EXPLORATORY SEARCH USING VECTOR MODEL AND LINKED DATA

by

Daeun Yim

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

Department of Computer and Information Technology

West Lafayette, Indiana

August 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL

STATEMENT OF COMMITTEE APPROVAL

Dr. Baijian Yang, Chair

Department of Computer and Information Technology

Dr. Julia M. Rayz

Department of Computer and Information Technology

Dr. James L. Mohler

Department of Computer and Graphics Technology

Approved by:

Dr. John A. Springer

Head of the Graduate Program

ACKNOWLEDGMENTS

I would like to express sincere appreciation and gratitude to Dr.Yang for his advice and guidance that helped me through my situation. I also wish to gratefully acknowledge Dr.Rayz and Dr.Mohler for their insightful comments and dedication.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IR	Information Retrieval
NLP	Natural Languagge Processing
RDF	Resource Description Framework
ARI	Adjusted Rand Index
NMI	Normalized Mutual Index
BERT	Bidirectional Encoder Representations from Transformers
GMM	Gaussian Mixture Model

ABSTRACT

The way people acquire knowledge has largely shifted from print to web resources. Meanwhile, search has become the main medium to access information. Amongst various search behaviors, exploratory search represents a learning process that involves complex cognitive activities and knowledge acquisition. Research on exploratory search studies on how to make search systems help people seek information and develop intellectual skills. This research focuses on information retrieval and aims to build an exploratory search system that shows higher clustering performance and diversified search results. In this study, a new language model that integrates the state-of-the-art vector language model (i.e., BERT) with human knowledge is built to better understand and organize search results. The clustering performance of the new model (i.e., RDF+BERT) was similar to the original model but slight improvement was observed with conversational texts compared to the pre-trained language model and an exploratory search baseline. With the addition of the enrichment phase of expanding search results to related documents, the novel system also can display more diverse search results.

CHAPTER 1. INTRODUCTION

The study is committed to building a prototype of an exploratory search engine. Exploratory search is a form of search that involves complex cognitive activities and knowledge acquisition that contributes to developing intellectual skills (White & Roth, 2009). This chapter provides the motivation of the research and an overview of the problem addressed by this research. A brief description of the solution for the problem follows.

1.1 Background

For the last few decades, the development of technology has profoundly changed the way people acquire and learn knowledge. The way people produce and consume information has shifted from print to digital formats and the amount of data produced daily has been growing exponentially. In the meantime, search engines have become the main medium to reach the abundant sources of information on the World Wide Web. Many researchers and companies have contributed to the evolution of information retrieval (IR) technology to harness the information effectively (Sanderson & Croft, 2012).

Although the popular search engines people use today consist of similar interface and features, search behaviors vary in terms of purposes and processes. In a broad sense, search activities can be divided into two categories: lookup search and exploratory search (Marchionini, 2006). Lookup search refers to searching for a direct answer to a question, e.g.”Who wrote the first English dictionary?”, for which a user can easily find out an answer from the search result.

Table 1.1. *An example of exploratory search session*

Intention	Search Behavior
Start search session	search ”word2vec”
Navigate search results	reads first few documents
Generate subtopic keyword	search ”skip-gram and CBOW”
Refine subtopic keyword	search ”Mikolov word2vec paper”
Discover parent concept	searh ”word embeddings”
Generate subtopic keyword	search ”transformer”
Discover similar subtopics	discover ”RNN, LSTM etc.”
Continue similar process	browse and focus

On the other hand, exploratory search is the opposite of lookup search in that it is an activity of information seeking whose purpose is learning. Table 1.1 shows an example of exploratory search interaction between a user and a conventional search engine. During the interaction, a user filters out the necessary information, actively discovers subtopics, refines search words based on the discoveries, returns to the previous search words, and iterates the process continuously. If the search engine could automate this process and ease users' intellectual burdens, users' exploratory search could be performed faster and efficiently.

Exploratory search has been developed along with the advancement of IR and natural language processing (NLP) technology. The research field of exploratory search encompass diverse subtopics such as, conceptual research of itself (White & Roth, 2009), human and computer interaction (Wongsuphasawat et al., 2016), evaluation methods (Palagi, 2018), and retrieval methods. The research on retrieval methods studies how to organize and represent the retrieved results. Previous studies organized retrieved results in forms of document hierarchy (Yang, 2015), subtopic facets (Athukorala, Medlar, Oulasvirta, Jacucci, & Glowacka, 2016), and related concept graph (Kejriwal & Szekely, 2019). Organizing information in these forms was mainly accomplished either by traversing linked data (Tzitzikas, Manolis, & Papadakos, 2017) or grouping documents (Ortiz, Kim, Wang, Seki, & Mostafa, 2019). These two main approaches will be studied further in order to capture problems to be resolved.

1.2 Problem Statement

When a user explores search results demanding to acquire knowledge for an unfamiliar area, the user does not have an end goal and hopes to encounter serendipitous information (i.e. knowledge discovery) not knowing what exactly she is looking for (Foster & Ford, 2003). Especially, when the search task is broad and pieces of information are scattered, users struggle more finding necessary information(Chi, He, Han, & Jiang, 2018). For these reasons, it is desirable to provide search results in diverse subtopics related to the users' search inputs and, more importantly, in an organized way.

The conventional search method (i.e., listing documents ranked by relevance based on keywords matching) is highly convenient to complete simple lookup tasks. However, when the intention to explore information is detected, optimizing display and interaction of search results is proven to have better performance with regards to efficiency and effectiveness in acquiring knowledge by user studies (Y. Zhang, Broussard, Ke, & Gong, 2014). Hence, It is more desirable to adjust the interface and interaction with users based on the user's intent to explore the topic (Athukorala, Głowacka, Jacucci, Oulasvirta, & Vreeken, 2016).

There have been a number of studies on exploratory search to help searchers reduce searchers' browsing burden. The very first attempt is the pioneering scatter/gather method (Cutting, Karger, Pedersen, & Tukey, 1992). Scatter/gather method is to scatter documents into multiple clusters and gather documents from a selected cluster and then scattering them into smaller clusters again. Starting from this study, clustering had been the most popular method and various methods of grouping a subset of documents have been developed. However, after the concept of Semantic Web was introduced by Burners Lee Berners-Lee, Hendler, and Lassila (2001), many exploratory search papers adopted linked data traversing (i.e., knowledge graph and Resource Description Framework (RDF)) to provide relevant knowledge from user input (Tzitzikas et al., 2017). Organizing search results as a graph-like structure became more popular since then (Nuzzolese, Presutti, Gangemi, Peroni, & Ciancarini, 2016).

However, both document grouping and linked data approaches are not perfect solutions for exploratory search. Many document clustering works incorporate semantic level analysis to enhance clustering performance (Di Marco & Navigli, 2011) or still use conventional term frequency based vectorization (Ortiz et al., 2019). While documents clustering area has made significant advancements along with the improvement of vector representation of natural languages (Park, Park, Kim, Cho, & Park, 2019), exploratory search is yet to incorporate state-of-the-art achievements in the NLP field.

Linked data approach help explore information using existing knowledge made by human intelligence. However, linked data cannot be comprehensive enough to cover every piece of knowledge, and building comprehensive linked data can be very costly. Limiting the database to linked data means giving up the majority of available information on the Internet. Moreover, graph exploration only lists related entities on the words and phrases level (Tzitzikas et al., 2017). Displaying a phrase list does not provide enough contexts for users to catch information. Lastly, it is hard to match random documents to existing graph entities and reorganize search results using linked data entities (Fafalios, Holzmann, Kasturia, & Nejdl, 2017).

Due to limitations in linked data exploration, many previous studies on exploratory search are implemented on domain-specific data or academic papers to exploit structured data and intrinsic networks from citations and authors (Abbasi & Frommholz, 2015; Kejriwal & Szekely, 2019; Mohajeri, Samuel, Zalane, & Rafiei, 2016). However, to build a real-world search application, a general-purpose and high-performance exploratory engine needs to be built. Another minor problem of the field is that previous works do not solve ambiguity problems. Because of constraints on keyword matching in understanding semantics of texts, conventional IR methods rely on users to solve the ambiguity (i.e., polysemy) in search keywords and results (Di Marco & Navigli, 2013; Navrat, 2012).

1.3 Purpose

The main purpose of the study is to complement and harness the two aforementioned approaches (i.e., documents clustering and linked data traversing) to build a better performing exploratory search engine. This is mainly achieved by incorporating the state-of-the-art natural language vector representation model to enhance document clustering performance and integrating the model with knowledge information available as a form of linked data. The representation of search results will follow the scatter/gather framework (Cutting et al., 1992) as it shows the exploratory search concept very well. The novel system is expected to have better clustering performance than existing methods and reflect knowledge included in linked data as well as ensuring diversity in the search results. This study also aims to build a general-purpose system that can work on input words from any domain.

1.4 Significance

With the development of the internet, every source of information has been aggregated in the World Wide Web (Case, 2012), which has become the most popular information-seeking channel (Savolainen & Kari, 2004). Along with this change, search engines became the first place people turned to in order to find information, and thus we expect to develop it to support more intensive exploratory information seeking.

Exploring the information efficiently and helping users understand available resources easily are critical competencies not only at the individual level but also for enterprises, governments, and non-profit organizations (Sabou et al., 2018). Enhancing the use of internal knowledge across the organization can reduce the waste of 47 million dollars that comes from inefficient knowledge share (Panopto, 2018).

The exploratory search is a relatively well-researched area. It is proven that exploratory search supporting tools can be very helpful in completing given tasks (Kules, Capra, Banta, & Sierra, 2009). The two main approaches to provide relevant knowledge from searchers' input are to group retrieved documents and traverse knowledge entities from linked data. This research wants to utilize both document grouping and knowledge exploring methods while making progress by bridging the gap between major NLP achievements and exploratory search. The integrated vector model is expected to reflect the textual context and human knowledge. To the best of the author's knowledge, there has been no research that combined the distributional vector representation of the language model and knowledge graph to build an exploratory search engine. Also, to guarantee the scalability and flexibility of the system, the system needs to be able to be adjusted with newly available knowledge.

1.5 Research Question

The following research question was addressed in this study. Can the proposed exploratory search system show better performance than baseline studies in terms of clustering performance and diversity of search results? The baselines of the clustering and diversity performance are the following:

- Clustering performance compared to pre-trained language model written by Devlin, Chang, Lee, and Toutanova (2019) and graph embedding language model inspired by Ristoski and Paulheim (2016).
- Clustering performance compared to a clustering-based exploratory search system written by Ortiz et al. (2019).
- Diversity of search results compared to the suggested system without the diversifying stage.

These three performance criteria were chosen to prove whether the suggested solution can provide more organized and diverse search results to users compared to the baseline methods.

1.6 Assumptions

This research has the following assumptions:

- The linked data resource is comprehensive enough to cover general knowledge.
- The performance of clustering models remain consistent across diverse datasets.
- The annotated datasets are correctly classified.

1.7 Limitations

This research has the following limitations:

- The linked data resource does not reflect every bit of knowledge created by human intelligence.
- The database of the search system does not contain every available piece of information from the Internet.

1.8 Delimitations

This research includes the delimitations:

- Only the documents fully accessible to the public and written in English were tested.
- The system only processed text data, not images and videos.
- Only Reuters (Lewis, 1997), Yahoo answers (Lewis, Yang, Rose, & Li, 2004), BBC news (Greene & Cunningham, 2006), AG News, 20 News Group datasets were used to test clustering performance.
- The system only accepts a single word as a search input.

1.9 Summary

This chapter explains the necessity and importance of exploratory search field and briefly analyzes the problems that have arisen from previous studies on exploratory search and the gaps between relevant NLP areas and the target area. To resolve the aforementioned problems, the study commits to building a prototype of exploratory search engine that complements the existing system and the research question is to test the performance of the novel system in terms of clustering and diversity of search results. Chapter 2 is a literature review on the problem and methodology.

CHAPTER 2. REVIEW OF LITERATURE

This chapter is a review of the literature relevant to the exploratory search. Firstly, the chapter examines conceptual studies and information retrieval frameworks on exploratory search. For the methodology literature review, it reviews the development of word embeddings, graph embeddings, and integration of those two embeddings.

2.1 Exploratory Search

This section explores studies on exploratory search concepts and its implementations which leads to an analysis of existing problems in this area. This section introduces the concept of exploratory search in detail and analyzes studies that implemented the abstract ideas.

2.1.1 Conceptual Work on Exploratory Search

Starting from (Marchionini, 2006), researchers started to recognize different types of search behaviors, i.e., lookup search and exploratory search. The author paid attention to the remarkable differences of exploratory search from lookup search and defined the distinguishing characteristics of exploratory search are similar to learning and investigating. The concept of exploratory search is discerned from lookup search which refers to looking up for a short answer to a question such as ”Who wrote the first English dictionary?”. In contrast, an exploratory search starts with a motivation to learn about a topic (e.g., ”I want to learn more about natural language processing”). During the search session for this question, the user would filter out necessary information from search results, actively discover subtopics, learn new information, refine search words based on the discoveries, return to the previous search words from time to time, and iterate the process continuously.

Further detailed reflection on the concept was performed by White and Roth (2009). The author explains that exploratory search represents the searches when a user is not familiar with the domain, not certain how to achieve their goals, or not certain about what their goals are. The goal of the exploratory search is to examine a topic and acquire knowledge during the search session.

To better understand exploratory search, we need to see it as a learning process. Information seeking can be learned in more detail by looking at how learning works. Bloom and Krathwohl (1966) introduces a taxonomy of learning by describing two cognitive levels. Cognition comprises of two levels. The lower level of cognition involves memorizing, recalling, and understanding facts while the higher level of cognition refers to the application, analysis, synthesis, and evaluation of acquired knowledge. The characteristics of exploratory search are related to the higher level of recognition. In an exploratory search session, searchers read and evaluate search results and aggregate necessary information. Then, they apply the acquired information to refine the search words and examine more information.

Similarly, conceptual work on traditional information-seeking behaviors can be extended to exploratory search. Information-seeking behaviors are metaphorically expressed as berry picking (Bates, 1989) or foraging (Pirolli & Card, 1995). These information-seeking behaviors were adopted to understand exploratory search after search became one of the main sources of learning. Now, information berry-picking and foraging represent a sequentially evolving process in which users repeat actions of browsing and focused search (Savolainen, 2018).

Other than finding similar features from traditional information-seeking behaviors, some researchers analyzed characteristics of exploratory search by looking at actual search logs. Rose and Levinson (2004) performed a user study to find that half of the number of informational searches are undirected search, which means that people searched a topic to learn more about it rather than to look for a specific answer. In a more recent study, the researchers found that there was a shift in search goals during searchers' exploratory search sessions (Ma & Zhang, 2018). Athukorala, Głowacka, et al. (2016) also proved a clear behavioral difference between simple lookup search, and exploratory search using search logs of an actual search engine. From the research, the authors recommend search engines adapt interaction with users evaluating whether the user is exploring information or searching for a simple answer since these search behaviors demand different needs.

