2021

Neural Representations of Concepts and Texts for Biomedical Information Retrieval

Jiho Noh
University of Kentucky, bornoriginal1@gmail.com
Author ORCID Identifier:
https://orcid.org/0000-0001-5734-9068
Digital Object Identifier: https://doi.org/10.13023/etd.2021.094

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Jiho Noh, Student

Dr. Ramakanth Kavuluru, Major Professor

Dr. Zongming Fei, Director of Graduate Studies

NEURAL REPRESENTATIONS OF CONCEPTS AND TEXTS FOR BIOMEDICAL INFORMATION RETRIEVAL

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the College of Engineering
at the University of Kentucky

By
Jiho Noh
Lexington, Kentucky

Director: Dr. Ramakanth Kavuluru, Associate Professor of Computer Science
Lexington, Kentucky

2021

https://orcid.org/0000-0001-5734-9068

ABSTRACT OF DISSERTATION

Neural Representations of Concepts and Texts
for Biomedical Information Retrieval

Information retrieval (IR) methods are an indispensable tool in the current landscape of exponentially increasing textual data, especially on the Web. A typical IR task involves fetching and ranking a set of documents (from a large corpus) in terms of relevance to a user's query, which is often expressed as a short phrase. IR methods are the backbone of modern search engines where additional system-level aspects including fault tolerance, scale, user interfaces, and session maintenance are also addressed. In addition to fetching documents, modern search systems may also identify snippets within the documents that are potentially most relevant to the input query. Furthermore, current systems may also maintain preprocessed structured knowledge derived from textual data as so called knowledge graphs, so certain types of queries that are posed as questions can be parsed as such; a response can be an output of one or more named entities instead of a ranked list of documents (e.g., "what diseases are associated with EGFR mutations?"). This refined setup is often termed as question answering (QA) in the IR and natural language processing (NLP) communities.

In biomedicine and healthcare, specialized corpora are often at play including research articles by scientists, clinical notes generated by healthcare professionals, consumer forums for specific conditions (e.g., cancer survivors network), and clinical trial protocols (e.g., www.clinicaltrials.gov). Biomedical IR is specialized given the types of queries and the variations in the texts are different from that of general Web documents. For example, scientific articles are more formal with longer sentences but clinical notes tend to have less grammatical conformity and are rife with abbreviations. There is also a mismatch between the vocabulary of consumers and the lingo of domain experts and professionals. Queries are also different and can range from simple phrases (e.g., "COVID-19 symptoms") to more complex implicitly fielded queries (e.g., "chemotherapy regimens for stage IV lung cancer patients with ALK mutations"). Hence, developing methods for different configurations (corpus, query type, user type) needs more deliberate attention in biomedical IR.

Representations of documents and queries are at the core of IR methods and retrieval methodology involves coming up with these representations and matching

queries with documents based on them. Traditional IR systems follow the approach of keyword based indexing of documents (the so called inverted index) and matching query phrases against the document index. It is not difficult to see that this keyword based matching ignores the semantics of texts (synonymy at the lexeme level and entailment at phrase/clause/sentence levels) and this has lead to dimensionality reduction methods such as latent semantic indexing that generally have scale-related concerns; such methods also do not address similarity at the sentence level. Since the resurgence of neural network methods in NLP, the IR field has also moved to incorporate advances in neural networks into current IR methods. This dissertation presents four specific methodological efforts toward improving biomedical IR. Neural methods always begin with dense embeddings in $\mathbb{R}^d$ for words and concepts to overcome the limitations of one-hot encoding in traditional NLP/IR. In the first effort, we present a new neural pre-training approach to jointly learn word and concept embeddings for downstream use in applications. In the second study, we present a joint neural model for two essential subtasks of information extraction (IE): named entity recognition (NER) and entity normalization (EN). Our method detects biomedical concept phrases in texts and links them to the corresponding semantic types and entity codes. These first two studies provide essential tools to model textual representations as compositions of both surface forms (lexical units) and high level concepts with potential downstream use in QA. In the third effort, we present a document reranking model that can help surface documents that are likely to contain answers (e.g, factoids, lists) to a question in a QA task. The model is essentially a sentence matching neural network that learns the relevance of a candidate answer sentence to the given question parametrized with a bilinear map. In the fourth effort, we present another document reranking approach that is tailored for precision medicine use-cases. It combines neural query-document matching and faceted text summarization. The main distinction of this effort from previous efforts is to pivot from a query manipulation setup to transforming candidate documents into pseudo-queries via neural text summarization. Overall, our contributions constitute nontrivial advances in biomedical IR using neural representations of concepts and texts.

KEYWORDS: Information Retrieval, Natural Language Processing, Deep Neural Networks, Information Extraction, Text Summarization, Question Answering

Author's signature: Jiho Noh

Date: May 5, 2021

Neural Representations of Concepts and Texts
for Biomedical Information Retrieval

By
Jiho Noh

Director of Dissertation: Ramakanth Kavuluru

Director of Graduate Studies: Zongming Fei

Date: May 5, 2021

Dedicated to my parents, Myeong-wan Noh and Cha-suk Lee, my wife, Dasom Lee.

ACKNOWLEDGMENTS

The work in this dissertation was performed under the direction of Dr. Ramakanth Kavuluru, my research advisor and committee chairman. I am most grateful to him for his continuous support and guidance. I have learned the essential methodologies for research from him that will benefit me in my future career. His thoroughness and attention to detail have been significantly helpful in every aspect of this research.

I am indebted to my committee members: Dr. Raphael Finkel, Dr. Qiang Ye, and Dr. Brent Harrison, for their kindest support and encouragement. Also, I thank Dr. Licong Cui for her participation in my committee for my Ph.D. proposal. Their comments and discussions have been incorporated into this work.

I would also want to express particular appreciation to my wife, Dasom, for her unconditional support and patience. Without her love and encouragement, I would not be able to finish this long academic journey. Furthermore, I owe my children, Mincheol (Mike) and Yuncheol (Eric), an apology for not being a full-time father. Finally, I want to thank my parents, Myeong-wan Noh and Cha-suk Lee, for their endless support over the years. This work would not have happened if it were not for these people I mentioned above, and I am forever thankful.

I gratefully share any credits that this study may receive with all the contributors, including anyone I could not mention above, and I am solely responsible for any deficiencies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

1.1 Challenges in Information Retrieval

The history of information retrieval (IR) can be traced back to much earlier than the Internet era. As a part of library science, the techniques of document indexing started at least two thousand years ago. Even before the day-to-day use of search engines, many research efforts in IR were found in different communities as early as the 1960s. Since then and up until recent years, the successes of IR were primarily built upon the use of the so called *bag-of-words* representations. Although computationally efficient and reasonably effective for building basic systems, *bag-of-words* offers a shallow 'understanding' of textual content.

Document retrieval (including Web documents) search engines are the most visible IR applications. These systems take a query, find relevant documents from the corpus, and rank them in an attempt to order by the relevance to the posed query. The query can be *ad hoc* (a set of words/phrases) or can sometimes be posted in the form of a question. Following is an example query and a relevant document.

```
Query:
  How to treat sciatic pain in pregnancy?
Document:
  Treatments for sciatic pain during pregnancy include massage,
  chiropractic care, and physical therapy. Self-treatment of sciatic
  pain during pregnancy includes exercises to help stretch the muscles
  of the leg, buttocks, and hip to decrease the pressure on the sciatic
  nerve. Some people also find nonweight-bearing exercises, such as
  swimming, to be helpful. This is because the water helps to support
  the weight of your baby.
```

Modern IR systems use the *bag-of-words* model to represent queries and documents. An input word sequence is converted into a vector of pre-defined index terms of words. Each dimension of the vector corresponds to the frequency of the corresponding word in the text. For example, the above query sentence can be converted into a vector `{treat: 1, sciatic: 1, pain: 1, pregnancy: 1}` excluding stopwords and punctuations. With this *index-term*-based information, a ranking method such as TF-IDF (Term Frequency-Inverse Document Frequency) measures the relevance of a document to the given query. TF-IDF is still the most popular method

in modern search engines; 83% of the textual resource recommender systems in digital libraries use some variants of the TF-IDF models [Beel et al., 2016]. In spite of the method's popularity, the ranking methods based on the term-level statistics have limitations including as follows:

- The model relies on the assumption that the statistics of word frequency can represent the theme/meaning of a query or document; In fact, the relevance between a query and a document is only partially measured by a proxy similarity measure using the bag-of-words representations.
- The retrieval quality heavily depends on how effectively the query is formed in terms of the methods used by the term-frequency based search engine. When a query contains aspects that hinder programs from capturing the intended meaning, we get lower precision and recall scores, and hence lower overall performance. Issues typically arise from polysemy (e.g., bodily discharge vs. hospital discharge), homonymy (e.g., cricket the sport as opposed to the insect), and synonymy (e.g., Georgia vs. peach state).

Nonetheless, the *index-term*-based IR systems are popular due to their efficiency of determining document relevance especially when leveraging auxiliary information (e.g., page popularity). This implies that there is room to improve the retrieval quality by utilizing the semantic aspects of natural language from unstructured free-texts.

The notion of *semantic search* is to retrieve relevant items by modeling the searcher's intent and the contextual meaning of the documents in the searchable data space. Some of the research objectives for the semantic search are listed below:

- handling generalization
- handling morphological variations
- handling concept/knowledge matching — going from terms to entities
- handling natural language questions (interrogative sentences)
- ability to find the most relevant sentence or text snippet in a document
- ability to utilize the metadata of document such as provenance or publication date

For biomedical IR, in particular, there exist additional challenges.

- Biomedical documents typically contain longer and more complex sentences with long range dependencies between different entities/words that may be discussed in them. To some extent, this has to do with the inherent complexity in conveying biomedical phenomena and also with the idiosyncrasies that maybe prevalent among medical professionals. It becomes increasingly difficult to map such sentences to vector representations without losing information.

- Prominent issues arising from polysemy, homonymy, and synonymy (including abbreviations) for entities such as genes, proteins, organisms, chemicals, and diseases add to compositional ambiguity when vectorizing queries and documents.

Due to the aforementioned challenges and the potential lack of IR system familiarity in particular domains, it becomes extremely difficult for general users to get the most relevant documents on the first querying attempt. For example, third-year medical students need to submit, on average, 14 separate queries to get the desired information using a typical biomedical document search engine [Wildemuth and Moore, 1995].

The fundamental issues that IR researchers have been addressing have not dramatically changed since the beginning of the search engine era. One of the primary issues is the difficulty in constructing a query representation that captures a user's intent. Compared to other forms of information requests, such as SQL statements to relational databases, free-text queries are very rough and implicit in representing what users really want to know. For example, a user might use a single word query like "rheumatism" in search for either the definition of the disease or the possible treatments of the disease, where the user's intent is not properly stated. Similarly, a query "moonshot" can mean the cancer-related initiative program or literally the launching of a spacecraft to the moon. This problem is commonly known as the *vocabulary mismatch* issue [Furnas et al., 1987]. There are typically no easy solutions for this when users formulate queries that are very incomplete relative to the actual information need they have in mind. This problem is definitely ameliorated when the queries are formulated as interrogative sentences.

Another common problem is to encapsulate the semantic information in a high-level language (e.g., natural language) and translate it into a low-level machine processable format (e.g., vector representation). Inadequate encapsulations directly influence relevance computations between documents and queries. This problem is also known as the *semantic gap* issue. Conventional document retrieval models have used term frequencies (the bag-of-words model) to represent the topic of query/document

and use the word-level frequency statistics to find the *relevant* documents to a query. The assumptions made in the typical retrieval methods are useful from a practical perspective in IR but not thoroughly valid especially in specialized fields such as biomedicine.

Figure 1.1 illustrates the sequential steps of the document retrieval pipeline and where the two fundamental challenges (vocabulary mismatch and semantic gap issues) exist. Typically, traditional IR systems consist of four main components.

1. **Source Data**: The IR pipeline starts with two data sources: user's information needs and the target information collections. The articles are unstructured texts encoded in natural language, and the form of queries often depends on the model of the IR system; It can be a SQL statement to a relational database or a set of keywords to a document search engine. Sometimes it can be semi-structured where the query is represented in the forms of various pieces of evidence each being a free-text field.

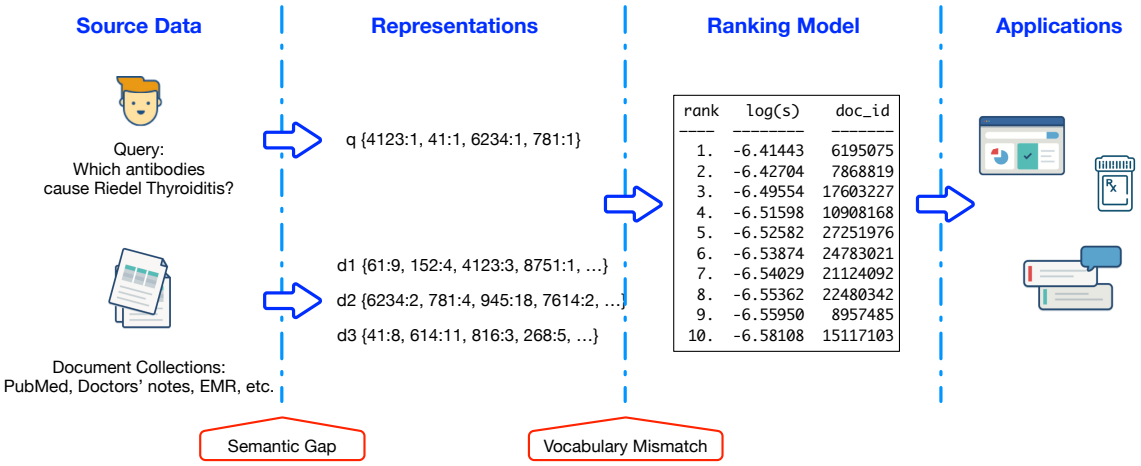
2. **Representations**: Queries and documents are converted to a machine-readable format, which is expected to capture the associated themes/topics. The representation format entirely depends on the assumptions made for the model concerning the way of reading and understanding text. In traditional IR systems, the bag-of-words represent the texts, whereby word frequencies are used as features, disregarding other available information such as the order of the words or additional syntactic structure. Modern IR systems, however, take various linguistics or functional constraints into consideration.

3. **Ranking Models**: Various ranking models exist, all of which aim to measure a target representation's relevance to the query representations. We categorize these methods by their learning methods (details of the methods are further explained in Appendix A):

    - *Unsupervised methods* — e.g., query likelihood language model, Okapi BM25, and sequential dependence model (SDM).
    - *Supervised methods* — e.g., learning-to-rank models such as logistic regression (LR), Support Vector Machine (SVM), RankSVM, LambdaMart, and ListMLE.

4. **Applications**: In the previous step, relevant documents are retrieved and ranked by their scores. Following is the user interface of a search application that displays the rankings or utilizes the results in other forms; it can be

Figure 1.1: Information Retrieval Pipeline



web search results, product recommendations, or formed as a textual response in a conversational AI chatbot system. Different post-processing visualizations may also be used to present the results in a summarized format as opposed to document lists.

1.2 Deep Learning for Natural Language Processing

Over the past decade, the broad field of unstructured data processing (including images, videos, text) has seen a major resurgence of interest in artificial neural networks (ANNs). Neural models with multiple non-linear layers are typically called deep neural networks (DNNs) and the field of their study applied to machine learning has been popularized as *deep learning*. DNNs form increasingly higher levels of abstraction of a raw input as the depth of network increases. This has been a boon for the field of computer vision where such a signal hierarchy (lines, arcs, patches, textures) is inherently natural to the organization of image data. Following the successes of DNNs in vision, we have seen a major uptake in the field of natural language processing (NLP). The key advantages of deep learning methods over traditional feature-based NLP techniques can be highlighted as below:

Rich representations for lexical semantics via distributed representations Traditional NLP often uses one-hot encoding to represent a word in a fixed vocabulary; here each word is represented by a vector of the size of the vocabulary with a '1' in a single location (representing the word's index) and zeroes at all other positions. With this crude representation, it is hard to capture relationships among words needed for

language understanding. For example, the vector for the word 'student' is at the same distance from vectors for words 'university' and 'movie' while it is clear to us that the proximity to the former should be higher. To address this, distributed representations typically represent words as dense vectors in $\mathbb{R}^d$ where d is usually a few hundred. These vectors are randomly initialized and are then pre-trained using NNs to predict neighbor words in larger corpora of free text. This imbues the resulting word vectors with distributional information of word usage resulting in distances in $\mathbb{R}^d$ that reflect semantic relatedness in the language.

Transfer learning with pre-trained language models The basic method of pre-training for word embeddings simply predicts neighbor words in a context window (of odd length) surrounding a given word in a corpus. Popularized in 2013 as *word2vec*, this window is moved across each sentence and the neighbor words are predicted based on the center word in the window and errors in prediction lead to updates in the parameters of the model which include the word vectors themselves. One can see this task captures distributional properties of words and leads to better dense representations, but it is still operating in a bag-of-words style setting within the context. Word order in the context is still being ignored. In this setting, we only get static embeddings. That is embedding for a polysemous word like 'bank' is the same regardless of whether it is a financial organization or a river bank. To address this, in 2017, a suite of methods were invented that leverage language models (LMs) to pre-train word embeddings.

Natural language structures are inherently ambiguous as there is no guarantee of a unique parse for a particular sentence following standard grammatical rules (e.g., NounPhrase $\rightarrow$ (Adjective) Noun). Thus it is not practical to characterize natural language generation with a fixed set of rules as language evolves with time. Hence, statistical LM approaches are used to generate new sentences in a natural language and use them in downstream tasks such as machine translation or speech recognition. The LM task can be simply described as learning a probability distribution of the next word in a sentence given a sequence of previous k words: $P(w_t|w_{t-k}, \ldots, w_{t-1})$. This distribution is typically learned from a large corpus of free text that is representative of the general patterns found in the particular language being modeled. Given a previous sequence, this naturally imposes a ranking on the next word to be selected from the full vocabulary.

In 2017, NN-based LM have been repurposed to pre-train word embeddings in a context dependent manner in contrast with the simpler Word2vec approach. A pre-

trained neural LM can then be used to come up with different embeddings for 'bank' depending on its surrounding context. Such a pretrained LM trained on large corpora (e.g. Wikipedia) can be used as a component in a downstream task (e.g., classification) where it is further fine-tuned with the downstream task's objective function (e.g, cross entropy of the classification). So in some sense, the general knowledge about the language as captured by the LM is being *transferred* to a downstream task that uses it. Since 2017, this has been touted as the most important advance in the field of NLP manifesting in the form of several pre-trained LM methods (e.g., BERT, ELMo, ULMFit, GPT, ALBERT, ROBERTa). Pre-trained LMs can be seen as accommodating the "missing information" required in natural language generation and understanding.

1.3 Thesis Statement

Given the intrinsic difficulties of representing free-texts in a machine processable manner, in the context of IR, it is critical to develop methods that are able to do this to also match the semantic contents of documents and queries. Given the information needs in biomedicine are typically closely tied to entities (e.g., diseases, genes, drugs), it becomes essential to bridge the information gaps between a query and document by utilizing available domain-specific knowledge bases and textual resources. The ability to exploit external knowledge (e.g., ontologies) is even more important in biomedicine because the amount of free text that is available in this discipline is not even close to what would be available on the Web for general English. Although there are over 30 million biomedical research articles published thus far, we only have full text access to about 4 million of them. For the remaining papers, we can only access titles, abstracts, and metadata. It is impractical to share clinical text because of presence of private health information (PHI) in them (e.g., names, phone numbers, addresses) and hence a researcher's exposure to clinical text is limited to their own organization's datasets or those from their collaborators. On the other hand, biomedicine is rich in knowledge bases (ontologies, terminologies) that are curated by human experts over decades keeping track of encyclopedic knowledge in particular subfields. This dissertation aims to leverage advances in neural networks and the available external knowledge resources in biomedicine to address the challenges mentioned earlier: vocabulary mismatch and semantic gap.

1.3.1 Bridging the *vocabulary mismatch*

How can we improve language understanding and provide means to represent text in biomedical IR systems?

As discussed in the previous section, DNNs help us build distributed representations for words providing richer semantics than the bag-of-words model. Let us consider the token 'TJS', a user provided query for "Tommy John Surgery". This procedure is named after a baseball player *Tommy John* who underwent this surgery for the first time. This surgery is formally known as "Ulnar Collateral Ligament (UCL) Reconstruction". As we can see, these two different names for the same entity do not share a single word. However, we expect the IR systems to treat them as identical. Even with our dense embedding methods, it is really not possible for these to be treated as identical phrases because the constituent tokens are very different (any relatedness ought to come from words 'surgery' and 'reconstruction'). Unless we have many different occurrences of these two phrases being used in similar contexts in our training corpus, it is tricky to capture that they refer to the same entity. Here it is more beneficial to leverage external knowledge bases such as the Medical Subject Headings (MeSH) where there is a unique code `D000070638` for the concept "Ulnar Collateral Ligament Reconstruction" with a link to well known synonymous terms curated through manual efforts.

In this dissertation, we present two approaches to address these concerns. First, we explore learning methods for distributed representations for biomedical terms that include the lexical units of both words and biomedical concept codes from standardized terminologies. The outcome of this effort is a way to learn both word embeddings and concept embeddings to help with downstream IR applications. The second approach explores methodologies for Named Entity Recognition (NER) and Entity Normalization (EN) in unstructured text. The ability to recognize biomedical mentions (especially those in a non-standard form) and provide the linkage to pre-defined concept codes from standardized terminologies is crucial in QA style applications.

1.3.2 Minimizing the *semantic gaps* between the query and documents

How can we build effective encoding methods that can reduce the gap between the query and document representations?

Traditional IR models rank documents based on exact lexical matches between query and document words. Hence, a typical means to minimize the semantic gap between the user's information needs and the document topics is through Query

Expansion (QE), introducing additional query terms to broaden the initial query to match more documents. This dissertation explores deep learning methods for Natural Language Understanding (NLU) to improve retrieval performance in the context of query (or document) understanding. First, we propose and investigate the efficacy of a neural sentence matching model for relevance measures. This model is evaluated in a biomedical QA task whereby we retrieve relevant documents to a given biomedical question. Secondly, we examine a novel approach to the query expansion technique; instead of manipulating the query, we read a candidate document, summarize it into query-like sentences, and compare them with the initial query. We adopt the neural machine translation (NMT) model to generate such pseudo-queries. We realize that it is not really plausible to completely understand a user's search intent based on the phrase they may use to represent a mental model of what they are seeking. We would like to point out that our research is not really about making those leaps about intent. It is about how to use the query phrases supplied in a more effective manner using better representations of documents and queries.

1.4 Organization

The remaining chapters of this dissertation are organized as follows.

Chapter 2 describes the methodology in modern IR systems and presents background information to understand the specific tasks introduced in the rest of the dissertation. This chapter introduces an example of a typical approach to document ranking tasks for enhanced retrieval quality. We also highlight some of the previous efforts to address the *vocabulary mismatch* issue and *semantic gaps* in the IR pipeline system. This chapter concludes with a brief discussion on the challenges and limitations of using neural networks for IR.

Chapter 3 focuses on the methods of learning distributed representations for biomedical terms. Here, we extend the lexical units from words to their semantic counterparts — biomedical concepts in standardized vocabularies (e.g., UMLS, MeSH, ICD). In this study, we repurpose the transformer architecture to improve pretrained *static word embeddings* using concept correlations in distant supervision learning. Our proposed model achieved the best performances across several word embeddings evaluation benchmarks. Qualitative observations also indicate the model's efficacy in fine-tuning distributed representations to better capture semantics.

Chapter 4 presents the second approach for bridging the vocabulary mismatch. This chapter explores a different vein of NLP research — named entity recognition (NER) and normalization (EN) tasks. These tasks are the subtasks of Information Extraction (IE), which gives a different perspective in IR tasks. We present a joint learning neural model for recognizing the mentions of biomedical concepts from free-texts and linking them to a standardized ontology. We show the Zero-Shot Learning (ZSL) capability for unseen entities by utilizing the dense vector representations for entity aliases. We also propose a novel approach to the sequence segmentation task and evaluate the model against conventional methods.

Chapter 5 presents the third study in which we adopt neural network models for encoding the question and candidate answer sentences in a biomedical question-answering setup. We utilize Siamese-style networks [Mueller and Thyagarajan, 2016] with RNN encoders and a self-attention mechanism to measure the candidate answer sentence's relevance to the given question sentence. This study shows the effectiveness of relevance measures from the neural representations in document retrieval and their usefulness in the question-answering framework. It also introduces two additional meta-document features leveraging the external knowledge bases, such as the Medical Subject Headings (MeSH) vocabulary and SemMedDB, a repository of semantic predications extracted from the biomedical literature.

Chapter 6 presents the fourth and the final study of this dissertation focused on the IR tasks for Precision Medicine (PM). We propose a hybrid document scoring and reranking model composed of three different neural models: (1). a document relevance classification model, (2). a neural extractive summarization model for identifying keywords, and (3). a facet-conditioned neural abstractive summarization model for generating a pseudo-query given the document context and a facet type (e.g., genetic variation). Our main innovation is in pivoting the focus from query manipulation techniques (e.g., pseudo-relevance feedback) by previous methods to transforming candidate documents into pseudo-queries via neural text summarization models.

Chapter 7 concludes the dissertation by summarizing the artifacts of our research efforts and highlighting potential future work.

Chapter 2 Background and Related Work

2.1 Modern IR Systems

A modern IR system typically involves the following four processes along the information retrieval pipeline as described in the previous chapter:

Indexing Documents and corresponding metadata are processed and stored in advance to improve the efficiency in the following retrieval process; Often, the information is quantified through term frequencies, the positions of terms in a document, and document lengths. An index is a data structure storing a mapping from an elementary searchable unit, such as a tokenized word, to its location in a document or a set of documents — an *inverted index* maps from words to documents, and a *forward index* maps back from documents to words. Generally speaking, the purpose of constructing these indices is to provide an efficient way of full-text searches.

Query analysis/transformation The language (terms, phrases) used for queries is typically different from the one in documents. Due to the ambiguity issues caused by the use of different languages, additional work such as query manipulation is often unavoidable. The methods for adjusting query split into two classes: global methods and local methods. A representative global method is query expansion whereby more informative terms are added to the query using independently constructed knowledge bases. Local methods include (pseudo-) relevance feedback (PRF) techniques whereby the original query representation is adjusted according to the initially retrieved documents by the original query.

Initial retrieval Once a user provide a (modified) search query, an IR system retrieves a set of candidate documents and rank them in order by using one of the available document scoring functions that measure document relevance to the query (e.g., BM25). Most of the modern IR systems use a variant of *TF-IDF* formula, where *TF* (short for term frequency) measures the weight of a term that occur in a document and *IDF* (short for inverse document frequency) measures how much information the term provides across the collection.

Reranking The ranked list of documents obtained from the previous process can be presented to the user as is, or it can be refined and filtered using more accurate (but less efficient) document scoring functions.

Conventionally, the modern IR framework relies on the bag-of-words approaches wherein indexing is based on the term/document IDs and their occurrence statistics. Although the predominant use of this method in real-world applications proves its effectiveness in IR, the bag-of-words retrieval model has inherent limitations: incapable of understanding (or representing) the meanings of queries/documents and ranking documents based on the semantic relations to the query. Much attention has been drawn to this problem. The following sections present some of the previous efforts which aim to address this problem using neural network-based models.

2.2 Related Work: Neural Networks for IR

Neural Networks for IR (NN4IR) is an emerging field that focuses on leveraging the advantages of deep learning over the traditional machine learning methods to enhance the retrieval performance. Various components of IR system adopt the neural network techniques. Training and utilizing word embeddings permits the influx of expert knowledge from external resources. Numerous neural ranking models have been proposed along with the advances of natural language understanding via NN approaches to rank documents in a different perspective. As an alternative to bag-of-words, different neural models have been proposed to capture the semantic evidence in the indexing stage. In the following subsections, we present previous efforts in NN4IR, in particular those aiming to address the two primary issues in IR. Related work that is only pertinent to a specific method is presented in its corresponding section.

2.2.1 Previous efforts to bridge *vocabulary mismatch*

Distributed vector representations for words

The success in applying neural networks to natural language processing can be attributed to the development of distributed vector representations for words, also known as *word embeddings*. Word embeddings represent a word as a low-dimensional dense vector in $\mathbb{R}^d$ where d is usually several hundred. One assumption is that contextually similar words have similar vector representations; hence the cosine similarity or dot product between two representations is likely to be relatively high when the words share common meanings.

Word embeddings are learned from a large text corpus with the object of predicting a word given its context using a neural network. The learning methodology of word embeddings has evolved through attempts to address certain linguistic challenges, including the following:

- The number of examples with a rare word or not-indexed word (Out of Vocabulary, OOV) is extremely small.
- The morphological analysis, which is crucial in understanding lexical semantics, is less exploited in building word embeddings.
- Polysemy and homonymy cannot be implemented with the single vector representation scheme.
- Certain types of words, such as acronyms, chemical identifiers, or entity codes, have a different expression format.

The following list highlights the related work on the development of word embeddings:

Word2vec [Mikolov et al., 2013a] The implementation of Word2vec drew massive attention in the field and had an extensive influence on NLP applications due to its efficient training techniques (i.e., negative sampling and stochastic gradient descent) for speed and scalability. This method is based on the distributional hypothesis [Harris, 1954]: that is, words that are used and occur in the same contexts tend to purport similar meanings.

Word2vec has two variant models that depend on the target of the probabilities: *Skip-gram* and *CBOW*. These models are implemented in a shallow neural network consisting of an input layer, a linear projection layer, and an output score layer. The input layer is a one-hot input vector representing the target word, the projection layer contains the dense vector of the word, and the output layer contains the scores which can be interpreted as a probability distribution of words that are likely to be seen in the target word's context. The goal of *Skip-gram* is to predict context words by the given target word. These models learn a classification task; *CBOW* answers "Which word is missing in the context?" and *Skip-gram* answers "which word is in the context of a specific word?"

GloVe [Pennington et al., 2014] Global Vectors for Word Representation extends the Word2vec method. The model aims to approximate the co-occurrence counts among words, which shows fast training and comparable performance even with a small corpus. GloVe differs in that Word2vec is a "predictive" model, whereas this model is a "count-based" model.

CharCNN embeddings [Zhang et al., 2015, Kim et al., 2016] Another approach that considers the subword-level meanings is the embeddings in the Character CNN models. With a set of characters (e.g., 26 English characters, 10 digits, and 33 special characters), the local (subword) semantic information can be used as features for downstream NLP tasks. The model proved its strength with OOV words, misspelled words, rare words, and emoticons. It also reduced model complexity by using a relatively small number of vector representations for the combinations of characters.

fastText [Bojanowski et al., 2017a] fastText took a further step by constructing word representations by its constituent character-level n-grams (where n is usually between 3 and 6). It represents a word as the sum of its character n-grams representations. With this model, grammatical variations that share most n-grams and compound nouns are easy to model. As to the rest, it shares the same architecture of the Word2vec Skip-gram model.

Embeddings from language models In pursuit of building *contextualized* word embeddings, researchers have proposed to train NN-based generalized language models (GLMs) on the large corpora and use the model outputs as contextualized embeddings; representative works are ELMo [Peters et al., 2018], BERT [Devlin et al., 2019], RoBERTa [Liu et al., 2019], and XLNet [Yang et al., 2019]. For previous embeddings methods such as word2vec or GloVE, word order is not relevant. But LM's objective specifically deals with predicting the next word based on a prefix. Hence, word order is crucial here. This means, the same word can lead to a different contextual embedding using this approach based on what other words occurred before it. Also, GLMs employ deep neural networks, most of which utilize the attention mechanisms, whereas the previous models are shallow neural networks. These GLMs are capable of approximating complex functions and consequently higher-level language understanding compared to the previous models for building word embeddings.

Distributed vector representations for biomedical terms

Learning distributed vector representations for words has been extended to their semantic counterparts — biomedical concepts in standardized vocabularies (e.g., UMLS, MeSH, ICD).

