TrCP	4	0.667	0.400	0.500	0.500	0.571	0.444	0.997	0.994
TrIP	5	0.333	1.000	0.200	0.400	0.250	0.571	0.993	0.997
TrNAP	1	0.000	0.500	0.000	1.000	0.000	0.667	0.999	0.999
TrWP	1	0.000	0.000	0.000	0.000	0.000	0.000	0.999	0.999
false	137	0.401	0.259	0.474	0.803	0.435	0.392	0.812	0.621
Total	900	0.757	0.847	0.730	0.528	0.742	0.650	0.931	0.904

Table 5.2: Model Comparison

GPT [199] and Pythia [200] were both trained on The Pile [201], which includes a lot of text from PubMed.

5.7 Conclusion

We compare the performance of two models on the test subset of a volunteer-labeled FXS data set. Our model is fine-tuned on the training subset of the same data set, and we compare to a baseline model fine-tuned on the i2b2 2010 data set. Our model has much higher recall than the baseline, but has lower precision. Our model could be improved by collecting more data or fine-tuning on both data sets. We believe there now exist better alternatives for extracting triples from text, such as prompting GPT-4.

Chapter 6

Conclusion

We built a domain-specific KG about NDDs to assist medical professionals, caretakers, and patients. We had multiple medical professionals in the loop during the creation of our KG, which combines academic knowledge along useful to professionals using the KG and online information that is more useful to patient and caretakers who require information in non-academic areas such as financing and services. We hope our methodology can be applied to other medical domains on the same level as NDDs, and that these lower-level KGs can be combined into a general medical KG that contains more accurate information that is more useful to non-professionals.

While creating the KG, we noticed that the topic modeling annotation we used as one of the annotation methods while creating the KG was not to the level we would have liked. To remedy this, we began to explore potential improvements to topic modeling algorithms. We eventually came to the idea of using RL as a topic model. Our RL topic model outperformed all other unsupervised topic models on 11 different data sets, and even performed favourably against topic models using supervised labeling. This new RL topic model can be used to augment our KG creation process, and also for any other application requiring topic modeling.

We also noticed that recent advances in NLP could be applied to directly extract triples from text. To do this, we created a training set from student annotations of the abstracts of medical literature about FragileX. We then used this training

set to fine-tune ouBioBERT, a medical LLM. Due to the rapid advances in NLP, this method is likely already outdated, as there are better medical LLMs such as BioGPT. Alternatively, the zero-shot or few-shot learning power of GPT-3 and GPT-4 may be able to be used to extract triples from text with only a few annotated examples rather than the many annotated examples we required for fine-tuning.

Bibliography

[1] International Data Corporation. "Global datasphere." (2023), [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=IDC_P38353 (visited on 03/06/2023).

[2] E. Burgener and J. Rydning. "High data growth and modern applications drive new storage requirements in digitally transformed enterprises." (2022), [Online]. Available: https://www.delltechnologies.com/asset/en-us/products/storage/industry-market/h19267-wp-idc-storage-reqs-digital-enterprise.pdf (visited on 03/06/2023).

[3] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[4] P. Villalobos, J. Sevilla, L. Heim, T. Besiroglu, M. Hobbhahn, and A. Ho, "Will we run out of data? an analysis of the limits of scaling datasets in machine learning," *arXiv preprint arXiv:2211.04325*, 2022.

[5] P. Norvig. "On chomsky and the two cultures of statistical learning." (), [Online]. Available: https://norvig.com/chomsky.html (visited on 03/06/2023).

[6] N. Chomsky, "Three models for the description of language," *IRE Transactions on information theory*, vol. 2, no. 3, pp. 113–124, 1956.

[7] N. Chomsky and D. W. Lightfoot, *Syntactic structures*. Walter de Gruyter, 2002.

[8] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[9] B. F. Skinner, *Science and human behavior*. Simon and Schuster, 1965.

[10] B. F. Skinner, *Verbal behavior*. New York: Appleton-Century-Crofts, 1957.

[11] J. Kaplan *et al.*, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[12] J. Hestness *et al.*, "Deep learning scaling is predictable, empirically," *arXiv preprint arXiv:1712.00409*, 2017.

[13] T. Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[14] UBS. "Let's chat about ChatGPT." (2023), [Online]. Available: https://www.ubs.com/global/en/wealth-management/our-approach/marketnews/article.1585717.html (visited on 03/06/2023).