After looking at the features of exploratory search, one can conclude that different search behaviors require different search results. However, modern web search engines do not tell the difference in users' intent nor provide enough help for users' exploratory information seeking (Aula & Russell, 2007). They interact with users by sending input keywords and search results back and forth. Users struggle more finding necessary information when the search task is broad and pieces of information are scattered (Chi et al., 2018) and it is a user's responsibility to refine her search words and iterate the same process multiple times (Aula, Khan, & Guan, 2010). Discovering related topics were the least supported by conventional search engines (Singer, Danilov, & Norbisrath, 2012) while exploratory search tools can flatten the gap between experts and novice users Kang and Fu (2010). These are representative reasons why research works on exploratory search want to incorporate the process of learning itself into improving search results (Tibau, Siqueira, Nunes, Bortoluzzi, & Marenzi, 2018).

2.1.2 Exploratory Search System

Among the various research fields within exploratory search, this research focuses on how to organize and represent information using natural language processing. Previous studies have developed exploratory search systems with diverse approaches such as document hierarchy, subtopic facets, and concept graphs. Organizing information in these forms was mainly accomplished by either grouping retrieved documents or traversing linked data.

2.1.2.1 Document Clustering Approaches

Most of the application papers and domain-specific exploratory search engine papers revisit the pioneering scatter/gather methods (Cutting et al., 1992). Scatter/gather refers to an information retrieval framework that iterates document clustering and scattering the selected cluster repetitively.

The paper is revisited and re-evaluated by domain-specific applications (Bascur, van Eck, & Waltman, 2019; Mohajeri et al., 2016; Ortiz et al., 2019). Other papers used sophisticated clustering methods that were applied to implement a scatter-gather exploratory search system. Di Marco and Navigli (2013) uses the semantic similarity between documents using word sense induction to reduce ambiguity. Abbasi and Frommholz (2015) clusters scholarly science papers using the optimum clustering framework (OCF) technique which computes relevance between documents based on input words from a user.

Clustering is frequently chosen by researchers to implement an exploratory search engine. Vector space clustering methods are the most frequently used and easy to implement. In most studies, researchers transformed documents to vector-matrix using term frequency-inverse document frequency (TF-IDF) methods or Doc2Vec model. This leads to severe problems that semantics are not reflected and training documentation embeddings for every text can be highly inefficient. They used classic vector space clustering techniques such as K-means, DBSCAN, and LDA to cluster the vectorized documents (Hall, Clough, & Stevenson, 2012; Ortiz et al., 2019; Ortiz, Seki, & Mostafa, 2018; Soares, Campello, Nourashrafeddin, Milios, & Naldi, 2019).

Another common approach to cluster search results is to use a graph data structure. A graph is constructed from a corpus by texts comprising vertices and relevance measures comprising edges. Measuring semantic similarity between document pairs is a popular method to connect edges among document vertices. The semantic similarity metric and clustering methods vary with papers. Angelova and Siersdorfer (2006) links document based on content similarity and groups neighbors. David and Kosala (2018) uses Ant Colony Optimization in which multi-agents find local optima by stochastic optimization. Yang (2015) builds a hierarchy from search results by semantic distances between document nodes. He et al. (2016) composes a network using the relationship between words and documents.

2.1.2.2 Linked Data Approaches

Linked data refers to a knowledge graph or network of data in which pieces of information are linked based on relations. Two entities are linked with an edge, which represents a relation. An entity can be connected with many other entities with many different relations. Linked data is hard and costly to build because it requires human intelligence, but it can be a great tool to help users expand their knowledge by enabling them to traverse from an anchor point to other entities understanding their relations.

For this reason, Semantic Web was initiated by Tim Berners Lee, the inventor of World Wide Web (Berners-Lee et al., 2001). The mission of the Semantic Web is to make resources on the Internet understandable to machines. Resource Description Framework (RDF) is a framework introduced to implement this idea (W3C, 1998). RDF is used to describe sources and merge data with a different underlying schema. DBPedia (Auer et al., 2007) and Wikidata (Vrandečić & Krötzsch, 2014) are the representative examples of comprehensive RDF database. This notion is also called as ‘linked data’ since it represents connected data of the Internet.

Consumer search engines such as Google, Yahoo, and Facebook, also adopted linked data to understand web resources at the semantic level. Google uses a knowledge graph to provide relevant information to named entities (Singer et al., 2012) and Facebook encourages web pages to include Open Graph (OG) protocol in their HTML metadata. Search engines recommend web pages to add RDF information in the HTML header to help their crawlers understand the contents of web pages better.

DBPedia and Wikidata are the most frequently used and comprehensive RDF database. Mirizzi, Ragone, Di Noia, and Di Sciascio (2010) built one of the first exploratory search systems that applied Semantic Web concept for. The paper presents a primitive graph exploring technique that allows constrained moves from an anchor point and rank neighbor nodes based on relevance by calculating the similarity of word frequency in documents.

Tzitzikas et al. (2017) analyzed 31 exploratory search engines using RDF/S. The paper introduces three types of transitions which are class-based browsing, property-based browsing, property path-based browsing, and entity type switch. The transition means how to show sub-graphs from a selected entity. Sherkhonov, Cuenca Grau, Kharlamov, and Kostylev (2017) adds multi-level exploration features. Cheng, Zhang, and Qu (2014) clusters related concepts using entities in linked data. The related concepts are computed based on pattern frequency.

Not all linked data adopts RDFs as resources. Nuzzolese et al. (2016) aggregates different linked datasets and heterogeneous sources. They apply their algorithm, Encyclopedic Knowledge Patterns, to compute the most relevant information from linked data and mainly use DBPedia to select and organize information. Klouche, Ruotsalo, and Jacucci (2018) suggests an idea of hypercues, which can substitute hyperlinks by adding entities information to support interactive entity-based search. Xu et al. (2015) makes use of its own relevance relations to extract semantic relations between word pairs from documents. It links entities extracted from crawled documents. On the other hand, Fafalios et al. (2017) uses RDF to match entities in RDF to elements in documents by adding RDF layers on documents.

2.1.2.3 Linked Data and Clustering Combined Approaches

In conjunction with the advancements of the document grouping and linked data approaches in exploratory search field, a few studies attempted to integrate the two approaches to enhance performance.

One way to integrate the two approaches is to use clustering using a knowledge graph. Rieh, Collins-Thompson, Hansen, and Lee (2016) attempt to cluster relations in the linked data and represents them as facets. The clustering works on similarity level and clusters are labeled central topics of clusters. The clustering algorithm merges and builds hierarchies on the knowledge graph based on a similarity measure.

Another way is to compute semantic similarity for clustering using a knowledge graph. Tutek, Glavas, Šnajder, Milić-Frayling, and Dalbelo Basic (2016) uses a knowledge graph to compute the similarity between documents seeing a document as a sub-graph of a knowledge graph. Rupasingha, Paik, and Kumara (2017) generates ontology from domain-specific corpora and uses it to calculate the similarity between a pair of documents.

2.2 Natural Language Vector Representation Models

This section links to the main contribution of this research, integrating linked data and state-of-the-art vector representations of natural language. The first part of the section examines the development of vector representation models of natural language based on texts. It shows how vector space language models have been developed to reflect on more contexts. The second subsection is a review of embedding linked data into vector space. The last part is a review of the integration of the first two approaches. It shows research on how to embed linked data to vector models trained on text datasets.

2.2.1 Word Embeddings from Text Data

A neural network language model for vector representations of natural language was proposed by Mikolov, Sutskever, Chen, Corrado, and Dean (2013) for the first time. Since then, the technology attracted a lot of attention from researchers and they have contributed to the development of vector representation of language model.

The first significant breakthrough of (Mikolov, Sutskever, et al., 2013) was making a probability language model using n-gram, skip-gram, and continuous bag-of-words (CBOW) methods. N-gram is to give neighboring words probability based on sequences. Skip-gram and CBOW are both prediction models for a natural language but the prediction is performed in the opposite direction. For skip-gram, the model takes an input word and generates surrounding words, while the CBOW model gives surrounding words so that the model outputs a predicted word that fits the context (Mikolov, Chen, Corrado, & Dean, 2013).

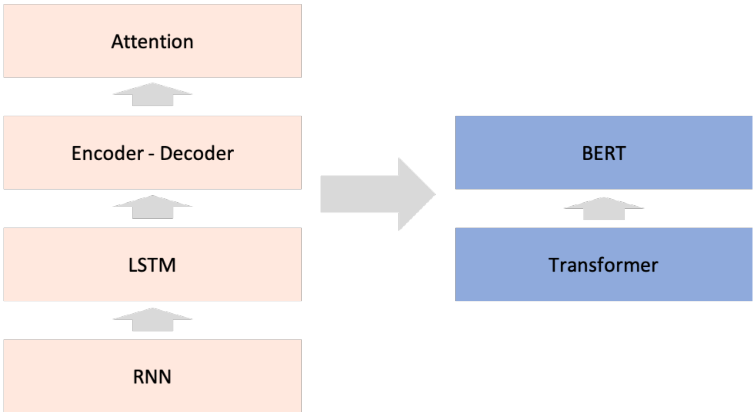


Figure 2.1. Development of Vector Representation of Language Model

The neural network has evolved together with the development of deep learning area. 2.1 shows how vector representation of language models has been developed. The motivation under the development is to reflect more contexts more efficiently. Starting from RNN, LSTM and Encoder-Decoder model further developed the concept of recurrent training on sequential data. The attention model was built upon that to make the model more efficiently remember important pieces of information. However, the transformer model was not in the linear path of the recurrent neural network. It only utilizes self-attention scheme to train a language model and BERT continues this approach recording the highest performance in many downstream NLP tasks.

Recurrent neural network (RNN) had changed the language model by looking back at all of the words in sentences. In RNN, each hidden layer consists of neurons that generate hidden states and outputs. Hidden states are dependent on previous steps. They capture information from earlier time steps to memorize information in long sequences. The hidden states are fed into the next layers. Although the RNN model can reflect more context by feeding a sentence as a sequence and remember more words than conventional neural networks, it still has vanishing gradient problem which means prediction performance reduces gradually as a sentence input becomes longer.

Long Short Term Memory (LSTM) model is a specially devised RNN model to resolve the vanishing gradient problem. The fundamental idea of LSTM is to optionally add and remove information along the path. LSTM sequentially input words in a sentence which makes it inevitable to have directions. Bi-LSTM puts two independent RNNs together to give sequence both backward and forward. Encoder-Decoder is another pivotal advancement that uses two recurrent neural networks, encoder and decoder (Cho et al., 2014). Encoder transforms sequential inputs into vectors and the produced vectors are fed into decoder again. The vectors are changed into a sequence of tokens passing through the decoder (Sutskever, Vinyals, & Le, 2014).

The attention model gives attention to each word by attention-weighted positions. In this way, the model can focus on content words that are more relevant to the sentences (Luong, Pham, & Manning, 2015). These methods send every produced position weighted context vector into the decoder. The decoder combines hidden states from the encoder to give positional weights to outputs.

The biggest disadvantage of the aforementioned models was that it takes too much time to produce and process sequential inputs. Also, the model cannot solve the dependencies in long sentences due to its sequential characteristics. To solve this problem, Vaswani et al. (2017) introduces the Transformer model insisting that recurrences are redundant and attention logic is all training needs. The Transformer model uses a self-attention structure that builds relations between words in a sentence which gives weights to an important word to predict the next word.

Bidirectional Encoder Representations from Transformers (BERT) is the state-of-the-art neural language model built upon the Transformer. BERT learns from bidirectional representations and can be fine-tuned for special purpose Devlin et al. (2019). However, BERT uses 12 layers of encoders to build a vector language model instead of using a decoder model to use the model for translation. BERT was trained on two tasks; masked language model and next sentence prediction. Unlike feature based learning, BERT can be fine-tuned upon pre-trained vectors for a specific task.

The outstanding performance of BERT raised the development of many variations. RoBERTa (Y. Liu et al., 2019) analyzed the effect of tuning key hyper-parameters when training the BERT model. The authors tested the BERT model with various modifications. Since BERT uses masked language model (MLM) and next sentence prediction (NSP) for training objects, (Y. Liu et al., 2019) re-train the model from scratch with different tactics: removing NSP or changing MLM masking pattern. They also tried training the model with longer batches, more data, and longer sequences. They conclude that modifying the BERT model to dynamic masking, applying longer and bigger batches, removing NSP loss objective can lead to significantly better performance.

Similar to RoBERTa paper, Raffel et al. (2019) worked on an empirical survey that examines when it comes to transfer learning, how changes in parameters can affect the performance of downstream NLP tasks. Transfer learning refers to fine-tuning self-supervised pre-trained models on smaller datasets. Language models like BERT can be used for transfer learning. The authors apply insights gained from these experiments to build Text-To-Text Transfer Transformer (T5) model. Their findings on the impact of architecture are that encoder-decoder models outperform encoder-only (BERT) or decoder-only models, the best performing pre-training object was a fill-in-the-blank-style loss. When it comes to fine-tuning parameters, all of the parameters should be tuned rather than changing a subset of them. Lastly, training a smaller model with larger data outperforms training a bigger model with smaller data.

Unlike the RoBERTa and T5 model that focused on improving model performance, ALBERT focuses more on improving training efficiency (Lan et al., 2020). Even though it is common sense that larger corpus can enhance downstream results, it is hard to increase the size of the corpus due to hardware limitations. This study addresses this problem by reducing parameters. ALBERT also uses 12 encoders as the BERT model but the encoders share parameters, unlike BERT. ALBERT also reduces parameters by dividing the vocabulary embedding matrix into two submatrices. It is said that ALBERT has 18x fewer parameters and can be trained 1.7x faster than its BERT counterpart.

ELECTRA (Clark, Luong, Le, & Manning, 2020) also slightly modify the original BERT pre-training object to reduce computation. ELECTRA introduces ’replaced token detection’ instead of MLM. They replace the tokens using detection instead of prediction. Two models, generator and discriminator, are used to implement this idea. The generator model is a small MLM and predicts what chosen tokens should be. Then the discriminator model predicts whether a token is an original token or replaced token. Using this architecture, while MLM only masks 15% of tokens, it can learn from 100% of tokens which significantly enhances model performance as well as reduce computing time.

2.2.2 Word Embeddings from Linked Data

There has been many research works on embedding entities in linked data into vector space in order to test entity classification and knowledge graph completion (Cai, Zheng, & Chang, 2018). The early works focus more on preserving the associations of entities and relations and transforming them into vector representation. The studies use unique learning models to minimize the global loss function of entities and relations.