De Vine et al. [2014] exploited the Unified Medical Language System (UMLS) concepts from clinical patients records (MedTrack [1]) and medical literature (OSUMED [2]) in building the UMLS concept embeddings. They used a *Word2vec Skip-gram* model to train word embeddings over its temporal occurrences in the records. Choi et al. [2016a] expanded the work mentioned earlier by applying the same architecture on three different corpora: *Medline* abstracts, medical claims, and clinical narratives. Choi et al. [2016b] proposed *Med2vec*, multi-layer representations that learn the biomedical codes and clinic-visiting information of patients which appear in the electronic health record (EHR) data. This model is evaluated with healthcare prediction problems. They evaluated the model by predicting the future medical codes given the observed ones and also predicting the clinical risk groups (CRG) level. Cai et al. [2018] trained embeddings of diagnosis codes in electronic medical records (EMRs). They incorporated the temporal information into the medical concepts with a neural architecture utilizing the attention mechanisms. All of these efforts focus on training entity (or similar meta-text) embeddings within a model for predictive tasks.

Another important line of research is the methodology of adjusting the pre-trained word embeddings (also known as *semantic specialization of word vector space*) for a specific task. The key idea is to refine word representations using external knowledge resources for improved lexical semantics. It is an attempt to combine two worlds of knowledge; (1) the probabilistic semantic knowledge from a large text corpus and (2) the relational knowledge from carefully curated and structured ontologies.

In training word embeddings using latent semantic analysis (LSA), Yih et al. [2012] constructed the word co-occurrence matrix such that it incorporates relational information like antonymy from a thesaurus. This method is a joint learning model whose objective is regulated by a specific lexical-semantic relation. Yu and Dredze [2014] combined synonymy constraints with the CBOW distributional objective to train the *Word2vec* embeddings. Liu et al. [2015] transformed linguistic knowledge into ordinal constraints and used them in training word embeddings (i.e., word similarity in the following order: (1) *synonyms*, (2) *co-hyponyms*, and (3) *shorter distance in a semantic hierarchy*).

[1] A collection of 17,198 clinical patient records used in the TREC 2011/12 Medical Records Track
[2] A collection of 348,566 MEDLINE medical journal abstracts used in TREC 2000 Filtering Track

These works aim to adjust the entire vocabulary (*joint specialization*). In contrast, we can also fine-tune only the words that appear in external constraints (*post-processing specialization*) [Faruqui et al., 2015, Mrkšic et al., 2016].

Evaluation methods for word embeddings

Unsupervised learning methods which induce vector representations from large corpora have their drawbacks; the semantic features of the embeddings heavily depend on the corpus and the context type [Melamud et al., 2016]. For example, popular large corpora, such as Google News or Wikipedia, have different word distributions to a domain-specific corpus, such as PubMed. Prior work, described earlier, has demonstrated that using the corpus-specific embeddings for a particular task in that domain, such as document retrieval, outperforms that of using globally-trained embeddings. However, we cannot assert that a "locally-trained" embeddings is superior than other "globally-trained" embeddings.

Despite the popularity of word embeddings, the evaluation methods for those have not been fully established. Although researchers have proposed various evaluation methods, we have not had a standardized way of measuring the quality of word embeddings. For example, the authors of Word2vec used a *word analogy* task for its evaluation method. However, we cannot use the same evaluation method in evaluating other word embeddings for examining various linguistic features. In 2015, Schnabel et al. categorized the existing evaluation approaches into extrinsic and intrinsic evaluation methods [Schnabel et al., 2015]. In 2016, the Association for Computational Linguistics (ACL) opened the first workshop on the methodology of evaluating vector-space representations used in NLP tasks. The goal was to discuss the critical problems of assessing the performance of word embeddings in NLP tasks.

Intrinsic evaluations Intrinsic evaluations test model's performance in measuring certain type of relationships between words. Typically, these evaluation datasets contain a set of semantically related word pairs, and human experts score them for serving as reference. The model's performance is evaluated by computing an aggregated score such as a correlation coefficient. The following lists commonly used benchmarks for evaluating the quality of word embeddings grouped by their relationship types.

- Similarity/relatedness
  - WordSim-353 [Finkelstein et al., 2001]: similar/related words
  - RW [Luong et al., 2013]: rare words and morphologically complex words

  - MEN [Bruni et al., 2014]: related words such that they occur as annotations in an image dataset

- Semantic similarity (decoupled from relatedness and association)
  - TOEFL Synonym Questions [Landauer and Dumais, 1997]
  - SimLex-999 [Hill et al., 2015]
  - SimVerb [Gerz et al., 2016]
  - SemEval 2017 Task 2 [Camacho-Collados et al., 2017]

- Lexical entailment
  - HyperLex [Vulić et al., 2017]: *type-of* relation also known as hyponymy-hypernymy or lexical entailment relation

- Word analogy
  - GRE Antonymy [Mohammad et al., 2008]
  - Microsoft and Google Analogy [Mikolov et al., 2013b]

The majority of the evaluation datasets for biomedical word embeddings target measuring the relatedness between biomedical concepts (often via a medical vocabulary). For example, each instance consists of a pair of biomedical concepts and the corresponding relatedness scores judged by human experts such as physicians and biomedical coders. Details of the quantitative evaluations with biomedical concepts are also presented in Section 3.4.2.

Extrinsic evaluations Extrinsic evaluation methods measure the contribution of word embeddings to a specific downstream NLP task as a linguistic feature. There is an implicit assumption made such that higher quality word embeddings will necessarily improve the results on any downstream task. However, it is difficult to say that the assumption holds in all cases. In general, we cannot assure that the optimized word embeddings on a specific system are superior to the ones which perform acceptably on multiple tasks. Empirical evidence supports that the extrinsic evaluations cannot be used as a proxy for the quality of word embeddings.

2.2.2 Previous efforts to reduce *semantic gaps*

The bag-of-words approach in IR causes a semantic gap between a user query and its computational representation, and eventually between a user query and documents. This problem in IR is also referred to as the *query-document mismatch* [Li and Xu, 2014]. A user query often consists of a list of few keywords or a simple interrogative sentence. Consequently, the user-provided information in a query is often insufficient to reason out the perfect matching documents. To address this issue, users may attempt to refine a query multiple times. Typically, the size of a document collection is often in millions. Regarding the query-document matching process, manipulating a user-provided query is more practical approach than dealing with millions of documents. Hence, query refinement (QR) has become a conventional method to enhance the retrieval quality.

In the following subsections, we present previous efforts on systematic approaches to automatically or interactively engage in this refinement process. We can categorize QR methods into two groups: *global* and *local* in terms for the systematic use of additional information in building queries. Global methods utilize external knowledge sources and provide additional information to the original user query. On the other hand, local methods leverage the initial search results or different types of feedback to adjust the original query. Followings are some of the commonly accepted QR methods.

Query expansion (QE) using controlled vocabularies — a global method

Query expansion (QE) is a typical QR method whereby we add or select (i.e., *query reduction*) more informative terms to the user's original query terms. As a global approach, we utilize external knowledge sources, such as a controlled vocabulary, to search for synonyms or various aliases of an ambiguous term in a user query. Controlled vocabularies allow us to find a restricted set of words representing common names, concepts, and domain terminologies. Widely used thesauri for general English texts include WordNet and the Library of Congress Subject Headings (LCSH). Examples of the ontology-like vocabularies frequently used in biomedical informatics are lists in Table 2.1. These vocabularies provide not only the definitions of terms but also the semantic relationships between concepts, such as the 'is-a' (subsumption) and 'part-of' (composition) relationship.

In one of our previous experiments, we used the UMLS thesaurus to obtain the synonyms and related words to the biomedical concepts that occur in a query [Noh

and Kavuluru, 2017]. For example, given a query, "What is LHON, also known as Leber's syndrome?", the term 'LHON' can be further described by its preferred name "Optic Atrophy, Hereditary, Leber". Figure 2.1 shows the concept relations of the term 'LHON' and its definition in the UMLS thesaurus.

Expanding queries using synonyms is the most typical way of using controlled vocabularies in QE, which has adopted and evaluated in many research works. For example, Lu et al. [2009] investigated the effectiveness of using MeSH in PubMed. They made use of the MeSH field of indexed MEDLINE citations and created a special term matching table for assigning MeSH terms to original query terms. Currently, PubMed uses QE for biomedical entities in MeSH (e.g., diseases, chemical names, and so on) leveraging this method.

Figure 2.1: UMLS concept relations of the term "LHON"

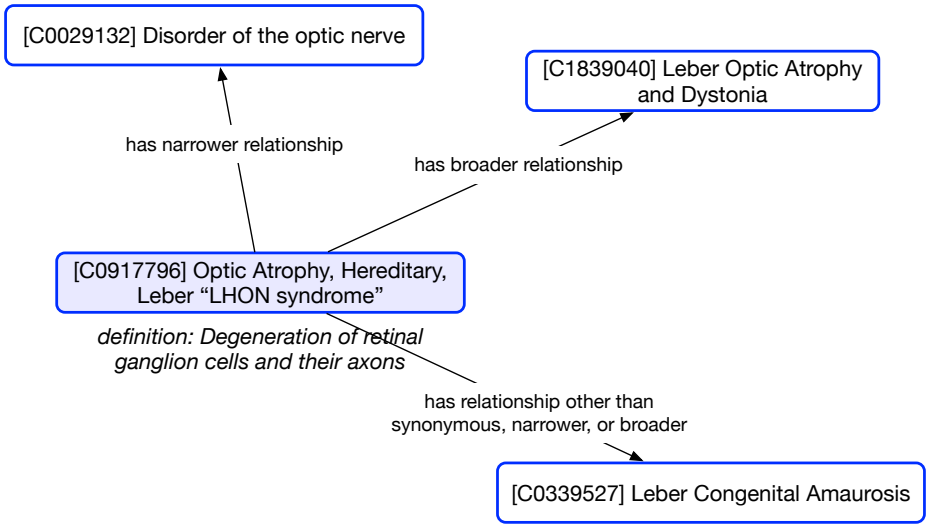


Table 2.1: Popular ontology-like biomedical vocabularies

Ontology name (Abbreviation)	Target entities
Systemized nomenclature of medicine-Clinical Terms (SNOMED CT)	General terminology for electronic health records, including clinical findings, symptoms, diagnoses, procedures, body structures, organisms and other etiologies, substances, pharmaceuticals, devices and specimens
RxNorm	Clinical drugs containing all medications available on the U.S. market
Medical Subject Headings (MeSH)	General terminology for indexing and cataloging of biomedical documents
International Classification of Diseases, version 10 — Clinical Modification (ICD10CM)	Diagnostic terminology including disease, symptoms, clinical signs and circumstances
Logical Observation Identifier Names and Codes (LOINC)	Laboratory observations and measurements
National Center for Biotechnology Information Taxonomy (NCBI Taxonomy)	Nomenclature for all of the organisms in the public sequence databases
National Drug File — Reference Terminology (NDFRT)	Electronic drug list used by the VHA hospitals and clinics
Unified Medical Language System (UMLS)	Meta-thesaurus combining many health and biomedical vocabularies and standards

Recently, more approaches of neural network-based methods for QE have been proposed:

Goodwin and Harabagiu [2014] utilized controlled biomedical vocabularies (i.e., UMLS, SNOMED-CT) for selecting query expansion terms. They search for candidate expansion terms from globally-trained word embeddings; 20 most similar word embeddings to a query term are chosen as candidate expansion terms. Almasri et al. [2016] proposed a QE method utilizing target corpus-based word embeddings. They use pre-trained embeddings with a domain-specific corpus (i.e., the CLEF medical document collection) and showed superior results over the method using globally-trained word embeddings for QE.

Using controlled vocabularies for query expansion is an almost necessary process in medical document search engines. Medical queries often contain abbreviations for diseases, chemical substances, or genes/proteins. Using controlled vocabularies in QE certainly helps the retrieval performance in settings like this. Nevertheless, there exist many challenges such as described below:

1. Identifying an entity mention in a user query and linking to a standard concept requires a model to learn the mapping function from biomedical annotated documents. In the biomedical domains, annotating documents with a controlled vocabulary by human experts is an expansive task. Without using an automated process, it is practically impossible to tag a large corpus with a controlled vocabulary.

2. Determining the criteria for systematically choosing related terms for QE is not a trivial task. The information gain by adding a synonym to a certain query term can be beneficial, or disadvantageous. In other words, adding additional terms can shift the topic to an undesirable one.

3. In most cases, the placement of entities in a hierarchical structure defines the relationships between them. When the entities do not occur on the same branch path, it is not easy to clarify the relationships between them. Furthermore, it is initially impossible to define the relationships between entities and words.

Query expansion using relevance feedback — a local method

QE using relevance feedback is another well-established approach. In this method, a system takes the initially retrieved documents explicitly or implicitly identified as relevant, analyzes the documents to extract candidate expansion terms or phrases,

and appends them to the original query to perform a new retrieval process. Depending on the source of feedback, we can categorize the methods into three groups:

1. *Explicit Feedback* — Users "explicitly" indicate the relevance of the results, and the system uses the feedback to improve the original query.

2. *Implicit Feedback* — User feedback can be inferred by monitoring user behaviors such as certain type of browsing actions like clicking, scrolling, or spending time for viewing documents.

3. *Blind ("Pseudo") Feedback* — If it is assumed that the top k documents of the initially returned results are *relevant* to the given query, then we use them as feedback. This method is called *pseudo-relevance feedback*.

The Rocchio Algorithm and Relevance-based Language Models (RMs) are the most representative relevance feedback methods. The Rocchio algorithm [Rocchio, 1971] is a classic method of explicit relevance feedback wherein the user's search query and documents are embedded in the vector space model (VSM). The original query is refined by incorporating the relevance feedback information to select an arbitrary percentage of relevant and non-relevant documents. Relevance-based language models, such as RM3 [Lavrenko and Croft, 2017], assume that retrieved documents (D) are random samples from either one of two classes: relevant (R) or irrelevant ($\overline{R}$). To optimize the retrieval performance, documents should be ranked by $P(R|D)$, which is rank equivalent to the ranking by log-odds, $\log \frac{P(D|R)}{P(D|\overline{R})}$. Further explanation of this derivation is provided in Appendix A.2. The common idea of these feedback-based methods is to select the expansion terms (or just select documents) from those with a higher likelihood of being relevant to the given query.

Many research efforts have been put into this group of approaches leveraging the NN methods. Roy et al. [2016] restricted the query expansion terms within the set of words found in the initially retrieved documents. They utilized corpus-trained word embeddings to select expansion terms by cosine similarity. Their proposed method showed improved performance over the unexpanded baseline models but inferior performance to the traditional query expansion methods, such as RM3 or PRF. Diaz et al. [2016] questioned whether it is an effective method to use globally-trained word embeddings for QE. They proved that the word embeddings trained on the pseudo-relevant documents performs better in finding topic-specific expansion terms than globally-trained word embeddings. Query refinement techniques include rewriting the entire query terms using the ones from automatically selected candidate query

terms. Nogueira and Cho [2017] proposed a neural reinforcement learning model that learns which query terms should be used to increase the document recalls. Another promising approach is to manipulate the logical operators or syntactic components of query [Kim et al., 2011, Scells et al., 2018]. For example, Boolean operators (e.g., AND, OR) or other factors of query expressions (e.g., adjacent term window size, field restrictions, etc.) can be optimized to maximize the retrieval quality. With these methods, candidate queries are automatically generated and ranked by the retrieval performance.

Reranking using large pre-trained language models — neural LM approach

Since 2018, the use of pre-trained language models such as BERT [Devlin et al., 2019] in downstream NLP tasks has been the predominant approach for transfer learning. BERT uses the Transformer [Vaswani et al., 2017] architecture, whose core function is to compute the input units' interactions through multiple attention-based layers. These models are pre-trained on a sizeable general-language corpus, by which the trained model supposedly captures the linguistic features commonly applicable to various NLP tasks. A common practice to apply these models includes token/sequence classification [Shelmanov et al., 2019, Munikar et al., 2019] and question-answering [Alberti et al., 2019]. We adopt a pre-trained Generalized Language Model (GLM) in our proposed models for the reranking tasks.

In the following chapters, we present our efforts pertaining to the research objectives: bridging the vocabulary mismatch (Chapter 3 and 4) and reducing semantic gaps (Chapter 5 and 6).

Chapter 3 BERT-CRel: Distributed Representations for Biomedical Terms and Concepts [1]

In traditional IR, using the bag-of-words scheme for exact query-document term matches is the primary cause of the vocabulary mismatch issue. For example, two drug names, *Aspirin* and *Ibuprofen*, are considered as two different things in this scheme even though they belong to the same drug class (i.e. nonsteroidal anti-inflammatory drugs) and share much of the common attributes. One of the benefits of using neural network-based approaches is the capability of representing a word as a dense vector. With the representations, the machine can measure the semantic similarity between two terms by computing the cosine similarity or dot product of those two corresponding representations. In this chapter, we study the methodology of learning and fine-tuning distributed representations for biomedical terms, including the corresponding biomedical concept codes.

At this juncture in computing for biomedicine, natural language processing research and applications almost exclusively deal with neural network methods. Central to these methods is the notion of dense word embeddings, which also have been extended to their semantic counterparts — biomedical concepts in standardized vocabularies (e.g., UMLS, MeSH, ICD). Prior to this thesis research, most of the methods for training biomedical word embeddings did not consider the potential advantages of learning embeddings for both words and biomedical concepts in the same vector space.

Pre-training with neural methods that capture local and global distributional properties (e.g., skip-gram, GLoVE) using free text corpora is often used to embed both words and concepts. Pre-trained embeddings are typically leveraged in downstream applications using various neural architectures that are designed to optimize task-specific objectives that might further tune such embeddings. Since 2018, however, there is a marked shift from these *static* embeddings to *contextual* embeddings motivated by contextualized language models (e.g., ELMo, BERT, and ULMFiT). These dynamic embeddings have the added benefit of being able to distinguish homonyms and acronyms given their context. However, static embeddings are still relevant in low resource settings (e.g., smart devices, IoT elements, edge computing) and to study lexical semantics from a computational linguistics perspective.

[1] This chapter is based on the paper [Noh and Kavuluru, 2020a] which is submitted to the *Journal of Biomedical Informatics* for peer review.

Furthermore, by jointly learning concept (and word) embeddings, some ambiguity issues maybe overcome even with static embeddings. Improved static embeddings can also be used as initial parameters in contextualized models to further improve them. In this chapter, we jointly learn word and concept embeddings by first using the skip-gram method and further fine-tuning them with correlational information manifesting in co-occurring Medical Subject Heading (MeSH) concepts in biomedical citations. This fine-tuning is accomplished with the BERT transformer architecture in the two-sentence input mode with a classification objective that captures MeSH pair co-occurrence. In essence, we repurpose a transformer architecture (typically used to generate dynamic embeddings) to improve static embeddings using concept correlations. We conduct evaluations of these tuned static embeddings using multiple datasets for word relatedness developed by previous efforts. Without selectively culling concepts and terms (as was pursued by previous efforts), we believe we offer the most exhaustive evaluation of static embeddings to date with clear performance improvements across the board.

3.1 Deep Neural Networks and Distributed Representations for Words

Biomedical natural language processing (BioNLP) continues to be a thriving field of research, garnering both academic interest and industry uptake. Its applications manifest across the full translational science spectrum. From extracting newly reported protein-protein interactions from literature to mining adverse drug events discussed in the clinical text, researchers have leveraged NLP methods to expedite tasks that would otherwise quickly become intractable to handle with a completely manual process. Computer-assisted coding tools such as 3M 360 Encompass, clinical decision making assistants such as IBM Micromedex with Watson, and information extraction API such as Amazon Comprehend Medical are popular use-cases in the industry. As textual data explodes in the form of scientific literature, clinical notes, and consumer discourse on social media, NLP methods have become indispensable in aiding human experts in making sense of the increasingly data heavy landscape of biomedicine. The rise of deep neural networks (DNNs) in computer vision and NLP fields has quickly spread to corresponding applications in biomedicine and healthcare. Especially, as of now, BioNLP almost exclusively relies on DNNs to obtain state-of-the-art results in named entity recognition (NER), relation extraction (RE), and entity/concept linking or normalization (EN) — the typical components in biomedical information

extraction[2].

3.1.1 Neural word embeddings

The central idea in DNNs for NLP is the notion of dense embeddings of linguistic units in $\mathbb{R}^d$ for d that generally ranges from a few dozen to several hundreds. The unit is typically a word [Bengio et al., 2003, Collobert and Weston, 2008, Mikolov et al., 2013a], but can also be a subword [Bojanowski et al., 2017b] (e.g., prefix/suffix) or even a subcharacter [Yu et al., 2017a] (for Chinese characters that can be broken down further). These dense embeddings are typically *pre-trained* using large free text corpora (e.g., Wikipedia, PubMed citations, public tweets) by optimizing an objective that predicts local context or exploits global context in capturing distributional properties of linguistic units. Based on the well-known distributional hypothesis that words appearing in similar contexts are semantically related or share meaning [Harris, 1954], this pre-training often leads to embeddings that exhibit interesting properties in $\mathbb{R}^d$ that correspond to shared meaning. Once pre-trained, word embeddings are generally fine-tuned in a supervised classification task (with labeled data) using a task-specific DNN architecture that builds on top of these embeddings. While the notion of dense word embeddings existed in the nineties (e.g., latent semantic indexing), neural embeddings together with task-specific DNNs have revolutionized the field of NLP over the past decade.

Since 2018, however, the static embeddings discussed thus far have been improved upon to address issues with polysemy and homonymy. Around the same time, transformers (such as BERT [Devlin et al., 2019] and RoBERTa [Liu et al., 2019]), ELMo [Peters et al., 2018], and UMLFiT [Howard and Ruder, 2018] have been developed to facilitate contextualized embeddings that generate the embedding of a word based on its surrounding context. This process typically generates different embeddings for polysemous occurrences of a word, such as when the word "discharge" is used to indicate bodily secretions or the act of releasing a patient from a hospital. Even for words that typically have a unique meaning, contextual embeddings might generate embeddings that more precisely capture the subtleties in how it is used in a particular context. Such contextualized embeddings might be better suited when predicting NER tags or composing word sequences toward a classification end-goal.

Although contextualized embeddings are an excellent addition to the neural NLP repertoire, we believe there is merit in improving the static embeddings for various

[2]Some exceptions exist when handling smaller datasets in highly specific domains where ensembles of linear models may prove to be better

reasons: (1) Contextualized models are based on language modeling and are more complex with multiple layers of recurrent units or self-attention modules. Base models tend to have tens of millions of parameters [Rogers et al., 2021] and using them without GPUs in low-resource settings such as smart devices used in edge computing or IoT is infeasible. Simpler models that use static embeddings can be built with 1–2 orders of magnitude fewer parameters and can run on smaller CPUs even in low resource settings. While leaner transformers are actively being investigated (e.g., DistilBERT [Sanh et al., 2019]), they offer nowhere near the model size reduction needed for usage in low resource settings. (2) Static embeddings can be of inherent utility for linguists to continue to study lexical semantics of biomedical language by looking into word or subword embeddings and how they may be indicative of lexical relations (e.g., hypernymy and meronymy). Another related use case is to study noun compound decomposition [Kavuluru and Harris, 2012] in the biomedical language, which is typically treated as a bracketing task that ought to rely only on the local context within the noun compound. For example, *candidate ((tumor suppressor) gene)* and *((tumor suppressor) gene) list* demonstrate two different decompositions of four-word compounds. (3) Contextualized embeddings typically only make sense in languages that have large digitized corpora. For less known languages that have smaller repositories, the language modeling objective such embeddings rely on can lead to significant overfitting compared to static approaches [Eisenschlos et al., 2019]. (4) Improved static word embeddings can also help initialize the embeddings before the process of language-modeling-based training ensues in the more expensive contextualized models [3] to further enhance them (when compute power is not a major limitation).

3.1.2 BERT-CRel: High-level intuition and overview

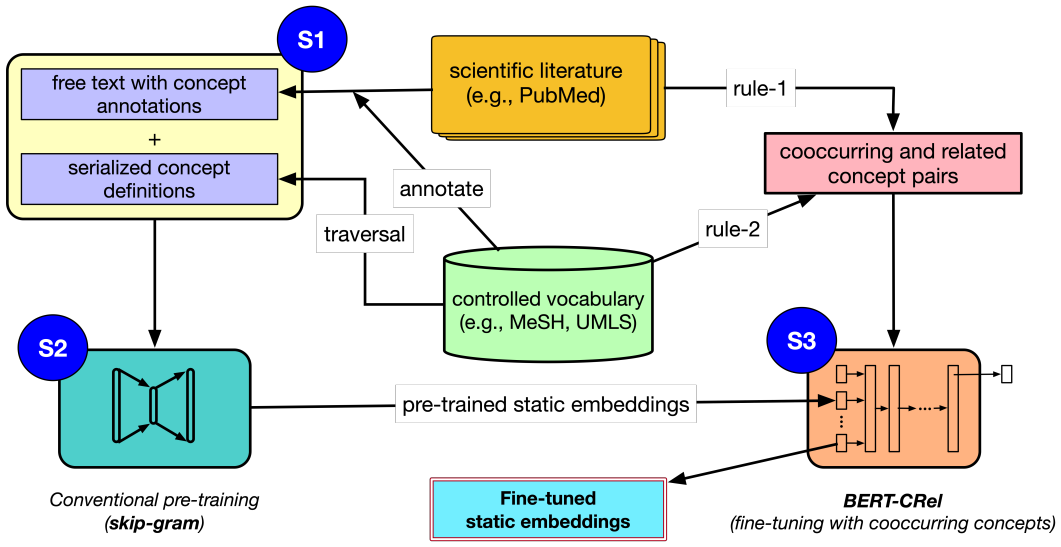
This chapter proposes and evaluates methods to improve biomedical word embeddings to be made publicly available for downstream use by the community. Before we outline the framework and intuition behind our methods, we first motivate the idea of jointly learning embeddings for biomedical concepts and words in the context of our goals. Our framework is depicted in Figure 3.1 whose components will be discussed in the rest of this section.

Biomedical concepts are analogous to named entities in general English. Names of genes, drugs, diseases, and procedures are typical examples of concepts. Just

[3] This clearly assumes that the same tokenization is appropriately maintained in both static and the subsequent contextualized models

like entity linking in general NLP research, concept mapping is typically needed in BioNLP where concepts are to be mapped to their standardized counterparts in some expert curated terminology. This mapping part is harder in BioNLP given the variety of ways a concept can be referred to in running text. Often, there might not be much lexical overlap between different aliases that point to the same concept. For example, the procedure *ulnar collateral ligament reconstruction* is also called *Tommy John surgery* and they both refer to the same medical subject heading (MeSH) concept code D000070638. These aliases are provided in the corresponding terminology and the Unified Medical Language System (UMLS) metathesaurus that integrates many such terminologies.

Figure 3.1: The schematic of our approach to improve word embeddings. S1 deals with pre-processing steps to create a concept enhanced corpus. S2 involves conventional pre-training using local context prediction objectives. S3 constitutes fine-tuning with distributional regularities based on co-occurrence. For S3, entity pairs are constructed based on two relevance rules: rule-1 is concept co-occurrence in a PubMed citation and rule-2 is proximity in a concept hierarchy



Our first main idea is to use a well-known concept mapping tool to spot concepts in large biomedical corpora and insert those concept codes adjacent to the concept spans. This step is indicated as the (S1) portion in Figure 3.1. Subsequently, run a pre-training method to embed both words and concepts in the same space in $\mathbb{R}^d$. This jointly learns embeddings for both words and concepts and enables two-way sharing of semantic signal: first word embeddings are nudged to predict surrounding concepts, and as the pre-training window moves along the running text, concept embeddings are also nudged to predict neighboring words. In fact, this phenomenon has been

exploited by multiple prior efforts [Cai et al., 2018, Choi et al., 2016b, De Vine et al., 2014] including in our prior work [Sabbir et al., 2017]. Most of these efforts aim to learn concept embeddings that can be used in downstream applications. Here, we demonstrate that this process also improves the word embeddings themselves. This process is indicated through the (S2) part of Figure 3.1. Our choice for biomedical concepts to be jointly learned is the set of nearly 30,000 MeSH codes that are used on a daily basis at the National Library of Medicine (NLM) by trained coders who assign 10–15 such codes per biomedical article.

On top of this joint pre-training approach, we introduce a novel application of the BERT transformer architecture to further fine-tune the word and concept embeddings with a classification objective that discriminates "co-occurring" MeSH codes (from PubMed citations) from random pairs of MeSH terms. Here, co-occurrence refers to the two terms appearing in the same citation as determined by human coders who annotated it. That is, the positive examples are derived from a set of MeSH codes assigned to a sampled biomedical citation, and negative examples are random pairs of MeSH codes from the full terminology. Intuitively, if two codes are assigned to the same article, they are clearly related in some thematic manner. Besides this, we also derive additional positive pairs from the MeSH hierarchy by choosing those that are separated by at most two hops. "Jointness" is incorporated here by appending each code with its preferred name. Specifically, in the two-sentence input mode for BERT, each sentence is a code and its preferred name appended next to it. This code pair "relatedness" classification task further transfers signal between words and codes leading to demonstrable gains in intrinsic evaluations of resulting word embeddings. These steps are captured through (S3) in Figure 3.1. We present more specifics and implementational details in Sections 3.2 and 3.3.

The resulting embeddings are evaluated for their semantic representativeness using intrinsic evaluations with well-known datasets and also through qualitative analyses. The results show a substantial improvement in evaluations compared to prior best approaches. Overall, we present an effective novel application of transformer architectures originally developed for contextualized embeddings to improve static word embeddings through joint learning and fine-tuning word/concept embeddings.

3.2 Training Strategies for BERT-CRel

For (S1) and (S2) (in Figure 3.1), to carry out conventional pre-training and learn word/concept embeddings, we seek a free publicly available resource that comes with

annotations of biomedical concepts from a well-known terminology. This is readily made available through the PubTator [Wei et al., 2019] initiative from BioNLP researchers at the NLM. It has over 30 million PubMed citations (abstracts and titles from the 2020 baseline) and over 3 million full-text articles with high-quality annotations for genes (and their variants), diseases, chemicals, species, and cell lines. Our choice for the concept vocabulary was MeSH (2020 version) because the diseases and chemicals from PubTator have mappings to MeSH codes; furthermore, with nearly 30K concepts, MeSH is fairly representative of the general concept space in biomedicine. Additionally, MeSH concepts also come with brief definitional blurbs describing their meaning in general-purpose English (more later). We use these blurbs in pre-training especially for MeSH concepts that do not appear in PubTator annotations.

3.2.1 Pre-training static word embeddings

Pre-training step (S2) in Figure 3.1 uses fastText [Bojanowski et al., 2017b] for training static embeddings. FastText improves upon the basic skip-gram model by learning word embeddings as compositions of constituent character n-grams and their representations. The corpus for this is a sample subset (1%) of the PubTator dataset such that each PubMed citation sampled contains at least two annotations with MeSH concepts. MeSH codes from the annotations are inserted immediately after the corresponding concept spans in texts. To distinguish MeSH codes from regular words, we represent them as `ConceptCode||SourceVocab`, essentially a concatenation of the concept code and `SourceVocab`, an abbreviation for the source terminology. Although MeSH codes are unique enough, we chose this formatting to be amenable to a general setup with multiple terminologies. With this, consider the example title: "A multi-centre international study of salivary hormone oestradiol and progesterone measurements in ART monitoring." With the corresponding codes inserted, this title is transformed into: *A multi-centre international study of salivary hormone oestradiol* `D004958MeSH` *and progesterone* `D011374MeSH` *measurements in ART monitoring.* The two codes inserted next to "oestradiol" and "progesterone" were identified by PubTator.