[15] OpenAI. "Introducing ChatGPT." (2022), [Online]. Available: https://openai.com/blog/chatgpt (visited on 03/06/2023).

[16] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[17] J. Hoffmann *et al.*, "Training compute-optimal large language models," *arXiv preprint arXiv:2203.15556*, 2022.

[18] OpenAI. "Gpt-4." (2023), [Online]. Available: https://openai.com/research/gpt-4 (visited on 03/14/2023).

[19] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. ARTICLE, pp. 2493–2537, 2011.

[20] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[21] 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.

[22] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[23] Hugging Face. "Language modeling." (), [Online]. Available: https://huggingface.co/docs/transformers/tasks/language_modeling (visited on 03/14/2023).

[24] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[25] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[26] L. Chen *et al.*, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15 084–15 097, 2021.

[27] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[28] W. L. Taylor, "Cloze procedure: A new tool for measuring readability," *Journalism quarterly*, vol. 30, no. 4, pp. 415–433, 1953.

[29] J. Firth, "A synopsis of linguistic theory, 1930-1955," *Studies in linguistic analysis*, pp. 10–32, 1957.

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

[31] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[32] Z. Ji *et al.*, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, 2022.

[33] A. Aghajanyan *et al.*, "Cm3: A causal masked multimodal model of the internet," *arXiv preprint arXiv:2201.07520*, 2022.

[34] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001.

[35] E. Miller, "An introduction to the resource description framework.," *D-lib Magazine*, 1998.

[36] A. Hogan *et al.*, "Knowledge graphs," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–37, 2021.

[37] B. AlKhamissi, M. Li, A. Celikyilmaz, M. Diab, and M. Ghazvininejad, "A review on language models as knowledge bases," *arXiv preprint arXiv:2204.06031*, 2022.

[38] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "Ernie: Enhanced language representation with informative entities," *arXiv preprint arXiv:1905.07129*, 2019.

[39] W. Xiong, J. Du, W. Y. Wang, and V. Stoyanov, "Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model," *arXiv preprint arXiv:1912.09637*, 2019.

[40] F. Moiseev, Z. Dong, E. Alfonseca, and M. Jaggi, "Skill: Structured knowledge infusion for large language models," *arXiv preprint arXiv:2205.08184*, 2022.

[41] S. Hao, B. Tan, K. Tang, H. Zhang, E. P. Xing, and Z. Hu, "Bertnet: Harvesting knowledge graphs from pretrained language models," *arXiv preprint arXiv:2206.14268*, 2022.

[42] J. Youn and I. Tagkopoulos, "Kglm: Integrating knowledge graph structure in language models for link prediction," *arXiv preprint arXiv:2211.02744*, 2022.

[43] J. Lee *et al.*, "Biobert: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[44] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[45] S. Wada, T. Takeda, S. Manabe, S. Konishi, J. Kamohara, and Y. Matsumura, "Pre-training technique to localize medical bert and enhance biomedical bert," *arXiv preprint arXiv:2005.07202*, 2020.

[46] H.-C. Shin *et al.*, "Biomegatron: Larger biomedical domain language model," *arXiv preprint arXiv:2010.06060*, 2020.

[47] G. Miolo, G. Mantoan, and C. Orsenigo, "Electramed: A new pre-trained language representation model for biomedical nlp," *arXiv preprint arXiv:2104.09585*, 2021.

[48] L. N. Phan *et al.*, "Scifive: A text-to-text transformer model for biomedical literature," *arXiv preprint arXiv:2106.03598*, 2021.

[49] K. raj Kanakarajan, B. Kundumani, and M. Sankarasubbu, "Bioelectra: Pretrained biomedical text encoder using discriminators," in *Proceedings of the 20th Workshop on Biomedical Language Processing*, 2021, pp. 143–154.

[50] S. Alrowili and K. Vijay-Shanker, "Biom-transformers: Building large biomedical language models with bert, albert and electra," in *Proceedings of the 20th Workshop on Biomedical Language Processing*, 2021, pp. 221–227.

[51] R. Luo *et al.*, "Biogpt: Generative pre-trained transformer for biomedical text generation and mining," *Briefings in Bioinformatics*, vol. 23, no. 6, 2022.

[52] T. L. Scao *et al.*, "Bloom: A 176b-parameter open-access multilingual language model," *arXiv preprint arXiv:2211.05100*, 2022.