(Bordes, Usunier, Garcia-Duran, Weston, & Yakhnenko, 2013) translates the triples of graph data into vector embeddings by putting the two entities close and other vectors that depend on the relation. The model, TransE, learns to minimize loss function that involves relations and entities. Other studies develop the idea by mapping relations to hyper-plane to better capture the properties (Bordes et al., 2013) or by separating entity vector space and relation vector space (Lin, Liu, Sun, Liu, & Zhu, 2015). He, Liu, Ji, and Zhao (2015) uses Gaussian distribution to represent the components of the triples and train the model.

Another approach representing a knowledge graph is to use semantic matching. Socher, Chen, Manning, and Ng (2013) predicts relations between entities by giving entities an average value of constituting word vectors. Yang, tau Yih, He, Gao, and Deng (2015) uses a scoring function that captures relations using bi-linear mapping. Ristoski, Rosati, Noia, Leone, and Paulheim (2019) specifically uses RDF database. The authors extract text-like corpus from the knowledge graph by traversing the graph and train skip-gram model on the corpus.

2.2.3 Integrating Word Embeddings of Text Data and Linked Data

Recently, a novel field of research to train knowledge graph together with the conventional word embeddings has attracted attention among researchers. This approach is to integrate knowledge into word embeddings and complement knowledge graphs considering a very comprehensive knowledge graph is far from completion. To integrate the two heterogeneous datasets, it is important to suggest a decent probabilistic model train the heterogeneous data or training strategy to feed linked data into the language model.

Some papers jointly train linked data entities and texts. Wang, Zhang, Feng, and Chen (2014) enables this by training the knowledge model, text model, and alignment model together. For the knowledge model, they slightly change the relation formula of TransE paper, (Bordes et al., 2013), to the probabilistic model to let the model learns to maximize the likelihood of triple facts. The text model uses a similar model to skip-gram model and the align model is the one that bridges the knowledge and text model. The authors utilize entity anchors by connecting a word from the Wikipedia page and a word from a linked data entity. They joint-learn the three models to maximize the likelihood function. Yamada, Shindo, Takeda, and Takefuji (2016) also uses the skip-gram model and jointly train the model with the knowledge graph model that will predict neighboring entities and anchor context model that will predict anchor words. The knowledge base model is slightly different in that the paper uses the likelihood of relatedness between entities as an objective function.

After the BERT language model was published and proven to have better performance in downstream NLP tasks, the graph embedding field started to adopt the BERT model and retrain the model with texts and knowledge graphs. Interestingly, Petroni et al. (2019) proves that BERT already captures a good amount of information from the knowledge graph, but many studies even improved the performance by retraining or fine-tuning the model with knowledge graph for named entity recognition tasks.

Z. Zhang et al. (2019) trains the BERT model by aggregating input of a sentence and entities found from the sentence. The Knowledgeable encoder integrates the entities' information into the textual information. The authors utilized the TransE method to transform linked data into the trainable probabilistic formula. Yin et al. (2019) fine-tunes BERT pre-trained model by giving a pair of sentences that consist of a sentence including entities and entities. Gillick et al. (2019) also takes a similar approach by feeding a pair of mention and entity into BERT-like dual encoder.

On the other hand, Yamada and Shindo (2019) uses the masking model from BERT. The model randomly masks entities and train the model to predict the entities. The input format is a pair of context and entities separated by separator token. Broscheit (2019) adds classifier layer on the top of the BERT model to compute the probability of linking a token of BERT sub-words to an entity in the knowledge base. W. Liu et al. (2020) insert knowledge into a sentence by designing a knowledge layer and feed a transformed sentence into the model. Yao, Mao, and Luo (2019) is the only paper that did not retrain BERT from scratch. The authors fine-tuned BERT pre-trained model with sequential tokens made from triples of the knowledge graph. The entity and relation were separated by the separator token.

2.2.4 Clustering

Clustering algorithms are mainly divided into three categories, agglomerative and hierarchical clustering, distance-based clustering, and graph clustering (Aggarwal & Zhai, 2012). The agglomerative way is to merge texts into clusters based on similarity. Also, merging texts into clusters and merging clusters into bigger clusters enable building cluster hierarchy from test data (Murtagh, 1983). BIRCH model (T. Zhang, Ramakrishnan, & Livny, 1996) stands for balanced iterative reducing and clustering using hierarchies. This model generates a summary of the information distribution, and then cluster them instead of the original hefty dataset.

The representative examples of distance-based clustering are K-means and DBSCAN algorithms. K-means algorithm finds k number of groups from a set by finding centroids of k clusters and the coherence of each cluster is calculated based on distances between objects (Lloyd, 1982). DBSCAN algorithm stands for the density-based spatial clustering of applications with noise. DBSCAN algorithm locates a cluster with high density which is separated from other clusters. DBSCAN does not require an input of the number of clusters and a large number of sample space (Ester, Kriegel, Sander, & Xu, 1996). One can also group documents into clusters using topic modeling methods. Topic modeling generates latent themes from a group of documents assuming that co-occurrences of similar words mean they have the same topic. Latent Dirichlet Allocation (LDA) model finds the topics of a document in an unsupervised way using a generative probabilistic model (Blei, Ng, & Jordan, 2003). Li, Kuo, and Lin (2011) used the LDA technique to cluster documents.

Recently, along with the emergence of the vector space model with abundant information embedded in matrices and deep learning techniques, a few pioneering research works utilizing autoencoders were introduced. DEC model (Xie, Girshick, & Farhadi, 2016) is a representative model among those. This model learns feature representations and cluster labels from deep neural networks. It learns to map a piece of high dimensional information into lower-dimensional vectors space and then optimize the clusters.

Previous works mostly leveraged TF-IDF vectors to represent documents numerically. However, recently, many studies are using a vector language model to substitute document pair semantic similarity calculation. Kutuzov and Kuzmenko (2016) adopts the classic semantic similarity-based clustering but uses neural embeddings to compute the similarity. The most recent achievements from the clustering field are Park et al. (2019). The authors directly use the vector representation from the language model to preserve the rich embeddings. They initialize clusters and update the centroids using cosine similarity between document pairs. The suggested clustering method showed better performance than other clustering methods.

CHAPTER 3. RESEARCH METHODOLOGY

This chapter describes the novel system intended to solve the disconnection between the two major exploratory search systems (i.e., clustering-based and linked data-based systems) and enhance clustering performance by employing state-of-the-art vector representation.

The chapter consists of the solution section which contains system design and component details and the experiment section which explains dataset and evaluation methods. The solution part focuses on the main contribution of this study: retrieving and organizing search results. An experimental setup follows to evaluate the performance of the suggested system. The setup includes a description of dataset and evaluation metrics.

3.1 System Design

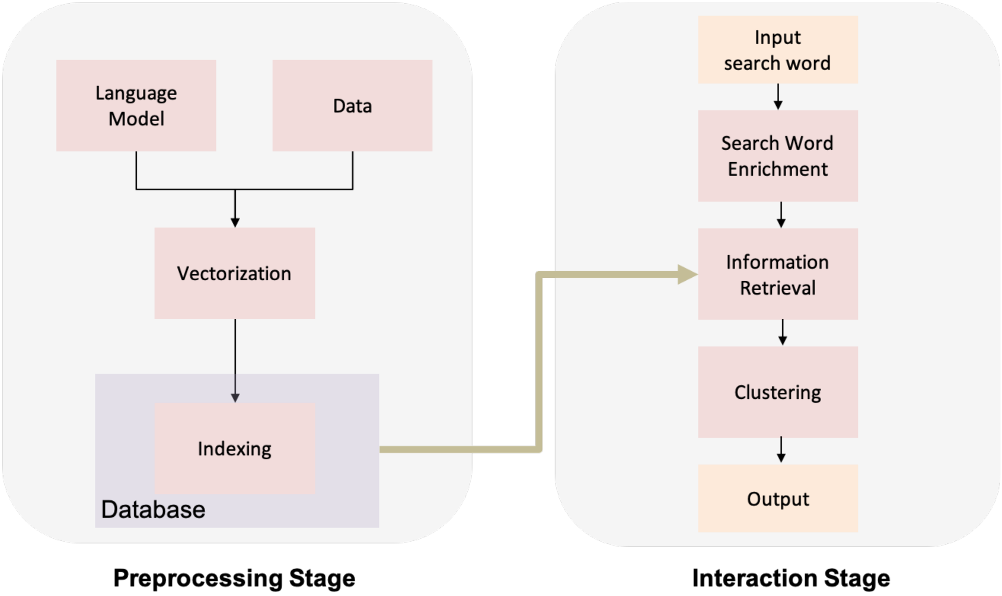


Figure 3.1. System Design of the Architecture

A typical search engine is composed of three main functions: crawling, indexing, and ranking. Crawling refers to robots exploring in World Wide Web finding new and updated web sites. Indexing crawled documents means machines processing and storing the documents into the engine's database. Finally, ranking refers to deciding priorities among retrieved results of a search word (Brin & Page, 1998). For this study, a prototype of an exploratory search engine is built with an emphasis on the retrieval stage to enhance explore search results.

Figure 3.1 describes the system architecture components of the suggested exploratory search engine. Crawling, language modeling and document indexing belong to the preprocessing stage. Unlike typical search engines, the novel system wants to expand the simple process of preprocessing to have exploratory search features. The novel system preprocesses documents by a new language model. The new language model is a crucial part of the system since the research aims to incorporate vector space and knowledge graphs to understand information more effectively. The interaction stage is composed of search words enriching, information retrieval, and clustering. Once a user inputs a search word, the system dynamically finds relevant documents, clusters retrieved items, and returns them. The vectorized documents by language model are clustered in this step.

3.2 Language Model

Bidirectional Encoder Representations from Transformers (BERT) is the most recent neural language representation model built upon the recent achievements of the NLP field, the attention model and the Transformer model. BERT learns from bidirectional representations and can be used for transfer learning (Devlin et al., 2019). BERT inherits the idea of the Transformer model written by Vaswani et al. (2017), but instead of using a decoder for translation purposes, BERT uses 12 layers of encoders.

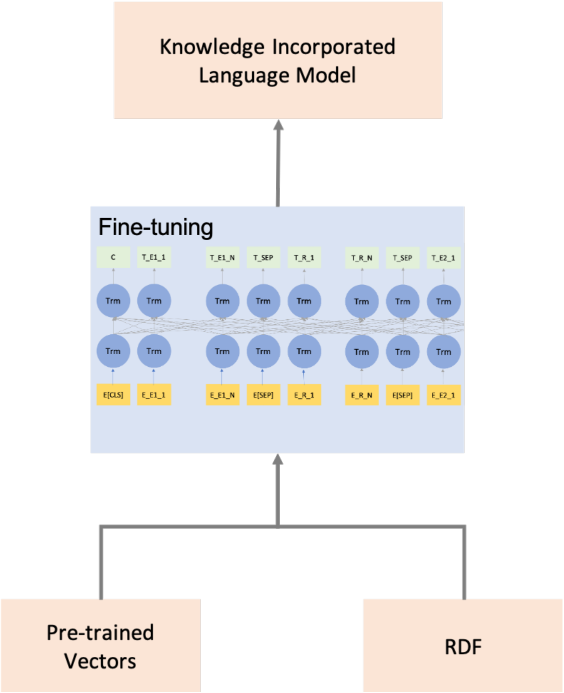


Figure 3.2. Fine-tuned BERT Language Model

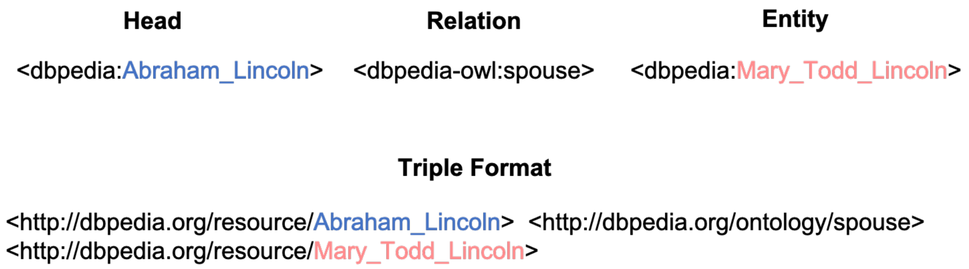
BERT was trained on two tasks: the masked language model (MLM) and the next sentence prediction (NSP). Unlike feature based learning, BERT can be fine-tuned upon pre-trained vectors for a specific task. BERT showed significant performance improvement in downstream NLP tasks. This research also uses BERT model vectorizing documents expecting state-of-the-art achievements.

Along with the development of vector representation of language models, another method to understand natural languages has emerged on the Internet. Semantic Web was initiated by Tim Berners Lee, the inventor of World Wide Web (Berners-Lee et al., 2001). The mission of the Semantic Web is to make resources on the Internet understandable to machines. Resource Description Framework (RDF) is a framework introduced to implement this idea (W3C, 1998). RDF is used to describe web sources and merge data with various underlying schemas. This concept is also called as ‘linked data’ since it represents connected data of the Internet.

DBPedia (Auer et al., 2007) and Wikidata (Vrandečić & Krötzsch, 2014) are the representative examples of comprehensive RDF databases. Out of available RDFs, DBPedia is the most comprehensive since it extracts entities and linkages from the Wikipedia corpus (Auer et al., 2007). The data consists of 1.3 billion English triples which contain 6.0 million entities including 2.0 million named entities. Figure 3.3 shows an example of RDF triple. An RDF triple consists of head entity, tail entity, and the relation between them. For this study, DBPedia 2016-10 dataset was used.

The main contribution of this novel language model is to re-train on the BERT pre-trained model with DBPedia linked data. By incorporating the vector represented language model and knowledge graph in the pre-existing RDF dataset, the new language model is expected to understand documents better than the original vector model and linked data. In other words, the new language model is expected to reflect both the contextual meaning of words from texts and human knowledge from linked data. Figure 3.2 shows the holistic view of fine-tuning process. The parameters in the last layer of the pre-trained BERT model will be adjusted while training the model on the new linked data dataset.

The key question raised within the integration of a vector model and a linked data is: how to encode the linked data into the vector space. This research adopts an idea from (Ristoski et al., 2019) to extract corpus from RDF using graph walks and Weisfeiler-Lehman Subtree RDF Graph Kernels. In this way, RDF triples will be transformed into a sequential words list maintaining the ”entity-relation-entity” order.



Head	Relation	Entity
<dbpedia:Abraham_Lincoln>	<dbpedia-owl:spouse>	<dbpedia:Mary_Todd_Lincoln>

Triple Format

<http://dbpedia.org/resource/Abraham_Lincoln> <http://dbpedia.org/ontology/spouse> <http://dbpedia.org/resource/Mary_Todd_Lincoln>

Figure 3.3. RDF Triple Example

For a given graph $G = (R, E)$ where R refers to relations and E refers to entities, the neighboring triples will be extracted into a list of triples by Breadth First Search (BFS) algorithm. When extracting corpus from linked data, neighboring triples should stay close to each other since BERT takes the order of sentences into account when training. This was the main reason why BFS algorithm was chosen instead of the Depth First Search (DFS) algorithm.