Our goal is to imbue a two-way semantic signal between all types of concepts and related words. However, only a portion of the MeSH headings (9,415 out of 29,640) is referred to in the PubTator annotations. Hence, we ought to supplement PubTator based training data with additional texts that contain the missing MeSH codes. This is where we exploit the definitional information of concepts provided

by MeSH creators. With this, each MeSH concept provides a textual snippet for fastText. The snippet supplied is the concatenation of the preferred name, source code, and definition of the concept. For example, the MeSH code D008654 for the concept Mesothelioma results in the textual input: "**Mesothelioma** `D008654MeSH` *A tumor derived from mesothelial tissue (peritoneum, pleura, pericardium). It appears as broad sheets of cells, with some regions containing spindle-shaped, sarcoma-like cells and other regions showing adenomatous patterns. Pleural mesotheliomas have been linked to exposure to asbestos.*" This means, for codes that may never show up in any annotated PubTator documents, we guarantee a single document that is constructed in this manner tying the concept with words that are highly relevant to its meaning. These are the "serialized concept definitions" referred to in the (S1) component of Figure 3.1. These additional documents are supplied in an in-order traversal sequence of the MeSH hierarchy to fastText as a "mega" document where adjacent documents correspond to hierarchically related concepts.

3.2.2 Fine-tuning embeddings for concept relatedness classification

Component (S3) of Figure 3.1 involves model BERT-CRel to further fine-tune word and concept embeddings by capturing concept relatedness (CRel). It is a canonical transformer [Vaswani et al., 2017] model for a binary classification task. In essence, this is repurposing the BERT architecture without any pre-training for the language modeling objective; we retain the classification objective with an additional feedforward layer and sigmoid unit feeding off of the `[CLS]` token output. The input is a pair (m^i, m^j) of "related" MeSH concepts in the two-sentence input mode following the format

$$\texttt{[CLS]}\ m^i\, w_1^i \cdots w_n^i\ \texttt{[SEP]}\ m^j\, w_1^j \cdots w_m^j\ \texttt{[SEP]}$$

where m^i and m^j are related MeSH codes and $w_1^i \cdots w_n^i$ is the *preferred name* of m^i. `[CLS]` and `[SEP]` are well-known special tokens used in BERT models.

Positive training pairs (m^i, m^j) are generated using two rules. Rule-1 deems the pair to be related if both codes were assigned to some document in the sample corpus C by coders at the NLM. More formally, the set of all such positive pairs

$$R_C = \bigcup_{c \in C} \{(m^i, m^j) : \forall_{i \neq j}\, m^i, m^j \in \mathcal{M}(c)\},$$

where $\mathcal{M}(c)$ is the set of MeSH concepts assigned to citation c. Rule-2 considers a pair to be related if the codes are connected by at most two hops in the directed-acyclic MeSH graph G_{MeSH}. These would capture parent/child, grand parent/child,

and sibling connections between concepts. Specifically,

$$R_{MeSH} = \{(m^i, m^j) : d^{G_{MeSH}}(m^i, m^j) \leq 2, \forall_{i \neq j}\, m^i, m^j \in G_{MeSH}\} \cup R_{SA}^{MeSH} \cup R_{PA}^{MeSH},$$

where d is graph distance, R_{SA}^{MeSH} is the set of "see also" relations, and R_{PA}^{MeSH} is the set of "pharmacological action" relations defined between MeSH concepts by the NLM. These auxiliary relations are not part of the MeSH hierarchy but are publicly available to mine. For instance, the concept *Multiple Myeloma* has a see-also link to the concept *Myeloma Proteins*, which in turn has a pharm-action connection to the concept *Immunologic Factors*. It is not difficult to see that these relations also capture strong semantic relatedness between concepts. $R_C \cup R_{MeSH}$ is the full set of positive relations used to fine-tune word/concept embeddings with BERT-CRel. To generate the same number of negative examples, we randomly sample the MeSH concept pairs across the entire vocabulary, retaining the term frequency distribution.

3.3 Methodology

3.3.1 fastText+: Adjustments for word/concept pre-training

As indicated in Section 3.2.1 we use fastText [Bojanowski et al., 2017b] for the initial pre-training on the concept-annotated corpus created through PubTator and MeSH definitional information. Building on the skip-gram model [Mikolov et al., 2013a], fastText additionally models and composes character n-grams to form word embeddings, thus accounting for subword information. This can capture relatedness among morphological variants and in exploiting regularities in lexical meaning manifesting in word forms through suffixes, prefixes, and other lemmata. It also helps in forming better embeddings on the fly for some unseen words (through the constituent character n-grams) instead of relying on the catch-all UNK embeddings that are typically used. However, we do not want this subword decomposition to occur when dealing with concept embeddings because they are atomic units, and there is no scope for unseen tokens given we know the full code set upfront. Hence we impose the following two constraints.

1. Concept codes (e.g., D002289MeSH) are not decomposed into subword vectors; the model thus is forced to recognize the concept codes from the corresponding tokens by the unique format `ConceptCode||SourceVocab`.

2. The output vocabulary must contain the full set of concept codes (here, MeSH descriptors) regardless of their frequencies in the corpus unlike the default case where fastText imposes a minimum frequency for character n-grams.

For the full implementation details of fastText, we refer to the original paper by Bojanowski et al. [Bojanowski et al., 2017b]. Here, we only highlighted the modifications we sought to handle concept tokens. This adapted version of fastText is henceforth called fastText$^+$ in this chapter. Table 3.1 lists the empirically chosen hyperparameters for training fastText for our concept-annotated corpus. Note that the dimensionality of word vectors (*dim*) is intentionally chosen to be divisible by 12, the number of transformer blocks in the subsequent fine-tuning phase through the BERT architecture.

Table 3.1: Model hyperparameters for *fastText* training

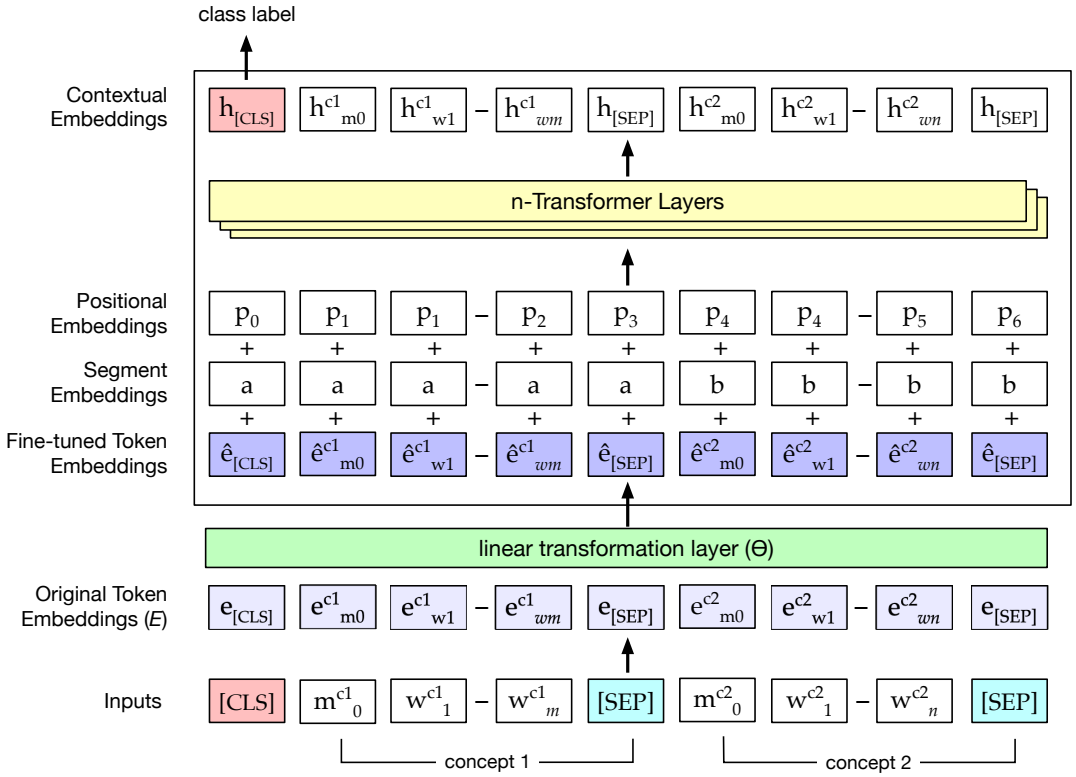
Parameters	Values
minCount (required number of word occurrences)	5
dim (dimensionality of word vectors)	396
ws (size of context window)	30
epoch (number of epochs)	5
minn (min. length of character ngrams)	3
maxn (max. length of character ngrams)	6

3.3.2 Optimization details for post-processing specialization

We introduced BERT-CRel in Section 3.2.2 to further fine-tune pre-trained word/-concept embeddings learned with fastText$^+$. BERT-CRel is a shallow transformer encoder, which reads the textual representations of a concept pair and predicts their relatedness as a binary classification task. Note that is unlike the original purpose of BERT — to build contextualized embeddings. Furthermore, we do not use any pre-trained BERT model (such as SciBERT) because our framework does not suit the *WordPiece* tokenization that is typically used. What is available at this stage are the pre-trained word/concept embeddings from fastText$^+$. So we repurpose BERT as shown in Figure 3.2. Here we apply a linear transformation on the initial pre-trained static embeddings.

The input texts are tokenized using a simple white space-based split function followed by a text clean-up process. Initially, we load the original token embeddings with the pre-trained static embeddings from fastText$^+$. We provide examples of concept pairs (as outlined in Section 3.2.2) along with their binary relatedness labels to the model. Each input sequence starts with `[CLS]`, followed by a pair of concept phrases (code token followed by the preferred name for each concept) separated by

Figure 3.2: BERT-CRel: concept relatedness classification model



[SEP]. While training, the first `[CLS]` token collects all the features for determining the relatedness label between two concepts. We add a linear transformation layer following the original token embeddings to apply subtle adjustments to the given token embeddings. This linear layer is initialized with the identity matrix.

Two-step optimization

We take a two-step optimization approach where during the first step, we focus on optimizing the classification model before fine-tuning the pre-trained embeddings. To accomplish this, during the first step, only the transformer layers are updated with the specified range of learning rates $[lr^{\alpha}_{\max}, lr^{\alpha}_{\min}]$, starting with $lr^{\alpha}_{\max}$ and decreasing with time. Once the optimizer reaches the minimum learning rate ($lr^{\alpha}_{\min}$), we initiate the next optimization schedule by applying another range of learning rates $[lr^{\beta}_{\max}, lr^{\beta}_{\min}]$ and start computing gradients of the linear transformation layer. This new range is to update the linear transformation layer (Θ) and the pre-trained embeddings from fastText^{+} (E).

This second step is implemented using multi-stage annealing within learning rate range $[lr^{\beta}_{\max}, lr^{\beta}_{\min}]$. That is, we first update the linear layer with fixed embeddings

from the previous stage. This stops when the learning rate decreases to lr^{β}_{min}. At this point, the embeddings are updated ($E_{i+1} = \Theta_i E_i$) at once using the state of the parameters and Θ_{i+1} is set back to I (identity matrix). The learning rate is then reset to a higher value that starts at $lr_{i+1} = \gamma^{i+1} \cdot lr^{\beta}_{\text{max}}$ ($\gamma < 1$); and the process of updating Θ_{i+1} continues with fixed E_{i+1}. This alternating process of freezing E and updating Θ and then updating E after reaching minimum learning rate is repeated until lr_{i+1} reaches lr^{β}_{min} (which is the default manner in which PyTorch's *ReduceLROnPlateau* operates). E_1 is the pre-trained set of embeddings from fastText^{+} and Θ_1 is initialized with I. Intuitively, this lets the learning rate bob within the $[lr^{\beta}_{\text{max}}, lr^{\beta}_{\text{min}}]$ range inspired by cyclical learning rate schedules [Smith, 2017] designed to overcome saddle point plateaus.

Implementation details

We use PyTorch and HuggingFace's *BertForSequenceClassification* model to implement BERT-CRel. The model is evaluated on the validation set every 10,000 steps. Binary cross-entropy is the loss function used. We save the improved word embeddings of the best model according to the UMNS dataset (more later) evaluation results. We use *ReduceLROnPlateau* with the initial learning rate lr^{α}_{max} = `3e-5` and the minimum learning rate lr^{α}_{min} = `2e-5` with decay $\gamma = 0.9$ for the initial step of updating just the transformer layers. The scheduler reduces learning rates by γ once it sees no improvement on the validation results three consecutive times. While fine-tuning static embeddings, during the multi-stage annealing process, we set the learning rates from `3e-5` (lr^{β}_{max}) to `1e-5` (lr^{β}_{min}) with $\gamma = 0.8$.

3.4 Evaluation Scenarios

3.4.1 Qualitative evaluations

As a qualitative evaluation, we examine the representation learning quality of the embeddings produced by BERT-CRel. This is done in the context of other prior approaches for generating biomedical word embeddings. For the sake of comparison, we use the same set of biomedical query terms (usually noun phrases) used in Wang et al.'s study [Wang et al., 2018]. The task is to retrieve five *closest* terms in the word/concept embedding space to each query term and assess how related they actually are to the query term. For example, given the word 'aspirin,' we expect to see related terms such as 'blood thinner', 'anti-inflammatory drug', or 'clopidogrel' (shares functionality with aspirin). These typically include hyponyms, hypernyms, or

co-hyponyms. Besides terms by Wang et al. [Wang et al., 2018], we also examine the neighbors of most popular acronyms used in biomedical literature; we find up to five closest terms to the acronym and the corresponding MeSH codes. We used two available algorithms for acronym extraction, the Schwartz and Hearst algorithm [Schwartz and Hearst, 200] and ALICE [Ao and Takagi, 2005], and obtained 331 most frequently used acronyms in the PubMed citations for this purpose. We note that for multi-word terms, we simply take the average of constituent word embeddings before retrieving the closest words and concepts.

3.4.2 Quantitative evaluations

Intrinsic evaluations for word embeddings examine the quality of representativeness that is independent of downstream tasks. We use publicly available reference datasets for measuring the relatedness between biomedical concepts. With the reference standards, we can evaluate the quality of vector representations for computing relatedness between biomedical terms compared to human judgments. Each instance within a dataset consists of a pair of biomedical concepts and the corresponding relatedness score judged by human experts such as physicians and medical coders. Some of the datasets also provide corresponding UMLS concept codes. The terms that occur in these datasets are more often seen in the biomedical domains than in other fields. Table 3.2 enumerates the reference datasets we use, where the middle column indicates the number of concept pairs within each dataset.

Table 3.2: Datasets of biomedical concept pairs for similarity/relatedness evaluations.

Dataset name (alias)	Size	Judged by
UMNSRS-Sim (`UMNS`) [Pakhomov et al., 2010]	566	medical residents
UMNSRS-Rel (`UMNR`) [Pakhomov et al., 2010]	587	medical residents
MayoSRS (`MAYO`) [Pakhomov, 2018]	101	physicians and coders
MiniMayoSRS (`MMY[P/C]`) [Pedersen et al., 2007]	29	physicians and coders
Pedersen's (`PDS[P/C]`) [Pedersen et al., 2007]	30	physicians
Hliaoutakis' (`HLTK`) [Hliaoutakis, 2005]	36	mostly physicians

We expand the instances by linking the concepts to corresponding MeSH codes. We utilize the UTS (UMLS Terminology Services) API to find the most similar MeSH codes to the concepts. When available, we exploit the UMLS codes provided along with the datasets; otherwise, we query by the concept name. We use the cosine vector similarity to measure the semantic match between two concepts/terms. Here also, if the concept name is composed of multiple words, we take the mean vector of its

constituent word representations. If the word is an OOV, the `[UNK]` token vector learned in BERT-CRel training process is used. If `[UNK]` token is not available, for the fastText$^+$ pre-trained embeddings, we assume the relatedness score of the pair to be 0 as default. Finally, a ranked list of concept pairs based on cosine scores is compared against the ground truth expert ranking using the Spearman's rank correlation coefficient ρ.

Table 3.3: Five most similar terms to selected biomedical concepts trained from different models and textual resources, MeSH names: (i) Diabetes Melitus (ii) Diabetes Mellitus, Type 2 (iii) Ulcer (iv) Peptic Ulcer (v) Stomach Neoplasms (vi) Colorectal Neoplasms (vii) neoplasms (viii) Dyspnea (ix) Pharyngeal Diseases (x) Opioid-Related Disorders (xi) Aspirin

Query term	**fastText**$^+$ (PubMed)	**BERT-CRel** (PubMed)	Wang et al.'s (EHR)	Wang et al.'s (PMC)	GloVe (Wiki+Giga)	W2V (Google News)
diabetes	D003920 [i]	D003920 [i]	mellitus	cardiovascular	hypertension	diabetics
	mellitus	mellitus	uncontrolled	nonalcoholic	obesity	hypertension
	nondiabetes	nondiabetes	cholesterolemia	obesity	arthritis	diabetic
	diabetic	D003924 [ii]	dyslipidemia	mellitus	cancer	diabetes_mellitus
	D003924 [ii]	diabetic	melitis	polycystic	alzheimer	heart_disease
peptic ulcer disease	D014456 [iii]	D014456 [iii]	scleroderma	gastritis	ulcers	ichen_planus
	ulcers	D010437 [iv]	duodenal	alcoholism	arthritis	Candida_infection
	D010437 [iv]	ulcers	crohn	rheumatic	diseases	vaginal_yeast_infections
	gastroduodenitis	D013274 [v]	gastroduodenal	ischaemic	diabetes	oral_thrush
	ulceration	gastroduodenitis	diverticular	nephropathy	stomach	dermopathy
colon cancer	colorectal	D015179 [vi]	breast	breast	breast	breast
	D015179 [vi]	colorectal	ovarian	mcf	prostate	prostate
	cancers	cancers	prostate	cancers	cancers	tumor
	D009369 [vii]	colorectum	postmenopausally	tumor_suppressing	tumor	pre_cancerous_lesion
	colorectum	D009369 [vii]	caner	downregulation	liver	cancerous_polyp
dyspnea	D004417 [viii]	D004417 [viii]	palpitations	sweats	shortness	dyspnoea
	dyspnoea	dyspnoea	orthopnea	orthopnea	breathlessness	pruritus
	shortness	shortness	exertional	breathlessness	cyanosis	nasopharyngitis
	breathlessness	breathlessness	doe	hypotension	photophobia	symptom_severity
	dyspnoeic	dyspnoeic	dyspnoea	rhonchi	faintness	rhinorrhea
sore throat	pharyngitis	pharyngitis	scratchy	runny	shoulder	soreness
	throats	D010608 [ix]	thoat	rhinorrhea	stomach	bruised
	pharyngolaryngitis	pharyngolaryngitis	cough	myalgia	nose	inflammed
	tonsillopharyngitis	pharyngotonsillitis	runny	swab_fecal	chest	contusion
	rhinopharyngitis	rhinopharyngitis	thraot	nose	neck	sore_triceps
opioid	opioids	opioids	opiate	opioids	analgesic	opioids
	opiate	opiate	benzodiazepine	nmda_receptor	opiate	opioid_analgesics
	nonopioid	nonopioid	opioids	affective_motivational	opioids	opioid_painkillers
	nonopioids	morphine	sedative	naloxone_precipitated	anti-inflammatory	antipsychotics
	D009293 [x]	nonopioids	polypharmacy	hyperlocomotion	analgesics	tricyclic_antidepressants
aspirin	D001241 [xi]	D001241 [xi]	ecotrin	chads	ibuprofen	dose_aspirin
	acetylsalicylic	acetylsalicylic	uncoated	vasc	tamoxifen	ibuprofen
	nonaspirin	nonaspirin	nonenteric	newer	pills	statins
	aspirinate	aspirinate	effcient	cha	statins	statin
	aspirinated	antiplatelet	onk	angina	medication	calcium_supplements

Table 3.4: Nearest neighbors of the frequently used biomedical abbreviations in BERT-CRel embeddings

Acronyms	Close to Word	Close to Code
MRI (MeSH: D008279 Name: Magnetic Resonance Imaging)	imaging	D066235 (Fluorine-19 Magnetic Resonance Imaging)
	mris	D038524 (Diffusion Magnetic Resonance Imaging)
	weighted	D000074269 (Resonance Frequency Analysis)
	tesla	D000081364 (Multiparametric Magnetic Resonance Imaging)
	magnetic	D017352 (Echo-Planar Imaging)
BMI (MeSH: D015992 Name: Body Mass Index)	overweight	D065927 (Waist-Height Ratio)
	waist	D049629 (Waist-Hip Ratio)
	circumference	D049628 (Body Size)
	whr	D064237 (Lipid Accumulation Product)
	D009765 (Obesity)	D001823 (Body Composition)
CT (MeSH: D014057 Name: Computed Tomography)	scans	D014056 (Tomography, X-Ray)
	tomographic	D055114 (X-Ray Microtomography)
	computed	D000072078 (Positron Emission Tomography Computed Tomography)
	scan	D055032 (Electron Microscope Tomography)
	tomography	D014055 (Tomography, Emission-Computed)
NO (MeSH: D009569 Name: Nitric Oxide)	significant	nitric
	any	oxide
	did	inos
	not	nos
	both	D013481 (Superoxides)
ROS (MeSH: D017382 Name: Reactive Oxygen Species)	D017382 (Reactive Oxygen Species)	ros
	oxidative	oxidative
	h2o2	h2o2
	oxidant	D006861 (Hydrogen Peroxide)
	D013481 (Superoxides)	D013481 (Superoxides)
PCR (MeSH: D016133 Name: Polymerase Chain Reaction)	polymerase	D054458 (Amplified Fragment Length Polymorphism Analysis)
	qpcr	D020180 (Heteroduplex Analysis)
	primers	D022521 (Ligase Chain Reaction)
	taqman	D060885 (Multiplex Polymerase Chain Reaction)
	rt	D024363 (Transcription Initiation Site)
AD (MeSH: D000544 Alzheimer Disease)	D000544 (Alzheimer Disease)	alzheimer
	alzheimer	alzheimers
	alzheimers	ad
	abeta	abeta
	dementias	D003704 (Dementia)

3.5 Results and Discussion

We first discuss observations from the qualitative assessments conducted. Table 3.3 shows the five most related terms to a given biomedical term across several available embeddings. Sample query terms are in three groups: disease name, symptoms, and drug names. In the table, the fastText$^+$ column denotes the results obtained from the pre-trained static embeddings with the joint learning of word and concept embeddings (Section 3.3.1). The BERT-CRel column indicates the results obtained from the improved static embeddings by the concept-relatedness classification task with the BERT encoder model. We notice that both of our approaches (fastText$^+$ and BERT-CRel) surface a coherent set of words and concepts related to the query terms. Also, corresponding MeSH codes returned allow us to interpret input terms in an indirect but more precise way. For example, D015179 (Colorectal Neoplasms) exactly matches the query term "colon cancer" while other words are indicating relevant words but may not be as specific (e.g., "cancers"). The returned words for the query term "sore throat" also demonstrate better ability in finding related terms. We were able to retrieve specific related disease names such as *pharyngitis*, *pharyngolaryngitis*, and *rhinopharyngitis*. The more primitive methods do not produce terms that are as tightly linked with the theme conveyed by query terms compared with our methods. Between our fastText$^+$ and BERT-CRel rankings, there is a non-trivial overlap of terms, but the relative order seems to have changed due to the fine-tuning process. We see more examples where BERT-CRel ranks MeSH codes that precisely match the query term higher than the fastText$^+$ ranking. Also, BERT-CRel appears to surface related terms that are not just morphological variants of the query term. For example, for the "opioid" query, it returns morphine, which is not returned in any other methods. However, other methods also seem to surface some interesting related terms such as "analgesics", a broader term that refers to pain relievers.

Table 3.4 shows the mapping between some commonly used biomedical acronyms and their nearest terms; the second column lists terms that are close to the acronym, and the third column contains terms close to the corresponding MeSH code. The results in the third column show how the distributed representations of MeSH codes are affected by the training sources. As mentioned earlier, PubTator annotates biomedical concepts that only belong to the following categories: gene, mutation, disease names, chemical substances, and species. Consequently, the MeSH codes for some acronyms (e.g., MRI, BMI, CT, PCR) had to learn associated representations just from MeSH definitions and the BERT-CRel objective; their nearest neighbors, hence, tend to be other MeSH codes. However, other acronyms with enough annotation examples in the PubTator dataset (e.g., NO, ROS, AD) mapped to more of the related regular words. Among top five matches for AD and its MeSH code is "abeta" (stands for amyloid beta), the main component in plaques in brains of people with Alzheimer's disease.

We now focus on quantitative evaluations based on expert curated datasets in Table 3.2. MiniMayoSRS and Pedersen's datasets are judged by two different groups of experts: physicians and medical coders. We compare our model against several state-of-the-art methods across all the reference datasets. Table 3.5 shows the results

of our pre-trained embeddings (fastText$^+$) and the fine-tuned embeddings (BERT-CRel). The metric is Spearman's ρ comparing methods' rankings with human relevance scores. Before we delve into the scores, we note that the correlation coefficients may not be directly comparable in all cases. Most of the previous studies evaluated the models on a subset of the original reference standards. We specify the number of instances used in each evaluation in parentheses next to the score; a score without the number of instances means that the evaluation used the full dataset.

Table 3.5: Results of intrinsic evaluations measured with Spearman's correlation coefficient. Note, the number in parenthesis indicates the number of examples used for the evaluation (with the header row indicating the total number of instances in the original datasets). Scores without parenthesis use the full set of instances and top scores for each dataset are shown in bold font. The ranking for the word+MeSH rows is computed by the reciprocal rank fusion with the rankings generated by the "word" and "MeSH" embeddings.

Approach	UMNS (n=566)	UMNR (n=587)	MAYO (n=101)	MMYP (n=29)	MMYC (n=29)	PDSP (n=30)	PDSC (n=30)	HLTK (n=36)
Word2vec (baseline)	0.568	0.499	0.508	0.744	0.748	0.738	0.736	0.434
Wang et al. [Wang et al., 2018]	0.440	—	0.412	—	—	0.632	—	0.482
Park et al. [Park et al., 2019]	—	—	—	—	—	0.795	—	0.633
Chiu et al. [Chiu et al., 2016]	0.652 (459)	0.601 (561)	—	—	—	—	—	—
Zhang et al. [Zhang et al., 2019]	0.657 (521)	0.617 (532)	—	—	—	—	—	—
Yu et al. [Yu et al., 2016, 2017b]	0.689 (526)	0.624 (543)	—	0.696 (25)	0.665 (25)	—	—	—
Henry et al. [Henry et al., 2019]	0.693 (392)	0.641 (418)	—	0.842	0.816	—	—	—
fastText$^+$ (word)	0.654	0.609	0.630	0.851	0.853	0.820	0.831	0.513
fastText$^+$ (MeSH)	0.648	0.568	0.608	0.739	0.701	0.612	0.612	**0.846**
fastText$^+$ (word+MeSH)	0.689	0.623	0.685	0.836	0.832	0.756	0.769	0.753
BERT-CRel (word)	0.683	**0.643**	0.667	**0.890**	0.844	**0.850**	**0.849**	0.537
BERT-CRel (MeSH)	0.659	0.576	0.610	0.710	0.712	0.678	0.678	0.823
BERT-CRel (word+MeSH)	**0.708**	0.637	**0.695**	0.847	**0.857**	0.803	0.835	0.743

As indicated in Section 3.4.2, we use all instances of all datasets in the evaluation; for any OOV term, we use a fallback mechanism that returns a score either using the `[UNK]` embedding or the default score 0. We believe this is a more robust way of evaluating methods instead of selectively ignoring some instances[4]. All rows except those that involve "MeSH" in the first column use word-embedding based rankings. Rows that involve MeSH are comparisons that directly compute cosine score with the MeSH code embedding generated by our method. Rows with "word+MeSH" modeling involve reciprocal rank fusion [Cormack et al., 2009] of rankings generated by "word" and "MeSH" configurations in the previous two rows.

Digging into the scores from Table 3.5, with very few exceptions, BERT-CRel correlates better with human judgments compared with fastText$^+$ across datasets,

[4]In our observation, this was mostly done by other efforts when dealing with terms that are very rare, hence OOV, and hence cannot be readily compared for lack of a proper representation. To some extent, we overcame OOV by using MeSH definitions in fastText$^+$ and the concept pair relevance setup in BERT-CRel

and improves by around 2.5% in ρ on average. The most comparable scores with previous efforts are from the third row from the end (BERT-CRel with "word" level comparison) given they are word-based measures. This BERT-CRel configuration wins outright for the UMNR dataset even when compared to methods that fuse rankings from word and concept level scores. It also is better than almost all other prior methods across all datasets even when they use selected subsets from the full dataset. Our effort provides the most robust evaluation by exhaustively considering all instances across all well-known datasets developed for evaluating embeddings. Overall, we demonstrate that jointly learning word and concept embeddings by leveraging definitional information for concepts provides better embeddings; further enhancing these embeddings by exploiting distributional correlations across concepts (obtained from MeSH co-occurrences and hierarchical links), through transformer-based classifiers, offers more noticeable gains in embedding quality.

3.6 Related Work

In this section, we briefly discuss previously proposed methods for training domain-specific word/concept embeddings, which we evaluated for this study as shown in Table 3.5, then, in the following section, we conclude this chapter with the summary of this study.

Wang et al. [Wang et al., 2018] trained word embeddings on unstructured electronic health record (EHR) data using fastText. The subword embeddings of the fastText model enabled them to obtain vector representations of OOVs. Park et al. [Park et al., 2019] proposed a model for learning UMLS concept embeddings from their definitions combined with corresponding Wikipedia articles [Park et al., 2019]. The degree of relatedness between two concepts is measured by the cosine similarity between the corresponding concept vectors. Zhang et al. [Zhang et al., 2019] proposed a similar method to ours for preparing the training corpus. They also used the MeSH RDF-based graph from which they sampled random paths to generate sequences of MeSH terms and used them to train word embeddings; in our work, we traverse the MeSH hierarchy to obtain single in-order path of MeSH concepts of which each node is represented by its preferred concept name, unique MeSH code, and its definition. Yu et al. [Yu et al., 2017b] also trained UMLS concept embeddings and fine-tuned them using a "retrofitting" method developed by Faruqui et al. [Faruqui et al., 2015]. They improved pre-trained embeddings using concept relationship knowledge defined in the UMLS semantic lexicon. Among different relationships, they claim that RO (*has other relationship*) and RQ (*related and possibly synonymous*) relationships returned the most improvements on the UMNSRS evaluation dataset. Henry et al. [Henry et al., 2019] computed several association measures, such as *mutual information*, with concept co-occurrence counts and measured the semantic similarity and relatedness between concepts. Overall, the Pearson's Chi squared association measure (χ^2) performed the best.

3.7 BERT-CRel Summary

In this effort, we proposed a method for training and improving static embeddings for both words and domain-specific concepts using a neural model for the concept-relatedness classification task. To incorporate the relational information among biomedical concepts, we utilize document metadata (i.e., MeSH assignments to the PubMed articles) in corpus and the hierarchical relationships of the concepts defined in a controlled vocabulary (i.e., MeSH hierarchy structures). Our approach achieved the best performances across several benchmarks. Qualitative observations indicate that our methods may be able to nudge embeddings to capture more precise connections among biomedical terms.

Our proposed method for training and improving static embeddings can be utilized in many BioNLP tasks. The use of joint word/concept embeddings can potentially benefit neural models that need mutual retrievability between multiple embeddings spaces. In one of our recent studies (also presented in Chapter 6), we leveraged embeddings generated with these methods in a neural text summarization model for information retrieval [Noh and Kavuluru, 2020b]. Exploiting the joint embeddings of words and MeSH codes, we were able to summarize a document into a sequence of keywords using either regular English words or MeSH codes that are then compared with query words and codes.

Chapter 4 JEREN: Joint Learning for Biomedical NER and Entity Normalization: Encoding Schemes, Counterfactual Examples, and Zero-Shot Evaluation

This chapter explores the methods for named entity recognition (NER) and entity normalization (EN) in free-texts. In pursuit of bridging the vocabulary mismatch, recognizing biomedical entities and providing the linkage into standardized concept codes is crucial in understanding the vocabulary semantics.