[53] R. Taylor *et al.*, "Galactica: A large language model for science," *arXiv preprint arXiv:2211.09085*, 2022.

[54] K. Singhal *et al.*, "Large language models encode clinical knowledge," *arXiv preprint arXiv:2212.13138*, 2022.

[55] W. D. Heaven. "Why meta's latest large language model survived only three days online." (2022), [Online]. Available: https://www.technologyreview.com/2022/11/18/1063487/meta-large-language-model-ai-only-survived-three-days-gpt-3-science/ (visited on 03/14/2023).

[56] Y. Tian, W. Shen, Y. Song, F. Xia, M. He, and K. Li, "Improving biomedical named entity recognition with syntactic information," *BMC bioinformatics*, vol. 21, no. 1, pp. 1–17, 2020.

[57] Z. Yuan, Y. Liu, C. Tan, S. Huang, and F. Huang, "Improving biomedical pretrained language models with knowledge," *arXiv preprint arXiv:2104.10344*, 2021.

[58] M. Yasunaga, J. Leskovec, and P. Liang, "Linkbert: Pretraining language models with document links," *arXiv preprint arXiv:2203.15827*, 2022.

[59] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs.," *SEMANTiCS (Posters, Demos, SuCCESS)*, vol. 48, no. 1-4, p. 2, 2016.

[60] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.

[61] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 2, pp. 494–514, 2021.

[62] J. Lehmann *et al.*, "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic web*, vol. 6, no. 2, pp. 167–195, 2015.

[63] M. Fabian, K. Gjergji, W. Gerhard, *et al.*, "Yago: A core of semantic knowledge unifying wordnet and wikipedia," in *16th International world wide web conference, WWW*, 2007, pp. 697–706.

[64] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic web*, vol. 8, no. 3, pp. 489–508, 2017.

[65] D. Jurafsky and J. H. Martin, *Speech and language processing*, 3rd ed. draft. Jan. 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/ (visited on 03/27/2023).

[66] C. Manning and H. Schutze, *Foundations of statistical natural language processing*. MIT press, 1999.

[67] R. Churchill and L. Singh, "The evolution of topic modeling," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–35, 2022.

[68] 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.

[69] H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du, and W. Buntine, "Topic modelling meets deep neural networks: A survey," *arXiv preprint arXiv:2103.00498*, 2021.

[70] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine learning*, vol. 39, pp. 103–134, 2000.

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

[72] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 113–120.

[73] J. Chang, S. Gerrish, C. Wang, J. Boyd-Graber, and D. Blei, "Reading tea leaves: How humans interpret topic models," *Advances in neural information processing systems*, vol. 22, 2009.

[74] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker, "Perplexity—a measure of the difficulty of speech recognition tasks," *The Journal of the Acoustical Society of America*, vol. 62, no. S1, S63–S63, 1977.

[75] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," in *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, 2010, pp. 100–108.

[76] N. Aletras and M. Stevenson, "Evaluating topic coherence using distributional semantics," in *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, 2013, pp. 13–22.

[77] J. H. Lau, D. Newman, and T. Baldwin, "Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 530–539.

[78] A. B. Dieng, F. J. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 439–453, 2020.

[79] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[80] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[81] N. K. Arora *et al.*, "Neurodevelopmental disorders in children aged 2–9 years: Population-based burden estimates across five regions in india," *PLoS medicine*, vol. 15, no. 7, e1002615, 2018.

[82] E. Emerson, "Deprivation, ethnicity and the prevalence of intellectual and developmental disabilities," *J Epidemiol Community Health*, vol. 66, no. 3, pp. 218–224, 2012.

[83] E. Taylor, "Developing adhd," *Journal of Child Psychology and Psychiatry*, vol. 50, no. 1-2, pp. 126–132, 2009.

[84] S. Johnson *et al.*, "Neurodevelopmental disability through 11 years of age in children born before 26 weeks of gestation," *Pediatrics*, vol. 124, no. 2, e249–e257, 2009.

[85] L. H. Zauche, A. E. D. Mahoney, and M. K. Higgins, "Predictors of co-occurring neurodevelopmental disabilities in children with autism spectrum disorders," *Journal of pediatric nursing*, vol. 35, pp. 113–119, 2017.