To extract and process a massive amount of data from the RDF dataset, the conventional BFS algorithm was modified to reflect this attribute. The BFS algorithm sees the immense DBPedia linked data as a forest, i.e., a large group of graphs. The algorithm visits a graph, tracks visited nodes in every graph, and move to a new graph when nodes in the visiting graph are exhausted. This tactic was used to ensure no single vertex is left without being visited. The graph structure was formed from DBPedia triples using Algorithm 1. Algorithm 2 is the pseudocode of the modified RDF traversing BFS algorithm.

Algorithm 1: Building RDF Graph

```
Input: $KnowledgeGraph$, $Triples$
Output: G = (V, E)
//initialization
Array $Triples$
foreach $OneTriple \in Triples$ do
    $HeadEntity$, $Relation$, $TailEntity$ = $OneTriple$.split()
    $HeadVertex$ = $KnowledgeGraph$.addVertex($HeadEntity$)
    $TailVertex$ = $KnowledgeGraph$.addVertex($TailEntity$)
    $KnowledgeGraph$.addEdge($HeadVertex$, $Relation$, $TailVertex$)
end
return $KnowledgeGraph$;
```

Algorithm 2: RDF Graph Walk algorithm

```
Input: $G = (V, E)$
Output: Array $triples$
//initialization
Array $Triples$, Queue $Q$, Set $Visited$, Set $Vertices$
$Q$.enqueue(random anchor)
while $Q$ do
    $HeadEntity \leftarrow Q$.dequeue()
    if $HeadEntity \in Vertices$ then
        $Vertices$.remove($HeadEntity$)
    end
    $Visited$.add($HeadEntity$)
    $Neighbors \leftarrow$ FindNeighbors($HeadEntity$)
    foreach $TailEntity \in Neighbors$ do
        $Triple \leftarrow (HeadEntity, Relation, TailEntity)$
        if $TailEntity \notin Visited$ then
            $Q$.enqueue($TailEntity$)
        end
    end
    if $not\ Q\ and\ Vertices$ then
        //current graph is exhausted
        $NewRoot$ = random.choice($Vertices$)
        $Q$.append($NewRoot$)
    end
end
```

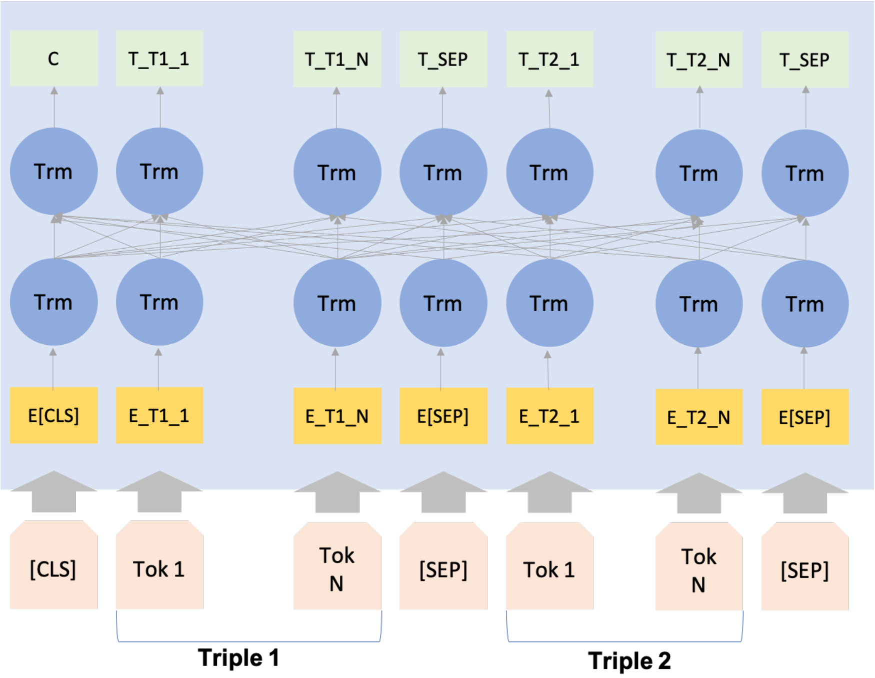


Figure 3.4. Fine-tuning Procedure

Figure 3.4 presents the detailed view of fine-tuning. The three components of triples are tokenized using the WordPiece algorithm and neighboring triples are separated by a separator token. The transformed sequence tokens will be fed into the BERT model together with the next sequence. The training details are same as the original BERT model; the model is trained to predict randomly masked words and the next sentence. The weights are updated by the negative sampling method. Negative random sampling allows the model to update a small number of weights rather than all of the weights for each sample.

The model aims to fine-tune the pre-trained model rather than re-train the whole model from scratch. Re-training from scratch is studied by a few researchers and showed good results in named entity recognition tasks (Broscheit, 2019; Yamada & Shindo, 2019). However, fine-tuning a model instead of re-training can maintain updating the model simple and fast when adding additional knowledge graphs in the future. Gillick et al. (2019) re-trained BERT vector from scratch, but re-training the model every time the knowledge graph expands is costly and not efficient.

3.3 Preprocessing Stage

The rest of the preprocessing stage is to collect available documents from the Internet, embed the documents into vector space, and store the information of the documents. The simplest way to embed a document is to use an average vector value of the final hidden layer of parameters as recommended by the authors of the BERT paper, (Devlin et al., 2019). This method gives a single vector representation of a document that enables to run numerical clustering models. This process retains the simplicity of the framework while preserving the original embeddings.

To elaborate the process, let the collected documents as a set $D = \{d_1, d_2, \ldots, d_n\}$ and n represents the number of documents. Each document, d_i, can be represented with the words composing it. $d_i = \{w_{i1}, w_{i2}, \ldots, w_{i_m}\}$ when m is the number of words in a document. Using the new language model, each word can be tokenized into a vector using the final hidden layer. The transformed tokens can be expressed as $d_i = \{Tok_{i1}, Tok_{i2}, \ldots, Tok_{i_m}\}$. A document is transformed into a matrix that contains the vector values of every word in the document. The vector values in the matrix take average to make one single value to represent the document. The final vector representation of *i*th document can be mathematically shown as $\frac{1}{m}\sum_{n=1}^{m}(Tok_{i_n})$.

The embeddings and the original texts of the collected documents are stored together. From the interaction step, the original texts are used to extract relevant documents to the user input and the embeddings are used for the clustering step.

3.4 Interaction Stage

In the interaction stage, the system interacts with the users' input words to retrieve search results. This stage consists of two steps, enriching input words and clustering search results. Enriching words is responsible for diversifying search results and clustering is for organizing the results. Clustering documents, the key feature of the exploratory research system, helps users explore and seek information.

3.4.1 Document Retrieval

Diversifying search results align with the purpose of the exploratory search to guarantee the diversity of search results and users exposed to many relevant topics they were unaware of Nuzzolese et al. (2016). While linked data-based exploratory search systems can diversify search results by expanding traversing routes, clustering-based search systems need an extra feature to implement the diversification.

In this novel system, the search results expand to not only the string matched documents to search words but also potentially relevant documents. Relevant documents can be found by enriching search words into a set of related/similar words. The enrichment method is inspired by classification paper written by Haj-Yahia, Sieg, and Deleris (2019). This study developed an enrichment method to aggregate relevant words of the classification category to capture the topic of documents. The enrichment method expands words using human knowledge, word embeddings, and knowledge graphs. Inheriting this idea, the novel system also enriches input words using knowledge graphs and word embeddings. Firstly, the input word is expanded using WordNet (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990) associated synonyms. Then, word embeddings are used to generate semantically similar words from vector space. Words were extracted from pre-trained Glove vector (Pennington, Socher, & Manning, 2014) based on cosine similarity. This study retrieves documents from the database using the enriched set of words. The output list of documents then streams into the clustering step.

3.4.2 Clustering

The study revisits the pioneering scatter/gather exploratory search framework (Cutting et al., 1992) to organize retrieved documents into clusters. The engine utilizes the preprocessed document embeddings to apply the numerical clustering model. Reaching the clustering stage, the system will have extracted relevant documents to the user input and had preprocessed data of each document. Once a document is represented as a vector value, documents can be clustered using any kind of numerical clustering models such as K-means, Gaussian Mixture Model (GMM), and BIRCH.

K-Means is one of the most frequently used clustering models. The model optimizes clusters by minimizing the average distances between points in a cluster. K-Means++ algorithm improves K-means clustering further by randomly seeding the centers of clusters (Arthur & Vassilvitskii, 2007). GMM uses a similar strategy to K-Means but it takes accounts for variance and distance. GMM also can perform soft clustering but for the research, only the hard clustering algorithm was used. BIRCH stands for balanced iterative reducing and clustering using hierarchies. The BIRCH model minimizes memory usage by summarizing the information in dense regions. The dense regions create a tree data structure and the cluster centroids are detached from the leaf. BIRCH is better than other models when processing large datasets.

3.5 Experiments

This study mainly evaluates the clustering performance of the system. In addition to clustering, experiments on the diversity of search results are conducted. The two areas of evaluation are selected to prove whether the new search engine can provide more organized and diverse search results to users.

To test these criteria of the novel system, three different experiments were tested. The first experiment measures how much the RDF tuned BERT model (RDF+BERT) enhances clustering performance than the original BERT model and the RDF model. For the second experiment, the clustering performance of the novel system is compared to a recent exploratory search study. Lastly, the third experiment is to evaluate the diversity of the system's search results. The novel system with and without enrichment logic are compared to each other.

The first experiment compares the clustering quality of its RDF+BERT model with the pre-trained BERT model (Devlin et al., 2019) and the RDF model inspired by (Ristoski & Paulheim, 2016). This experiment shows how adding linked data to the BERT embeddings affect the performance of downstream NLP tasks compared to the regular BERT model and the RDF embeddings. The three language models vectorize test datasets and the document vectors are clustered. To ensure the models are tested by datasets with various attributes (e.g., formal and informal English) and various clustering algorithms, the experiments are conducted with three clustering models, K-means, GMM, and BIRCH, and with five different datasets. The pre-trained BERT model used is BERT-Medium configuration which was trained with 8 hidden layers and 512 token sizes. The vocabulary size is 30522. The hardware specification the model was trained on is 8 Intel Xeon processors, with 30 GB RAM. Test results are based on three repetitive experiments. The experiments were repeated 10 times on the randomly divided datasets to validate the results.

The second experiment evaluates the new model compared against a recent clustering-based exploratory search engine, (Ortiz et al., 2019). The clustering system from (Ortiz et al., 2019) used the keyword discovery algorithm to make documents only contain important information. This method significantly reduces vector dimensions and noise in the next processes. Then they applied latent semantic analysis (LSA) to further reduce the dimension size and cluster the vectors using the k-means++ model. Since this baseline model used the k-means model, only the k-means model was tested other than GMM and BIRCH models. Test results are based on three repetitive experiments.

The last experiment evaluates whether the enrichment step in the retrieving stage can diversify search results. The diversity metric aims to measure the separation between clusters and homogeneity within clusters. Having these two criteria is to measure the separation of clusters from each other and coherence within clusters at the same time. To evaluate this, two kinds of exploratory search engines were prepared. They are built based on the architecture demonstrated from figure 3.1. However, one system has enrichment logic while the other runs without it. Once a search word is fed in, both systems retrieve the same number of search results from the database and cluster the results. From the clustered search results, the separation and homogeneity of clusters are measured. This process was repeated with four different input words.

This study interprets the separation between a pair of clusters as the distance between the centers of the clusters. Therefore, separation is measured by two distance metrics: cosine distance and Manhattan distance (i.e., L1 norm). These metrics work significantly better than Euclidean distance for high dimensional vector space (Aggarwal, Hinneburg, & Keim, 2001). Homogeneity is another important factor of diversity. This study assumes the homogeneity of a cluster corresponds to the density of plots in the cluster.

3.5.1 Dataset

Five different datasets were used to test clustering performance. Yahoo Answers and 20 News Group datasets are conversational texts and consist of relatively informal vocabulary. In contrast, Reuters, BBC, and AG News datasets are news articles. Table 3.1 shows the statistics of datasets and 3.2 shows an example text from each of the test dataset.

Table 3.1. *Dataset Statistics*

Dataset	Number of Texts	Number of Clusters
Yahoo Answers	10,000	10
20 News Group	18,000	20
Reuters	747	5
BBC	4,452	5
AG News	10,000	4

- **Yahoo Answers**: Yahoo test dataset originally has 60,000 texts, but for clustering performance, the texts were filtered out by length. The processed dataset includes 46806 question and answer sets which have more than 30 words. The texts are annotated with 10 classes including Society & Culture, Science & Mathematics, Health, Education &, and more (X. Zhang, Zhao, & LeCun, 2015). 10,000 texts were randomly extracted from the training and testing dataset.

Table 3.2. *Dataset example*

Dataset	Example
Yahoo Answers	”What makes friendship click?”,”How does the spark keep going?”,”good communication is what does it. Can you move beyond small talk and say what’s really on your mind. If you start doing this, my experience is that potentially good friends will respond or shun you. Then you know who the really good friends are.”
20 News Group	”I’ve got a very nice collection of historical books on medical quackery, and on the topic of massage this is a recurring theme. Ordinary massage is intended to make a person feel better, especially if they have muscular or joint problems. But – like chiropracty – there are some practitioners who take the technique to a far extreme, invoking what seems to me to be quack science to justify their technique.”
Reuters	”ZAMBIA DOES NOT PLAN RETAIL MAIZE PRICE HIKE The Zambian government has no immediate plans to follow last week ’ s increase in the producer price of maize with a hike in the retail price of maize meal , an official of the ruling party said . Last December , a 120 pct increase in the consumer price for refined maize meal , a Zambian staple , led to food riots in which at least 15 people died. ...”
BBC	”Time Warner said on Friday that it now owns 8 percent of search-engine Google. But its own internet business, AOL, had has mixed fortunes. It lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding three quarters. However, the company said AOL’s underlying profit before exceptional items rose 8 percent on the back of stronger internet advertising revenues. ...”
AG News	”Calif. Aims to Limit Farm-Related Smog (AP)”,”AP - Southern California’s smog-fighting agency went after emissions of the bovine variety Friday, adopting the nation’s first rules to reduce air pollution from dairy cow manure.”

- **20 News Group**: 20 News Group dataset is collected by Ken Lang. The dataset is a collection of conversation threads from newsgroup documents. The 18,000 threads are organized into 20 different topics in a hierarchy (e.g., comp.sys.ibm.pc.hardware and comp.sys.mac.hardware) which enables to classify them into subtopics.
- **Reuters**: The Reuters dataset is a part of Reuters-21578 dataset collected by Lewis (1997). Five topics that are frequent and mutually exclusive were chosen from the original dataset. 747 documents were chosen from the five topics: money-supply, grain, livestock, trade, and gold.
- **BBC**: BBC dataset is news articles released by BBC. It has 4,452 number of articles and the topic clusters include business, sport, entertainment, politics, and tech (Greene & Cunningham, 2006). The topic classes are divided and the average length of articles is relatively longer than other datasets.
- **AG News**: AG dataset is a collection of news articles from more than 2000 news sources collected by ComeToMyHead, an academic news search engine. The dataset consists of very short news texts from 4 different topics: world, sports, business, and sci/tech. The original corpus contains 120,000 training texts and 7,600 testing texts. 10,000 texts were randomly extracted from the training and testing dataset.