NER (or just ER) and EN form an indispensable first step to many biomedical natural language processing applications. In biomedical information science, recognizing entities (e.g., genes, diseases, or drugs) and normalizing them to concepts in standard terminologies or thesauri (e.g., Entrez, ICD-10, or RxNorm) is crucial for identifying more informative relations among them that drive disease etiology, progression, and treatment. In this effort we pursue two high level strategies to improve biomedical ER and EN. The first is to decouple standard entity encoding tags (e.g., "B-Drug" for the beginning of a drug) into type tags (e.g., "Drug") and positional tags (e.g., "B"). A second strategy is to use additional counterfactual training examples to handle the issue of models learning spurious correlations between surrounding context and normalized concepts in training data. We conduct elaborate experiments using the MedMentions dataset, the largest dataset of its kind for ER and EN in biomedicine. We find that our first strategy performs better in entity normalization when compared with the standard coding scheme. The second data augmentation strategy uniformly improves performance in span detection, typing, and normalization. The gains from counterfactual examples are more prominent when evaluating in zero-shot settings, for concepts that have never been encountered during training.

4.1 Biomedical NER and EN

Biomedical information extraction (BIE) from free text is at the heart of many downstream biomedical natural language processing (BioNLP) applications including knowledge discovery, search systems, and Question-Answering (QA) models. Niche applications such as automatic clinical cohort selection and evidence based medicine through patient similarity computing may also rely on the output of BIE systems. Given an input text (sentence or paragraph), at a high level BIE consists of two mains steps: (1) spotting biomedical entities (e.g., genes, diseases, and drugs) in text and linking them to standardized concepts in ontologies, terminologies, or other thesauri (e.g., Entrez, ICD-10, and RxNorm). (2) identifying any relations between concepts identified in step (1) as asserted in the text. Once these inter-concept relations are identified, they can be stored in structured databases as knowledge graphs. As more and more concepts and inter-concept relations are being discussed in scientific literature, clinical text, and even social media these days, BIE is the only scalable way of curating relational information being presented in textual data. There are obvious caveats regarding BIE methods given any NLP method has associated accuracy is-

sues. However, if the same relation is obtained from multiple research articles, risks associated with imperfect methods can be alleviated.

4.1.1 Components of NER and EN

Our current effort concerns step (1) of BIE discussed in the previous paragraph. This step is actually composed of two different but related tasks:

1. First is to identify spans of text in the input representing an entity of interest. This means determining the exact location where the span starts and ends (via character offsets) and then assigning an entity type (e.g., drug, disease, or gene) that typically comes from a set of predetermined fixed types. This *mention detection* and *entity typing* subtasks are together typically called named entity recognition.

2. Often the same entity is referred by multiple aliases (text strings) that essentially point to the same biomedical concept. Abbreviations or other short forms and synonyms are obvious sources of aliases. Mapping equivalent aliases to a unique biomedical entity, concept, or code in a standard terminology (e.g., RxNorm, NDC, Multum, Micromedex for drugs) is often called entity normalization (EN), entity linking, or concept mapping.

To summarize, NER consists of Mention Detection (MD) and Entity Typing (ET); and EN consists of mapping the span detected to an actual concept in a standardized terminology. We note MD and ET go together in the sense that identifying a span as representing an entity without figuring out the type of such an entity is mostly not meaningful. The EN step is also crucial because just knowing something is a drug may not be enough and any downstream task can only operate in a concrete way if we also identify the exact drug, by mapping to a standard terminology. Henceforth we refer to the standardized entities (unique codes in a terminology) as *concepts* and typed spans of text as just *entities*. As such, our overall task in this chapter can be simply stated as identifying entities and mapping them to concepts in an input text.

4.1.2 Challenges in biomedical NER/EN

Conventionally, BIE systems handle the NER and EN tasks independently and sequentially in a pipeline setup. That is, entity mentions (spans) are detected first along with their types. Subsequently, those spans are mapped to concepts in a terminology. Given the type is already known before EN happens in this setup, one needs to typically look for concepts that satisfy the type constraint set by the outcome of the preceding ER task. A well-known issue of this pipeline approach is the error propagation over the series of tasks. That is, errors made in mention detection or entity typing will automatically snowball to create errors in the EN step. Another missed opportunity in such a pipeline setup is more effective learning of features (and associated weights) that may be shared and tuned more effectively across multiple tasks simultaneously (e.g., using a single objective function or shared parameters).

Pipelines inherently involve separate models for each constituent task and hence operate in disparate feature spaces that do not share any predictive signal. Some recent BIE efforts still seem to rely on this pipeline setup. However, multi-task models and joint approaches are also gaining popularity in the general NLP community and more recently in BioNLP too.

Another limiting factor is the target terminology size for the EN task. As the number of concepts increases, it becomes prohibitive to create high coverage training datasets. This is to be expected as manual efforts in biomedicine are more complex needing expert time, when compared with similar tasks in general domains where crowdsourcing is popular. Also, in terms of methodology, the *softmax* computation for the simpler multi-class modeling becomes very expensive with large target concept spaces. For example, the Unified Medical Language System (UMLS) Metathesaurus, one of the commonly used biomedical terminologies for entity annotation tasks, contains over 4.4 million concepts (as of 2020). The MedMentions [Mohan and Li, 2019] NER and EN dataset used in our effort is considered the largest resource in biomedicine for this task and has 352,496 annotated mentions; still that only covers $< 1\%$ $(= 34,724)$ of the full UMLS concept space. So on average there are around 10 instances for each of the unique concepts covered in the dataset. But we also notice that the test data split of MedMentions has concepts that are never encountered during training, leading to inevitable zero-shot scenarios. The sparsity of having very few or no training examples for a large portion of the target concept space can lead to overfitting outcomes with complex nonlinear models where spurious correlations between concepts and surrounding textual artifacts are sort of *memorized* by the model .

We handle error snowballing issues with a joint modeling approach that uses both shared parameter spaces and combined objective function. We address the sparsity concerns with two different strategies. The first is to experiment with a decoupled tagging scheme for representing training data for NER where type tags and positional tags are treated separately. The other strategy uses additional counterfactual training examples derived from the original training dataset to break spurious correlations between concepts and contextual artifacts. Next, we provide some related work pertinent to our contribution.

4.2 Related Work

Before the prevalent use of neural methods for NER, most prior approaches relied on feature engineering with rule-based heuristic decision models. NER features include word-level patterns (e.g., punctuation, presence of different special characters such as digits or capital letters, part-of-speech, or prefixes/suffixes of tokens) [Collins, 2002, Bick, 2004], list look-up features (e.g., gazetteer, lexicon, or dictionary) [McDonald, 1993, Rau, 1991]. *TaggerOne* [Leaman and Lu, 2016] which is widely used in biomedical NER utilizes a semi-Markov model with carefully designed NER features such as the ones described earlier. *Dictionary matching*, based on the string-matching methods, was another popular choice [Wei and Kao, 2011, Hakenberg et al., 2011, Wang et al., 2019, Loureiro and Jorge, 2020] for the EN task. The matching scores are

computed between a mention and entities in the controlled vocabulary by leveraging the character n-grams or tf-idf features.

Since 2016, researchers have been proposing neural network-based approaches for EN. Kolitsas et al. [Kolitsas et al., 2018] prepared fixed dense vector representations of concepts using the pre-trained *Word2vec* embeddings, which were then compared with a mention representation. Yamada et al. [Yamada et al., 2016] also proposed a method to jointly learn the embeddings of words and concepts using a knowledge graph; they are then used in the named entity disambiguation process.

Recently, approaches utilizing neural language models for EN are burgeoning. Liu et al. [Liu et al., 2018] trained a simple neural language model on the next word prediction task, taking the character sequence as inputs. They used this language model for encoding input sequence before passing to an LSTM-CRF framework for the sequence labeling task. Wu et al.'s idea [Wu et al., 2019], which is conceptually similar to ours, uses the popular BERT transformer [Devlin et al., 2019] to model a concept's representation using its title and short description, whereas our model represents each concept by its alias and categorical entity type. For EN, similarity scores are computed across the concept representations of k nearest neighbors of the mention representation.

Several neural network-based models for MD have also been proposed. A common approach is to consider all possible spans in a document as potential mentions and compute the mention scores using a feed-forward neural network layer [Lee et al., 2017, Zhang et al., 2018]. Because of quadratic complexity ($\mathbb{O}(T^2)$) in the number of tokens T, they had to rely on a heuristic rule to prune out certain unlikely mention candidates during both inference and training.

At least two previous neural NER models are evaluated on the MedMentions dataset, which can be considered state-of-the-art approaches as of now. Loureiro et al.'s model [Loureiro and Jorge, 2020] uses a pre-trained neural language model (a BERT variant) to encode the input sentence. A BiLSTM-CRF module follows to identify a candidate entity span. Once the span is specified, the language model's hidden outputs for the span are pooled to construct its contextual representation. This representation provides the contextual matching feature for EN. Also, they used SimString [Okazaki and Tsujii, 2010] to compare the spans to concepts as in dictionary matching. They use pre-trained categorical entity embeddings (i.e., 21 semantic types and 18,425 CUI entities) for matching. Our proposed models differ in how the entity representations are structured and computed, which will be further discussed in the following sections. Also, our models do not rely on the dictionary matching methods. Wiatrak et al.'s model [Wiatrak and Iso-Sipila, 2020] adopts a similar architecture, a pre-trained language model followed by BiLSTM. They explore the use of hierarchical multi-task learning using ER as an auxiliary task, whereby they aim for a joint learning objective similar to ours.

4.3 High Level Strategies

Before we elaborate on specifics of our models, we describe our strategies to convey high level intuition.

4.3.1 Decoupled labeling scheme for NER

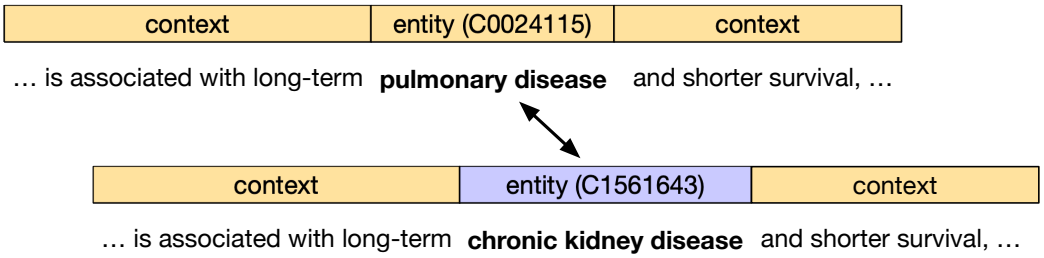
Sequence tagging problems are ubiquitous in NLP, especially for part-of-speech tagging and NER. Unlike for classification where class labels are assigned to each document, for NER (and other tagging problems), one needs to generate a label for each token in the input. We need a simple way to capture entity spans for NER. To this end, entity spans are typically indicated by tags (one per token) that correspond to the beginning of an entity and the rest of it. Tokens that are not part of any entity typically have a "other" or "outside" tag. There are some variants of this tagging scheme but they all have a set of entity related tags and an "outside" tag. NER training data is represented in this fashion to directly build models that learn to assign tags, which can be easily used to infer entity spans. The most popular one among such tagging schemes is the IOB (Inside-Outside-Beginning) format. The `B-` prefix indicates that the token is the beginning of a entity, and an `I-` prefix indicates that the token is inside an entity span. Other tokens are labeled with the `O` tag. An entity that is represented by a single token can be labeled with either `B-` or `I-` tags depending on the scheme used.

The IOB prefixes are typically combined with the entity type suffixes. For example, the `B-LOC` tag indicates the beginning token of a "location" entity. This is a natural way to model the learning process because tokens that indicate the beginning of location span may have different characteristics (e.g., different casing, prefixes) compared with tokens that represent the inside tokens of such an entity or even the beginning of another entity type, say, a disease. Combining the positional information ("B") and the type information ("LOC") will help the model capture these differences. However, sparsity issues with not enough training examples for a specific position/type combination tag may cause performance issues. The IOB prefix tags are specifically for determining span boundaries, while the following type suffixes (e.g., "LOC") are for entity typing. A sparsity related issue is the possibility of the model learning spurious associations between the lexical units and specific tags. We are not aware of any efforts that decouple MD (IOB tags) with ET (type tags) and hence we explores the use of decoupled IOB prefixes (i.e., B, I, O) compared against the conventional IOB tagging scheme (i.e., B-type*, I-type*, O).

4.3.2 Counterfactual training examples

Although MedMentions dataset we use provides a large number of annotated examples compared to the previously available datasets, $\sim 200k$ is considered "small" for training a language model for NER targeting a terminology of several million unique concepts. To overcome this limitation, we augment the observable examples by creating counterfactual examples as proposed by Zeng et al. [Zeng et al., 2020]. As illustrated in Figure 4.1, for each sentence example, we randomly choose one of the entity mentions and replace it with another entity that has the same semantic type of the original entity. The motivation of using this method is to eliminate the spurious correlations that may be established in a highly nonlinear model between the entity and its surrounding context.

Figure 4.1: Data augmentation with counterfactual examples



While data augmentation allows us to train the models with more examples, the increased compute needed with the extended name space of new concepts should be carefully managed. We do this with an augmentation factor (ξ), a model hyperparameter, that represents the additional number of counterfactual examples generated per each original example.

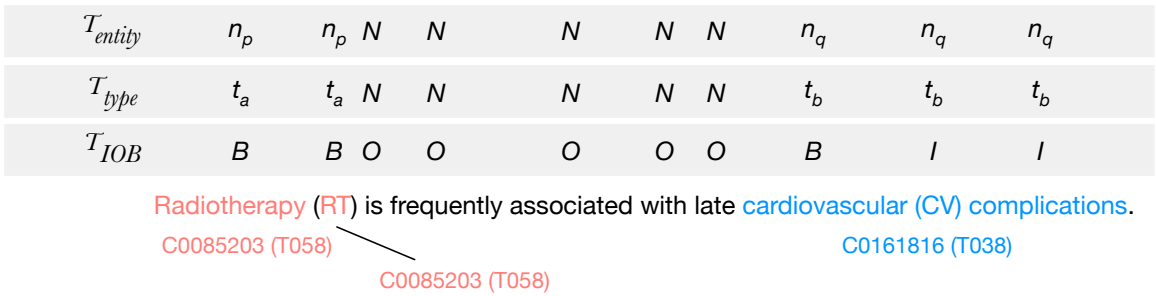
4.4 Methodology

We denote an input sequence as $\mathcal{X} = (x_1, x_2, \ldots, x_n)$, where x_i is the i-th token of a length n sequence. In our proposed approach, we have three different multiclass tag assignment problems one each for mention detection (MD), entity typing (ET), and entity normalization (EN). The sequence labeling task is to predict a tag y_i for each token x_i where $y_i \in \mathcal{T}_*$, where $*$ can be $\mathcal{IOB}$ for MD, *type* for the semantic type, and *concept* for EN. Here $\mathcal{T}_{IOB} = \{I, O, B\}$ has just three tags indicating mention boundaries. $\mathcal{T}_{type}$ has as many elements as there are semantic types (e.g., disease) and $\mathcal{T}_{concept}$ has as many elements as there are unique concepts in the target terminology. There is also an "other" class for $\mathcal{T}_{type}$ and $\mathcal{T}_{concept}$ for certain tokens that are not part of a named entity and hence not needing a type or concept. All three classification tasks are done at the token-level whereby each token is labeled with $y \in \mathcal{T}_*$. For an identified entity span (via IOB tags), a single entity type in $\mathcal{T}_{type}$ and a unique normalized concept in $\mathcal{T}_{concept}$ are chosen by majority vote across the corresponding per-token type and concept assignments in that span, respectively. A sample gold annotation for MD, ET, and EN are shown in Figure 4.2 where two unique concepts from the UMLS are annotated along with their semantic types in parentheses. The "O" tag is used for "other" annotations for type and concept tagging. We note that for the conventional NER scheme where MD and ET are combined, the annotation would naturally combine the IOB tag with the semantic type (e.g., B-t_a for the first word of the sentence).

4.4.1 Models

We devise two different joint neural models: IOBHI and ONETAG (Figure 4.3). Both architectures use a pre-trained language model, SciBERT [Beltagy et al., 2019], for encoding a sentence. To use a transformer based pre-trained language model as the base of a neural architecture is mostly standard practice at this time. Here we use

Figure 4.2: Example annotations for MD, ET, and EN ($n_a = C0085203$, $t_a = T058$, $n_b = C0161816$, $t_b = T038$)

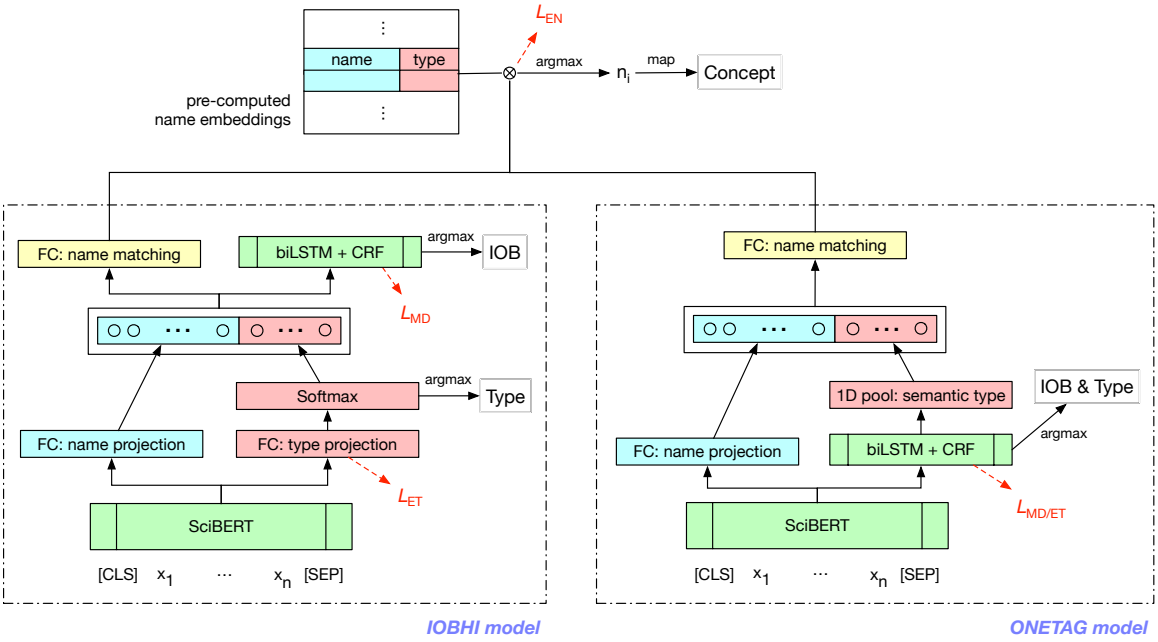


SciBERT which is trained on scientific literature both from biomedicine and computer science. At a high level, we pre-compute vector representations of concepts in the target terminology using their names (different synonymous aliases) and semantic types with the pre-trained SciBERT model (covered in Section 4.5.1). These vectors are then compared with vector representations of SciBERT hidden outputs for each token in an input sentence. Best matched concept from the target concept space is then chosen for every detected span. As may be expected, additional nuts and bolts level details are incorporated to ensure dimensions are reshaped as needed through feed forward layers as the input is passed through the network. IOBHI and ONETAG differ in details of how these representations are derived. With this basic setup in mind, we will move on to specific details.

In the IOBHI model (left section in Figure 4.3), we consider the decoupled IOB and type tagging task as discussed in Section 4.3.1. The IOB classifier is placed at the end of the network such that it can read in the processed name and type representations for identifying mention segmentations. In ONETAG (right section in Figure 4.3), we use the conventional IOB tagging scheme, where we have $2n+1$ target classes with n entity types (I and B tags for each semantic type). Note that the two networks are drawn in the same figure for the sake of brevity, but only the left or the right block is active at a time resulting in two different architectures.

Most of the BERT variants use WordPiece tokenization whereby the input sequence is split into subword units, which is expected to enhance the representations of rare words and morphological variations. The use of WordPiece demands additional pre-processing for annotation labels in subword units, which is further explained in Section 4.5.2. In IOBHI, *name* and *type projection* are dense layers that collect and transform the features from SciBERT's hidden outputs for each token into their representations. For the type representations, we apply the softmax function to obtain the semantic type probability distribution which can be directly compared with the one-hot vectors of the pre-computed name embeddings for semantic type. The name projection layer essentially transforms tokens in a span to the same space as the pre-computed segments of concept names. The concatenated vector of name and type representations are fed to the following biLSTM+CRF [Huang et al., 2015] for IOB tagging. The biLSTM+CRF module comprises of two bidirectional LSTM layers and

Figure 4.3: Model design of *IOBHI* (left) and *ONETAG* (right): In *IOBHI*, IOB classification is deferred to the end of the network (FC: fully connected layer, ⊗: dot product; note that IOBHI and ONETAG models are independent but are drawn in one figure for brevity.)



a CRF layer whose target tag set size is three for the IOB tags.

In the upper section of the figure, the same concatenated vector goes through a fully connected layer (*name matching*) followed by a dot product across the pre-computed concept name embeddings (details in Section 4.5.1). This process computes the bilinear interactions between the token-level feature representation and the concept name embeddings. Then, we assign the corresponding concept code to each token's normalized name representation using the pre-defined mapping between the concept name indices and actual concept codes. In the right section of Figure 4.3, we have the ONETAG model where the biLSTM+CRF module replaces the semantic type classifier, which predicts the IOB-prefixes and entity type simultaneously. Following that is the 1D average pooling layer with kernel size 2 to construct the semantic type probability distribution the same way as in the IOBHI model. The rest of the name projection and bilinear mapping components share the same design. The name ONETAG derives from using a single tag to represent both positional tags (IOB) and semantic types that were decoupled in the IOBHI model.

4.4.2 Optimization

We take three objective functions in this model as depicted in Figure 4.3. $\mathcal{L}_{MD}^{nll}$ is the negative log-likelihood (NLL) loss from the CRF layer and $\mathcal{L}_{ET}^{fl}$ and $\mathcal{L}_{EN}^{fl}$ are the focal losses for the tasks of entity typing and normalization, respectively. For the purpose of joint learning, we use the weighted sum of the three objective functions

where λ was empirically chosen to 1.0. The joint objectives for training the IOBHI and ONETAG models are

$$\mathcal{L}_{IOBHI}(\mathcal{X}, \mathcal{Y}; \theta) = \mathcal{L}^{nll}_{MD} + \lambda(\mathcal{L}^{fl}_{ET} + \mathcal{L}^{fl}_{EN}) \quad \text{and} \tag{4.1}$$

$$\mathcal{L}_{ONETAG}(\mathcal{X}, \mathcal{Y}; \theta) = \mathcal{L}^{nll}_{MD/ET} + \lambda\mathcal{L}^{fl}_{EN}. \tag{4.2}$$

Next, we provide some background and rationale for choosing the focal loss function $\mathcal{L}^{fl}$.

Focal loss [Lin et al., 2017] is a popular choice for the object detection tasks in computer vision. The motivation of this function is to minimize the gradient norms of easily classified examples (e.g., the pixels of background image in an object detection task). This function is especially effective with class imbalanced data such as in object detection where most of the pixels belong to the non-object class. As shown in Equation (4.3), the loss depends on the predicted probability distribution $\hat{p}$ on i-th individual sample where γ is a user-defined hyperparameter:

$$\mathcal{L}^{fl}(\mathcal{X}, \mathcal{Y}; \theta) = -\frac{1}{|\mathcal{Y}|} \sum_{y_i \in \mathcal{Y}} (1 - \hat{p}_{i,y_i})^{\gamma} \log \hat{p}_{i,y_i}, \tag{4.3}$$

where θ are network parameters.

As observed by Mukhoti et al. [Mukhoti et al., 2020], focal loss forms an upper bound on the regularized KL-divergence between the target distribution q and the predicted distribution $\hat{p}$, where the regulariser is the negative entropy of $\hat{p}$ (proof in [Mukhoti et al., 2020]):

$$\mathcal{L}^{fl}(\mathcal{X}, \mathcal{Y}; \theta) \geq \mathrm{KL}(q||\hat{p}) - \gamma\mathbb{H}(\hat{p}). \tag{4.4}$$

The optimization using focal loss, hence, minimizes KL divergence while increasing the entropy of the predicted distribution $\hat{p}$, whereas cross-entropy only minimizes KL divergence. In our case where the majority class is the unlabeled (*other*) class, a prediction with a higher confidence, such as an "obvious" *other* class token, decreases the gradient norms in updating model parameters. As recommended by Mukhoti et al. [Mukhoti et al., 2020], we choose γ dynamically with the threshold of the predicted probability, such that $\gamma = 5$ if $\hat{p}_{i,y_i} < 0.3$, else $\gamma = 3$.

4.5 Data Preparation

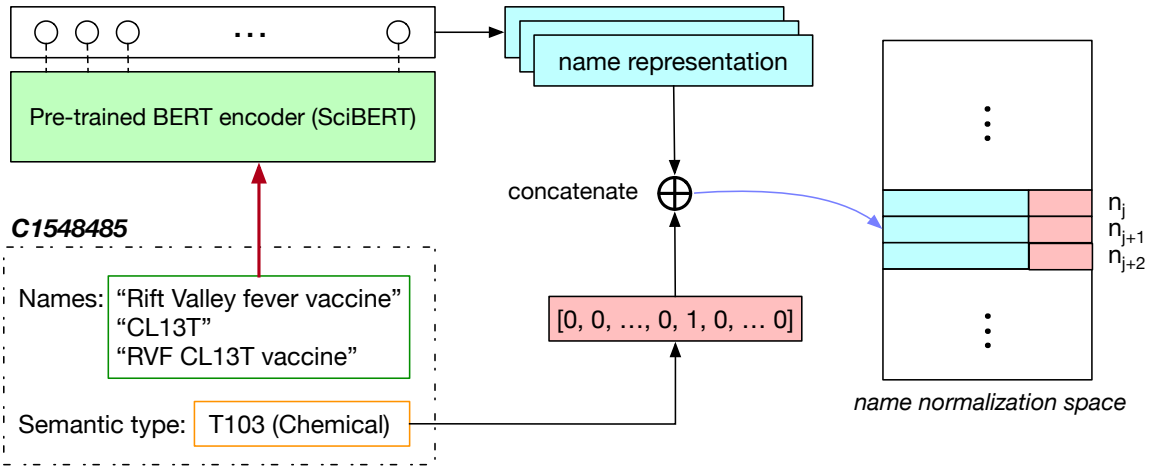
Recently, Mohan and Li introduced the *MedMentions* dataset with an extensive set of biomedical entity annotations targeting the UMLS concepts. UMLS is the metathesaurus that combines concepts from over 200 medical vocabularies — 4.4 million unique concepts in the 2020 AB release — making it one of the most comprehensive biomedical terminologies. MedMentions provides 352,496 annotated examples from 4,392 PubMed abstracts prepared by human experts in biomedical content curation. The authors of MedMentions selectively chose 21 semantic types of UMLS that are considered most useful for semantic indexing. They created the annotated corpus

MedMentions-ST21pv with the entities of the 21 semantic types and their descendent types. This corpus contains 203,282 mentions with 25,419 unique concepts from 4,392 documents. In this study, we use *MedMentions-ST21pv* as a benchmark.

4.5.1 Concept name embeddings

For the EN task, as indicated earlier, we look for the contextually most similar name from the pre-defined concept name embeddings given the encoded name representation. We independently compute these embeddings ahead of training time using the same pre-trained BERT model (i.e., SciBERT). Figure 4.4 illustrates these pre-processing steps listed as follows:

Figure 4.4: Pre-computed concept name embeddings for EN



(Step 1). We collect names from the UMLS definitions with the following constraints of the UMLS concept properties and add all the aliases mentioned in the MedMentions corpus together.

- the entity belongs to the 21 semantic types (T005, T007, T017, T022, T031, T033, T037, T038, T058, T062, T074, T082, T091, T092, T097, T098, T103, T168, T170, T201, T204) or the descendent types of those.
- `LAT` (language of term) is `EN` (English)
- `TS` (term status) is `P` (preferred)
- `STT` (string type) is `PF` (preferred form)
- `SAB` (source name) is one of preferred sources (CPT, FMA, GO, HGNC, HPO, ICD10, ICD10CM, ICD9CM, MDR, MSH, MTH, NCBI, NCI, NDDF, NDFRT, OMIM, RXNORM, SNOMEDCT_US)
- `SUPRESS` (supressible flag) is `N` (none)

(Step 2). We encode up to three names (typically multi-token phrases) for each concept using the BERT encoder. We take the encoder's last layer outputs and use the mean vectors as the name embeddings. We also append the concept's semantic type representation (i.e., the one-hot vector of the type) in order for the model to consider both the name and semantic type of the entity.

With the specified constraints, we can build 9,538,297 name embeddings for the entire UMLS concept set. For training and testing purposes, we build name embeddings only for the concepts that appear at least once in the MedMentions-ST21pv corpus. During training, we use the training subset with 42,836 name embeddings corresponding to concepts seen in the training examples; this number varies by the data augmentation factor (75,865 with $\xi = 1$, 106,561 with $\xi = 2$). At test time, we use the full set of 56,893 name embeddings for the entire set of concepts in the MedMentions corpus. The methodology for using a large scale name normalization space (such as the full 9.5+ million UMLS embeddings) is out of the scope in this current study.

4.5.2 Subword-level tokens to word-level labels

BERT models use the WordPiece tokenization [Wu et al., 2016] method, which is a subword segmentation algorithm. The vocabulary is constructed iteratively from the characters in the language by adding the most frequent combinations of entries in the current vocabulary. In our effort, the sequence labeling is modeled at the word level, which creates a discrepancy with the sequence tokenized using WordPiece given its subword focus. So, we use WordPiece to tokenize text and assign the given label to all the constituent subword tokens. In the inference step, we use the majority vote to determine the label of a word. For example, WordPiece would tokenize the word "hydrocodone" into (`hydro`, `##cod`, `##one`). If the model predicted the semantic types for the sequence as (T103, T168, T103), then we assign T103 to the full word.

4.6 Experiments

4.6.1 Datasets and baseline models

We use MedMentions-ST21pv to evaluate the performance of our models on the NER and EN tasks. Table 4.1 shows the statistics of the dataset. We experiment with the same *train-validation-test* splits (60-20-20%) provided by the creators of the MedMentions datasets.

Table 4.1: Statistics of the MedMentions-ST21pv dataset

	Training	Dev	Test
# of documents	2,635	878	879
# of mentions	122,241	40,884	40,157
# of unique concepts mentioned	18,520	8,643	8,457

We compare our models with the publicly available biomedical NER tools and some previous efforts. We outline four different models from this effort, given two variables: model design and data augmentation factor.

- **ONETAG**: Model using conventional IOB format (i.e., combined IOB prefixes and type suffixes)
- **ONETAG** ($\xi = n$): ONETAG trained with data augmentation (n is the augmentation factor)
- **IOBHI**: Model with decoupled IOB format (i.e., separate IOB tags and semantic types)
- **IOBHI** ($\xi = n$): IOBHI trained with data augmentation

Followings are the tools and the state-of-the-art models used for comparisons:

- *TaggerOne* [Leaman and Lu, 2016] has been a popular choice for the biomedical NER, which uses carefully designed rule-based algorithms.
- *QuickUMLS* [Soldaini and Goharian, 2016] utilizes an approximate dictionary matching algorithm, which outperforms other biomedical text processing tools such as MetaMap and cTAKES.
- *SciSpacy* [Neumann et al., 2019] is a package of specially designed tools for biomedical and scientific text processing leveraging the spaCy library. SciSpacy has shown superior results to QuickUMLS and MetaMap on the biomedical NER tasks.

Due to the recency of the MedMentions release, there are not many end-to-end models for NER and EN on this particular dataset. We identified two recent peer-reviewed efforts with similar evaluation setup as ours for comparison purposes.

- Loureiro et al. [Loureiro and Jorge, 2020] presented a BERT-biLSTM-CRF framework with an approximate dictionary matching method.
- Wiatrak et al. [Wiatrak and Iso-Sipila, 2020] proposed a model of a BERT-BiLSTM-MLP framework with a hierarchical structure of multiple tasks.

4.6.2 Training details

We adopt the SciBERT uncased model [Beltagy et al., 2019] as the sentence encoder whose model dimension is 768. The maximum sentence length of inputs is 256. The biLSTM+CRF consists of two layers of biLSTM networks with model dimension 256. All models are optimized using AdamW [Loshchilov and Hutter, 2017] controlled by a linear scheduler with warmup steps. The learning rate starts with 0, increases up to 3×10^{-5} during the warmup steps, and linearly decreases to 0 until the specified number of training steps. We apply dropout to biLSTM hidden states with a rate of 0.1. Training is done for 8 epochs with the batch size of 8.