[86] B. H. Hansen, B. Oerbeck, B. Skirbekk, B. É. Petrovski, and H. Kristensen, "Neurodevelopmental disorders: Prevalence and comorbidity in children referred to mental health services," *Nordic Journal of Psychiatry*, vol. 72, no. 4, pp. 285–291, 2018.

[87] N. Tatishvili, M. Gabunia, N. Laliani, and S. Tatishvili, "Epidemiology of neurodevelopmental disorders in 2 years old georgian children. pilot study–population based prospective study in a randomly chosen sample," *european journal of paediatric neurology*, vol. 14, no. 3, pp. 247–252, 2010.

[88] H. A. Zucker, *Tackling online misinformation: A critical component of effective public health response in the 21st century*, 2020.

[89] Y. Zhao, J. Da, and J. Yan, "Detecting health misinformation in online health communities: Incorporating behavioral features into machine learning based approaches," *Information Processing & Management*, vol. 58, no. 1, p. 102 390, 2021.

[90] M. Parker and M. Killian, "Autism spectrum disorder and complex healthcare needs: The role of healthcare experiences," *Research in Autism Spectrum Disorders*, vol. 73, p. 101 535, 2020.

[91] H. Eklund *et al.*, "Needs of adolescents and young adults with neurodevelopmental disorders: Comparisons of young people and parent perspectives," *Journal of Autism and Developmental Disorders*, vol. 48, pp. 83–91, 2018.

[92] J. S. Bloch and J. D. Weinstein, "Families of young children with autism," *Social Work in Mental Health*, vol. 8, no. 1, pp. 23–40, 2009.

[93] R. R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Transactions on systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[94] P. Ernst, C. Meng, A. Siu, and G. Weikum, "Knowlife: A knowledge graph for health and life sciences," in *2014 IEEE 30th International Conference on Data Engineering*, IEEE, 2014, pp. 1254–1257.

[95] L. Shi, S. Li, X. Yang, J. Qi, G. Pan, and B. Zhou, "Semantic health knowledge graph: Semantic integration of heterogeneous medical knowledge and services," *BioMed research international*, vol. 2017, 2017.

[96] M. Sheng *et al.*, "Dekgb: An extensible framework for health knowledge graph," in *Smart Health: International Conference, ICSH 2019, Shenzhen, China, July 1–2, 2019, Proceedings 7*, Springer, 2019, pp. 27–38.

[97] L. Li *et al.*, "Real-world data medical knowledge graph: Construction and applications," *Artificial intelligence in medicine*, vol. 103, p. 101 817, 2020.

[98] Y. Zhang *et al.*, "Hkgb: An inclusive, extensible, intelligent, semi-auto-constructed knowledge graph framework for healthcare with clinicians' expertise incorporated," *Information Processing & Management*, vol. 57, no. 6, p. 102 324, 2020.

[99] Z. Huang, J. Yang, F. van Harmelen, and Q. Hu, "Constructing knowledge graphs of depression," in *Health Information Science: 6th International Conference, HIS 2017, Moscow, Russia, October 7-9, 2017, Proceedings 6*, Springer, 2017, pp. 149–161.

[100] X. Chai, "Diagnosis method of thyroid disease combining knowledge graph and deep learning," *IEEE Access*, vol. 8, pp. 149 787–149 795, 2020.

[101] D. Flocco *et al.*, "An analysis of covid-19 knowledge graph construction and applications," in *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, 2021, pp. 2631–2640.

[102] AIDE Canada. "Resources for autism & intellectual disability." (2022), [Online]. Available: https://aidecanada.ca/ (visited on 03/08/2023).

[103] The Family & Community Resource Centre. "Ndd care coordination project: About." (2023), [Online]. Available: http://fcrc.albertahealthservices.ca/coordination/about/ (visited on 03/08/2023).

[104] Inform Alberta. "Alberta's province-wide service directory." (), [Online]. Available: https://informalberta.ca/public/common/search.do (visited on 03/08/2023).

[105] O. Bodenreider, "The unified medical language system (umls): Integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D267–D270, 2004.

[106] S. Köhler *et al.*, "The human phenotype ontology in 2021," *Nucleic acids research*, vol. 49, no. D1, pp. D1207–D1217, 2021.

[107] Institute of Education Services. "Education resources information center." (2023), [Online]. Available: https://eric.ed.gov/?ti=all (visited on 03/08/2023).

[108] Alliance of Information and Referral Systems. "The taxonomy." (2023), [Online]. Available: https://www.airs.org/i4a/pages/index.cfm?pageid=3386 (visited on 03/08/2023).