3.5.2 Evaluation

Clustering performance was evaluated by three different existing metrics, For diversity evaluation, a formula was designed to measure how much search results are diversified.

3.5.2.1 Clustering Evaluation Metrics

Clustering can be evaluated with a few different approaches. The simplest way to evaluate this is to compare the clustered results to the annotated class. ACC metric does this by matching predicted labels and ground-truth labels. In 3.1, y_i corresponds to the ground-truth label and c_i to the assigned cluster. M is a mapping function between the two labels. The most effective mapping function is the Hungarian algorithm (Min et al., 2018).

$$ACC = \max_{m} \frac{\sum_i (y_i = m(c_i))}{n} \tag{3.1}$$

Another approach is to use similarities between clusters. Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) and Adjusted Mutual Information (AMI) (Vinh, Epps, & Bailey, 2009) are the two most frequently used measures to evaluate cluster quality. AMI is based on information theory and ARI is based on a pair of objects counting. Adjusted Mutual Information(AMI) is more suitable to evaluate balanced clusters with a similar number of documents while the Adjusted Rand index (ARI) is more suitable to unbalanced clusters (Romano, Vinh, Bailey, & Verspoor, 2016). Both of them range from 0 to 1.

Rand Index (RI) measures similarity between two clustered data. The formula of RI is

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.2}$$

TP, TN, FP, and FN each represents true positive, true negative, false positive, and false negative (Rand, 1971). ARI corrects the Rand index for chance (Hubert & Arabie, 1985).

$$ARI = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex} \tag{3.3}$$

To put the equation differently, given two clusters $A = (A_1, A_2, ... , A_m)$, $B = (B_1, B_2, ... , B_n)$, n_{ij} refers $|A_i \cap B_j|$, a_i and b_j individually represents objects in A_i and B_j.

$$ARI(A,B) = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{N}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{N}{2}} \tag{3.4}$$

Mutual Information (MI) is a statistical measure of relatedness between a pair of objects. MI can be explained by entropy, the uncertainty of a random variable. The reduction in entropy of two random variable means it has higher mutual information (Cover & Thomas, 2001). Equation 3.5 represents entropy of random variable X, equation 3.6 represents mutual information formula. Normalized Mutual Information (NMI) equation 3.7 can be used to compare clusters with different numbers of clusters.

$$H(X) = -\sum_{x} p(x)\log_2 p(x) \tag{3.5}$$

$$I(X;Y) = H(X) - H(X|Y) = \sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{3.6}$$

$$NMI(X,Y) = \frac{I(A,B)}{\frac{1}{2}[H(A)+H(B)]} \tag{3.7}$$

3.5.2.2 Diversity Evaluation

Since there is no existing measure for diversity evaluation, a novel diversity measure for this research was devised. Let $C = \{c_1, c_2, \ldots, c_n\}$ are generated clusters and n represents the number of clusters. The centroids of each cluster can be expressed as $T = \{t_1, t_2, \ldots, t_n\}$ and each cluster is composed of points, $c_i = \{p_{i_1}, p_{i_2}, \ldots, p_{i_m}\}$. To measure separation in the clusters, it is assumed that the larger distance between the centroids of clusters means a higher degree of separation. The cosine distances and Manhattan distances between pairs of centers are summed and divided by the number of pairs. Let $Distance(t_i, t_j)$ refers to cosine distance between cluster i and j, distance value for a set of clusters can be computed as follows.

$$Distance(t_1, t_2) = 1 - \frac{t_1 \cdot t_2}{|t_1||t_2|} \tag{3.8}$$

$$Distance = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} Distance(i,j), (i \neq j)}{\binom{n}{2}} \tag{3.9}$$

This study interprets homogeneity as the density within clusters. Higher density refers to distances between the cluster elements and the center are small. The density of a cluster can be represented with summation of cosine similarity between the pair of a plot and the center. Mathematically this concept can be represented as follows:

$$Density(c_i) = \frac{\sum_{j=1}^{m}(\frac{c_i \cdot p_{i_j}}{|c_i||p_{i_j}|})}{m} \tag{3.10}$$

$$Density = \frac{\sum_{i=1}^{n} Density(c_i)}{n} \tag{3.11}$$

The final Diversity value is the product of overall distance and density. The distance is multiplied by 10 first since the values are smaller than 1 which makes the outcome of the product even smaller.

$$Diversity = 10 * Distance * Density \tag{3.12}$$

CHAPTER 4. RESULTS

This chapter presents the results of the three experiments introduced in the previous chapter. The clustering experiments (i.e., experiments 1 and 2) measure cluster accuracy of different language models. In these experiments, the language models vectorize test datasets and cluster the vectors using three different clustering models. The results show findings for RDF+BERT performance compared to the baseline models. The diversity experiment measures the separation and homogeneity of the retrieved clusters. The results show findings for how the enrichment step diversified search results.

4.1 Experiment 1. Clustering Compared to BERT and RDF Embeddings

The three metrics, ACC, NMI, and ARI, are calculated for five different datasets. The clustering results of the three models (i.e., BERT+RDF, BERT, RDF) are evaluated based on these three metrics. The datasets are listed in order of news articles (i.e., BBC and Reuters), conversational texts (i.e., 20 News Group and Yahoo Answers), and short news summary, AG news. The following plotted graphs show performance of each language model through 10 times of experiments. These graphs only contains K-Means clustering model and ACC measure since other values also show similar trends.

For simplicity, the new model is represented as ’RDF+BERT’ model. In the result tables, the underline indicates the best performing language model for each clustering model and the bold text indicates the best performing language model out of all clustering models.

4.1.1 News Articles: BBC and Reuters

The first two datasets are BBC and Reuters news articles. These datasets consist of news articles in formal written English which are expected to match with the corpus BERT was trained on. Table 4.1 shows mean values of experiment results for the BBC dataset and table 4.2 for the Reuters dataset.

Table 4.1. *Clustering Results - BBC*

Model	Clustering	ACC	NMI	ARI
RDF+BERT	K-means	0.892	0.701	0 701
	GMM	0.830	0.657	0.623
	BIRCH	0.889	0.704	0.702
BERT	K-means	**0.908**	**0.735**	**0.740**
	GMM	0.783	0.653	0.608
	BIRCH	0.876	0.692	0.681
RDF	K-means	0.520	0.209	0.170
	GMM	0.464	0.164	0.134
	BIRCH	0.504	0.193	0.152

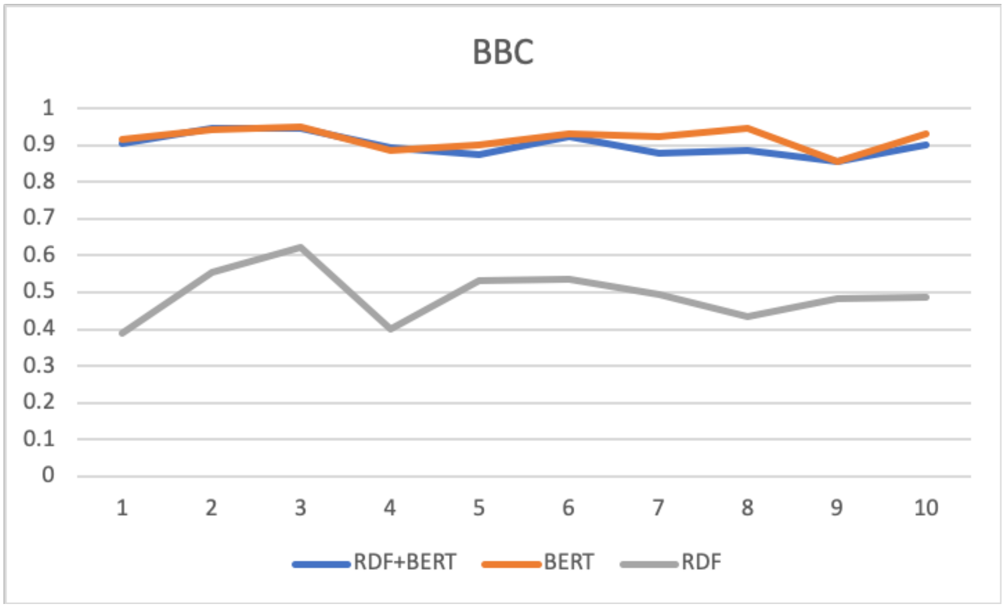


Figure 4.1. BBC

Table 4.2. *Clustering Results - Reuters*

Model	Clustering	ACC	NMI	ARI
RDF+BERT	K-means	**0.544**	0.407	0.328
	GMM	0.529	0.372	0.300
	BIRCH	0.537	0.392	0.293
BERT	K-means	0.559	**0.418**	0.329
	GMM	0.529	0.380	**0.300**
	BIRCH	0.555	0.409	0.320
RDF	K-means	0.366	0.148	0.088
	GMM	0.353	0.110	0.071
	BIRCH	0.351	0.144	0.064

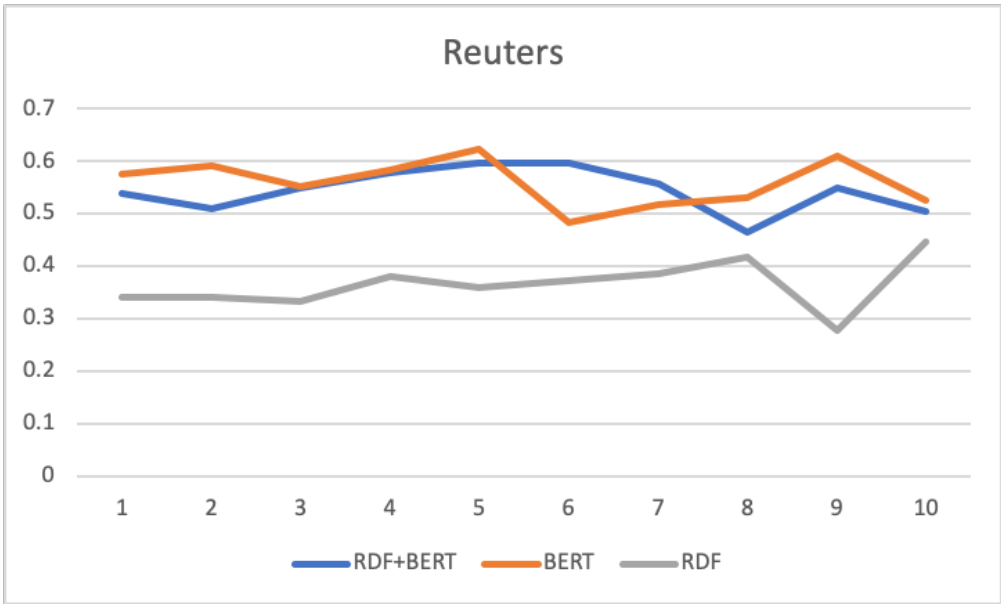


Figure 4.2. Reuters

The best clustering accuracy was from the BERT model. The GMM clustering result was higher for the RDF+BERT while all other highest values were from BERT. Figure 4.1 shows that the performance of the BERT model consistently similar to or better than the RDF+BERT model.

The Reuters dataset showed relatively lower accuracy than the BBC one due to high exclusivity among clusters in the BBC dataset. The Reuters dataset shows mixed signals. The K-means and BIRCH models show high accuracy of the RDF+BERT model; the global maximum NMI and ARI are drawn from the BERT model. Also, the gap between the maximum value and other values is small.

Overall, for news articles, RDF+BERT model did not enhance clustering performance. Both of the tables indicate the BERT model shows a similar or slightly higher performance than the RDF+BERT model.

4.1.2 Conversational Data: 20 News Group and Yahoo Answers

The three language models were also evaluated by conversational texts: 20 News Group and Yahoo Answers dataset. The most distinguishing properties of these two datasets are that they have less density of information and consist of relatively casual and diverse vocabulary. The example of these datasets can be seen in the dataset section in the methodology chapter.

Table 4.3. *Clustering Results - 20 News Groups*

Model	Clustering	ACC	NMI	ARI
RDF+BERT	K-means	**0.405**	0.425	0.244
	GMM	0.394	0.424	0.245
	BIRCH	0.397	**0.420**	**0.237**
BERT	K-means	0.396	0.419	0.237
	GMM	0.380	0.410	0.227
	BIRCH	0.391	0.412	0.221
RDF	K-means	0.143	0.068	0.024
	GMM	0.139	0.067	0.024
	BIRCH	0.138	0.065	0.019

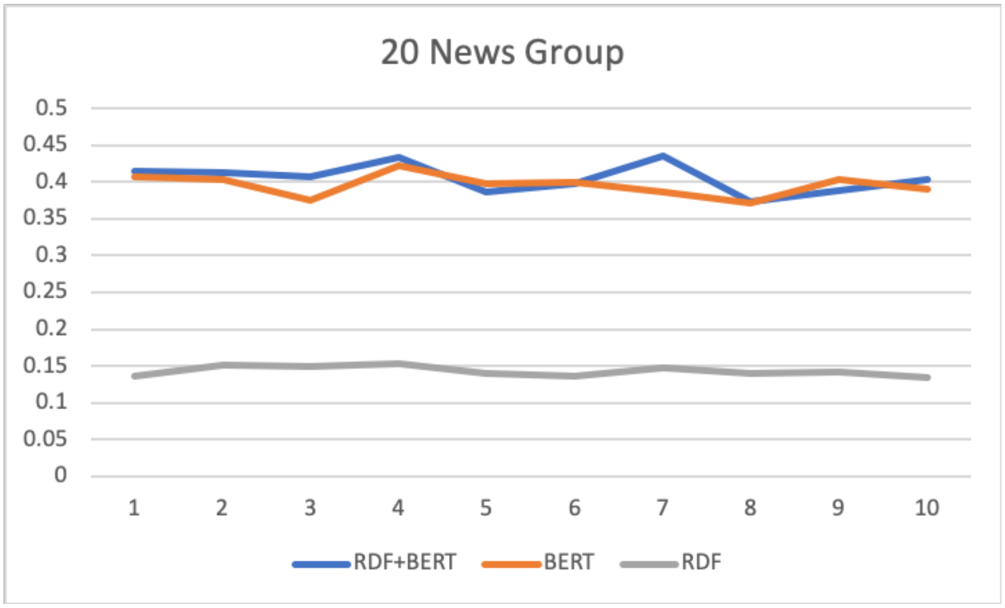


Figure 4.3. 20 News Group

Table 4.4. *Clustering Results - Yahoo Answers*

Model	Clustering	ACC	NMI	ARI
RDF+BERT	K-means	**0.432**	0.295	**0.204**
	GMM	0.422	0.292	0.197
	BIRCH	**0.406**	0.282	0.177
BERT	K-means	0.398	0.281	0.181
	GMM	0.394	0.278	0.179
	BIRCH	0.414	0.284	0.180
RDF	K-means	0.161	0.031	0.012
	GMM	0.163	0.031	0.013
	BIRCH	0.171	0.024	0.010

Table 4.3 shows the results for the 20 News Group and table 4.4 for the Yahoo Answers. Overall, clustering performance was lower than news articles datasets, but RDF+BERT slightly outperformed the baseline models: the BERT model and the RDF model.