Table 4.2: MD, ET, and EN performances on MedMentions-ST21pv dataset. The results marked with † are obtained from [Loureiro and Jorge, 2020].

	Mention Detection (MD)			Entity Typing (ET)			Entity Norm. (EN)		
Model	P	R	F_1	P	R	F_1	P	R	F_1
TaggerOne	n/a	n/a	n/a	n/a	n/a	n/a	0.471	0.436	0.453
QuickUMLS†	n/a	n/a	n/a	0.145	0.169	0.156	0.180	0.261	0.213
ScispaCy†	n/a	n/a	n/a	0.101	0.317	0.154	0.252	0.535	0.342
Loureiro et al.'s (CLF)	0.694	0.718	**0.706**	0.586	0.646	0.615	0.322	0.527	0.400
Loureiro et al.'s (STR_CLF+)	0.694	0.718	0.706	0.631	0.637	**0.634**	0.484	0.501	0.492
Wiatrak et al.'s	0.742	0.593	0.659	0.594	0.553	0.573	0.431	0.401	0.415
ONETAG	0.709	0.671	0.690	0.636	0.602	0.619	0.503	0.476	0.489
ONETAG ($\xi = 1$)	0.701	0.679	0.690	0.625	0.605	0.615	0.509	0.493	0.501
ONETAG ($\xi = 2$)	0.696	0.674	0.685	0.620	0.601	0.611	0.512	0.496	0.504
IOBHI	0.701	0.682	0.691	0.614	0.597	0.605	0.500	0.487	0.494
IOBHI ($\xi = 1$)	0.706	0.675	0.690	0.620	0.593	0.606	0.522	0.499	0.510
IOBHI ($\xi = 2$)	0.705	0.673	0.689	0.617	0.589	0.602	0.524	0.499	**0.511**

Table 4.3: Zero-shot evaluation of IOBHI and ONTAG models for NER and EN

	Mention Detection (MD)			Entity Typing (ET)			Entity Norm. (EN)		
Model	P	R	F_1	P	R	F_1	P	R	F_1
ONETAG	0.877	0.671	0.760	0.707	0.541	0.613	0.531	0.406	0.460
ONETAG ($\xi = 1$)	0.866	0.694	0.770	0.695	0.557	0.618	0.581	0.466	0.517
ONETAG ($\xi = 2$)	0.864	0.694	0.770	0.697	0.560	**0.621**	0.617	0.495	0.549
IOBHI	0.868	0.688	0.768	0.685	0.542	0.605	0.545	0.432	0.482
IOBHI ($\xi = 1$)	0.861	0.692	0.767	0.677	0.544	0.603	0.605	0.485	0.539
IOBHI ($\xi = 2$)	0.868	0.699	**0.775**	0.679	0.547	0.606	0.624	0.493	**0.551**

4.6.3 Evaluation metrics

Whereas the objective functions are computed at the token level during the training and validation steps, we measure the model's performance using the mention-level metrics. Following the well-known CONLL conventions for NER [Sang and De Meulder, 2003], we use the *exact-match evaluation* system, where the metrics are micro-averaged strict precision, recall, and F1 scores. That is, a predicted text span (MD step) is a true positive (*tp*) only if the starting position and the length exactly match the ground truth. For a predicted concept (or its type) to be a *tp*, its span should be an exact match with the corresponding class of the ground truth. Hence, the semantic type classification and entity normalization performances are upper-bounded by the performance of mention detection. The constraints are highest for the final EN step because the concept code, its type, and exact span in the input text ought to match the ground truth for us to consider it a *tp*.

To clarify our counting system, we define metrics

$$\text{precision} = \frac{\#tp}{\#tp + \#fp} \quad \textit{and} \quad \text{recall} = \frac{\#tp}{\#tp + \#fn},$$

where $(\#tp + \#fp)$ is the number of predictions and $(\#tp + \#fn)$ is the number of ground truth occurrences using the counts of False Positive (FP) and False Negative (FN). To hold these properties, we consider the cases where the model predicts correctly-bounded text spans with wrong labels as both *false positives* and *false negatives* (e.g., a mention that should have been labeled as "A" but was predicted as "B" is a *fn* for "A" and a *fp* for "B"). In the zero-shot evaluation setup (details in Section 4.6.4), we do not consider a prediction as an *fp* if it overlap the ground truth mentions of which the classes are seen in the training dataset. The model should identify the mentions regardless of whether the classes are seen or unseen in any dataset.

4.6.4 Results and discussion

Table 4.2 shows our models' performance against previous models and the biomedical NER tools on three different tasks: mention detection (MD), entity typing (ET, UMLS semantic type classification), and entity normalization (EN, UMLS concept code normalization) task. Our models outperform previous results on the end-to-end strictest EN task. But when restricted to the less stricter tasks, MD and ET, the Loureiro et al. result seems superior. All our models eventually appear to converge at the same performances regardless of the model architecture. On the ET task, Loureiro et al.'s *STR_ CLF+* model holds the best score; this model combines a neural network-method and the use of 3rd-party software for approximate dictionary matching. We believe that another of their proposed models (*CLF*, row four of Table 4.2) is more suitable for evaluating the end-to-end neural approaches against our models.

In particular, for the ET task, we see ONETAG models perform better than IOBHI models. We conclude that if the eventual goal is just NER (that is, MD and ET), the conventional tagging scheme is superior. Given there are only 21 semantic

types in the dataset, with over 122K mentions in the training dataset, there could be enough signal for the combined tags (position plus type) to render the models effective for NER compared with the decoupled setup. However, when it comes to EN performance, the decoupled approach (IOBHI) that delegates IOB tagging to the end performs better than the conventional ONETAG approach. IOBHI uses type prediction before matching with pre-computed concept name embeddings and this type-only signal (without the IOB hints) seems to better help the model match correct concepts for the EN part. On the other hand, the IOB tagging in the IOBHI model performs well enough to spot the boundaries to an extent that renders the end-to-end EN evaluation superior for the overall architecture.

All our models achieved higher scores than the previous efforts on the EN task. The consistent increase in F1 score for the EN task with models using data augmentation supports its efficacy of providing counterfactual examples that break spurious correlations of concepts with surrounding textual context. However, the gains by increasing the augmentation factor from 1 to 2 is minimal. We deduce that using an adequate amount of augmented examples improves learning for entity normalization but not for other tasks.

Zero-shot evaluation

We also wanted to analyze what happens to our performance metrics when we look at zero-shot (ZS) scenarios. That is, what happens if we evaluate performances over those concepts that only occur in the test set and never show up in the training and development subsets. We find that there are 3,247 such unique concepts and 8,180 mentions of them in the test set. These ZS concepts account for 38% of all concepts in the test set (but only around 20% of all test mentions, which is reasonable since these concepts are expected to be rare).

Table 4.3 outlines the model performance in the ZS setting for concepts that exclusively occur only in the test dataset. We see similar patterns to the overall results (from Table 4.2) on the test set: (1) ONETAG models perform better on ET, and IOBHI models perform better on EN, (2) the data augmentation techniques enhance the performance on EN but not others. It is important to recall that in this setting an *fp* can only arise out of mistakes made for ZS concepts. That is, an *fp* for a predicted ZS concept occurs if (a) either its type or span is incorrect or (b) the ground truth concept is a different ZS concept. Note that the ZS concept set size is relatively smaller (only 20% of test set mentions) and the universe of possibilities for false positives for a ZS concept arises out of only other ZS concepts. Hence, the precision values in the ZS setting in Table 4.3 are higher than those in the larger full test dataset (Table 4.2). Recall values for MD and EN are similar to general results but ET recall is markedly lower compared to general results[1]. We also observe that counterfactual data augmentation only increases F1-score by around 2% in overall results but in the ZS settings the gains are from 7–9%. This is not surprising since

[1]Please note that type prediction is independent of concept prediction although parameter sharing for the two tasks is in place. So this possibility of differences in recall is plausible although not expected.

breaking the spurious correlations between concepts and surrounding contexts during training is expected to help with ZS concepts that were never encountered in that process. Intuitively, we suspect, without augmentation some ZS concepts would be mismatched to potentially similar concepts seen during training.

Probing for semantic type affinities using the *name matching* layer parameters

Recall that a concept name embedding in our model is the concatenated vector of a dense vector from the name projection layer and a softmax probability distribution from the type projection layer. The following name matching layer computes the similarity scores with the pre-computed concept name embeddings, which are constructed in the same way via SciBERT. We deliberately designed the structure of a concept's name embedding in this manner to incorporate features corresponding to its name and also its type (in an explicit manner). This allows us to analyze the model regarding how it interprets the semantic type features from the hidden outputs on the EN task.

We particularly look at the parameter matrix of the name matching fully connected (FC) component (yellow boxes in Figure 4.3). Let the vector $u = \langle u_n, u_t \rangle$ be the hidden output from the model, which is the input vector of the FC component, expressed as a concatenation of vectors u_n and u_t, where u_n is for the name space in $\mathbb{R}^p$ and u_t is for the entity type space in $\mathbb{R}^q$. Let $v = \langle v_n, v_t \rangle$ be the pre-computed concept name embedding with the same structure as u. With W as the bilinear transformation function of the FC component, for u and v as defined earlier, we have $sim(u, v) = vWu$. Let's denote $\widetilde{u}$ be Wu, the transformed vector of u. Thus we rearrange the similarity measure of an input token and a pre-computed concept name embedding as the dot product of v and $\widetilde{u}$:

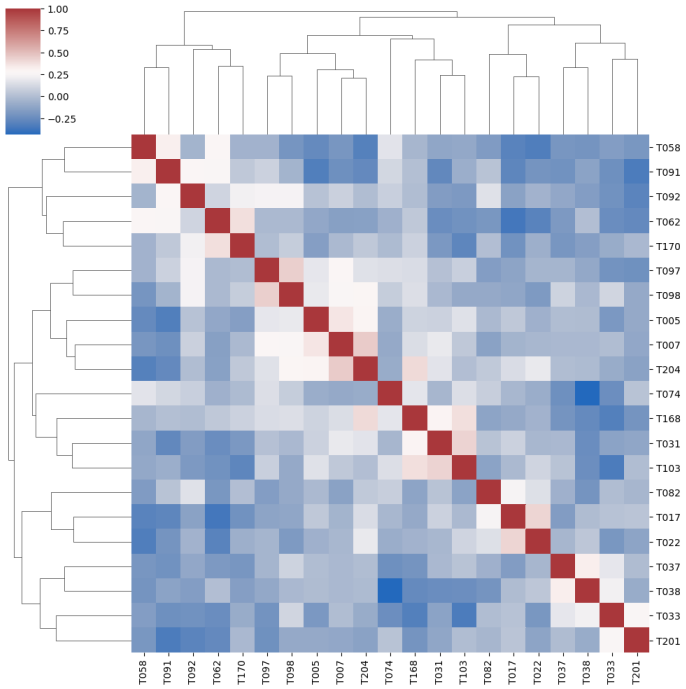
$$v \cdot \widetilde{u} = \sum_{k=1}^{p+q} v_k \widetilde{u}_k = \sum_{k=1}^{p} v_k \widetilde{u}_k + \sum_{k=p+1}^{p+q} v_k \widetilde{u}_k \tag{4.5}$$

$$= \sum_{k=1}^{p} \sum_{i=1}^{p+q} v_k W_{k,i} u_i + \sum_{k=p+1}^{p+q} \sum_{i=1}^{p+q} v_k W_{k,i} u_i \tag{4.6}$$

The second term in equation 4.6 clearly influences the type segment similarity in the end and is parametrized by a submatrix of W, specifically, $W_{type} = W[p + 1, p + q; 1, p + q]$. Thus, the multiplication of W_{type} and its transpose gives us the affinity matrix among entity type representations that the model learned. Figure 4.5 is the cluster map of the 21 semantic types obtained from the correlation matrix $W_{type}(W_{type})^\top$ where rectangular blocks of (shades of) red indicating clusters; this was derived from hierarchical clustering using the *Scipy* [Virtanen et al., 2020] cluster package. Below we display some of the interesting type clusters our probing has surfaced.

- Cluster 1:

Figure 4.5: Type affinity matrix derived from the *name matching* layer's bilinear function



  - T058: Health Care Activity,
  - T091: Biomedical Occupation or Discipline

- Cluster 2:
  - T092: Organization,
  - T062: Research Activity,
  - T170: Intellectual Product

- Cluster 3:
  - T097: Professional or Occupational Group,
  - T098: Population Group

- Cluster 4:
  - T005: Virus,
  - T007: Bacterium,
  - T204: Eukaryote

- Cluster 6:
  - T082: Spatial Concept,
  - T017: Anatomical Structure,

- T022: Body System

These clusters appear to indicate that W has nicely converged to capture similarities among types. For example Cluster 4 seems to be grouping different types of organisms and Cluster 6 appears to capture anatomical locations. This probing provides a peek into the inner workings of what the model is learning and provides additional confidence that it is teasing out reasonable representations of concepts.

4.7 Error Analysis

We manually analyzed randomly selected error causing instances to track different types of errors. We highlight a few high level classes of errors we found. We first focus on partial matches of spans that cause both *fp* and *fn* errors.

- Adjective or noun compound modifiers that our models predicted to be part of a span turned out to be incorrect in several error causing examples. The following examples contain *italicized* ground truth spans and the full and incorrect spans we predicted.
  - activating *mutations*
  - benign *parathyroid adenoma*
  - familial isolated *hyperparathyroidism*
  - leptin *gene promoters*
  - ZFX *oncogenes*

  Considering these examples, it does not appear that our spans are blatantly wrong because the adjectives and compounds we included in the predictions (e.g, benign, familial, leptin) seem pertinent to the ground truth spans. They appear to bring about more specificity to the concepts being tagged compared with ground truth annotations. We are not sure if these are errors in the MedMentions dataset or if this is really a nuanced phenomenon that our models are unable to capture.

- While the earlier examples indicated that we erroneously made spans more specific, we also encountered errors where less specific spans are somehow mapped to more specific concepts in MedMentions. Consider these examples:
  - (enzyme) *activity*
  - (bacterial culture) *medium*
  - *benign* (thyroid) *nodules*
  - *clinical* ... findings
  - *persistent* ... asthma

The first three examples show the italicized spans in the ground truth but were mapped at the EN level to ground truth concepts that are more specific (as indicated with the corresponding full preferred name elements in parentheses). In the last two examples, terms that render more preciseness to concepts (e.g., findings, asthma) were actually present in the context but appeared with a gap (that included other tokens) from the ground truth spans. Maybe in these cases, during the EN task, our models were unable to latch on to the implied/latent signal present in the surrounding context.

We also accrued several errors when we missed or erroneously tagged broad themed concepts. For example, concepts with preferred names *men, women, results, predictors, group, trials* are sometimes mapped to specific concepts and sometimes not in the MedMentions dataset. The models had trouble figuring out which contexts warrant a mapping. Additionally *fp* errors also occurred with several abbreviations were deemed incorrectly mapped as per ground truth but seemed appropriate upon manual examination. Our analysis revealed that at times, only the first occurrence of an abbreviation was annotated in the ground truth with some subsequent mentions left untagged. This particular scenario does not seem to be an outcome of model's issues but due to inconsistencies in MedMentions test set.

4.8 JEREN Summary

In this effort, we evaluated two high level strategies in the context of a multi-task learning framework for named entity recognition and entity normalization for biomedicine. First, we explored the effect of decoupling IOB-prefixes from the type tags in the combined conventional tagging scheme. Results show that separating the task of identifying the boundaries of mentions from the entity type classification enhance the entity normalization performance but not for entity typing and mention detection. We also demonstrate that using an adequate number of counterfactual training examples helps in the EN task, more so in the zero-shot evaluation setting. Parameter probing showed meaningful clusters of semantic types and error analyses surfaced interesting issues that warrant deeper exploration of the MedMentions dataset and more advanced strategies that better exploit the context.

The focus of this effort was to assess specific strategies in the context of a joint modeling framework for biomedical NER and EN. Hence, we kept the model design fairly simple without resorting to more sophisticated methods such as transfer learning or domain adaptation. We did, however, leverage latest pre-trained language models (SciBERT) for biomedicine. More innovative methods to better capture context and implied intent of the writers are necessary to make additional progress. Our effort only uses concept aliases and semantic types for the EN task; well established knowledge bases that include explicit relationships among concepts or topic distributions across documents can be utilized in future efforts in NER and EN. We have not fully addressed the scalability aspect in this effort. Although zero-shot performance is decent in the MedMentions name space, more realistic systems would need to search in the target space of 4.4 million UMLS concepts and 11 million corresponding English

names. This problem demands parallel and distributed deep learning techniques and very fast nearest neighbor search (e.g., locality-sensitive hashing), aspects we intend to explore in the near future.

Chapter 5 QAMat: Document Retrieval in the Question-Answering Pipeline [1]

An effective IR system should encode a query and candidate document and produce an accurate relevance score, which requires a good understanding of the complex relations between the query and document inputs. Conventional IR features, including the bag-of-words scheme, do not adequately capture the signal for measuring the relevance. Hence, researchers have proposed using neural methods for IR to leverage the strong representation power of DNNs for encoding textual inputs.

In the following two chapters, we present two different IR systems that leverage a neural model for transforming textual inputs into non-traditional features for computing the relevance score of a candidate document to query. The essence of these two systems is the document reranking method using different relevance metrics by neural networks. As demonstrated in earlier chapters, words can be represented by a fixed-size dense vector, and so do sentences. This chapter examines the efficacy of a neural sentence matching component as an answer sentence retrieval model in the question-answering setup. With this basic setup in mind, we will move on to the detailed problem description and background knowledge pertinent to this problem.

Document retrieval (DR) forms an important component in end-to-end Question-Answering (QA) systems where particular answers are sought for well formed questions. DR in the QA scenario is also useful by itself even without a more involved natural language processing component to extract exact answers from the retrieved documents. This latter step may simply be done by humans like in traditional search engines granted the retrieved documents contain the answer. In this chapter, we take advantage of datasets made available through the BioASQ QA task and build an effective biomedical DR system that relies on relevant answer snippets in the BioASQ training datasets. At the core of our approach is a question-answer sentence matching neural network that learns the relevance measure of a candidate answer sentence to a question in the form of a matching score. In addition to this matching score feature, we also exploit two auxiliary features for scoring document relevance: the name of the journal in which a document is published and the presence/absence of semantic relations (subject-predicate-object triples) in a candidate document connecting biomedical entities mentioned in the question. We rerank our baseline sequential dependence model scores using these three additional features weighted via adaptive random research (ARS) and other learning-to-rank (L2R) methods. Our full system placed 2nd in the final batch of phase A (DR) of task B (QA) in the 2018 BioASQ QA task. Our ablation experiments highlight the significance of the neural matching network component in the full "DR in QA" system.

Question answering (QA) has emerged as an important field within Natural Language Processing (NLP) and information retrieval (IR) communities to handle the explosion in curated textual and structured data. Modern search engines heavily use

[1] This chapter is based on the previously published paper [Noh and Kavuluru, 2018] appears in the 2018 17th IEEE International Conference on Machine Learning and Applications.

QA methods under the hood to deliver precise answers to different types of questions. In Google, simple factoid questions whose answers are usually short texts (e.g., "What is the capital of USA?") directly result in a bold font phrase that captures the answer (e.g., Washington, D.C.) displayed just below the search box. More complex questions may result in small Web text snippets that are likely to contain the answer. For the question "What causes constipation?", Bing shows an HTML list from WebMD of various causes. In specialized fields such as biomedicine, questions can be much more complex where the answers may not be readily available on Web pages but may need to be gleaned from scientific literature indexed by NIH search engine PubMed. To address challenges in biomedical QA, the U.S. National Library of Medicine (NLM) has been sponsoring a series of community shared tasks under the name BioASQ since 2013 [Tsatsaronis et al., 2015]. For a recent BioASQ example question, "Which currently known mitochondrial diseases have been attributed to POLG mutations?", Google and Bing do not have any straightforward responses but instead point to some research articles. However, what is expected as an answer in BioASQ is a list of diseases.

In the BioASQ QA task, the question types include *yes/no* (boolean response to a statement), *factoid* (answer is a single entity), *list* (response is a list of entities), and *summary* (answer is a detailed narrative response). Results are evaluated at various levels of granularity including the relevant documents (PubMed abstracts) retrieved, various snippets (small blurbs of text) retrieved from selected documents, specific biomedical concepts that may directly answer a question, and a so called "ideal" answer to a question (which is usually a precise English description of the answer). That is, although the eventual goal is the ideal answer(s), documents that contain answers, smaller snippets in them that contain the answer, and biomedical concepts relevant to the answer are also expected as output and evaluated separately. The corpus available for all retrieval tasks in BioASQ tasks is the set of all PubMed indexed citations (provided with its title, abstract, and additional metadata such as authors, journal name, and indexing terms). Hence throughout this chapter, by document, we mean the title+abstract and any other associated metadata.

In this chapter, we particularly focus on the high-level document retrieval model for a biomedical question-answering task. This is a natural first step because most end-to-end QA systems first need to identify documents that potentially contain answers. Subsequently, more sophisticated NLP methods are used to identify smaller snippets and next spans of particular phrases pertinent to the ideal answer within them. Also, superior performance in the DR task will lead to overall better end-to-end system performance, given all other factors being equal. Hence we focus on this task in our preliminary foray into the BioASQ task. Our approach to DR involves a traditional IR model to retrieve a list of documents and then rerank this list using neural question-answer sentence matching and two additional auxiliary features involving journal names (of documents) and an external knowledge base of relations extracted from biomedical articles. Specifically, we make the following contributions.

1. We train a neural sentence matching network to learn a matching score of the question sentence with each sentence in a candidate relevant document. We

do this by exploiting the training data that includes the relevant snippets from prior years in the BioASQ series.

2. We devise a feature that exploits the thematic overlap of a journal in which a candidate document is published and the question at hand, using medical subject headings (MeSH terms) as proxies for thematic content.

3. We also use an external knowledge base of relations called SemMedDB extracted by applying rule-based relation extraction algorithms to the BioASQ corpus. The main intuition is that documents containing binary relations involving a pair of entities mentioned in the question may have a higher chance of being relevant.

4. With features discussed thus far in this list, by using adaptive random search and learning-to-rank algorithms, we rerank documents retrieved by a traditional sequential dependence model implemented as part of the open source Galago search engine [Croft et al., 2010].

5. Overall, we find that our reranking approach performs consistently better than the baseline retrieval system when tested on the 2016 and 2017 BioASQ test sets. We also participated in BioASQ 2018 and our system[2] came in 2nd (among 26 different entries) in the final batch as shown in Table 5.1 (based on the mean average precision (MAP[3]) measure used by the task organizers).

Table 5.1: The official BioASQ results of top 5 retreival systems (2018, task 6b phaseA batch 5)

System	Precision	Recall	F1	MAP	GMAP
aueb-nlp-4	0.1145	0.3790	0.1590	0.0695	0.0012
ours	0.1085	0.3539	0.1513	0.0680	0.0009
sys2	0.1055	0.3331	0.1458	0.0633	0.0008
ustb_prir4	0.1105	0.3441	0.1532	0.0622	0.0009
testtext	0.1115	0.3540	0.1550	0.0618	0.0009

5.1 Methodology

We use the BioASQ [Tsatsaronis et al., 2015] QA datasets from years 2014 through 2017. When using a certain year's dataset as test set, we use all preceding years' datasets for training.

[2]We are not violating the double blind review criteria in this review phase given the BioASQ website does not reveal our affiliation details.

[3]The MAP values in Table 5.1 are much smaller than what they ought to be due to the special way BioASQ organizers compute AP for which they always divide the p@k sum by 10 instead of the actual number of relevant documents. This makes the MAP value much smaller given many questions have < 10 relevant documents. In our experiments in the rest of this chapter, we use the standard MAP formula to give realistic scores.

5.1.1 Baseline document retrieval model

We use the Sequential Dependence Model (SDM) [Metzler and Croft, 2005] in the initial document retrieval process as implemented in the open source Galago search engine [Croft et al., 2010]. Unlike the traditional bag-of-words models, the order of terms in a query is also taken into account in the SDM model. SDM is based on the Markov random field model, in which not only the unigrams but also the ordered and unordered bi-grams in a posed query are considered in the retrieval score computation. The term frequency score is

$$f_T(q_i, D) = \log P(q_i|\theta_D) = \log \frac{tf_{q_i,D} + \mu \frac{cf_{q_i}}{|C|}}{|D| + \mu} \tag{5.1}$$

where q_i is a query term, D is the document, θ_D is a language model built using D, $tf_{q_i,D}$ is the term frequency of q_i in D, cf_{q_i} is the collection frequency of q_i, $|C|$ is the total number of terms across all the documents, $|D|$ is the document length, and μ is the Dirichlet prior for the smoothing effect. Likewise, the functions for the ordered and unordered bi-grams are defined in a similar way:

$$f_O(q_i, q_{i+1}, D) = \log \frac{tf^N_{o(q_i,q_{i+1},D)} + \mu \frac{cf^N_{o(q_i,q_{i+1},D)}}{|C|}}{|D| + \mu} \tag{5.2}$$

$$f_U(q_i, q_{i+1}, D) = \log \frac{tf^M_{u(q_i,q_{i+1},D)} + \mu \frac{cf^M_{u(q_i,q_{i+1},D)}}{|C|}}{|D| + \mu} \tag{5.3}$$

where $tf^N_{o(q_i,q_{i+1},D)}$ and $tf^M_{u(q_i,q_{i+1},D)}$ indicate the frequencies of the terms q_i and q_{i+1} within an ordered window of N word positions and within a unordered window of M word positions respectively. The final scoring function is the weighted sum of the the three constituent functions

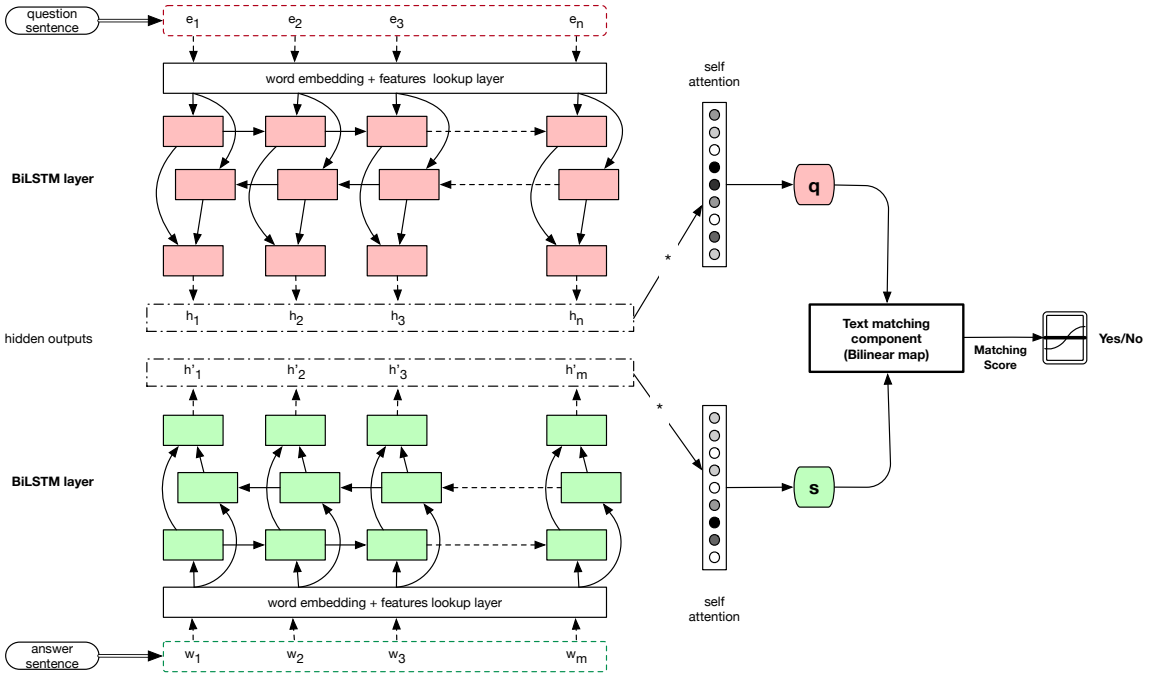
$$score(Q, D) = \lambda_T \sum_{i=1}^{|Q|} f_T(q_i, D) + \lambda_O \sum_{i=1}^{|Q|-1} f_O(q_i, q_{i+1}, D) + \lambda_U \sum_{i=1}^{|Q|-1} f_U(q_i, q_{i+1}, D) \tag{5.4}$$

where $Q = q_1, \ldots, q_{|Q|}$ is the query and λ_T, λ_O, and λ_U are weights for the unigram, ordered bigram, and unordered bigram components respectively. This SDM scoring function is the baseline throughout all our experiments where we measure the effectiveness of our matching score feature and other auxiliary features.

5.1.2 Question-Answer matching model

Our QA matching (QAMat) model is an attention-based neural network based on prior efforts on Siamese networks in NLP [Mueller and Thyagarajan, 2016]. However, the main difference is that we use separate parameters for encoding the question and candidate sentences while the original Siamese network uses the same parameters until the final distance layer. Given the linguistic (lexical and syntactic) layout of

Figure 5.1: Question-Answering Text Matching (QAMat) Model Architecture



a question and the importances of various words in it are different in nature from the relevance of different tokens observed in a candidate answer sentence, different parameter sets for encoding them separately are necessary. Due to this, we see our network as "matching" sentences instead of computing similarity between them.

As outlined earlier, the BioASQ training datasets provide a list of human adjudicated text snippets that are relevant to each question. As such, we train the QAMat model with the pairs of questions and relevant sentences in the ground truth training text snippets. Thus we expect the model to score sentences in a document with regards to their potential for containing an answer to a specific question. We first outline the architecture and subsequently elaborate on training dataset generation.

Beyond simple averaging of word embeddings in a sentence, researchers have attempted to build neural models that encode a phrase [Yin and Schütze, 2015], a sentence, or a document [Le and Mikolov, 2014] into a discriminative low dimensional vector representation. For QA in particular, a paragraph can be matched to a question sentence to find an answer phrase span in that paragraph [Chen et al., 2017]. We follow a similar approach where given a question sentence and a candidate answer sentence, the neural net estimates the probability that the answer sentence contains information pertinent to answer the question. We train two bidirectional long short-term memory networks (BiLSTMs [Hochreiter and Schmidhuber, 1997]), one for encoding a question sentence and the other for encoding a candidate answer sentence as shown in Figure 5.1.

Question Sentence Encoding All the tokens in a question sentence Q are mapped to corresponding word embeddings. The word embeddings are then fed into the

question BiLSTM to produce hidden node outputs

$$\{h_1, \ldots, h_n\} = BiLSTM(\{e_1, \ldots, e_n\}), \tag{5.5}$$

where e_i are embeddings of words in the question and h_j are concatenations of the forward and backward LSTM hidden outputs for the j-th position. All h_i are subsequently combined into a single fixed-size vector specifically in the form of a weighted sum with the weights

$$\alpha_j = \frac{\exp(w \cdot h_j)}{\sum_{t=1}^{n} \exp(w \cdot h_t)}, \tag{5.6}$$

determined via self-attention and where α_j quantifies the attention that needs to be put on the corresponding question word and w is the attention parameter vector learned as part of training. To this weighted sum representation of the question, we concatenate a one-hot 4-bit vector indicating the type of a question to encode the set *{yes/no, factoid, list, summary}* given the question type may effect the matching process.

Answer Sentence Encoding Similarly, we encode a candidate answer sentence representation using a second BiLSTM using word embeddings for the answer sentence tokens. The hidden outputs of the candidate answer sentence are combined using another attention layer just like for the question sentence. Then, the resulting two sentence representations are compared to each other in the next text matching component.