[109] National Library of Medicine. "Umls metathesaurus browser." (2022), [Online]. Available: https://uts.nlm.nih.gov/uts/umls/home (visited on 03/08/2023).

[110] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[111] M. Neumann, D. King, I. Beltagy, and W. Ammar, "Scispacy: Fast and robust models for biomedical natural language processing," *arXiv preprint arXiv:1902.07669*, 2019.

[112] N. Polettini, "The vector space model in information retrieval-term weighting problem," *Entropy*, vol. 34, pp. 1–9, 2004.

[113] A. Singhal, C. Buckley, and M. Mitra, "Pivoted document length normalization," in *Acm sigir forum*, ACM New York, NY, USA, vol. 51, 2017, pp. 176–184.

[114] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Nov. 2019. [Online]. Available: https://arxiv.org/abs/1908.10084.

[115] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[116] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[117] W. He, B. Dutta, R. M. Rodríguez, A. A. Alzahrani, and L. Martínez, "Induced owa operator for group decision making dealing with extended comparative linguistic expressions with symbolic translation," *Mathematics*, vol. 9, no. 1, p. 20, 2020.

[118] F. Gong, M. Wang, H. Wang, S. Wang, and M. Liu, "Smr: Medical knowledge graph embedding for safe medicine recommendation," *Big Data Research*, vol. 23, p. 100 174, 2021.

[119] X. Wu, J. Duan, P. Yi, and M. Li, "Medical knowledge graph: Data sources, construction, reasoning, and applications," *Big Data Mining and Analytics*, 2022.

[120] P. Ernst, A. Siu, and G. Weikum, "Knowlife: A versatile approach for constructing a large knowledge graph for biomedical sciences," *BMC bioinformatics*, vol. 16, pp. 1–13, 2015.

[121] T. Yu *et al.*, "Knowledge graph for tcm health preservation: Design, construction, and applications," *Artificial intelligence in medicine*, vol. 77, pp. 48–52, 2017.

[122] E. R. et al. "Big-O algorithm complexity cheat sheet." (2019), [Online]. Available: https://www.bigocheatsheet.com/ (visited on 03/08/2023).

[123] Neo4j. "Indexes for search performance." (2019), [Online]. Available: https://neo4j.com/docs/cypher-manual/5/indexes-for-search-performance/ (visited on 03/08/2023).

[124] J. Jin, J. Luo, S. Khemmarat, and L. Gao, "Querying web-scale knowledge graphs through effective pruning of search space," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 8, pp. 2342–2356, 2017.

[125] Y. Lv and C. Zhai, "Lower-bounding term frequency normalization," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 7–16.

[126] F. Rousseau and M. Vazirgiannis, "Graph-of-word and tw-idf: New approach to ad hoc ir," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 59–68.

[127] M. Lissandrini, D. Mottin, T. Palpanas, and Y. Velegrakis, "Graph-query suggestions for knowledge graph exploration," in *Proceedings of The Web Conference 2020*, 2020, pp. 2549–2555.

[128] C. Xiong and J. Callan, "Query expansion with freebase," in *Proceedings of the 2015 international conference on the theory of information retrieval*, 2015, pp. 111–120.

[129] S. Balaneshinkordan and A. Kotov, "An empirical comparison of term association and knowledge graphs for query expansion," in *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, Springer, 2016, pp. 761–767.

[130] A. Füßl and V. Nissen, "Interpretability of knowledge graph-based explainable process analysis," in *2022 IEEE Fifth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, IEEE, 2022, pp. 9–17.

[131] Y. Xian, Z. Fu, S. Muthukrishnan, G. De Melo, and Y. Zhang, "Reinforcement knowledge graph reasoning for explainable recommendation," in *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 2019, pp. 285–294.

[132] N. Ammar, A. Shaban-Nejad, *et al.*, "Explainable artificial intelligence recommendation system by leveraging the semantics of adverse childhood experiences: Proof-of-concept prototype development," *JMIR medical informatics*, vol. 8, no. 11, e18752, 2020.

[133] K. Cheng, N. Wang, and M. Li, "Interpretability of deep learning: A survey," in *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*, Springer, 2021, pp. 475–486.

[134] L. Douze, S. Pelayo, N. Messaadi, J. Grosjean, G. Kerdelhué, and R. Marcilly, "Designing formulae for ranking search results: Mixed methods evaluation study," *JMIR Human Factors*, vol. 9, no. 1, e30258, 2022.