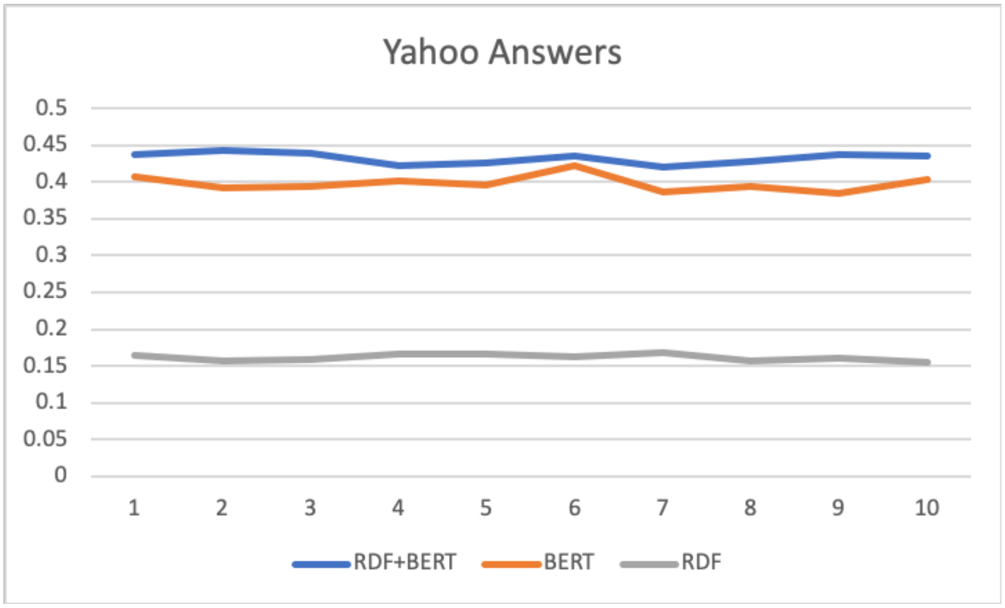


Figure 4.4. Yahoo Answers

4.1.3 Short News Articles: AG News

Finally, the three language models were evaluated by short news summary, AG News. The most distinguishing attribute of this test data is that each entry from AG News Dataset is very short, consisting of 2 to 5 sentences.

Table 4.5. *Clustering Results - AG News*

Model	Clustering	ACC	NMI	ARI
RDF+BERT	K-means	0.620	0.521	0.454
	GMM	**0.625**	0.517	0.465
	BIRCH	0.612	0.488	0.425
BERT	K-means	0.620	**0.528**	**0.457**
	GMM	0.623	0.519	0.466
	BIRCH	0.616	0.487	0.427
RDF	K-means	0.377	0.080	0.078
	GMM	0.362	0.076	0.073
	BIRCH	0.354	0.064	0.051

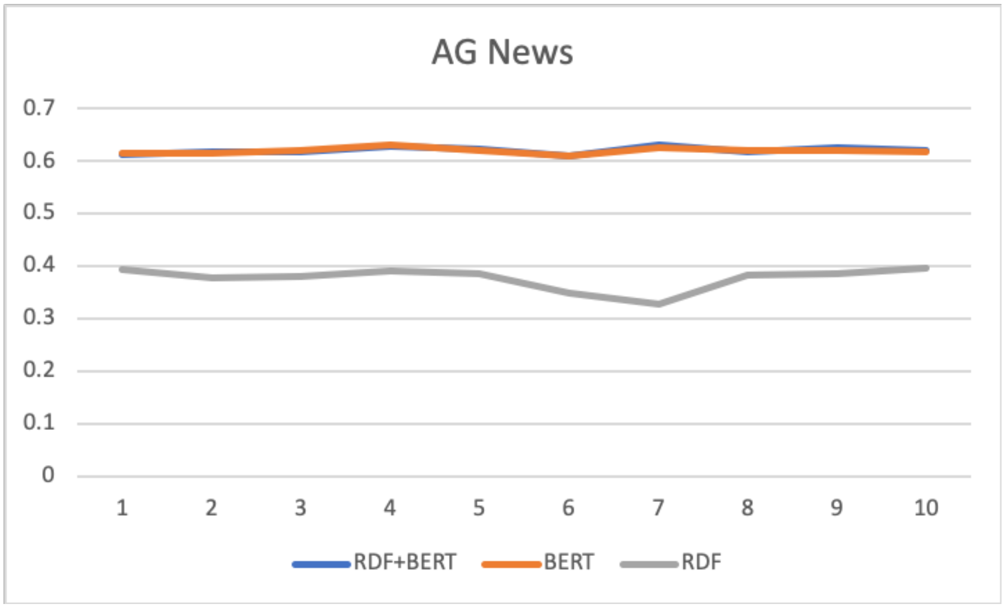


Figure 4.5. AG News

The results for the AG News test data are similar to that of the Reuters dataset. Overall, BERT baseline evaluation results showed very similar outcome. Figure 4.5 shows tuning the BERT model did not affect clustering short news articles. In general, the results showed higher numbers than conversational texts.

4.1.4 Discussion

While BERT showed slightly better performance with news dataset, the RDF+BERT model slightly outperformed BERT for conversational datasets. Since the original BERT embeddings were trained with the Wikipedia and BookCorpus(Zhu et al., 2015), the news articles should already be well represented with the BERT corpus. Figure 4.6 shows that the vocabulary of news datasets has much more overlaps with the vocabulary of Wikipedia than the conversational datasets do. Considering DBPedia RDFs are also extracted from Wikipedia corpus, the RDF model showing significantly higher performance with news articles dataset than with a conversational dataset is natural. When the BERT model already contains the vast majority of words from a certain dataset, there can be little room for improvement.

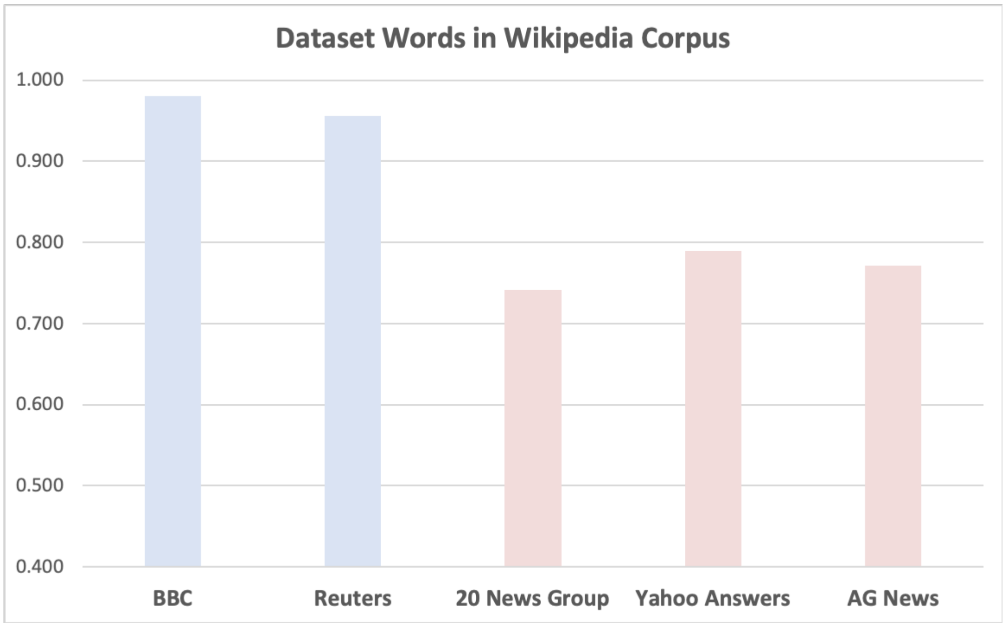


Figure 4.6. Vocabulary of Dataset Overlaps with Wikipedia Corpus

The experiment results prove that when texts consist of more informal, casual, and diverse words, the RDF+BERT model slightly outperformed the BERT model. This improvement demonstrates that incorporating human knowledge with the BERT model increased the capability to understand texts with these characteristics. Considering the RDF+BERT model shows consistent performance with every dataset, the RDF+BERT model is more suitable to build a general-purpose exploratory search engine.

4.2 Experiment 2. Clustering Compared to Clustering Exploratory Search Baseline

This section shows the clustering results of the RDF+BERT model compared to that of an exploratory search baseline, (Ortiz et al., 2019). Overall, the baseline model shows better performance with news articles, while the novel system shows far superior performance with conversational texts and short news articles.

4.2.1 News Articles: BBC and Reuters

Table 4.6. *Clustering Results - BBC*

Model	ACC	NMI	ARI
RDF+BERT	0.924	0.790	0.823
Baseline	**0.954**	**0.863**	**0.893**

Table 4.7. *Clustering Results - Reuters*

Model	ACC	NMI	ARI
RDF+BERT	0.573	0.404	0.327
Baseline	**0.589**	**0.486**	**0.407**

Table 4.6 shows the clustering results for the BBC dataset. The baseline method outperformed the RDF+BERT model. Table 4.7 shows the clustering results for the Reuters dataset. The baseline method slightly outperformed RDF+BERT. The gap between the baseline and RDF+BERT was similar to that of BBC dataset results while the overall values are lower.

4.2.2 Conversational Data: 20 News Group and Yahoo Answers

Table 4.8. *Clustering Results - 20 News Group*

Model	ACC	NMI	ARI
RDF+BERT	**0.428**	**0.424**	**0.256**
Baseline	0.330	0.355	0.165

Table 4.9. *Clustering Results - Yahoo Answers*

Model	ACC	NMI	ARI
RDF+BERT	**0.368**	**0.240**	**0.151**
Baseline	0.320	0.172	0.091

For the conversational dataset, the RDF+BERT system significantly outperformed the baseline while the overall results are lower than those of news articles. Table 4.8 shows the clustering results for the 20 News Group and table 4.9 for Yahoo Answers. For both datasets, the RDF+BERT method shows significantly better accuracy than the baseline.

4.2.3 Short News Articles: AG News

Table 4.10. *Clustering Results - AG News*

Model	ACC	NMI	ARI
RDF+BERT	**0.808**	**0.542**	**0.567**
Baseline	0.4048	0.1899	0.05393

The table 4.10 shows the clustering results for AG News dataset. RDF+BERT method shows slightly better performance than the baseline. The low performance of the baseline method is due to the logic that utilizes term frequency which requires longer texts to have higher clustering accuracy.

4.2.4 Discussion

The RDF+BERT model showed significantly better performance for conversational texts and short texts while the baseline excelled in tests with news articles. Frequency-based embeddings slightly outperformed the competitor for news articles that address similar topic and vocabulary but was significantly underperformed for texts with diverse vocabulary (i.e., conversational texts) or shorter texts.

4.3 Diversity Evaluation

For the diversity experiment, four random words were streamed into the system as input words. To simulate the search results with various sizes of the outcome, the test results were retrieved multiple times with two changing parameters: the number of texts and the number of clusters.

The cluster results have a varying number of texts, 100, 500, 2000, and 5000, and clusters, 5, 10, 20, and 20. Each table shows cosine distances and L1 distances between the center of clusters, density within the clusters, and the final diversity value which is the product of cosine distance and density.

Table 4.11. *Diversity Results - Test 1*

		Cosine Dist		L1 Dist		Density		Diversity	
Texts	Clusters	Enrich	No	Enrich	No	Enrich	No	Enrich	No
100	5	0.108	0.062	86.247	65.375	0.928	0.940	**1.004**	0.579
500	10	0.094	0.058	78.984	63.033	0.919	0.938	**0.866**	0.540
2000	20	0.103	0.067	82.843	67.034	0.915	0.939	**0.945**	0.627
5000	20	0.101	0.069	81.303	67.023	0.911	0.937	**0.919**	0.643

Out of four experiment results, only the table 4.11 shows a big difference in both distance and density. The system with enrichment has better separation between clusters while plots in the clusters are more scattered. For the enriched system, the average cosine distance is larger by 0.38 and Manhattan distance by 16.728. However, the system without enriching had a higher density by approximately 0.021. In turn, the overall diversity of the enriching system improved diversity by 0.34.

Table 4.12. *Diversity Results - Test 2*

		Cosine Dist		L1 Dist		Density		Diversity	
Texts	Clusters	Enrich	No	Enrich	No	Enrich	No	Enrich	No
100	5	0.092	0.052	80.418	62.540	0.916	0.935	**0.841**	0.482
500	10	0.082	0.079	76.638	73.936	0.919	0.923	**0.755**	0.732
2000	20	0.088	0.075	79.713	73.121	0.921	0.923	**0.811**	0.696
5000	20	0.087	0.103	78.122	85.948	0.916	0.929	0.796	**0.954**

Table 4.13. *Diversity Results - Test 3*

		Cosine Dist		L1 Dist		Density		Diversity	
Texts	Clusters	Enrich	No	Enrich	No	Enrich	No	Enrich	No
100	5	0.094	0.120	80.566	86.984	0.939	0.936	0.884	**1.120**
500	10	0.099	0.081	80.712	72.961	0.937	0.940	**0.930**	0.764
2000	20	0.121	0.104	87.690	80.425	0.930	0.936	**1.124**	0.977
5000	20	0.119	0.098	86.820	78.479	0.926	0.930	**1.104**	0.916

Table 4.14. *Diversity Results - Test 4*

		Cosine Dist		L1 Dist		Density		Diversity	
Texts	Clusters	Enrich	No	Enrich	No	Enrich	No	Enrich	No
100	5	0.099	0.077	82.431	72.362	0.917	0.919	**0.912**	0.712
500	10	0.110	0.078	85.764	72.807	0.915	0.920	**1.005**	0.718
2000	20	0.116	0.103	87.375	81.998	0.918	0.921	**1.064**	0.951
5000	20	0.110	0.126	84.070	89.605	0.913	0.923	**1.001**	1.166

In contrast, table 4.12, 4.13, and 4.14 showed slightly different results. They present a meaningful difference between enriching and no enriching in separation while showing no major impact on density. For the enriched system, the average cosine distance is larger by 0.18 to 0.21 and Manhattan distance by 7.4 to 8.7. However, the system without enriching had a higher density by approximately 0.000 to 0.003. In turn, this improved diversity measure by 0.16 to 0.19.