Semantic Matching Our matching component is based on well known metric learning constructs to measure relatedness or similarity between two vectors [Kulis et al., 2013]. We tested approaches ranging from simple dot product to bilinear maps and recent neural tensor networks [Socher et al., 2013]. Based on experiments, we finalized the bilinear map metric $g(\mathbf{s}, \mathbf{q}) = \mathbf{s}^T W \mathbf{q}$ where $\mathbf{s}$ and $\mathbf{q}$ are candidate answer sentence and question embeddings respectively as defined in the previous two paragraphs and W is the parameter matrix for the bilinear transformation. In the end, the output scores $g(\mathbf{s}, \mathbf{q})$ are passed to the logistic function. The network in Figure 5.1 is trained with the binary cross-entropy loss function to evaluate the prediction quality.

5.1.3 Training examples for QAMat

Each instance to train the QAMat model takes the form of a pair of sentences, one representing the question and the other representing the candidate answer sentence. An instance is positive if the second sentence in the pair is relevant to answering the question represented by the first sentence. We use the BioASQ data from previous years for training this. Specifically, all sentences of human curated text snippets in BioASQ data are labelled as the *relevant* group. To populate the *irrelevant* group, we randomly select from the relevant documents those sentences that do not appear in the relevant text snippets. Since the examples are from the relevant documents,

we expect the context to be related to the topic of the document but not directly containing content to glean the answer. We also sample *irrelevant* examples from the entire document collection given the chance of the random samples from over 27 million documents being relevant to the question is extremely low. The proportions for training are as follows:

- 50% of the sentences are relevant examples, and the other half are irrelevant examples.
- Among irrelevant examples, half are sampled from the relevant documents (but outside snippets that contain answers) and half are from the rest of the corpus (irrelevant documents).

5.1.4 MeSH distribution across questions and journals

QAMat component from Section 5.1.2 is our main explicit feature directly comparing question and document contents. Here we discuss an auxiliary feature involving thematic overlap between question contents and the journal in which a candidate document is published. The medical subject headings (MeSH) is a well-known standardized hierarchical vocabulary used to tag biomedical articles (just like keywords) to facilitate future thematic search by researchers who use acrshortnlm's PubMed search engine. Besides individual articles, a journal name is also assigned a set of MeSH terms. The MeSH terms for an article or journal can be treated as a thematic abstraction of the content in them. MeSH terms can also be extracted using NLM's medical text indexer (MTI) tool that outputs MeSH terms for any piece of text. Our intuition is that if we can build a distribution of MeSH terms occurring across questions and journals, we can use it to design a feature that takes as input the question and candidate document (thus its journal) and output a score for it based on thematic overlap.

We build a distribution matrix M where the rows are MeSH terms from questions in the training data and the columns are MeSH terms of the journals of the corresponding relevant training documents. Here $M[m_i][m'_j]$ contains the number of times in the training data we encountered a question with MeSH term m_i with a corresponding answer document whose journal has the associated MeSH term m'_j. More specifically, let $\mathcal{Q}$ is the set of questions in the training data. Let $R(Q)$ be the set of relevant documents for $Q \in \mathcal{Q}$. Let $t(Q)$ be the MeSH terms mentioned in Q and let $t(D)$ be the set of MeSH terms for the journal of document D. We fill the table M as below

$$\forall_{Q \in \mathcal{Q}} \, \forall_{D \in R(Q)} \, \forall_{m_i \in t(Q)} \, \forall_{m'_j \in t(D)} \, [M[m_i][m'_j] \mathrel{+}= 1], \tag{5.7}$$

where '$+= k$' refers to increment-by-k operation. We subsequently normalize each row by dividing each cell value with the sum of all elements in that row. With this, $M[m_i][m'_j]$ now represents $P(m'_j|m_i)$ — the probability estimate of encountering an answer document whose journal has MeSH term m'_j given the question contains term

m_i. With this setup, given a new question Q, for a candidate document D, the score is

$$\mu(Q, D) = \frac{1}{|t(Q)|} \sum_{m_i \in t(Q)} \sum_{m'_j \in t(D)} M[m_i, m'_j]. \tag{5.8}$$

It is straightforward to note $\mu \in [0, 1]$ given the normalization step in building M and the $1/t(Q)$ factor in computing μ.

5.1.5 Semantic predications in SemMedDB

SemMedDB [Kilicoglu et al., 2012] is a repository of semantic *predications* (subject-predicate-object triples) that are extracted from the biomedical scientific literature indexed by PubMed using rule-based NLP techniques. The acrshortnlm provides an updated SemMedDB every year to include predications from newer articles. In each predication, the subject and object are biomedical entities (e.g., diseases, drugs, and procedures) represented by concepts from the unified medical language system (UMLS). The predicates (e.g., *treats* and *causes*) that connect the subject and object come from an extended *semantic network*. For example, for a PubMed document sentence "We conclude that tamoxifen therapy is more effective for early stage breast cancer patients", SemMedDB would contain the predication (Tamoxifen Citrate [C0079589], *treats*, Breast Carcinoma [C0678222]) where the C codes in square braces represent UMLS unique concept identifiers for the entities. We note that relations in SemMedDB have corresponding provenance information of particular sentences (in PubMed citations) they came from. Given the BioASQ search corpus is also PubMed citations, we design features that capture semantic links between concept mentions in the question. Specifically, from a question sentence, we use NLM's MetaMap software to extract UMLS concepts $C(Q)$ for question Q. For a candidate document D, let $C(D)$ be all UMLS concepts that participated (either as subject or object) in at least one predication in D and let $R(D)$ be set of all predications in D. Our first binary feature $\pi^1(Q, D)$ is set to 1 if and only if $|\{(i, j) : i, j \in C(Q) \, and \, (i, p, j) \in R(D)\}| > 0$ for some predicate p. That is, π^1 fires only if there exists at least one SemMedDB triple in D whose subject and object are both present in Q. The second feature $\pi^2(Q, D) = (|C(Q) \cap C(D)|)/|C(Q)|$ is a numerical feature ($\in [0, 1]$) that measures the proportion of number of concepts present in both Q and semantic predication based concept set $C(D)$ to the total number of concepts in Q.

5.1.6 Feature weighting methods

Finally, to rerank the top few documents returned by the SDM model, we need a way to combine all the five scores derived from the (1) preliminary SDM retrieval (Section 5.1.1), (2) QAMat (Section 5.1.2), (3) MeSH distribution (Section 5.1.4), (4) SemMedDB relation match, and (5) SemMedDB concept proportion (Section 5.1.5). We note that we scale features to $[0, 1]$ range before combining them for final document ranking. Except the QAMat score, all other features score the entire document. For QAMat, we produce a score for each sentence in the candidate document. To arrive

at the final document level score, we can consider the average of all QAMat scores for all sentences in it, just the maximum value among sentences, or both the average and max scores. Based on our experiments, we chose the simpler maximum score option as involving the average score did not improve the validation set performances.

Adaptive Random Search (ARS) The ARS method is a particular instance of a class of stochastic optimization methods where a weighted sum of feature scores is used as the final score for ranking documents. In this case, we have five weights $\alpha_1, \ldots, \alpha_5$ such that $\sum_i \alpha_i = 1$, so the final score is also in $[0, 1]$ since all constituent scores are in that range too. ARS starts with a random configuration of α_is and incrementally updates them as it proceeds to explore he search space. It does not require derivatives when performing updates. Instead of using a fixed step size, ARS dynamically increases or decreases the step size based on the observed difference between the performances on a validation dataset. Karnopp [Karnopp, 1963] discusses the details of the ARS algorithm, which we incorporated in our system to optimize the weights for the ranking features.

Learning-to-Rank Algorithms Learning-to-rank [Liu, 2009] (L2R) has emerged from the machine learning community as an automated way of learning functions that can rank a list of documents in response to an input query based on different query-specific features extracted from the documents. We also compare ARS against a variety of L2R algorithms as implemented in the RankLib library[4]. For the training data, we use all five feature scores and a binary judgement ('relevant' or 'irrelevant') for each item. Whether we use ARS or an L2R algorithm, the feature weighting model is built solely from the training dataset.

5.2 Experiments and Results

We perform experiments on the BioASQ QA datasets (years 2014 through 2017) focusing on the past two years for testing scenarios to examine the efficacy of the proposed approaches. Before we get into our results, we outline some system configuration details for experiments.

- SDM component (Section 5.1.1): For this initial document retrieval component, we used its implementation by the Galago search engine [Croft et al., 2010]. Indexing of the documents was done by the Krovetz stemmer, included in the Galago system. The window width for the ordered query tokens (N in Section 5.1.1) is increased from the default setting of 1 to 3. The unordered width is increased from the default setting of 4 to 8 (M in Section 5.1.1). Empirically, this setting improved the recall scores. We choose the default settings in the Galago implementation of SDM and set unigram score weight $\lambda_T = 0.8$, ordered distance score weight $\lambda_O = 0.15$, and unordered window weight $\lambda_U = 0.05$. Finally, the maximum number of documents to be retrieved using SDM is set to 30.

[4]Open source collection of learning-to-rank implementations part of the Lemur project

- QAMat component (Section 5.1.2): For the neural matching component, we use pre-trained word embeddings with 300 dimensions trained on Wikipedia using *fastText* [Bojanowski et al., 2017a]. The dimensionality of the BiLSTM hidden layers is set to 256 (determined via experiments). For regularization, we apply a dropout to the inputs of the LSTM layers with the dropout rate of 0.3. The attention layer output is 512 dimensional given the hidden layer output is 256 dimensions in each direction in the BiLSTM. In order to indicate the type of the given question, four additional bits are appended to the question representation, hence the parameter matrix W of the following bilinear matching function is set to (512×516). The maximum number of epochs is set to 30 with early stopping enabled and batch size is fixed at 128. We train the model using Adamax optimizer with an initial learning rate of 0.005 and a weight decay of 0.0005. Gradient clipping is set to 10 to avoid the exploding gradient problem. All other network weights are based on default initializations in PyTorch [Paszke et al., 2017].

5.2.1 Experiments for the QAMat feature

In Table 5.2, we show the counts of datasets created for training the QAMat model as discussed in Section 5.1.3. We chose the datasets to be balanced given we do not want to compromise too much on recall and because we have other evidences (SDM, MeSH distribution, SemMedDB match scores) to alleviate precision trade-off concerns. For each question, the positive examples in the datasets were based on those found in the BioASQ datasets and negative examples were generated randomly from the rest of the corpus. We achieved test set accuracies of $\approx 87\%$ for the QAMat component. Next, we look at a sample question and QAMat scores (before they are passed to the sigmoid function) for answer sentences.

Table 5.2: Number of examples in the QAMat datasets for year 2016/17

dataset	relevant	irrelevant
train (2014–15)	23,466	23,466
test (2016)	16,706	16,706

(a) datasets for testing on year 2016

dataset	relevant	irrelevant
train (2014–16)	33,075	33,075
test (2017)	9,582	9,582

(b) datasets for testing on year 2017

Table 5.3: QAMat scores for sentences of a relevant and an irrelevant document for an example question

Question: ***Orteronel was developed for treatment of which cancer?***

Score	Sentence of a **relevant** document
0.9673	Orteronel also known as TAK-700 is a novel hormonal therapy that is currently in testing for the treatment of prostate cancer.
0.4328	Orteronel inhibits the 17,20 lyase activity of the enzyme CYP17A1, which is important for androgen synthesis in the testes, adrenal glands and prostate cancer cells.
0.0918	Preclinical studies demonstrate that orteronel treatment suppresses androgen levels and causes shrinkage of androgen-dependent organs, such as the prostate gland.
0.5679	Early reports of clinical studies demonstrate that orteronel treatment leads to reduced prostate-specific antigen levels, a marker of prostate cancer tumor burden, and more complete suppression of androgen synthesis than conventional androgen deprivation therapies that act in the testes alone.
0.0931	Treatment with single-agent orteronel has been well tolerated with fatigue as the most common adverse event, while febrile neutropenia was the dose-limiting toxicity in a combination study of orteronel with docetaxel.
0.4054	Recently, the ELM-PC5 Phase III clinical trial in patients with advanced-stage prostate cancer who had received prior docetaxel was unblinded as the overall survival primary end point was not achieved.
0.9050	However, additional Phase III orteronel trials are ongoing in men with earlier stages of prostate cancer.

Score	Sentence of a random (**irrelevant**) document
0.0009	The dynamics of antibody response in guinea pigs infected with Coxiella burnetii was investigated by microagglutination MA and complement-fixation CF tests with different preparations of C. burnetii antigens.
0.0108	At the onset of antibody response the highest antibody titres were detected by the MA test with natural antigen 2, later on by the MA test with artificial antigen 2.
0.0008	Throughout the 1-year period of observation, the CF antibody levels were usually lower and, with the exception of the highest infectious doses, the CF antibodies appeared later than agglutinating antibodies.
0.0008	There was no difference in the appearance of agglutinating and CF antibodies directed to antigen 1.
0.0143	Inactivation of the sera caused a marked decrease in antibody titres when tested with artificial antigen 2, whereas the antibody levels remained unchanged when tested with natural antigen 2.

Table 5.3 shows how the QAMat model scores the sentences of an example relevant document and also the ones of another random irrelevant document for the question "Orteronel was developed for treatment of which cancer?". As we can see, the relevant document sentences that succinctly discuss treatment of cancer with orteronel have

scored high. Other sentences in the document that contain a lot more information do not have as high a score as smaller sentences that pointedly talk about orteronel drug therapy for cancer. All the sentences in the irrelevant document attain negative scores, all of which are worse than the lowest score achieved by the relevant sentences.

5.2.2 L2R vs. ARS for feature weighting

Table 5.4 shows the mean average precision (MAP) results when using different feature weighting methods. Surprisingly, ARS outperforms all other methods except for one out of ten batches considered. *MART*, *Coordinate Ascent*, and *Random Forests* more or less perform at the same level but trail behind ARS. We believe L2R algorithms may perform better in situations where features used have non-trivial correlations. In this case, it appears the features considered may be contributing complementary evidence.

5.2.3 Ablation study

We perform a feature ablation study to measure the contributions of different features discussed in Section 5.1. We first build a full model consisting of all features and subsequently drop each component, one at a time, to note the dip in performance (here MAP). Table 5.5 shows the results of these experiments for test sets from 2016 and 2017. The first rows in Table 5.5 (a) and Table 5.5 (b) have results from our full model and the last rows are based on the baseline SDM model (Section 5.1.1). Rows 2–4 indicate dropped components from Sections 5.1.2–5.1.5 respectively. The bold scores indicate the values that had the biggest drop from the corresponding full featured model score in the first row. We also note that the blue colored scores (1st rows) indicate the best performance achieved in each test batch. That is, in all batches, our fully featured model obtained the best scores.

We display the optimized $[0, 1]$ ARS weights in Table 5.5 in columns 2–6. We observe that QAMat score takes the highest weight by a large margin compared to other feature weights. Furthermore, QAMat's weight increases in 2017 compared with its weight in 2016 potentially due to the availability of more training data for 2017. However the baseline SDM model (last rows) by itself does reasonably well but scores around 2% below our full model's MAP. Moreover, our model can highlight sentences based on high QAMat scores that are expected to contain crucial information pertinent for answering the question. Coming to ablation results, from rows 2–4, we notice that dropping the QAMat component causes the biggest drop in MAP in most of the cases. Although the MeSH distribution and SemMedDB features were useful, the ablation results show that their contribution is much less than that of the baseline SDM scores and QAMat scores.

5.3 Related Work

Our main contribution here is retrieval of relevant documents with an end goal of finding answers to specific questions in biomedicine. Unlike other ad hoc IR tasks,

Table 5.4: Learning-to-rank methods comparison based on MAP (algorithms — A1: MART, A2: RankBoost, A3: AdaRank, A4: CoordAscent, A5: LambdaMART, A6: RandForests)

BioASQ test datasets		MAP for learning-to-rank algorithms					
	ARS	A1*	A2	A3	A4	A5	A6
year 2016, batch 1	**0.4438**	0.4181	0.3731	0.3792	0.4296	0.4025	0.4175
year 2016, batch 2	**0.4780**	0.4625	0.3698	0.4396	0.4493	0.4497	0.4736
year 2016, batch 3	**0.4534**	0.4198	0.3366	0.4009	0.4274	0.4026	0.4417
year 2016, batch 4	**0.4388**	0.4036	0.3490	0.3813	0.4127	0.4022	0.4296
year 2016, batch 5	0.3722	0.3563	0.2869	0.3263	0.3551	0.3314	**0.3729**
year 2017, batch 1	**0.4075**	0.3843	0.2616	0.1233	0.3786	0.3517	0.3975
year 2017, batch 2	**0.4363**	0.4334	0.3300	0.1457	0.4299	0.4227	0.4263
year 2017, batch 3	**0.4534**	0.4377	0.3223	0.1536	0.4456	0.4105	0.4434
year 2017, batch 4	**0.3891**	0.3693	0.2598	0.1193	0.3763	0.3362	0.3791
year 2017, batch 5	**0.2316**	0.2068	0.1226	0.0793	0.2170	0.1887	0.2216

Table 5.5: QAMat: Ablation study — Bold entries indicate biggest drop in MAP and blue entries correspond to best MAP values (w_1: SDM score, w_2: QAMat, w_3: MeSH distribution, w_4: SemMedDB1, w_5: SemMedDB2)

Models	Optimized ARS Weights					MAP				
	w_1	w_2	w_3	w_4	w_5	batch1	batch2	batch3	batch4	batch5
All	0.30	0.48	0.09	0.01	0.13	0.444	0.478	0.453	0.439	0.372
– QAMat	0.45	—	0.29	0.07	0.19	**0.420**	0.472	**0.423**	**0.415**	0.360
– MeSH distribution	0.33	0.58	—	0.03	0.06	0.440	**0.466**	0.448	0.431	0.361
– SemMedDB	0.29	0.54	0.17	—	—	0.435	0.468	0.433	0.416	**0.352**
Baseline	1.00	—	—	—	—	0.428	0.471	0.431	0.422	0.351

Models	Optimized ARS Weights					MAP				
	w_1	w_2	w_3	w_4	w_5	batch1	batch2	batch3	batch4	batch5
All	0.17	0.73	0.04	0.01	0.06	0.408	0.436	0.453	0.389	0.232
– QAMat	0.52	—	0.18	0.24	0.05	**0.378**	**0.419**	**0.437**	**0.369**	0.218
– MeSH distribution	0.31	0.55	—	0.06	0.08	0.396	0.422	0.447	0.377	**0.214**
– SemMedDB	0.30	0.52	0.18	—	—	0.381	0.432	0.446	0.376	0.215
Baseline	1.00	—	—	—	—	0.396	0.418	0.438	0.375	0.213

the BioASQ IR task is unique in the sense that it is part of a more complex set of tasks including snippet retrieval and QA. In this section we briefly discuss other efforts related to this study.

Biomedical information retrieval has benefited from multiple shared tasks including TREC genomics [Roberts et al., 2009], clinical decision support [Roberts et al., 2016], and precision medicine [Roberts et al., 2017] tracks, the CLEF user-centered

health information retrieval task [Zuccon et al., 2016], and the BioASQ retrieval and QA task [Tsatsaronis et al., 2015]. The use of neural approaches for IR is on the rise in general [Onal et al., 2018], also for question-answer matching [Tran and Niedereée, 2018] and biomedical QA [Mollá, 2017, Wiese et al., 2017]. However, classical non-neural IR approaches especially those that employ pseudo relevance feedback and extensions of SDM model are topping the BioASQ IR task during recent years [Jin et al., 2017]. Our immediate goal is to combine the best of both worlds to build a superior IR system as elaborated in future research directions in Section 5.4.

5.4 QAMat Summary

In this chapter, we examined the effectiveness of the three different relevance measures for a biomedical document retrieval task where the query is a biomedical question in the form of interrogative sentence. The first measure involves computing matching scores via dense neural representations of both the question sentence and candidate answer sentences. The second measure utilizes thematic overlap between a document and the question based on distributional information of MeSH terms in questions and journals of corresponding answer documents. The third prioritizes documents that contain relations between biomedical concepts found in the question. We demonstrate that our proposed features help improve the retrieval quality consistently, and the official results in the 2018 BioASQ task (Table 5.1) confirm the effectiveness of our approach. Next we discuss some future research directions.

- Based on the SDM model in Section 5.1.1, we limit the number of documents to retrieve for reranking to 30. Although it is important to limit the size of the candidate document set to be reranked, additional experiments where pseudo relevance feedback is employed on top of SDM might be beneficial. That is, based on the top scoring (using the QAMat model) sentences in the top 30 documents, we may be able to expand the query to obtain more highly relevant documents with a second SDM fetch operation. The expansion can be in the form of new query terms or entities that ought to be included in the query.

- We used the type of question (yes/no, factoid, list, or summary) as part of the question representation matching process in Section 5.1.2. However, the 4-bit vector that represents the question type is added *after* the attention mechanism is applied to form a weighted vector for the question. It would be interesting to see how the scoring would change if the question type information is used as part of the attention mechanism. This can be accomplished by choosing a different attention parameter vector for each question type. Although this would be more time consuming, it might help the attention mechanism to focus more on words that might matter based on the question type.

- Also, for factoid and list question types, we may be able to ascertain the semantic type of the entities that constitute the answer. For the example for the question in Table 5.3, through NLP methods involving dependency parsing, we might be able to determine that the answer entity is a disease (cancer,

specifically). We can then parametrize the attention mechanism for the answer sentence and also the matching process based on this additional piece of information about the answer type. For instance, a candidate sentence that has more entities of the answer type detected in the question ought to be scored higher than other sentences that do not contain answer type entities.

- If we are able to use NLP to abstract out the relations in a question, we can also exploit the external SemMedDB knowledge base (Section 5.1.5) in a more effective way. Again for the question in Table 5.3, we can see it as a graph pattern query that involves the edges (Orteronel, *treats*, $?x$) and ($?x$, *is_a*, `disease`). Given we have the set of relations obtained from every PubMed citation as part of SemMedDB, the question pattern can be matched against relation edges found in candidate answer documents. The degree of pattern match can be measured with the fraction of number of edges in the pattern that match with relations in the document. We believe this is a more powerful feature that can help us with future participation in BioASQ challenges.

- Finally, an ideal retrieval system would have a visualization component that helps us assess what evidences in the document the model figured were important to rank it higher than others. To this end, we can imagine a color coded scheme where the intensity of the color denotes the importance of a word and is set based on the attention weights (Section 5.1.2) for the answer sentence. This will help system builders debug their models and aid end users in ascertaining the true relevance of any document returned by the system as a top match.

Chapter 6 TASumm: Literature Retrieval for Precision Medicine with Neural Matching and Faceted Summarization [1]

This chapter presents the fourth and final study of this dissertation to address the semantic gap issue between a query and document. As discussed in Section 2.2.2, query refinement (QR) attempts to minimize the difference between the actual query statement and the user's original intent by manipulating the user query. Query expansion (QE) is the most representative method of QR. Much of the previous research efforts explored QE methods, but not many focused on manipulating documents to reduce the semantic gaps between them. In this chapter, we propose a neural model that translates a document into query-like sentences under various biomedical themes. We study the effectiveness of this method for the document retrieval task with a particular setup in mind.

Information retrieval (IR) for precision medicine (PM) often involves looking for multiple pieces of evidence that characterize a patient case. This typically includes at least the name of a condition and a genetic variation that applies to the patient. Other factors such as demographic attributes, comorbidities, and social determinants may also be pertinent. As such, the retrieval problem is often formulated as *ad hoc* search but with multiple facets (e.g., disease, mutation) that may need to be incorporated. In this chapter, we present a document reranking approach that combines neural query-document matching and text summarization toward such retrieval scenarios. Our architecture builds on the basic BERT model with three specific components for reranking: (a) document-query matching (b) keyword extraction and (c) facet-conditioned abstractive summarization. The outcomes of (b) and (c) are used to essentially transform a candidate document into a concise summary that can be compared with the query at hand to compute a relevance score. Component (a) directly generates a matching score of a candidate document for a query. The full architecture benefits from the complementary potential of document-query matching and the novel document transformation approach based on summarization along PM facets. Evaluations using NIST's TREC-PM track datasets (2017–2019) show that our model achieves state-of-the-art performance.

The U.S. National Institutes of Health (NIH)'s precision medicine (PM) initiative [Collins and Varmus, 2015] calls for designing treatment and preventative interventions considering genetic, clinical, social, behavioral, and environmental exposure variability among patients. The initiative rests on the widely understood finding that considering individual variability is critical in tailoring healthcare interventions to achieve substantial progress in reducing disease burden worldwide. Cancer was chosen as its near term focus with the eventual aim of expanding to other conditions. As the biomedical research enterprise strives to fulfill the initiative's goals, computing needs are also on the rise in drug discovery, predictive modeling for disease onset and

[1] This chapter is based on the previously published paper [Noh and Kavuluru, 2020b] appears in *Findings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*

progression, and in building NLP tools to curate information from the evidence base being generated.

Table 6.1: Example cases from 2019 TREC-PM dataset

Facet	Input
Disease	Melanoma
Genetic variation	BRAF (E586K)
Demographics	64-year-old female
Disease	Gastric cancer
Genetic variation	ERBB2 amplification
Demographics	64-year-old male

TREC Precision Medicine Series In a dovetailing move, the U.S. NIST's TREC (Text REtrieval Conference) has been running a PM track since 2017 with a focus on cancer [Roberts et al., 2020]. The goal of the TREC-PM task is to identify the most relevant biomedical articles and clinical trials for an input patient case. Each case is composed of (1) a disease name, (2) a gene name and genetic variation type, and (3) demographic information (sex and age). Table 6.1 shows two example cases from the 2019 track. So the search is *ad hoc* in the sense that we have a free text input in each facet but the facets themselves highlight the PM related attributes that ought to characterize the retrieved documents. We believe this style of faceted retrieval is going to be more common across medical IR tasks for many conditions as the PM initiative continues its mission.

The vocabulary mismatch problem is a prominent issue in medical IR given the large variation in the expression of medical concepts and events. For example, in the query *"What is a potential side effect for Tymlos?"* the drug is referred by its brand name. Relevant scientific literature may contain the generic name *Abaloparatide* more frequently. Traditional document search engines have clear limitations on resolving mismatch issues. The IR community has extensively explored methods to address the *vocabulary mismatch* problem, including query expansion based on relevance feedback, query term re-weighting, or query reconstruction by optimizing the query syntax.

Several recent studies highlight exploiting neural network models for query refinement in document retrieval (DR) settings. Nogueira and Cho [2017] address this issue by generating a transformed query from the initial query using a neural model. They use reinforcement learning (RL) to train it where an *agent* (i.e., reformulator) learns to reformulate the initial query to maximize the expected return (i.e., retrieval performance) through *actions* (i.e., generating a new query from the output probability distribution). In a different approach, Narayan et al. [2018] use RL for sentence ranking for extractive summarization.

In this chapter, building on the BERT architecture [Devlin et al., 2019], we focus on a different hybrid document scoring and reranking setup involving three components: (a) a *document relevance classification* model, which predicts (and inherently

scores) whether a document is relevant to the given query (using a BERT multi-sentence setup); (b) a *keyword extraction* model which spots tokens in a document that are likely to be seen in PM related queries; and (c) an *abstractive document summarization* model that generates a pseudo-query given the document context and a facet type (e.g., genetic variation) via the BERT encoder-decoder setup. The keywords (from (b)) and the pseudo-query (from (c)) are together compared with the original query to generate a score. The scores from all the components are combined to rerank top k (set to 500) documents returned with a basic Okapi BM25 retriever from a Solr index [Grainger and Potter, 2014] of the corpora.

Our main innovation is in pivoting from the focus on queries by previous methods to emphasis on transforming candidate documents into pseudo-queries via summarization. Additionally, while generating the pseudo-query, we also let the decoder output concept codes from biomedical terminologies that capture disease and gene names. We do this by embedding both words and concepts in a common semantic space before letting the decoder generate summaries that include concepts. Our overall architecture was evaluated using the TREC-PM datasets (2017–2019) with the 2019 dataset used as the test set. The results show an absolute 4% improvement in P@10 compared to prior best approaches while obtaining a small $\approx$ 1% gain in R-Prec. Qualitative analyses also highlight how the summarization is able to focus on document segments that are highly relevant to patient cases.

6.1 Neural Text Summarization and the *BERT-CRel* Embeddings

The basic reranking architecture we begin with is the Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2019] model. BERT is trained on a *masked language modeling* objective on a large text corpus such as *Wikipedia* and *BooksCorpus*. As a sequence modeling method, it has achieved state-of-the-art results in a wide range of natural language understanding (NLU) tasks, including machine translation [Conneau and Lample, 2019] and text summarization [Liu and Lapata, 2019]. With an additional layer on top of a pretrained BERT model, we can fine-tune models for specific NLU tasks. In our study, we utilize this framework in all three components by starting with a `bert-base-uncased` pretrained HuggingFace model [Wolf et al., 2020].

6.1.1 Neural text summarization

We plan to leverage both extractive and abstractive candidate document summarization in our framework. In terms of learning methodology, we view extractive summarization as a sentence (or token) classification problem. Previously proposed models include the RNN-based sequence model [Nallapati et al., 2017], the attention-based neural encoder-decoder model [Cheng and Lapata, 2016], and the sequence model with a global learning objective (e.g., ROUGE) for ranking sentences optimized via RL [Narayan et al., 2018, Paulus et al., 2018]. More recently, graph convolutional neural networks (GCNs) have also been adapted to allow the incorporation of global information in text summarization tasks [Sun et al., 2019, Prasad and Kan, 2019]. Ab-

stractive summarization is typically cast as a sequence-to-sequence learning problem. The encoder of the framework reads a document and yields a sequence of continuous representations, and the decoder generates the target summary token-by-token [Rush et al., 2015, Nallapati et al., 2016]. Both approaches have their own merits in generating comprehensive and novel summaries; hence most systems leverage these two different models in one framework [See et al., 2017, Liu and Lapata, 2019]. We use the extractive component to identify tokens in a candidate document that may be relevant from a PM perspective and use the abstractive component to identify potential terms that may not necessarily be in the document but nevertheless characterize it for PM purposes.

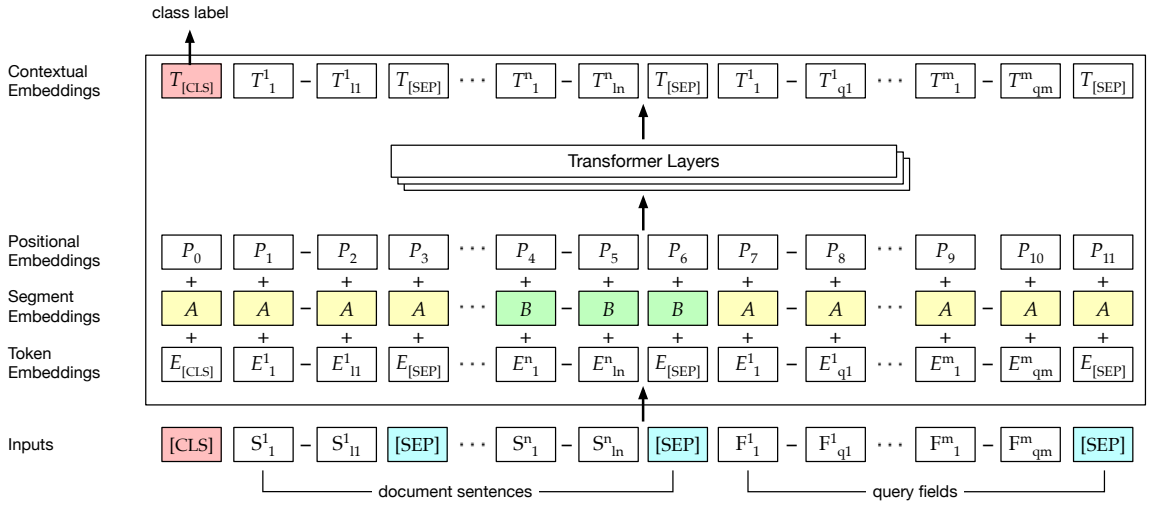
6.1.2 Word and entity embeddings

Most of the neural text summarization models, as described in the previous section, adopt the encoder-decoder framework that is popular in machine translation. As such the vocabulary on the decoding side does not have to be the same as that on the encoding side. We exploit this to design a summarization trick for PM where the decoder outputs both regular English tokens and also entity codes from a standardized biomedical terminology that captures semantic concepts discussed in the document. This can be trained easily by converting the textual queries in the training examples to their corresponding entity codes. This trick is to enhance our ability to handle vocabulary mismatch in a different way (besides the abstractive framing). We use specially optimized word/concept embeddings (BERT-CRel, see Chapter **??**) for this purpose. BERT-CRel embeddings are trained on biomedical literature abstracts that were annotated with entity codes in the Medical Subject Headings (MeSH) terminology; codes are appended to the associated textual spans in the training examples. So regular tokens and the entity codes are thus embedded in the same semantic space via pretraining with the *fastText* architecture [Bojanowski et al., 2017b]. Besides regular English tokens, the vocabulary of BERT-CRel thus includes 29,351 MeSH codes and a subset of supplementary concepts. In the dictionary, MeSH codes are differentiated from the regular words by a unique prefix; for example, *ϵmesh_d000123* for MeSH code *D000123*. With this, our summarization model can now translate a sequence of regular text tokens into a sequence of biomedical entity codes or vice versa. That is, we use MeSH as a new "semantic" facet besides those already provided by TREC-PM organizers. The expected output for the MeSH facet is the set of codes that capture entities in the disease and gene variation facets.