[135] A. Srivastava and C. Sutton, "Autoencoding variational inference for topic models," *arXiv preprint arXiv:1703.01488*, 2017.

[136] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[137] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *International conference on machine learning*, PMLR, 2016, pp. 1727–1736.

[138] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[139] P. Dayan and G. E. Hinton, "Using expectation-maximization for reinforcement learning," *Neural Computation*, vol. 9, no. 2, pp. 271–278, 1997.

[140] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," *arXiv preprint arXiv:1805.00909*, 2018.

[141] A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. Riedmiller, "Maximum a posteriori policy optimisation," *arXiv preprint arXiv:1806.06920*, 2018.

[142] M. Fellows, A. Mahajan, T. G. Rudner, and S. Whiteson, "Virel: A variational inference framework for reinforcement learning," *Advances in neural information processing systems*, vol. 32, 2019.

[143] T.-N. Doan and T.-A. Hoang, "Benchmarking neural topic models: An empirical study," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 4363–4368.

[144] D. Card, C. Tan, and N. A. Smith, "Neural models for documents with metadata," *arXiv preprint arXiv:1705.09296*, 2017.

[145] L. Liu, H. Huang, Y. Gao, Y. Zhang, and X. Wei, "Neural variational correlated topic modeling," in *The World Wide Web Conference*, 2019, pp. 1142–1152.

[146] Y. Miao, E. Grefenstette, and P. Blunsom, "Discovering discrete latent topics with neural variational inference," in *International Conference on Machine Learning*, PMLR, 2017, pp. 2410–2419.

[147] T. Lin, Z. Hu, and X. Guo, "Sparsemax and relaxed wasserstein for topic sparsity," in *Proceedings of the twelfth ACM international conference on web search and data mining*, 2019, pp. 141–149.

[148] R. Zhao, V. Tan, and H. Xu, "Online nonnegative matrix factorization with general divergences," in *Artificial Intelligence and Statistics*, PMLR, 2017, pp. 37–45.

[149] M. Hoffman, F. Bach, and D. Blei, "Online learning for latent dirichlet allocation," *advances in neural information processing systems*, vol. 23, 2010.

[150] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National academy of Sciences*, vol. 101, no. suppl_1, pp. 5228–5235, 2004.

[151] M. Grootendorst, "Bertopic: Neural topic modeling with a class-based tf-idf procedure," *arXiv preprint arXiv:2203.05794*, 2022.

[152] D. Angelov, "Top2vec: Distributed representations of topics," *arXiv preprint arXiv:2008.09470*, 2020.

[153] F. Bianchi, S. Terragni, and D. Hovy, "Pre-training is a hot topic: Contextualized document embeddings improve topic coherence," *arXiv preprint arXiv:2004.03974*, 2020.

[154] T. Nguyen and A. T. Luu, "Contrastive learning for neural topic model," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 974–11 986, 2021.

[155] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, IEEE, vol. 2, 2006, pp. 1735–1742.

[156] L. Gui, J. Leng, G. Pergola, Y. Zhou, R. Xu, and Y. He, "Neural topic model with reinforcement learning," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3478–3483.

[157] A. Kumar, N. Esmaili, and M. Piccardi, "A reinforced variational autoencoder topic model," in *International Conference on Neural Information Processing*, Springer, 2021, pp. 360–369.

[158] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[159] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[160] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[161] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[162] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, PMLR, 2015, pp. 1889–1897.

[163] S. Filippi, O. Cappé, and A. Garivier, "Optimism in reinforcement learning and kullback-leibler divergence," in *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2010, pp. 115–122.

[164] T. Kobayashi, "Optimistic reinforcement learning by forward kullback–leibler divergence optimization," *Neural Networks*, vol. 152, pp. 169–180, 2022.

[165] N. Vieillard, T. Kozuno, B. Scherrer, O. Pietquin, R. Munos, and M. Geist, "Leverage the average: An analysis of kl regularization in reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 163–12 174, 2020.

[166] H. J. Kappen, V. Gómez, and M. Opper, "Optimal control as a graphical model inference problem," *Machine learning*, vol. 87, no. 2, pp. 159–182, 2012.