4.3.1 Discussion

Table 4.11 to 4.14 clearly demonstrate that exploratory search system with enrichment logic shows higher diversity. In most cases, the enrichment logic ensured separation between clusters and did not have a substantially negative impact on the density of clusters. The results prove that the novel system significantly improves the diversity of search results by adding the enrichment step.

The results also show distances between pairs of cluster centers and the density of clusters can be in a trade-off relation. The degree of separation and density moved together. The bigger separation meant the smaller homogeneity within clusters. When the gap in separation was smaller, the difference in density was also small (e.g., see table 4.12, 4.13, and 4.14). In contrast, when the differences in separation were bigger, the difference in density was also big (e.g., see table 4.11).

Overall, the range of density values is smaller than cosine distance values. This difference in range makes the diversity measure more useful since diversity needs to put emphasis on distances between clusters than the density of each cluster. There was no conspicuous correlation of diversity to the number of texts and the number of clusters of the clustered dataset.

CHAPTER 5. CONCLUSION

This study suggests a novel language model that embeds comprehensive linked data (i.e., human knowledge) into the state-of-the-art language model (i.e., BERT) in order to understand search results better and present them in a cluster framework for exploratory searchers.

Three experiments were conducted to evaluate the efficiency of the RDF+BERT system. The study tested the new language model by clustering annotated datasets and measuring the diversity of retrieved search results. Overall, the novel system showed achievements compared to the baseline language models with regards to clustering accuracy and diversity of retrieved search results.

When clustering texts are informal, unstructured, and short, tuning BERT with knowledge linked data helped the embeddings understand the texts better. Although the improvement was not substantially big, the potential of integrating linked data and language models is expected to be worth studied further.

Besides, enriching input words diversified search results without hurting the cohesiveness of each result group. By prioritizing diversification in exploratory search results, searchers will be able to exposed to related concepts to the search words and expand their learning landscape further.

5.1 Future Work

In spite of these achievements, there are a few issues that need to be examined and studied further. The experiments result raised a question: how is the vocabulary in training corpus and test dataset relevant to the clustering performance. Conducting a more comprehensive study on how and why the linked data affect the clustering performance is necessary, especially with regards to the characteristics of corpus and test data. For this study, comparing the clustering results of RDF+BERT and BERT embeddings can present mixed signals since BERT is pre-trained on Wikipedia and BookCorpus and the DBPedia dataset is also extracted from Wikipedia.

Recognizing that variations of the BERT model, such as RoBERTa, ALBERT, ELECTRA, and T5, took the top spot on the benchmarks, similar experiments should be adapted to these new models to observe whether exploratory search capability is even more enhanced. Reduction in training time will be another significant factor when it comes to usability since some of these studies focus on enhancing training efficiency.

Lastly, since the DBPedia linked data is general and comprehensive, how domain-specific linked data can enhance clustering performance is another important research area. Tuning model parameters with additional domain-specific linked data can be assumed to give more capability to understand documents on that topic.

REFERENCES

Abbasi, M. K., & Frommholz, I. (2015, March). Cluster-based polyrepresentation as science modelling approach for information retrieval. *Scientometrics*, *102*(3), 2301–2322. Retrieved 2020-02-03, from `http://link.springer.com/10.1007/s11192-014-1478-1` doi: 10.1007/s11192-014-1478-1

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. *Lecture Notes in Computer Science*, 420–434.

Aggarwal, C. C., & Zhai, C. X. (2012). A survey of text clustering algorithms. *Mining Text Data*, 77–128.

Angelova, R., & Siersdorfer, S. (2006). A neighborhood-based approach for clustering of linked document collections. In *Proceedings of the 15th ACM international conference on Information and knowledge management - CIKM '06* (p. 778). Arlington, Virginia, USA: ACM Press. Retrieved 2020-02-03, from `http://portal.acm.org/citation.cfm?doid=1183614.1183726` doi: 10.1145/1183614.1183726

Arthur, D., & Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms* (pp. 1027–1035).

Athukorala, K., Głowacka, D., Jacucci, G., Oulasvirta, A., & Vreeken, J. (2016, November). Is exploratory search different? A comparison of information search behavior for exploratory and lookup tasks. *Journal of the Association for Information Science and Technology*, *67*(11), 2635–2651. Retrieved 2020-01-30, from `http://doi.wiley.com/10.1002/asi.23617` doi: 10.1002/asi.23617

Athukorala, K., Medlar, A., Oulasvirta, A., Jacucci, G., & Glowacka, D. (2016). Beyond Relevance: Adapting Exploration/Exploitation in Information Retrieval. In *Proceedings of the 21st International Conference on Intelligent User Interfaces - IUI '16* (pp. 359–369). Sonoma, California, USA: ACM Press. Retrieved 2020-01-30, from `http://dl.acm.org/citation.cfm?doid=2856767.2856786` doi: 10.1145/2856767.2856786

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. In D. Hutchison et al. (Eds.), *The Semantic Web* (Vol. 4825, pp. 722–735). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved 2020-02-06, from `http://link.springer.com/10.1007/978-3-540-76298-0_52` doi: 10.1007/978-3-540-76298-0_52

Aula, A., Khan, R. M., & Guan, Z. (2010). How does search behavior change as search becomes more difficult? In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10* (p. 35). Atlanta, Georgia, USA: ACM Press. Retrieved 2020-02-03, from `http://portal.acm.org/citation.cfm?doid=1753326.1753333` doi: 10.1145/1753326.1753333

Aula, A., & Russell, D. M. (2007). *Complex and exploratory web search.*

Bascur, J. P., van Eck, N. J., & Waltman, L. (2019). An interactive visual tool for scientific literature search: Proposal and algorithmic specification. *BIR@ECIR*, 76–87.

Bates, M. J. (1989, May). The design of browsing and berrypicking techniques for the online search interface. *Online Review*, *13*(5), 407–424. Retrieved 2020-01-30, from `https://www.emerald.com/insight/content/doi/10.1108/eb024320/full/html` doi: 10.1108/eb024320

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, *284*(5), 34–43.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Bloom, B. S., & Krathwohl, D. R. (1966). Taxonomy of educational objectives. handbook i: Cognitive domain. Retrieved from `https://academic.microsoft.com/paper/1622896514`

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems 26* (pp. 2787–2795).

Brin, S., & Page, L. (1998, April). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, *30*(1-7), 107–117. Retrieved 2020-02-03, from `https://linkinghub.elsevier.com/retrieve/pii/S016975529800110X` doi: 10.1016/S0169-7552(98)00110-X

Broscheit, S. (2019). Investigating Entity Knowledge in BERT with Simple Neural End-To-End Entity Linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)* (pp. 677–685). Hong Kong, China: Association for Computational Linguistics. Retrieved 2020-02-06, from `https://www.aclweb.org/anthology/K19-1063` doi: 10.18653/v1/K19-1063

Cai, H., Zheng, V. W., & Chang, K. C.-C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, *30*(9), 1616–1637.

Case, D. O. (Ed.). (2012). *Looking for information: a survey of research on information seeking, needs and behavior* (3ed ed.). Bingley: Emerald. (OCLC: 795008062)

Cheng, G., Zhang, Y., & Qu, Y. (2014). Explass: Exploring Associations between Entities via Top-K Ontological Patterns and Facets. In P. Mika et al. (Eds.), *The Semantic Web – ISWC 2014* (Vol. 8797, pp. 422–437). Cham: Springer International Publishing. Retrieved 2020-02-03, from `http://link.springer.com/10.1007/978-3-319-11915-1_27` doi: 10.1007/978-3-319-11915-1_27

Chi, Y., He, D., Han, S., & Jiang, J. (2018). What Sources to Rely on:: Laypeople's Source Selection in Online Health Information Seeking. In *Proceedings of the 2018 Conference on Human Information Interaction&Retrieval - CHIIR '18* (pp. 233–236). New Brunswick, NJ, USA: ACM Press. Retrieved 2020-02-10, from `http://dl.acm.org/citation.cfm?doid=3176349.3176881` doi: 10.1145/3176349.3176881

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). Doha, Qatar: Association for Computational Linguistics. Retrieved 2020-02-12, from `http://aclweb.org/anthology/D14-1179` doi: 10.3115/v1/D14-1179

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. In *Iclr 2020 : Eighth international conference on learning representations.*

Cover, T. M., & Thomas, J. A. (2001). *Elements of information theory*. Hoboken, N.J.: Wiley-Liss. (OCLC: 641886411)

Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey, J. W. (1992, August). Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. *ACM SIGIR Forum*, *51*(2), 148–159. Retrieved 2020-01-30, from `http://dl.acm.org/citation.cfm?doid=3130348.3130362` doi: 10.1145/3130348.3130362

David, & Kosala, R. R. (2018, August). Clustering Algorithm Comparison of Search Results Documents. In *2018 6th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1–6). Parapat, Indonesia: IEEE. Retrieved 2020-02-03, from `https://ieeexplore.ieee.org/document/8674246/` doi: 10.1109/CITSM.2018.8674246

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. Retrieved 2020-02-06, from `http://arxiv.org/abs/1810.04805` (arXiv: 1810.04805)

Di Marco, A., & Navigli, R. (2011). Clustering Web Search Results with Maximum Spanning Trees. In R. Pirrone & F. Sorbello (Eds.), *AI*IA 2011: Artificial Intelligence Around Man and Beyond* (Vol. 6934, pp. 201–212). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved 2020-01-30, from `http://link.springer.com/10.1007/978-3-642-23954-0_20` doi: 10.1007/978-3-642-23954-0_20

Di Marco, A., & Navigli, R. (2013, September). Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction. *Computational Linguistics*, *39*(3), 709–754. Retrieved 2020-02-03, from `http://www.mitpressjournals.org/doi/10.1162/COLI_a_00148` doi: 10.1162/COLI_a_00148

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 1996 int. conf. knowledg discovery and data mining (kdd '96)* (pp. 226–231).

Fafalios, P., Holzmann, H., Kasturia, V., & Nejdl, W. (2017, June). Building and Querying Semantic Layers for Web Archives. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)* (pp. 1–10). Toronto, ON, Canada: IEEE. Retrieved 2020-02-03, from `http://ieeexplore.ieee.org/document/7991555/` doi: 10.1109/JCDL.2017.7991555

Foster, A., & Ford, N. (2003, June). Serendipity and information seeking: an empirical study. *Journal of Documentation*, *59*(3), 321–340. Retrieved 2020-02-10, from `https://www.emerald.com/insight/content/doi/10.1108/00220410310472518/full/html` doi: 10.1108/00220410310472518

Gillick, D., Kulkarni, S., Lansing, L., Presta, A., Baldridge, J., Ie, E., & García-Olano, D. (2019). Learning dense representations for entity retrieval. In *Proceedings of the 23rd conference on computational natural language learning (conll)* (pp. 528–537).

Greene, D., & Cunningham, P. (2006). Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd international conference on machine learning (icml'06)* (pp. 377–384). ACM Press.

Haj-Yahia, Z., Sieg, A., & Deleris, L. A. (2019). Towards Unsupervised Text Classification Leveraging Experts and Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 371–379). Florence, Italy: Association for Computational Linguistics. Retrieved 2020-02-24, from `https://www.aclweb.org/anthology/P19-1036` doi: 10.18653/v1/P19-1036

Hall, M., Clough, P., & Stevenson, M. (2012). Evaluating the Use of Clustering for Automatically Organising Digital Library Collections. In D. Hutchison et al. (Eds.), *Theory and Practice of Digital Libraries* (Vol. 7489, pp. 323–334). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved 2020-02-03, from `http://link.springer.com/10.1007/978-3-642-33290-6_35` doi: 10.1007/978-3-642-33290-6_35

He, J., Huang, Y., Liu, C., Shen, J., Jia, Y., & Wang, X. (2016, December). Text Network Exploration via Heterogeneous Web of Topics. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)* (pp. 99–106). Barcelona, Spain: IEEE. Retrieved 2020-02-03, from `http://ieeexplore.ieee.org/document/7836653/` doi: 10.1109/ICDMW.2016.0022

He, S., Liu, K., Ji, G., & Zhao, J. (2015). Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th acm international on conference on information and knowledge management* (pp. 623–632).

Hubert, L., & Arabie, P. (1985, December). Comparing partitions. *Journal of Classification*, *2*(1), 193–218. Retrieved 2020-02-24, from `http://link.springer.com/10.1007/BF01908075` doi: 10.1007/BF01908075

Kang, R., & Fu, W.-T. (2010). Exploratory information search by domain experts and novices. In *Proceedings of the 15th international conference on Intelligent user interfaces - IUI '10* (p. 329). Hong Kong, China: ACM Press. Retrieved 2020-01-30, from `http://dl.acm.org/citation.cfm?doid=1719970.1720023` doi: 10.1145/1719970.1720023

Kejriwal, M., & Szekely, P. (2019). Knowledge Graphs for Social Good: An Entity-centric Search Engine for the Human Trafficking Domain. *IEEE Transactions on Big Data*, 1–1. Retrieved 2020-02-03, from `https://ieeexplore.ieee.org/document/8068229/` doi: 10.1109/TBDATA.2017.2763164

Klouche, K., Ruotsalo, T., & Jacucci, G. (2018). From Hyperlinks to Hypercues: Entity-Based Affordances for Fluid Information Exploration. In *Proceedings of the 2018 on Designing Interactive Systems Conference 2018 - DIS '18* (pp. 401–411). Hong Kong, China: ACM Press. Retrieved 2020-02-03, from `http://dl.acm.org/citation.cfm?doid=3196709.3196775` doi: 10.1145/3196709.3196775

Kules, B., Capra, R., Banta, M., & Sierra, T. (2009). What do exploratory searchers look at in a faceted search interface? In *Proceedings of the 2009 joint international conference on Digital libraries - JCDL '09* (p. 313). Austin, TX, USA: ACM Press. Retrieved 2020-02-03, from `http://portal.acm.org/citation.cfm?doid=1555400.1555452` doi: 10.1145/1555400.1555452

Kutuzov, A., & Kuzmenko, E. (2016). Neural embedding language models in semantic clustering of web search results. In *Lrec*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In *Iclr 2020 : Eighth international conference on learning representations*.

Lewis, D. D. (1997). Reuters-21578 text categorization test collection, distribution 1.0.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, *5*, 361–397.

Li, C.-H., Kuo, B.-C., & Lin, C.-T. (2011). Lda-based clustering algorithm and its application to an unsupervised feature extraction. *IEEE Transactions on Fuzzy Systems*, *19*(1), 152–163.

Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Aaai'15 proceedings of the twenty-ninth aaai conference on artificial intelligence* (pp. 2181–2187). Retrieved from `https://academic.microsoft.com/paper/2184957013`

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., & Wang, P. (2020). K-bert: Enabling language representation with knowledge graph. In *Aaai 2020 : The thirty-fourth aaai conference on artificial intelligence*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, *28*(2), 129–137.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*. Retrieved from `https://academic.microsoft.com/paper/2949335953`

Ma, C., & Zhang, B. (2018). A new query recommendation method supporting exploratory search based on search goal shift graphs. *IEEE Transactions on Knowledge and Data Engineering*, *30*(11), 2024–2036.