6.2 Methodology

In this effort, toward document reranking, we aim to measure the relevance match between a document and a faceted PM query. Each training instance is a 3-tuple (d, q, y_q^d) where q is a query, d is a candidate document, and y_q^d is a Boolean human adjudicated outcome: whether d is relevant to q. As mentioned earlier, first, we fine-tune BERT for a query-document relevance matching task modeled as a classification goal to predict y_q^d (`REL`). Next, we fine-tune BERT for token-level relevance classi-

Figure 6.1: BERT architecture for document relevance matching task `REL`



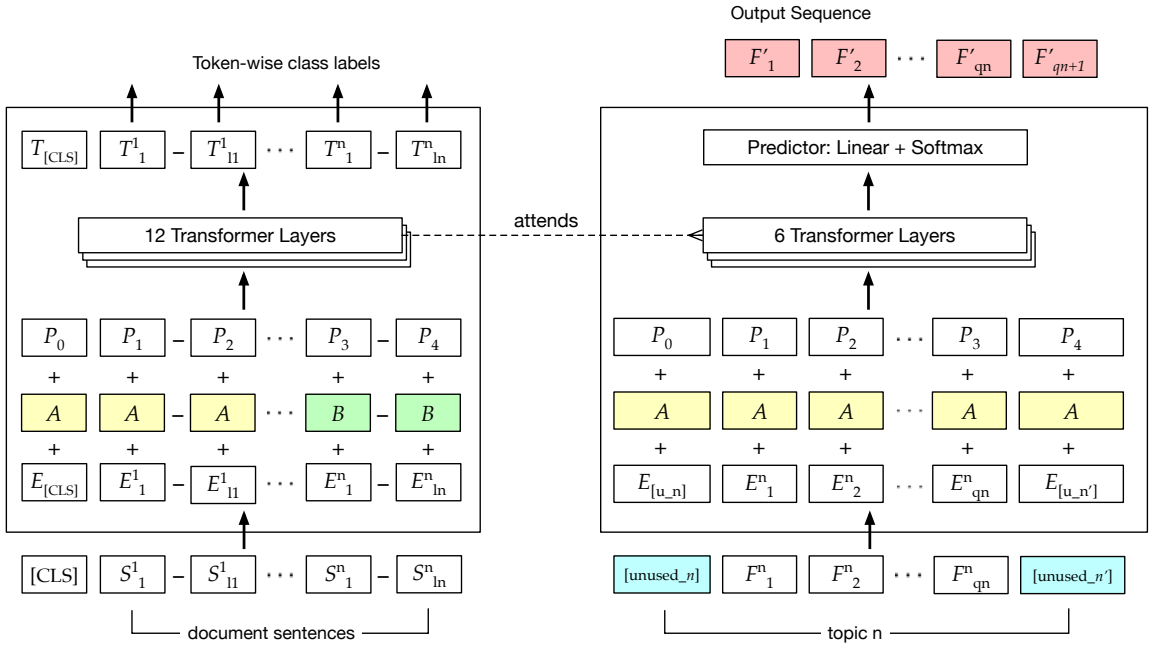
fication, different from `REL`, where a token in d is deemed relevant during training if it occurs as part of q. We name this model `EXT` for keyword extraction. Lastly, we train a BERT model in the seq2seq setting where the encoder is initialized with a pretrained `EXT` model. The encoder reads in d, and the decoder attends to the contextualized representations of d to generate a facet-specific pseudo-query sentence q_d, which is then compared with the original query q. We conceptualize this process as text summarization from a document to query sentences[2] and refer to it as `ABS`. All three models are used together to rerank a candidate d at test time for a specific input query.

6.2.1 Document relevance matching (`REL`)

Neural text matching has been recently carried out through Siamese style networks [Mueller and Thyagarajan, 2016], which also have been adapted to biomedicine [Noh and Kavuluru, 2018]. Our approach adapts the BERT architecture for the matching task in the multi-sentence setting as shown in Figure 6.1. We use BERT's tokenizer on its textual inputs, and the tokens are mapped to token embeddings. REL takes the concatenated sequence of a document and faceted query sentences. The functional symbols defined in the BERT tokenizer (e.g., `[CLS]`) are added to the input sequence. Each input sequence starts with a `[CLS]` token. Each sentence of the document ends with the `[SEP]` token with the last segment of the input sequence being the set of faceted query sentences, which end with another `[SEP]` token. In the encoding process, the first `[CLS]` token collects features for determining document relevance to the query. BERT uses segment embeddings to distinguish two sentences. We, however, use the them to distinguish multiple sentences within a document. For each sentence, we assign a segment embedding either A or B alternatively. The positional

[2]We note queries here are not grammatically well-formed sentences but are essentially sequences generated by the summarization model.

Figure 6.2: Architecture of the abstractive document summarization (ABS) model. The encoder (left component) is initialized with a pretrained EXT model. The class labels of the encoder are used for identifying keywords of the document, and the output sequences generated from the decoder (right component) are used to build a pseudo-query, which is later used in computing similarity scores for the user provided query.



embeddings encode the sequential nature of the inputs. The token embeddings along with the segment and positional embeddings pass through the transformer layers. Finally, we use the $[0, 1]$ output logit from the [CLS] token ($T_{[CLS]}$) as the matching score for the input document and query. We note that we don't demarcate any boundaries within different facets of the query.

6.2.2 Keyword extraction (EXT)

EXT model has an additional token classification layer on top of the pretrained BERT. The output of a token is the logit that indicates the log of odds of the token's occurrence in the query. With TREC-PM datasets, we expect to see the logits fire for words related to different facets with an optimized EXT at test time. Unlike the REL model, the input to EXT is a sequence of words in a document without any [SEP] delimiters. However, the model still learns the boundaries of the sentence via segment inputs. This component essentially generates a brief extractive summary of a candidate document. Furthermore, contextualized embeddings from EXT are used in the decoder of ABS to generate faceted abstractive document summaries.

6.2.3 Abstractive document summarization (ABS)

ABS employs a standard seq2seq attention model, similar to that by [Nallapati et al., 2016], as shown in Figure 6.2. We initialize the parameters of the encoder with a pretrained EXT model. The decoder is a 6-layer transformer in which the self-attention layers attend to only the earlier positions in the output sequence as is typical in auto-regressive language models. In each training phase step, the decoder takes each previous token from the reference query sentence; in the generation process, the decoder uses the token predicted one step earlier.

Table 6.2: Signals for different facets of the patient cases

Facets	(bos) / (eos)
Disease name	[unused_0]/[unused_100]
Genetic variations	[unused_1]/[unused_101]
Demographic info.	[unused_2]/[unused_102]
MeSH terms	[unused_3]/[unused_103]
Document keywords	[unused_4]/[unused_104]

We differentiate facets by the special pairs of tokens assigned to each topic. In a typical generation process, special tokens such as [bos] (begin) and [eos] (end) are used to indicate sequence boundaries. In this model, we use some special tokens in the BERT vocabulary with prefix 'unused_'. Specifically, [unused_i] and [unused_$(100+i)$] are used as bos and eos tokens respectively for different facets. These facet signals are the latent variables for which ABS is optimized. Through them, ABS learns not only the thematic aspects of the queries but also the meta attributes such as length. The special tokens for facets are listed in Table 6.2 (the last row indicates a new auxiliary facet we introduce in Section 6.3.1).

Each faceted query is enclosed by its assigned bos/eos pair, and the decoder of ABS learns $p_\theta(x_i|x_{<i}, x_0)$, where x_0 is the facet signal. As in the encoder and the original transformer architecture [Vaswani et al., 2017], we add the sinusoidal positional embedding P_t and the segment vector A (or B) to the token embedding E_t. Note that the dimension of the token embeddings used in the encoder (BERT embeddings) is different from that of the decoder (our custom BMET embeddings), which causes a discrepancy in computing context-attentions of the target text across the source document. Hence, we add an additional linear layer to project the constructed decoder embeddings ($E^n_j + A + P_i$ in the right hand portion of Figure 6.2) into the same space of embeddings of the encoder.

These projected embeddings are fed to the decoder's transformer layers. Each transformer layer applies multi-head attention for computing the self- and context-attentions. The attention function reads the input masks to preclude attending to future tokens of the input and any padded tokens (i.e., [PAD]) of the source text. Both attention functions apply a residual connection [He et al., 2016]. Lastly, each transformer layer ends with a position-wise feedforward network. Final scores for each token are computed from the linear layer on top of the transformer layers. In

training, these scores are consumed by a cross-entropy loss function. In generation process, the softmax function is applied over the vocabulary yielding a probability distribution for sampling the next token.

Finally to generate the pseudo-query, we use *beam search* to find the most probable sentence among predicted candidates. The scores are penalized by two measures proposed by Wu et al. [2016, Equation 14]: (1) The length penalty $lp(Y) = (5 + |Y|)^\alpha / (5 + 1)^\alpha$, where $|Y|$ is the current target length and $0 < \alpha < 1$ is the length normalization coefficient. (2) The coverage penalty

$$cp(X, Y) = \beta \sum_{i=1}^{|X|} \log(\min(\sum_{j=1}^{|Y|} p_{i,j}, 1.0)),$$

where $p_{i,j}$ is the attention score of the j-th target word y_j on the i-th source word x_i, $|X|$ is the source length, and $0 < \beta < 1$ is the coverage normalization coefficient. Intuitively, these functions avoid favoring shorter predictions and yielding duplicate terms. We tune the parameters of the penalty functions ($\alpha = \beta = 0.4$), with grid-search on the validation set for TREC-PM.

6.2.4 Reranking with `REL`, `EXT`, and `ABS`

The main purpose of the models designed in the previous subsections is to come up with a combined measure for reranking. For a query q, let $d_1, \ldots, d_r$, be the set of top r (set to 500) candidate documents returned by the Solr BM25 eDisMax query. It is straightforward to impose an order on d_j through `REL` via the output probability estimates of relevance. Given q, for each d_j we generate the pseudo-query (summary) q_{d_j} by concatenating all distinct words in the generated pseudo-query sentences by `ABS` along with the words selected by `EXT`. Repeating words and special tokens are removed. Although faceted summaries are generated through ABS, in the end q_{d_j} is essentially the set of all unique terms from `ABS` and `EXT`. Each d_j is now scored by comparing q and q_{d_j} via two similarity metrics: The ROUGE-1 recall score, s_{ROUGE} [Lin, 2004], and a cosine similarity based score computed as

$$s_{\cos}(q, q_{d_j}) = \frac{1}{|q|} \sum_{y \in q} \max_{x \in q_{d_j}} (\cos(e_y, e_x)),$$

where e_i denote vector representations from BMET embeddings (Section 6.1.2).

Overall, we compute four different scores (and hence rankings) of a document: (1) the retrieval score returned by Solr, (2) the document relevance score by `REL`, (3) pseudo-query based ROUGE score, and (4) pseudo-query similarity score $s_{\cos}$. In the end we merge the rankings with *reciprocal rank fusion* [Cormack et al., 2009] to obtain the final ranked list of documents. The results are compared against the state-of-the-art models from the 2019 TREC-PM task.

6.3 Experimental Setup

6.3.1 Data

Across 2017–2019 TREC-PM tasks, we have a total of 120 patient cases and 63,387 qrels (document relevance judgments) as shown in Table 6.3.

Table 6.3: Number of queries and pooled relevance judgments in the 2017–19 TREC-PM tracks

Year	Queries	Documents (rel. / irrel.)
2017	30	3,875 / 18,767
2018	50	5,588 / 16,841
2019	40	5,544 / 12,772

We create two new auxiliary facets, *MeSH terms* and *Keywords*, derived from any training query and document pair. We already covered the MeSH facet in Section 6.1.2. *Keywords* are those assigned by authors to a biomedical article to capture its themes and are downloadable from NIH's NCBI website. If no keywords were assigned to an article, then we use the set of preferred names of MeSH terms (assigned to the articles by trained NIH coders) for that example. The following list shows associated facets for a sample training instance:

- **Disease**: prostate cancer
- **Genetic variations**: ATM deletion
- **Demographics**: 50-year-old male
- **MeSH terms**: D011471, D064007
- **Keywords**: Aged, Ataxia Telangiectasia mutated Proteins

Each model consumes data differently, as shown in Table 6.4. `REL` takes a document along with the given query as the source input and predicts document-level relevance. We consider a document with the human judgment score either 1 (partially relevant) or 2 (totally relevant) as relevant for this study. Note that we do not include MeSH terms in the query sentences for `REL`. `EXT` reads in a document as the source input and predicts token-level relevances. During training, a relevant token is one that occurs in the given patient case. A pseudo-query is the output for `ABS` taking in a document and a facet type.

6.3.2 Implementation details

For all three models, we begin with the pretrained `bert-base-uncased` HuggingFace model [Wolf et al., 2020] to encode source texts. We use BERT's *WordPiece* [Schuster and Nakajima, 2012] tokenizer for the source documents.

Table 6.4: TASumm: Types of source and target for each model.

Model	Source	Target
REL	doc+query_sentences	doc relevance
EXT	doc	token relevances
ABS	doc+facet_signal	a pseudo-query

`REL` and `EXT` are trained for 30,000 steps with batch size of 12. The maximum number of tokens for source texts is limited to 384. As the loss function of these two models, we use *weighted* binary cross entropy. That is, given high imbalance with many more irrelevant instances than positive ones, we put different weights on the classes in computing the loss according to the target distributions (proportions of negative examples are 87% for `REL` and 93% for `EXT`). The loss is

$$l(x, y; \theta) = -w_y[y \log p(x) + (1 - y) \log(1 - p(x))], \tag{6.1}$$

where $w_0 = 13/87 = 0.15, w_1 = 1$ for `REL` and $w_0 = 7/93 = 0.075, w_1 = 1$ for `EXT`. Adam optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, starting learning rate $lr = 1e^{-5}$, and fixed weight decay of 0.0 was used. The learning rate is reduced when a metric has stopped improving by using the *ReduceLROnPlateau* scheduler in *PyTorch*.

For the decoder of `ABS`, multi-head attention module from OpenNMT [Klein et al., 2017] was used. To tokenize target texts, we use the NLTK word tokenizer unlike the one used in the encoder; this is because we use customized word embeddings, the *BMET* embeddings (Section 6.1.2), trained with a domain-specific corpus and vocabulary. The vocabulary size is 120,000 which includes the 29,351 MeSH codes. We use six transformer layers in the decoder. Model dimension is 768 and the feed-forward layer size is 2048. We use different initial learning rates for the encoder and decoder, since the encoder is initialized with a pretrained `EXT` model: $1e^{-5}$ (encoder) and $1e^{-3}$ (decoder). Negative log-likelihood is the loss function for `ABS` on the ground-truth faceted query sentences. For beam search in `ABS`, `beam_size` is set to 4. At test time, we select top two best predictions and merge them into one query sentence. The max length of target sentence is limited to 50 and a sequence is incrementally generated until `ABS` outputs the corresponding `eos` token for each facet. All parameter choices were made based on best practices from prior efforts and experiments to optimize P@10 on validation subsets.

6.4 Evaluations and Results

We conducted both quantitative and qualitative evaluations with example outcomes. The final evaluation was done on the 2019 TREC-PM dataset while all hyperparameter tuning was done using a training and validation dataset split of a shuffled combined set of instances from 2017 and 2018 tracks (20% validation and the rest for training).

6.4.1 Quantitative evaluations

We first discuss the performances of the constituent `REL` and `EXT` models that were evaluated using train and validation splits from 2017–2018 years. Table 6.5 shows their performance where `REL` can recover ≈ 92% of the relevant documents and `EXT` can identify ≈ 88% of the tokens that occur in patient case information, both at precisions over 90%. We find that learning a model for identifying document/token-level relevance is relatively straightforward even with the imbalance.

Table 6.5: Retrieval performance of `REL` and `EXT`.

	`REL`			`EXT`		
	P	R	F1	P	R	F1
Train	0.9814	0.9384	0.9594	0.9624	0.8877	0.9236
Valid	0.9266	0.9147	0.9206	0.9413	0.8732	0.9060

Next we discuss the main results comparing against the top two teams (rows 1–2) in the 2019 track in Table 6.6. Before we proceed, we want to highlight one crucial evaluation consideration that applies to any TREC track. TREC evaluates systems in the *Cranfield* paradigm where pooled top documents from all participating teams are judged for relevance by human experts. Because we did not participate in the original TREC-PM 2019 task, our retrieved results are not part of the judged documents. Hence, we may be at a slight disadvantage when comparing our results with those of teams that participated in 2019 TREC-PM. Nevertheless, we believe that at least the top few most relevant documents are typically commonly retrieved by all models. Hence we compare with both P@10 and R-Prec (P@all-relevant-doc-count) measures.

Table 6.6: Model performances compared with the top entries in 2019 TREC-PM.

Model	R-Prec	P@10
julie-mug [Faessler et al., 2020]	0.3572	0.6525
BITEM_PM [Caucheteur et al., 2020]	0.3166	0.6275
Baseline: Solr eDisMax	0.2307	0.5200
Baseline + Solr MLT	0.1773	0.2625
Baseline + `REL`	**0.3912**	0.6750
Baseline + `ABS`	0.2700	0.5625
Baseline + `REL`+`ABS`	0.3627	**0.6985**

Our baseline Solr query results are shown in row 3 with subsequent rows showing results from additional components. Solr *eDisMax* is a document ranking function which is based on the BM25 [Jones et al., 2000] probabilistic model. We also evaluate

eDisMax with Solr MLT (`MoreLikeThis`), in which a new query is generated by adding a few "interesting" terms (top TF/IDF terms) from the retrieved documents of the initial *eDisMax* query. This traditional relevance feedback method (row 4) method has decreased the performance from the baseline and hence has not been used in our reranking methods.

All our models (rows 5–7) present stable baseline scores in P@10 and the combined method (`+REL+ABS`) tops the list with a 4% improvement over the prior best model [Faessler et al., 2020]. Baseline with `REL` does the best in terms of R-Prec. Both prior top teams rely heavily on query expansion through external knowledge bases to add synonyms, hypernyms, and hyponyms of terms found in the original query.

6.4.2 Qualitative analysis

Table 6.7: Sample facet-conditioned document summarizations by `ABS`

Input document:

> Association between BRAF v600e mutation and the clinicopathological features of solitary papillary thyroid microcarcinoma. (PMID: 28454296)

Generated sentences with facet signals:

`[unused_0]` papillary intrahepatic cholangiocarcinoma
`[unused_1]` braf v600e
`[unused_2]` D018281 C535533
`[unused_3]` papillary thyroid braf clinicopathological v600e

Input document:

> Identification of differential and functionally active miRNAs in both anaplastic lymphoma kinase (ALK)+ and ALK− anaplastic large-cell lymphoma. (PMID: 20805506)

Generated sentences with facet signals:

`[unused_0]` lymphoma
`[unused_1]` anaplastic lymphoma alk cell bradykinin
`[unused_2]` D002471 D017728 D000077548
`[unused_3]` lymphoma alk receptor tyrosine kinase

Table 6.7 presents sample pseudo-queries generated by `ABS`. The summaries of the first document show some novel words, *intrahepatic* and *cholangiocarcinoma*, that do not occur in the given document (we only show title for conciseness, but the abstract also does not contain those words). The model may have learned the close relationship between *cholangiocarcinoma* and *BRAF v600e*, the latter being part of the genetic facet of the actual query for which PMID: 28454296 turns out to be relevant. Also embedding proximity between *intrahepatic* and *cholangiocarcinoma* may

have introduced both into the pseudo query, although they are not central to this document's theme. Still, this maybe important in retrieving documents that have an indirect (yet relevant) link to the query through the pseudo-query terms. This could be why, although `ABS` underperforms `REL`, it still complements it when combined (Table 6.6). The table also shows that `ABS` can generate concepts in a domain-specific terminology. For example, the second document yields following MeSH entity codes, which are strongly related to the topics of the document: *D002471* (Cell Transformation, Neoplastic), *D017728* (Lymphoma, Large-Cell, Anaplastic), and *D000077548* (Anaplastic Lymphoma Kinase).

Figure 6.3 depicts words highlighted by `EXT`. Evidently, we see terms related to the regulations of gene expressions, proteins, or disease names featuring more prominently. Figure 6.4 shows how `ABS` reads the source document differently depending on which facet signal it starts with, in the process of query generation; compared to `[unused0]` (disease facet), the attention heat map by `[unused1]` (genetic facet) focuses more on the words related to gene regulations.

Figure 6.3: Heatmap of the classification scores by `EXT`. Darker red indicates relatively higher probability of the token being relevant to the theme of the TREC-PM datasets.

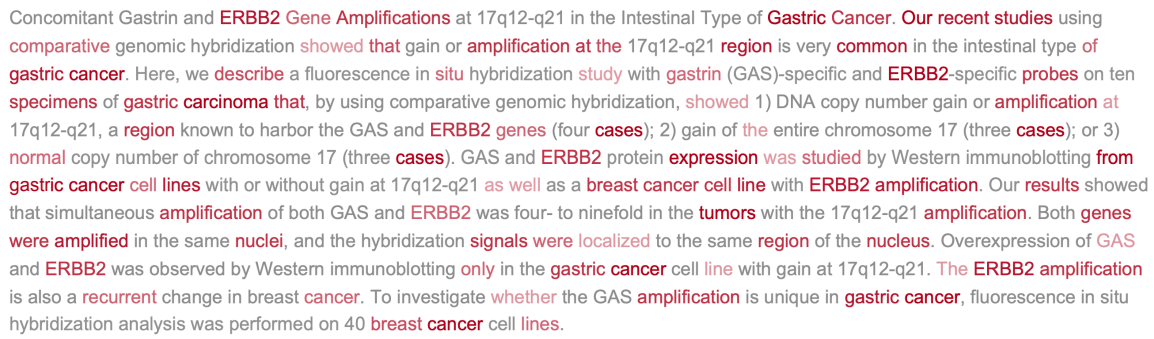
Efficacy of the dual PI3K and mTOR inhibitor NVP-BEZ235 in combination with nilotinib against BCR-ABL-positive leukemia cells involves the ABL kinase domain mutation. Imatinib, an ABL tyrosine kinase inhibitor (TKI), has shown clinical efficacy against chronic myeloid leukemia (CML). However, a substantial number of patients develop resistance to imatinib treatment due to the emergence of clones carrying mutations in the protein BCR-ABL. The phosphoinositide 3 kinase (PI3K)/Akt/mammalian target of rapamycin (mTOR) pathway regulates various processes, including cell proliferation, cell survival, and antiapoptosis activity. In this study, we investigated the efficacy of NVP-BEZ235, a dual PI3K and mTOR inhibitor, using BCR-ABL-positive cell lines. Treatment with NVP-BEZ235 for 48 h inhibited cell growth and induced apoptosis. The phosphorylation of the AKT kinase, eukaryotic initiation factor 4-binding protein 1 (4E-BP1), and p70 S6 kinase were decreased after NVP-BEZ235 treatment. The combination of NVP-BEZ235 with a BCR-ABL kinase inhibitor, imatinib, or nilotinib, induced a more pronounced colony growth inhibition, whereas the combination of NVP-BEZ235 and nilotinib was more effective in inducing apoptosis and reducing the phosphorylation of AKT, 4E-BP1, and S6 kinase. NVP-BEZ235 in combination with nilotinib also inhibited tumor growth in a xenograft model and inhibited the growth of primary T315I mutant cells and ponatinib-resistant cells. Taken together, these results suggest that administration of the dual PI3K and mTOR inhibitor NVP-BEZ235 may be an effective strategy against BCR-ABL mutant cells and may enhance the cytotoxic effects of nilotinib in ABL TKI-resistant BCR-ABL mutant cells.

6.4.3 Machine configuration and runtime

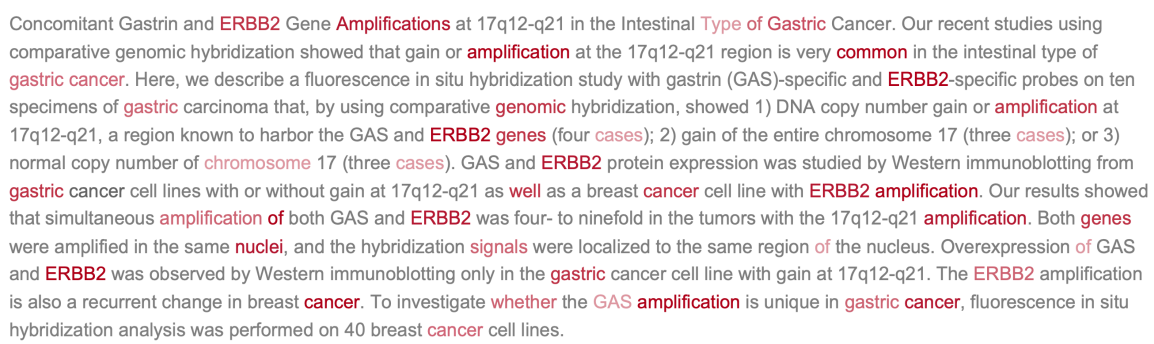
All training and testing was done on a single Nvidia Titan X GPU in a desktop with 64GB RAM. The corpus to be indexed had 30,429,310 biomedical citations (titles and abstracts of biomedical articles[3]). We trained the three models for five epochs and the training time per epoch (80,319 query, doc pairs) is 69 mins for `REL`, 72 mins for `EXT`, and 303 mins for `ABS`. Coming to test time, per query, the Solr eDisMax query returns top 500 results in 20 ms. Generating pseudo-queries for 500 candidates via `EXT` and `ABS` takes 126 seconds and generating `REL` scores consumes 16 seconds. So per query, it takes nearly 2.5 mins at test time to return a ranked list of documents. Although this does not facilitate real time retrieval as in commercial search engines, given the complexity of the queries, we believe this is at least near real time offering

[3]Due to copyright issues with full-text, TREC-PM is only conducted on abstracts/titles of articles available on PubMed.

Figure 6.4: Comparison between the attention heatmaps on a sample document conditioned by field signals in ABS model.



(a) Attention heatmap produced by `[unused0]` signal (topic of disease)



(b) Attention heatmap produced by `[unused1]` signal (topic of generic variants and gene regulations)

a convenient way to launch PM queries. Furthermore, this comes at an affordable configuration for many labs and clinics with a smaller carbon footprint.

6.5 TASumm Summary

In this chapter, we proposed an ensemble document reranking approach for PM queries. It builds on pretrained BERT models to combine strategies from document relevance matching and extractive/abstractive text summarization to arrive at document rankings that are complementary in eventual evaluations. Our experiments also demonstrate that entity embeddings trained on an annotated domain specific corpus can help in document retrieval settings. Both quantitative and qualitative analyses throw light on the strengths of our approach.

One scope for advances lies in improving the summarizer to generate better pseudo-queries so that ABS starts to perform better on its own. At a high level, training data is very hard to generate in large amounts for IR tasks in biomedicine and this holds for the TREC-PM datasets too. To better train ABS, it may be better to adapt other biomedical IR datasets. For example, the TREC clinical decision support (CDS) task that ran from 2014 to 2016 is related to the PM task [Roberts et al., 2016]. A future goal is to see if we can apply our neural transfer learning [Rios

and Kavuluru, 2019] and domain adaptation [Rios et al., 2018] efforts to repurpose the CDS datasets for the PM task.

Another straightforward idea is to reuse generated pseudo-query sentences in the eDisMax query by Solr, as a form of pseudo relevance feedback. The $s_{\cos}$ expression in Section 6.2.4 focuses on an asymmetric formulation that starts with a query term and looks for the best match in the pseudo-query. Considering a more symmetric formulation, where, we also begin with the pseudo-query terms and average both summands may provide a better estimate for reranking. Additionally, a thorough exploration of how external biomedical knowledge bases [Wagner et al., 2020] can be incorporated in the neural IR framework for PM is also important [Nguyen et al., 2017].

Chapter 7 Conclusion and Future Work

This chapter summarizes the work presented in this dissertation. We will complete this dissertation by highlighting our main contributions and discussing the study limitations and future work directions.

7.1 Summary of Dissertation Results and Contributions

Until recently, modern IR systems relied on the bag-of-words approaches wherein indexing is based on the term/document IDs and their occurrence statistics. With such systems, document ranking methods depend on the exact lexical matching between the query and document texts, not considering any semantic relationships among lexically different words. This creates the vocabulary mismatch issue and related semantic gaps between a query and its relevant documents. This dissertation proposes various methods to resolve these two problems via a full range of neural network-based solutions.

The emergence of word embeddings is a significant milestone for NLP. General-purpose vector representations for words are used in most neural network models for various downstream NLP tasks; Word2vec, GloVe, fastText, ELMo, and different embeddings of the pre-trained language models (e.g., BERT) are among the most popular ones. In biomedical informatics, constructing vector representations for the biomedical concepts provides extensive ways of adapting carefully curated external knowledge sources. Chapter 3 (BERT-CRel) proposes the methods of learning distributed representations for biomedical terms, which include words and biomedical concepts in a standardized vocabulary. We find-tune the pre-trained static embeddings on entity relevance classification tasks in a weakly supervised manner. Experiments demonstrated the proposed model's efficacy on several quantitative evaluation benchmarks. With this, we can describe a term by any of its nearest words and concept codes in the vector space. Chapter 6 shows a use-case of the BERT-CRel embeddings in a document reranking task whereby a neural text summarizer generates a document summary in terms of the MeSH codes.

Our other approach to the vocabulary mismatch problem is the biomedical named entity recognition and entity normalization. This improved method allows us to understand deeper contextual meanings and the relationships among the terms of interest. Chapter 4 presents a joint learning methodology for NER and EN, which have been addressed sequentially and independently in the traditional pipeline models for information extraction. In this study, we examined the efficacy of a joint neural model for NER and EN on entity annotation performance. We explored different chunking schemes (e.g., IOB-tagging) to learn how the NN model learns the chunks' boundary information. We also studied the effects of data augmentation with counterfactual examples in training a NN model. We probed how the model learns linguistic features (e.g., the relationships among the entities' semantic types) in different downstream tasks.

Particularly, we are interested in the document retrieval tasks for answering biomedical inquiries. The question can be a complete interrogative sentence, such as "How to ease sciatic nerve pain?" Our job is to find the most relevant scientific document that is likely to contain a specific answer to the given question. In Chapter 5, we aim to devise a neural network-based model which can provide a deeper understanding and consequently reduce the semantic gaps between the question and a candidate answer sentence. We showed improved retrieval performance by incrementally applying our new document scoring features to a traditional IR model.

Another major need in biomedical IR is in the field of precision medicine that considers individual variability in genes, environment, and lifestyle for each patient to tailor treatment regimens. The U.S. NIH's precision medicine (PM) initiative and associated research endeavors warrant an informatics component — designing an IR system with the specific objectives of PM discussed earlier. Chapter 6 proposes a novel document reranking method that combines neural networks for a text summarization model and a document-query matching model in the frame of DR for PM scenario. The aim is to pivot the focus on query manipulation by previous conventional methods to transforming documents into pseudo-queries by employing the neural text summarization models. We utilize the transformer-based seq2seq model toward this. This proposed method begins an open discussion on the possibilities of using neural text summarization models for DR.