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

[168] F. Bianchi, S. Terragni, D. Hovy, D. Nozza, and E. Fersini, "Cross-lingual contextualized topic models with zero-shot learning," *arXiv preprint arXiv:2004.07737*, 2020.

[169] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 976–11 986.

[170] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[171] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[172] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, PMLR, 2015, pp. 448–456.

[173] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[174] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[175] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, PMLR, 2013, pp. 1310–1318.

[176] I. Higgins *et al.*, "Beta-vae: Learning basic visual concepts with a constrained variational framework," 2016.

[177] K. Lang, "Newsweeder: Learning to filter netnews," in *Machine Learning Proceedings 1995*, Elsevier, 1995, pp. 331–339.

[178] E. Sandhaus, "The new york times annotated corpus," *Linguistic Data Consortium, Philadelphia*, vol. 6, no. 12, e26752, 2008.

[179] N. Ueda and K. Saito, "Parametric mixture models for multi-labeled text," *Advances in neural information processing systems*, vol. 15, 2002.

[180] T.-A. Hoang, K. D. Vo, and W. Nejdl, "W2e: A worldwide-event benchmark dataset for topic detection and tracking," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1847–1850.

[181] J. Qiang, Z. Qian, Y. Li, Y. Yuan, and X. Wu, "Short text topic modeling techniques, applications, and performance: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 1427–1445, 2020.

[182] R. McCreadie, I. Soboroff, J. Lin, C. Macdonald, I. Ounis, and D. McCullough, "On building a reusable twitter corpus," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012, pp. 1113–1114.

[183] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 142–150.

[184] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

[185] M. Figurnov, S. Mohamed, and A. Mnih, "Implicit reparameterization gradients," *Advances in neural information processing systems*, vol. 31, 2018.

[186] Y. Tian, M. Han, C. Kulkarni, and O. Fink, "A prescriptive dirichlet power allocation policy with deep reinforcement learning," *Reliability Engineering & System Safety*, vol. 224, p. 108 529, 2022.

[187] A. J. Verkerk *et al.*, "Identification of a gene (fmr-1) containing a cgg repeat coincident with a breakpoint cluster region exhibiting length variation in fragile x syndrome," *Cell*, vol. 65, no. 5, pp. 905–914, 1991.

[188] D. C. Crawford, J. M. Acuña, and S. L. Sherman, "Fmr1 and the fragile x syndrome: Human genome epidemiology review," *Genetics in medicine*, vol. 3, no. 5, pp. 359–371, 2001.

[189] K. B. Garber, J. Visootsak, and S. T. Warren, "Fragile x syndrome," *European journal of human genetics*, vol. 16, no. 6, pp. 666–672, 2008.

[190] I. Segura-Bedmar, P. Martínez Fernández, and M. Herrero Zazo, "Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013)," Association for Computational Linguistics, 2013.

[191] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets," *arXiv preprint arXiv:1906.05474*, 2019.

[192] Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall, "2010 i2b2/va challenge on concepts, assertions, and relations in clinical text," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 552–556, 2011.

[193] Informatics for Integrating Biology & the Bedside. "NLP research data sets." (2023), [Online]. Available: https://www.i2b2.org/NLP/DataSets/ (visited on 04/12/2023).

[194] DBMI Data Portal. "n2c2 NLP research data sets." (2023), [Online]. Available: https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/ (visited on 04/12/2023).

[195] C. Van Rijsbergen, "Information retrieval: Theory and practice," in *Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems*, vol. 79, 1979.

[196] N. Chinchor and B. M. Sundheim, "Muc-5 evaluation metrics," in *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*, 1993.

[197] D. A. Hall *et al.*, "Open-label pilot clinical trial of citicoline for fragile x-associated tremor/ataxia syndrome (fxtas)," *Plos one*, vol. 15, no. 2, e0225191, 2020.

[198] H. Nori, N. King, S. M. McKinney, D. Carignan, and E. Horvitz, "Capabilities of gpt-4 on medical challenge problems," *arXiv preprint arXiv:2303.13375*, 2023.

[199] N. Dey *et al.*, "Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster," *arXiv preprint arXiv:2304.03208*, 2023.

[200] S. Biderman *et al.*, "Pythia: A suite for analyzing large language models across training and scaling," *arXiv preprint arXiv:2304.01373*, 2023.

[201] L. Gao *et al.*, "The pile: An 800gb dataset of diverse text for language modeling," *arXiv preprint arXiv:2101.00027*, 2020.