Marchionini, G. (2006, April). Exploratory search: from finding to understanding. *Communications of the ACM*, *49*(4), 41. Retrieved 2020-01-30, from `http://portal.acm.org/citation.cfm?doid=1121949.1121979` doi: 10.1145/1121949.1121979

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Iclr (workshop poster).*

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems 26* (pp. 3111–3119).

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An On-line Lexical Database *. *International Journal of Lexicography*, *3*(4), 235–244. Retrieved 2020-02-24, from `https://academic.oup.com/ijl/article-lookup/doi/10.1093/ijl/3.4.235` doi: 10.1093/ijl/3.4.235

Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., & Long, J. (2018). A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, *6*, 39501–39514.

Mirizzi, R., Ragone, A., Di Noia, T., & Di Sciascio, E. (2010). Semantic Wonder Cloud: Exploratory Search in DBpedia. In F. Daniel & F. M. Facca (Eds.), *Current Trends in Web Engineering* (Vol. 6385, pp. 138–149). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved 2020-02-06, from `http://link.springer.com/10.1007/978-3-642-16985-4_13` doi: 10.1007/978-3-642-16985-4_13

Mohajeri, S., Samuel, H. W., Zalane, O. R., & Rafiei, D. (2016, April). BubbleNet: An innovative exploratory search and summarization interface with applicability in health social media. In *2016 International Conference on Digital Economy (ICDEc)* (pp. 37–44). Carthage, Tunisia: IEEE. Retrieved 2020-02-06, from `http://ieeexplore.ieee.org/document/7563143/` doi: 10.1109/ICDEC.2016.7563143

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, *26*(4), 354–359.

Navrat, P. (2012, January). Cognitive traveling in digital space: from keyword search through exploratory information seeking. *Open Computer Science*, *2*(3). Retrieved 2020-01-30, from http://www.degruyter.com/view/j/comp.2012.2.issue-3/s13537-012-0024-6/s13537-012-0024-6.xml doi: 10.2478/s13537-012-0024-6

Nuzzolese, A. G., Presutti, V., Gangemi, A., Peroni, S., & Ciancarini, P. (2016, November). Aemoo: Linked Data exploration based on Knowledge Patterns. *Semantic Web*, *8*(1), 87–112. Retrieved 2020-02-03, from https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/SW-160222 doi: 10.3233/SW-160222

Ortiz, M. S., Kim, H., Wang, M., Seki, K., & Mostafa, J. (2019). Dynamic Cluster-based Retrieval and Discovery for Biomedical Literature. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics - BCB '19* (pp. 390–396). Niagara Falls, NY, USA: ACM Press. Retrieved 2020-02-23, from http://dl.acm.org/citation.cfm?doid=3307339.3342191 doi: 10.1145/3307339.3342191

Ortiz, M. S., Seki, K., & Mostafa, J. (2018). Toward exploratory search in biomedicine: Evaluating document clusters by mesh as a semantic anchor. *arXiv preprint arXiv:1812.02129*.

Palagi, E. (2018). Evaluating exploratory search engines : designing a set of user-centered methods based on a modeling of the exploratory search process.

Panopto. (2018, Jul). *Panopto workplace knowledge and productivity report.* Author. Retrieved from https://www.panopto.com/about/news/inefficient-knowledge-sharing-costs-large-businesses-47-million-per-year/

Park, J., Park, C., Kim, J., Cho, M., & Park, S. (2019, December). ADC: Advanced document clustering using contextualized representations. *Expert Systems with Applications*, *137*, 157–166. Retrieved 2020-02-11, from https://linkinghub.elsevier.com/retrieve/pii/S0957417419304762 doi: 10.1016/j.eswa.2019.06.068

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019). Language models as knowledge bases. In *2019 conference on empirical methods in natural language processing* (pp. 2463–2473).

Pirolli, P., & Card, S. (1995). Information foraging in information access environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95* (pp. 51–58). Denver, Colorado, United States: ACM Press. Retrieved 2020-01-30, from `http://portal.acm.org/citation.cfm?doid=223904.223911` doi: 10.1145/223904.223911

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Rand, W. M. (1971, December). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, *66*(336), 846–850. Retrieved 2020-02-24, from `http://www.tandfonline.com/doi/abs/10.1080/01621459.1971.10482356` doi: 10.1080/01621459.1971.10482356

Rieh, S. Y., Collins-Thompson, K., Hansen, P., & Lee, H.-J. (2016, February). Towards searching as a learning process: A review of current perspectives and future directions. *Journal of Information Science*, *42*(1), 19–34. Retrieved 2020-01-19, from `http://journals.sagepub.com/doi/10.1177/0165551515615841` doi: 10.1177/0165551515615841

Ristoski, P., & Paulheim, H. (2016). Rdf2vec: Rdf graph embeddings for data mining. In *International semantic web conference (1)* (pp. 498–514). Retrieved from `https://academic.microsoft.com/paper/2523679382`

Ristoski, P., Rosati, J., Noia, T. D., Leone, R. D., & Paulheim, H. (2019). Rdf2vec: Rdf graph embeddings and their applications. *Social Work*, *10*(4), 721–752. Retrieved from `https://academic.microsoft.com/paper/2888634710`

Romano, S., Vinh, N. X., Bailey, J., & Verspoor, K. (2016). Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research*, *17*(1), 4635–4666.

Rose, D. E., & Levinson, D. (2004). Understanding user goals in web search. In *Proceedings of the 13th conference on World Wide Web - WWW '04* (p. 13). New York, NY, USA: ACM Press. Retrieved 2020-01-19, from `http://portal.acm.org/citation.cfm?doid=988672.988675` doi: 10.1145/988672.988675

Rupasingha, R. A. H. M., Paik, I., & Kumara, B. T. G. S. (2017, June). Improving Web Service Clustering through a Novel Ontology Generation Method by Domain Specificity. In *2017 IEEE International Conference on Web Services (ICWS)* (pp. 744–751). Honolulu, HI, USA: IEEE. Retrieved 2020-02-03, from `http://ieeexplore.ieee.org/document/8029831/` doi: 10.1109/ICWS.2017.134

Sabou, M., Ekaputra, F. J., Ionescu, T., Musil, J., Schall, D., Haller, K., … Biffl, S. (2018). Exploring Enterprise Knowledge Graphs: A Use Case in Software Engineering. In A. Gangemi et al. (Eds.), *The Semantic Web* (Vol. 10843, pp. 560–575). Cham: Springer International Publishing. Retrieved 2020-02-03, from `http://link.springer.com/10.1007/978-3-319-93417-4_36` doi: 10.1007/978-3-319-93417-4_36

Sanderson, M., & Croft, W. B. (2012, May). The History of Information Retrieval Research. *Proceedings of the IEEE*, *100*(Special Centennial Issue), 1444–1451. Retrieved 2020-01-30, from `http://ieeexplore.ieee.org/document/6182576/` doi: 10.1109/JPROC.2012.2189916

Savolainen, R. (2018, October). Berrypicking and information foraging: Comparison of two theoretical frameworks for studying exploratory search. *Journal of Information Science*, *44*(5), 580–593. Retrieved 2020-01-30, from `http://journals.sagepub.com/doi/10.1177/0165551517713168` doi: 10.1177/0165551517713168

Savolainen, R., & Kari, J. (2004, September). Placing the Internet in information source horizons. A study of information seeking by Internet users in the context of self-development. *Library & Information Science Research*, *26*(4), 415–433. Retrieved 2020-02-12, from `https://linkinghub.elsevier.com/retrieve/pii/S0740818804000520` doi: 10.1016/j.lisr.2004.04.004

Sherkhonov, E., Cuenca Grau, B., Kharlamov, E., & Kostylev, E. V. (2017). Semantic Faceted Search with Aggregation and Recursion. In C. d'Amato et al. (Eds.), *The Semantic Web – ISWC 2017* (Vol. 10587, pp. 594–610). Cham: Springer International Publishing. Retrieved 2020-02-06, from `http://link.springer.com/10.1007/978-3-319-68288-4_35` doi: 10.1007/978-3-319-68288-4_35

Singer, G., Danilov, D., & Norbisrath, U. (2012). Complex search: aggregation, discovery, and synthesis. *Proceedings of the Estonian Academy of Sciences*, *61*(2), 89. Retrieved 2020-01-30, from `http://www.kirj.ee/?id=20506&tpl=1061&c_tpl=1064` doi: 10.3176/proc.2012.2.02

Soares, V. H. A., Campello, R. J. G. B., Nourashrafeddin, S., Milios, E., & Naldi, M. C. (2019, December). Combining semantic and term frequency similarities for text clustering. *Knowledge and Information Systems*, *61*(3), 1485–1516. Retrieved 2020-02-03, from http://link.springer.com/10.1007/s10115-018-1278-7 doi: 10.1007/s10115-018-1278-7

Socher, R., Chen, D., Manning, C. D., & Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems 26* (pp. 926–934).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems 27* (pp. 3104–3112). Retrieved from https://academic.microsoft.com/paper/2130942839

Tibau, M., Siqueira, S. W. M., Nunes, B. P., Bortoluzzi, M., & Marenzi, I. (2018). Modeling exploratory search as a knowledge-intensive process. In *2018 ieee 18th international conference on advanced learning technologies (icalt)* (pp. 34–38).

Tutek, M., Glavas, G., Šnajder, J., Milić-Frayling, N., & Dalbelo Basic, B. (2016). Detecting and Ranking Conceptual Links between Texts Using a Knowledge Base. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM '16* (pp. 2077–2080). Indianapolis, Indiana, USA: ACM Press. Retrieved 2020-02-03, from http://dl.acm.org/citation.cfm?doid=2983323.2983913 doi: 10.1145/2983323.2983913

Tzitzikas, Y., Manolis, N., & Papadakos, P. (2017, April). Faceted exploration of RDF/S datasets: a survey. *Journal of Intelligent Information Systems*, *48*(2), 329–364. Retrieved 2020-01-30, from http://link.springer.com/10.1007/s10844-016-0413-8 doi: 10.1007/s10844-016-0413-8

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In *Nips'17 proceedings of the 31st international conference on neural information processing systems* (pp. 6000–6010). Retrieved from https://academic.microsoft.com/paper/2963403868

Vinh, N. X., Epps, J., & Bailey, J. (2009). Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09* (pp. 1–8). Montreal, Quebec, Canada: ACM Press. Retrieved 2020-02-24, from http://portal.acm.org/citation.cfm?doid=1553374.1553511 doi: 10.1145/1553374.1553511

Vrandečić, D., & Krötzsch, M. (2014, September). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, *57*(10), 78–85. Retrieved 2020-02-06, from `http://dl.acm.org/citation.cfm?doid=2661061.2629489` doi: 10.1145/2629489

W3C. (1998, July). *https://www.w3.org/TR/1998/WD-rdf-syntax-19980720/.* Author. Retrieved 2020-02-19, from `https://www.w3.org/TR/1998/WD-rdf-syntax-19980720/`

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014). Knowledge Graph and Text Jointly Embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1591–1601). Doha, Qatar: Association for Computational Linguistics. Retrieved 2020-02-06, from `http://aclweb.org/anthology/D14-1167` doi: 10.3115/v1/D14-1167

White, R. W., & Roth, R. A. (2009, January). Exploratory Search: Beyond the Query-Response Paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, *1*(1), 1–98. Retrieved 2020-01-30, from `http://www.morganclaypool.com/doi/abs/10.2200/S00174ED1V01Y200901ICR003` doi: 10.2200/S00174ED1V01Y200901ICR003

Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., & Heer, J. (2016, January). Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics*, *22*(1), 649–658. Retrieved 2020-01-30, from `http://ieeexplore.ieee.org/document/7192728/` doi: 10.1109/TVCG.2015.2467191

Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *Icml'16 proceedings of the 33rd international conference on international conference on machine learning - volume 48* (pp. 478–487).

Xu, Z., Wei, X., Luo, X., Liu, Y., Mei, L., Hu, C., & Chen, L. (2015, February). Knowle: A semantic link network based system for organizing large scale online news events. *Future Generation Computer Systems*, *43-44*, 40–50. Retrieved 2020-02-06, from `https://linkinghub.elsevier.com/retrieve/pii/S0167739X14000636` doi: 10.1016/j.future.2014.04.002

Yamada, I., & Shindo, H. (2019). Pre-training of deep contextualized embeddings of words and entities for named entity disambiguation. *arXiv preprint arXiv:1909.00426*.

Yamada, I., Shindo, H., Takeda, H., & Takefuji, Y. (2016). Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* (pp. 250–259). Berlin, Germany: Association for Computational Linguistics. Retrieved 2020-02-06, from `http://aclweb.org/anthology/K16-1025` doi: 10.18653/v1/K16-1025

Yang, B., tau Yih, W., He, X., Gao, J., & Deng, L. (2015). Embedding entities and relations for learning and inference in knowledge bases. In *Iclr 2015 : International conference on learning representations 2015.*

Yang, H. (2015, March). Browsing Hierarchy Construction by Minimum Evolution. *ACM Transactions on Information Systems*, *33*(3), 1–33. Retrieved 2020-01-30, from `http://dl.acm.org/citation.cfm?doid=2737814.2714574` doi: 10.1145/2714574

Yao, L., Mao, C., & Luo, Y. (2019). Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Yin, X., Huang, Y., Zhou, B., Li, A., Lan, L., & Jia, Y. (2019). Deep Entity Linking via Eliminating Semantic Ambiguity With BERT. *IEEE Access*, *7*, 169434–169445. Retrieved 2020-02-06, from `https://ieeexplore.ieee.org/document/8911323/` doi: 10.1109/ACCESS.2019.2955498

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). Birch: an efficient data clustering method for very large databases. In *Proceedings of the 1996 acm sigmod international conference on management of data* (Vol. 25, pp. 103–114).

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Nips'15 proceedings of the 28th international conference on neural information processing systems - volume 1* (pp. 649–657).

Zhang, Y., Broussard, R., Ke, W., & Gong, X. (2014, May). Evaluation of a scatter/gather interface for supporting distinct health information search tasks: Evaluation of a Scatter/Gather Interface for Supporting Distinct Health Information Search Tasks. *Journal of the Association for Information Science and Technology*, *65*(5), 1028–1041. Retrieved 2020-02-11, from `http://doi.wiley.com/10.1002/asi.23011` doi: 10.1002/asi.23011

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 1441–1451). Florence, Italy: Association for Computational Linguistics. Retrieved 2020-02-06, from `https://www.aclweb.org/anthology/P19-1139` doi: 10.18653/v1/P19-1139

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 ieee international conference on computer vision (iccv)* (pp. 19–27).