7.2 Study Limitations and Future Work Directions

Seeking and finding answers to questions in a fundamental aspect of human intelligence and hence a major milestone for artificial intelligence. In this current deep learning era, researchers have shown the capabilities and potentials of using neural networks (e.g., neural language models) for NLP tasks including QA. However, not all researchers agree that a neural model trained only with observable examples, without having a fundamental set of rules for inference, represents human intelligence. Plato's problem, given by Noam Chomsky, "the problem of explaining how we can know so much given our limited experience" is still an open question.

For example, the current neural language models cannot easily infer a simple logical rule such as the transitive property. Given two example sentences, "a ball is in a bag" and "a bag is in a car", without any specific example of "a ball is in a car", it is hard to infer that the car is larger than the ball in size. "I need more data" should not be the solution to this issue. Many of the current research efforts are attempting to address this fundamental problem under the names of commonsense reasoning, representation learning, explainability, interpretability, and inference with knowledge graphs. Our future research work will involve the following themes.

7.2.1 Feature representations for document indexing and retrieval

A wide-ranging body of prior work in representation learning demonstrated the empirical usefulness of various representations (for words, entities, sentences, topics, documents, or even subgraphs of a knowledge structure). However, the representativeness

of document embeddings (or anything related to the document-level encodings) is not fully matured for the information retrieval tasks.

This study will extend our previous work on document processing. The first goal of this study is to explore different methods of building distributed representations for documents, eventually in pursuit of the use in document indexing and retrieval. Most of the current techniques rely on the lexical representations for documents; for example, we build a representation for document abstract using contextualized embeddings (or the sum/average of them) in terms of the vocabulary given. A possible approach for improvement is to construct a document from multiple perspectives and build representations for each constituent view (e.g., entity relationships, metadata, author relationships, and publication information).

7.2.2 Vector similarity search problem

A consequent computational challenge is how to find the most similar document embeddings in the information retrieval systems. Modern machine learning models transform inputs (i.e., queries and documents) into high dimensional vectors. The problem is the size of the search space, which is often too large for exhaustive search; the number of entities for entity linking task, the number of documents for document retrieval task, or the number of words for text generation task are often too large to run exhaustive similarity searches. Studies [Guo et al., 2016, Malkov and Yashunin, 2020, Guo et al., 2020] specifically focusing on this problem are gaining more attention. The similarity functions used in this dissertation are mostly based on inner-product (also known as maximum inner-product search, MIPS). We believe, further investigation of the nearest neighbor search methods is inevitable.

Chapter A Appendix A. Unsupervised Ranking Models

A.1 Query Likelihood (QL) Model

The QL model ranks documents by $P(d|q)$, the probability of a document that it is relevant to the query. From the Bayes' rule,

$$P(d|q) = P(q|d)P(d)/P(q),$$

$P(q)$ is the same for all documents, thus can be ignored. $P(d)$ is uniform across all the documents, which can also be ignore. $P(d)$ is a document score that is independent from the given query. It is possible to use metadata of a document and use it as a prior probability.

With all the simplifications, $P(d|q) \propto P(q|d)$. Essentially, the QL model is a query generation model given a respective document.

We construct a language model (θ_d) from each document (d).

Using the multinomial unigram language model where the documents are classes, we can defined $P(q|d)$ as below:

$$P(q|d) \stackrel{\text{def}}{=} P(q|\theta_d) = \prod_{t \in q} P(t|\theta_d)$$

The likelihood function in the log space can be computed as follows:

$$\log P(q|\theta_d) = \sum_{t \in q} \log P(t|\theta_d)$$

MLE is an optimization problem to maximize the likelihood of seeing a document d given an observed sample d and the estimated language model θ_d:

$$argmax_{\theta_d}(\log P(d|\theta_d)) = argmax_{\theta_d}(\sum_{t \in V} c(t,d) \cdot \log P(t|\theta_d))$$

To use the method of Lagrange multipliers, we have a constraint function,

$$\sum_{t \in V} P(t|\theta_d) = 1$$

This yields a Lagrange functions as below:

$$L = \sum_{t \in V} c(t,d) \cdot \log P(t|\theta_d) - \lambda(\sum_{t \in V} P(t|\theta_d) - 1)$$

The partial derivatives with respect to the variables are,

$$\frac{\partial L}{\partial P(t|\theta_d)} = \frac{c(t,d)}{P(t|\theta_d)} - \lambda = 0$$

$$\frac{\partial L}{\partial \lambda} = \sum_{t \in V} P(t|\theta_d) - 1 = 0$$

$$\sum_{t \in V} c(t,d) = \lambda \sum_{t \in V} P(t|\theta_d) = \lambda$$

Therefore, the likelihood function can be defined as

$$P(t|\theta_d) = \frac{c(t,d)}{\sum_{t \in V} c(t,d)}$$

In a nutshell,

1. MLE methods are counting the frequency in a scope and divided by the total size
2. The Query Likelihood probability is the product of all the term probability in the given language model, one zero occurrence query term results in zero probability. To solve this issue, smoothing techniques (e.g., discounting methods, interpolation methods, the Jelinek-Mercer smoothing, the Dirichlet smoothing) are necessary
3. QL is just an estimate of the relevance between a query and a document. Despite the few "weaknesses", QL still works well, comparable to BM25.

A.2 Ranking documents by relevance-based Language models

Following demonstrates how relevance-based models can be used to optimize retrieval performance. The proof was demonstrated earlier by S. Robertson [Robertson, 1977] and V. Lavrenko [Lavrenko and Croft, 2017].

First, let's denote several events related to the actions of retrieving documents and measuring the relevance of documents.

- R: document is relevant
- $\overline{R}$: document is irrelevant
- D: retrieved document

Hence, $P(R|D)$ is the probability of a retrieved document being relevant.

The application of Bayes's theorem give the following:

$$P(R|D)P(D) = P(D|R)P(R)$$

Similarly,

$$P(\overline{R}|D)P(D) = P(D|\overline{R})P(\overline{R})$$

Hence,

$$\frac{P(R|D)}{P(\overline{R}|D)} = \frac{P(D|R)P(R)}{P(D|\overline{R})P(\overline{R})}$$

We apply log on both sides:

$$\log \frac{P(R|D)}{P(\overline{R}|D)} = \log \frac{P(D|R)}{P(D|\overline{R}))} + \log \frac{P(R)}{P(\overline{R})}$$

LHS can be simplified using the logistic transformation.

$$\log \frac{P(R|D)}{P(\overline{R}|D)} = \log \frac{P(R|D)}{1 - P(R|D)} = \operatorname{logit} P(R|D)$$

Thus,

$$\operatorname{logit} P(R|D) = \log \frac{P(D|R)}{P(D|\overline{R}))} + \log \frac{P(R)}{P(\overline{R})}$$

Let's parameterize the probabilities, such that

- $\theta_R = P(D|R)$, which corresponds to the *recall* metric
- $\theta_{\overline{R}} = P(D|\overline{R})$, which corresponds to the *fallout* metric
- $\phi = P(R|D)$, which corresponds to the *precision* metric
- $\gamma = P(R)$

From the previous equation, we have:

$$\operatorname{logit} \phi = \log \frac{\theta_R}{\theta_{\overline{R}}} + \operatorname{logit} \gamma$$

The optimal retrieval performance can be achieved if the documents are ranked by ϕ, which is rank-equivalent to $\theta_R/\theta_{\overline{R}}$ given that γ is constant per document in an individual retrieval request.

The essence of the retrieval method using the relevance-based language models is that we can rank documents by computing the log-odds from user provided relevance feedback. For example, a document (d_i) score can be computed using a relevance language model as such:

$$\log \frac{P(D|R)}{P(D|\overline{R})} \sim \sum_{w \in d_i} \log \frac{P(w|R)}{P(w|\overline{R})},$$

where $w \in d_i$ represents the words in a document.

A.3 Okapi BM25

Okapi BM25 is a bag-of-words document ranking function based on the query terms appearing in each documents, regardless of the inter-relationship between the query terms within a document. One of the most prominent instantiations of the function is as follows.

$$w_i = \frac{(k_1+1)\cdot tf}{k_1\left((1-b)+b\cdot\frac{dl}{avgdl}\right)+tf}\cdot\log\frac{N-n(q_i)+0.5}{n(q_i)+0.5},$$

where

- the first component is TF part and the latter is IDF part
- k_1 determines the upper bound of the TF component in BM25, free parameter $k_1 \in [1.2, 2.0]$
- b is another free parameter, usually set to 0.75
- dl is the document length
- $avgdl$ is the average document length in the collection
- N is the total number of documents
- $n(q_i)$ is the number of documents that contain the query term q_i

A.4 Sequential Dependence Model (SDM)

Unlike the traditional bag-of-words models, the order of terms in a query is also taken into account in the SDM model. SDM is based on the Markov random field model, in which not only the unigrams but also the ordered and unordered bi-grams in a posed query are considered in the retrieval score computation. The term frequency score is

$$f_T(q_i, D) = \log P(q_i|\theta_D) = \log\frac{tf_{q_i,D}+\mu\frac{cf_{q_i}}{|C|}}{|D|+\mu}$$

where q_i is a query term, D is the document, θ_D is a language model built using D, $tf_{q_i,D}$ is the term frequency of q_i in D, cf_{q_i} is the collection frequency of q_i, $|C|$ is the total number of terms across all the documents, $|D|$ is the document length, and μ is the Dirichlet prior for the smoothing effect. Likewise, the functions for the ordered and unordered bi-grams are defined in a similar way:

$$f_O(q_i, q_{i+1}, D) = \log\frac{tf^N_{o(q_i,q_{i+1},D)}+\mu\frac{cf^N_{o(q_i,q_{i+1},D)}}{|C|}}{|D|+\mu}$$

$$f_U(q_i, q_{i+1}, D) = \log\frac{tf^M_{u(q_i,q_{i+1},D)}+\mu\frac{cf^M_{u(q_i,q_{i+1},D)}}{|C|}}{|D|+\mu}$$

where $tf^N_{o(q_i,q_{i+1},D)}$ and $tf^M_{u(q_i,q_{i+1},D)}$ indicate the frequencies of the terms q_i and q_{i+1} within an ordered window of N word positions and within a unordered window of M word positions respectively. The final scoring function is the weighted sum of the three constituent functions.

$$score(Q,D) = \lambda_T \sum_{i=1}^{|Q|} f_T(q_i, D) + \lambda_O \sum_{i=1}^{|Q|-1} f_O(q_i, q_{i+1}, D) + \lambda_U \sum_{i=1}^{|Q|-1} f_U(q_i, q_{i+1}, D)$$

where $Q = q_1, \ldots, q_{|Q|}$ is the query and λ_T, λ_O, and λ_U are weights for the unigram, ordered bigram, and unordered bigram components respectively. This SDM scoring function is the baseline throughout all our experiments where we measure the effectiveness of our matching score feature and other auxiliary features.

Acronyms

ACL Association for Computational Linguistics. 16

ARS Adaptive Random Search. 75, 77

BioNLP Biomedical Natural Language Processing. 17, 24, 27, 29, 42

CNN Convolutional Neural Networks. 14

CRF Conditional Random Field. 52

DR Document Retrieval. 66, 67, 101

EHR Electronic Health Record. 37

EN Entity Normalization. 8, 45

ET Entity Typing. 44

FN False Negative. 57

FP False Positive. 57

GLM Generalized Language Model. 14, 22

IE Information Extraction. iii, 9

IOB Inside, Outside, Beginning Tagging Scheme. 48, 49, 59, 101

IR Information Retrieval. ii, 1–3, 7, 8, 11, 17, 23, 67, 78, 84, 98, 100

LM Language Model. 6

LSTM Long Short-Term Memory. 76

MAP Mean Average Precision. 68, 78, 82

MD Mention Detection. 44, 47, 57

MeSH Medical Subject Headings. 18, 28, 31, 35, 41, 42, 78, 79, 93

MLP MultiLayer Perceptron. 56

NCBI U.S. National Center for Biotechnology Information. 19, 92

NER Named Entity Recognition. 8, 25, 44–46

NIH The U.S. National Institutes of Health. 67, 83, 92

NIST The U.S. National Institute for Standards and Technology. 83

NLM The U.S. National Library of Medicine. 28–31

NLP Natural Language Processing. iii, 24, 45, 66, 100

NLU Natural Language Understanding. 8

NN Neural Networks. 5, 6

OOV Out Of Vocabulary. 13, 14, 36, 39

PM Precision Medicine. 10, 85, 86

PRF Pseudo-Relevance Feedback. 22

QA Question-Answering. iii, 8, 43, 66, 78

Bibliography

Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4): 305–338, 2016.

Barbara M Wildemuth and Margaret E Moore. End-user search behaviors and their relationship to search effectiveness. *Bulletin of the Medical Library Association*, 83 (3):294, 1995.

George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, 1987.

Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, volume 16, pages 2786–2792, 2016.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *AAAI*, pages 2741–2749, 2016.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, 5(1):135–146, 2017a.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. doi: 10.18653/v1/N19-1423.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

Lance De Vine, Guido Zuccon, Bevan Koopman, Laurianne Sitbon, and Peter Bruza. Medical semantic similarity with a neural language model. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1819–1822. ACM, 2014.

Youngduck Choi, Chill Yi-I Chiu, and David Sontag. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41, 2016a.

Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1495–1504. ACM, 2016b.

Xiangrui Cai, Jinyang Gao, Kee Yuan Ngiam, Beng Chin Ooi, Ying Zhang, and Xiaojie Yuan. Medical concept embedding with time-aware attention. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3984–3990, 2018.

Wen-tau Yih, Geoffrey Zweig, and John C Platt. Polarity inducing latent semantic analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1212–1222. Association for Computational Linguistics, 2012.

Mo Yu and Mark Dredze. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 545–550, 2014.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1501–1511, 2015.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, 2015.

Nikola Mrkšic, Diarmuid OSéaghdha, Blaise Thomson, Milica Gašic, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. In *Proceedings of NAACL-HLT*, pages 142–148, 2016.

Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, 2016.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, 2015.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414, 2001.

Minh-Thang Luong, Richard Socher, and Christopher D Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the seventeenth conference on computational natural language learning*, pages 104–113, 2013.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of artificial intelligence research*, 49:1–47, 2014.

Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. Simverb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, 2016.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26, 2017.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43 (4):781–835, 2017.

Saif Mohammad, Bonnie Dorr, and Graeme Hirst. Computing word-pair antonymy. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 982–991, 2008.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013b.

Hang Li and Jun Xu. Semantic matching in search. *Foundations and Trends in Information retrieval*, 7(5):343–469, 2014.

Jiho Noh and Ramakanth Kavuluru. Team uknlp at trec 2017 precision medicine track: a knowledgebased ir system with tuned query-time boosting. *TREC, Gaithersburg, MD*, 2017.

Zhiyong Lu, Won Kim, and W John Wilbur. Evaluation of query expansion using mesh in pubmed. *Information retrieval*, 12(1):69–80, 2009.

Travis Goodwin and Sanda M Harabagiu. Utd at trec 2014: Query expansion for clinical decision support. Technical report, TEXAS UNIV AT DALLAS RICHARDSON, 2014.

Mohannad Almasri, Catherine Berrut, and Jean-Pierre Chevallet. A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information. In *European conference on information retrieval*, pages 709–715. Springer, 2016.

Joseph Rocchio. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323, 1971.

Victor Lavrenko and W Bruce Croft. Relevance-based language models. In *ACM SIGIR Forum*, volume 51, pages 260–267. ACM, 2017.

Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608*, 2016.

Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 367–377, 2016.

Rodrigo Nogueira and Kyunghyun Cho. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 574–583, 2017. doi: 10.18653/v1/d17-1061.

Youngho Kim, Jangwon Seo, and W Bruce Croft. Automatic boolean query suggestion for professional search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 825–834. ACM, 2011.

Harrisen Scells, Leif Azzopardi, Guido Zuccon, and Bevan Koopman. Query variation performance prediction for systematic reviews. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1089–1092, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Artem Shelmanov, Vadim Liventsev, Danil Kireev, Nikita Khromov, Alexander Panchenko, Irina Fedulova, and Dmitry V Dylov. Active learning with deep pre-trained models for sequence tagging of clinical and biomedical texts. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 482–489. IEEE Computer Society, 2019.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, pages 1–5. IEEE, 2019.

Chris Alberti, Kenton Lee, and Michael Collins. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*, 2019.

Jiho Noh and Ramakanth Kavuluru. Improved biomedical word embeddings in the transformer era. *ArXiv*, abs/2012.11808, 2020a.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017b. doi: 10.1162/tacl_a_00051.

Jinxing Yu, Xun Jian, Hao Xin, and Yangqiu Song. Joint embeddings of chinese words, characters, and fine-grained subcharacter components. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 286–291, 2017a.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, 2018.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2021.

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

Ramakanth Kavuluru and Daniel Harris. A knowledge-based approach to syntactic disambiguation of biomedical noun compounds. In *Proceedings of COLING 2012: Posters*, pages 559–568, 2012.

Julian Eisenschlos, Sebastian Ruder, Piotr Czapla, Marcin Kadras, Sylvain Gugger, and Jeremy Howard. MultiFiT: Efficient multi-lingual language model fine-tuning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5706–5711, 2019.

AKM Sabbir, Antonio Jimeno-Yepes, and Ramakanth Kavuluru. Knowledge-based biomedical word sense disambiguation with neural concept embeddings. In *2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 163–170. IEEE, 2017.

Chih-Hsuan Wei, Alexis Allot, Robert Leaman, and Zhiyong Lu. PubTator central: automated concept annotation for biomedical full text articles. *Nucleic acids research*, 47(W1):W587–W593, 2019.

Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.

Yanshan Wang, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, and Hongfang Liu. A comparison of word embeddings for the biomedical natural language processing. *Journal of biomedical informatics*, 87:12–20, 2018.

Ariel S Schwartz and Marti A Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Biocomputing*, pages 451–462. World Scientific, 200.

Hiroko Ao and Toshihisa Takagi. ALICE: an algorithm to extract abbreviations from medline. *Journal of the American Medical Informatics Association*, 12(5):576–586, 2005.

Serguei Pakhomov, Bridget McInnes, Terrence Adam, Ying Liu, Ted Pedersen, and Genevieve B Melton. Semantic similarity and relatedness between clinical terms: an experimental study. In *AMIA annual symposium proceedings*, volume 2010, page 572, 2010.

Serguei Pakhomov. Semantic relatedness and similarity reference standards for medical terms. `https://bit.ly/2WxjiR8`, 2018.

Ted Pedersen, Serguei VS Pakhomov, Siddharth Patwardhan, and Christopher G Chute. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of biomedical informatics*, 40(3):288–299, 2007.

Angelos Hliaoutakis. Semantic similarity measures in mesh ontology and their application to information retrieval on medline. *Master's thesis*, 2005.

Junseok Park, Kwangmin Kim, Woochang Hwang, and Doheon Lee. Concept embedding to measure semantic relatedness for biomedical information ontologies. *Journal of biomedical informatics*, 94:103182, 2019.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th workshop on biomedical natural language processing*, pages 166–174, 2016.

Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. BioWordVec, improving biomedical word embeddings with subword information and mesh. *Scientific data*, 6(1):1–9, 2019.

Zhiguo Yu, Trevor Cohen, Elmer V Bernstam, and Byron C Wallace. Retrofitting word vectors of MeSH terms to improve semantic similarity measures. *EMNLP 2016*, page 43, 2016.

Zhiguo Yu, Byron C Wallace, Todd Johnson, and Trevor Cohen. Retrofitting concept vector representations of medical concepts to improve estimates of semantic similarity and relatedness. *Studies in health technology and informatics*, 245:657, 2017b.

S Henry, A McQuilkin, and BT McInnes. Association measures for estimating semantic similarity and relatedness between biomedical concepts. *Artificial intelligence in medicine*, 93:1, 2019.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759, 2009. doi: 10.1145/1571941.1572114.

Jiho Noh and Ramakanth Kavuluru. Literature retrieval for precision medicine with neural matching and faceted summarization. In *Findings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3389–3399, 2020b. doi: 10.18653/v1/2020.findings-emnlp.304. URL `https://www.aclweb.org/anthology/2020.findings-emnlp.304`.

Sunil Mohan and Donghui Li. Medmentions: a large biomedical corpus annotated with umls concepts. *arXiv preprint arXiv:1902.09476*, 2019.

Michael Collins. Ranking algorithms for named entity extraction: Boosting and the votedperceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 489–496, 2002.

Eckhard Bick. A named entity recognizer for danish. In *LREC*. Citeseer, 2004.

David D McDonald. Internal and external evidence in the identification and semantic categorization of proper names. In *Acquisition of Lexical Knowledge from Text*, 1993.

Lisa F Rau. Extracting company names from text. In *Proceedings the Seventh IEEE Conference on Artificial Intelligence Application*, pages 29–30. IEEE Computer Society, 1991.

R Leaman and Z Lu. Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinformatics (Oxford, England)*, 32(18):2839–2846, 2016.

Chih-Hsuan Wei and Hung-Yu Kao. Cross-species gene normalization by species inference. *BMC bioinformatics*, 12(S8):S5, 2011.

Jörg Hakenberg, Martin Gerner, Maximilian Haeussler, Illés Solt, Conrad Plake, Michael Schroeder, Graciela Gonzalez, Goran Nenadic, and Casey M Bergman. The gnat library for local and remote gene mention normalization. *Bioinformatics*, 27(19):2769–2771, 2011.

Xi Wang, Jiagao Lyu, Li Dong, and Ke Xu. Multitask learning for biomedical named entity recognition with cross-sharing structure. *BMC bioinformatics*, 20(1):427, 2019.

Daniel Loureiro and Alípio Mário Jorge. Medlinker: Medical entity linking with neural representations and dictionary matching. In *European Conference on Information Retrieval*, pages 230–237. Springer, 2020.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, 2018.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. In *20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016*, pages 250–259. Association for Computational Linguistics (ACL), 2016.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Xu, Huan Gui, Jian Peng, and Jiawei Han. Empower sequence labeling with task-aware neural language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*, 2019.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, 2017.

Rui Zhang, Cicero dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of the 56th Annual Meeting of the*

Association for Computational Linguistics (Volume 2: Short Papers), pages 102–107, 2018.

Naoaki Okazaki and Jun'ichi Tsujii. Simple and efficient algorithm for approximate dictionary matching. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 851–859, 2010.

Maciej Wiatrak and Juha Iso-Sipila. Simple hierarchical multi-task neural end-to-end entity linking for biomedical text. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 12–17, 2020.

Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. Counterfactual generator: A weakly-supervised method for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7270–7280, 2020.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611, 2019.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

T Lin, P Goyal, R Girshick, K He, and P Dollar. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33, 2020.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Luca Soldaini and Nazli Goharian. Quickumls: a fast, unsupervised approach for medical concept extraction. In *MedIR workshop, SIGIR*, 2016.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. Scispacy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Erik Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Jiho Noh and Ramakanth Kavuluru. Document retrieval for biomedical question answering with neural sentence matching. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 194–201. IEEE, 2018. doi: 10.1109/icmla.2018.00036.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138, 2015.

W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.

Donald Metzler and W Bruce Croft. A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479. ACM, 2005.

Wenpeng Yin and Hinrich Schütze. Discriminative phrase embedding for paraphrase identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1368–1373, 2015.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1870–1879, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.

Halil Kilicoglu, Dongwook Shin, Marcelo Fiszman, Graciela Rosemblat, and Thomas C Rindflesch. Semmeddb: A pubmed-scale repository of biomedical semantic predications. *Bioinformatics (Oxford, England)*, 28(23):3158–3160, 2012.

Dean C Karnopp. Random search techniques for optimization problems. *Automatica*, 1(2-3):111–121, 1963.

Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. *NIPS 2017 Workshop on Autodiff*, 2017.

Phoebe M Roberts, Aaron M Cohen, and William R Hersh. Tasks, topics and relevance judging for the TREC genomics track: five years of experience evaluating biomedical text information retrieval systems. *Information Retrieval*, 12(1):81–97, 2009.

Kirk Roberts, Matthew Simpson, Dina Demner-Fushman, Ellen Voorhees, and William Hersh. State-of-the-art in biomedical literature retrieval for clinical cases: a survey of the TREC 2014 CDS track. *Information Retrieval Journal*, 19(1-2): 113–148, 2016.

Kirk Roberts, Dina Demner-Fushman, Ellen M Voorhees, William R Hersh, Steven Bedrick, Alexander J Lazar, and Shubham Pant. Overview of the TREC 2017 precision medicine track. In *Proceedings of TREC Conference*, pages 1–13, 2017.

Guido Zuccon, Joao Palotti, Lorraine Goeuriot, Liadh Kelly, Mihai Lupu, Pavel Pecina, Henning Mueller, Julie Budaher, and Anthony Deacon. The IR task at the CLEF eHealth evaluation lab 2016: user-centered health information retrieval. In *CLEF 2016-Conference and Labs of the Evaluation Forum*, volume 1609, pages 15–27, 2016.

Kezban Dilek Onal, Ye Zhang, Ismail Sengor Altingovde, Md Mustafizur Rahman, Pinar Karagoz, Alex Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten McNamara, et al. Neural information retrieval: At the end of the early years. *Information Retrieval Journal*, 21(2-3):111–182, 2018.

Nam Khanh Tran and Claudia Niedereée. Multihop attention networks for question answer matching. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 325–334. ACM, 2018.

Diego Mollá. Macquarie university at BioASQ 5B–query-based summarisation techniques for selecting the ideal answers. *BioNLP 2017*, pages 67–75, 2017.

Georg Wiese, Dirk Weissenborn, and Mariana Neves. Neural question answering at BioASQ 5B. *BioNLP 2017*, pages 76–79, 2017.

Zan-Xia Jin, Bo-Wen Zhang, Fan Fang, Le-Le Zhang, and Xu-Cheng Yin. A multi-strategy query processing approach for biomedical question answering: USTB_PRIR at BioASQ 2017 Task 5B. *BioNLP*, pages 373–380, 2017.

Francis S Collins and Harold Varmus. A new initiative on precision medicine. *New England journal of medicine*, 372(9):793–795, 2015. doi: 10.1056/NEJMp1500523.

Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, William R. Hersh, Steven Bedrick, Alexander J. Lazar, Shubham Pant, and Funda Meric-Bernstam. Overview of the TREC 2019 precision medicine track, 2020. URL `https://trec.nist.gov/pubs/trec28/papers/OVERVIEW.PM.pdf`.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. In *NAACL-HLT*, pages 1747–1759, 2018. doi: 10.18653/v1/n18-1158.

Trey Grainger and Timothy Potter. *Solr in action*. Manning Publications Co., 2014.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067, 2019.

Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3721–3731, 2019. doi: 10.18653/v1/d19-1387.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, 2016. doi: 10.18653/v1/p16-1046.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *International Conference on Learning Representations*, 2018.

Zhiqing Sun, Jian Tang, Pan Du, Zhi-Hong Deng, and Jian-Yun Nie. Divgraph-pointer: A graph pointer network for extracting diverse keyphrases. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 755–764, 2019. doi: 10.1145/3331184.3331219.

Animesh Prasad and Min-Yen Kan. Glocal: Incorporating global information in local convolution for keyphrase extraction. In *NAACL-HLT*, pages 1837–1846, 2019. doi: 10.18653/v1/N19-1182.

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *EMNLP*, pages 379–389, 2015. doi: 10.18653/v1/d15-1044.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, 2016. doi: 10.18653/v1/k16-1028.

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083, 2017. doi: 10.18653/v1/p17-1099.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. doi: 10.1109/cvpr.2016.90.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W04-1013`.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE, 2012.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, 2017. doi: 10.18653/v1/p17-4012.

Erik Faessler, Michel Oleynik, and Udo Hahn. JULIE lab & Med Uni Graz @ TREC 2019 precision medicine track, 2020. URL `https://trec.nist.gov/pubs/trec28/papers/julie-mug.PM.pdf`.

Deborah Caucheteur, Emilie Pasche, Julien Gobeill, Anais Mottaz, Luc Mottin, and Patrick Ruch. Designing retrieval models to contrast precision-driven ad hoc search vs. recall-driven treatment extraction in precision medicine, 2020. URL `https://trec.nist.gov/pubs/trec28/papers/BITEM_PM.PM.pdf`.

K Sparck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information processing & management*, 36(6):809–840, 2000.

Anthony Rios and Ramakanth Kavuluru. Neural transfer learning for assigning diagnosis codes to emrs. *Artificial Intelligence in Medicine*, 96:116–122, 2019. doi: 10.1016/j.artmed.2019.04.002.

Anthony Rios, Ramakanth Kavuluru, and Zhiyong Lu. Generalizing biomedical relation classification with neural adversarial domain adaptation. *Bioinformatics*, 34 (17):2973–2981, 2018. doi: 10.1093/bioinformatics/bty190.

Alex H Wagner, Brian Walsh, Georgia Mayfield, David Tamborero, Dmitriy Sonkin, Kilannin Krysiak, Jordi Deu-Pons, Ryan P Duren, Jianjiong Gao, Julie McMurry, et al. A harmonized meta-knowledgebase of clinical interpretations of somatic genomic variants in cancer. *Nature genetics*, 52(4):448–457, 2020. doi: 10.1101/366856.

Gia-Hung Nguyen, Laure Soulier, Lynda Tamine, and Nathalie Bricon-Souf. Dsrim: A deep neural information retrieval model enhanced by a knowledge resource driven representation of documents. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 19–26, 2017. doi: 10.1145/3209978.3210081.

J. Guo, Y. Fan, Qingyao Ai, and W. Croft. A deep relevance matching model for ad-hoc retrieval. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016.

Yury A. Malkov and D. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:824–836, 2020.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, 2020.

Stephen E Robertson. The probability ranking principle in ir. *Journal of documentation*, 1977.

Vita

Jiho Noh

Education

2011–2013, *B.S. in Computer Science*, Indiana University, Bloomington, USA

Experience

2020–Present, *Graduate Research Assistant*, University of Kentucky
2019–2020, *Lecturer (Discrete Mathematics)*, University of Kentucky, Lexington, USA
2018–2019, *Teaching Assistant (Discrete Mathematics)*, University of Kentucky, Lexington, USA
2017–2018, *Teaching Assistant (Introduction to Program Design, Abstraction, and Problem Solving)*, University of Kentucky, Lexington, USA
2015–2017, *Graduate Research Assistant*, Endangered Language Alliance (ELA) and University of Kentucky

Awards

2020 — Graduate School Congress (GSC) Travel Awards, University of Kentucky
2020 — Runner-up, 2020 Research Presentation Competition, Commonwealth Computational Summit, University of Kentucky
2018 — Department Conference Travel Award, Department of Computer Science, University of Kentucky
2018 — Winner, 2018 BioASQ Challenge (Large-scale biomedical semantic indexing and question answering)
2017 — Top 5 Retrieval System, 2017 TREC (Precision Medicine Track Competition)
2011–2012 — Dean's List and Founders Scholar, Indiana University at Bloomington
2011–2012 — Lindley Scholarship, Indiana University at Bloomington

Publications

1. Noh J and Kavuluru R. Improved Biomedical Word Embeddings in the Transformer Era. *arXiv preprint* arXiv:2012.11808. 2020 Dec 22. (Journal of Biomedical Informatics in review)

2. Noh J and Kavuluru R. Literature Retrieval for Precision Medicine with Neural Matching and Faceted Summarization. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 2020 Nov.

3. Kavuluru R, Noh J, Rose SW. Twitter Discourse on Nicotine as Potential Prophylactic or Therapeutic for COVID-19. *medRxiv*. 2021 Jan 1.

4. Noh J and Kavuluru R. Team UKNLP at TREC 2017 Precision Medicine Track: A Know ledge-Based IR System with Tuned Query-Time Boosting. *In Proceedings of Text REtrieval Conference* 2017 Dec.

5. Noh J and Kavuluru R. Document Retrieval for Biomedical Question Answering with Neural Sentence Matching. *In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* 2018 Dec 17 (pp. 194–201). IEEE.