© 2021 Tarek Sakakini

KNOWLEDGE BASE INTEGRATION IN BIOMEDICAL NATURAL LANGUAGE PROCESSING APPLICATIONS

BY

TAREK SAKAKINI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Assistant Professor Suma Bhat, Chair
Professor Pramod Viswanath
Professor Daniel G. Morrow
Professor Mark Hasegawa-Johnson

ABSTRACT

With the progress of natural language processing in the biomedical field, lack of annotated data due to regulations and expensive labor remains an issue. In this work, we study the potential of knowledge bases for biomedical language processing to compensate for the shortage of annotated data. Accordingly, we experiment with integration of a rigorous biomedical knowledge base, the Unified Medical Language System, in three different biomedical natural language processing applications: text simplification, conversational agents for medication adherence, and automatic evaluation of medical students' chart notes.

In the first task, we take as a use case simplifying medication instructions to enhance medication adherence among patients. Given the lack of an appropriate parallel corpus, the Unified Medical Language System provided simpler synonyms for an unsupervised system we devise, and we show positive impact on comprehension through a human subjects study.

As for the second task, we devise an unsupervised system to automatically evaluate chart notes written by medical students. The purpose of the system is to speed up the feedback process and enhance the educational experience. With the lack of training corpora, utilizing the Unified Medical Language System proved to enhance the accuracy of evaluation after integration into the baseline system.

For the final task, the Unified Medical Language System was used to augment the training data of a conversational agent that educates patients on their medications. As part of the educational procedure, the agent needed to assess the comprehension of the patients by evaluating their answers to predefined questions. Starting with a small seed set of paraphrases of acceptable answers, the Unified Medical Language System was used to artificially augment the original small seed set via synonymy. Results did not show increase in quality of system output after knowledge base integration due to

the majority of errors resulting from mishandling of counts and negations.

We later demonstrate the importance of a (lacking) entity linking system to perform optimal integration of biomedical knowledge bases, and we offer a first stride towards solving that problem, along with conclusions on proper training setup and processes for automatic collection of an annotated dataset for biomedical word sense disambiguation.

To my parents, and brothers, for their love and guidance.

ACKNOWLEDGMENTS

I thank my dad, Jamal Sakakini, for instilling in me the importance of education and science, and my mom, Amal El Sayyed, for her continuous nourishment of all those around her. I thank my brother, Mohamad Sakakini, for his guidance and mentorship, and my brother, Abdulrahman Sakakini, for his embracing love.

I am grateful for having Prof. Suma Bhat as an advisor on this dissertation journey. Her contagious excitement for NLP led us on many NLP endeavors. I would also like to thank my collaborators, Prof. Pramod Viswanath, Prof. Dan Morrow, Prof. Mark Hasegawa-Johnson, Prof. Jinjun Xiong, and Prof. Wen-Mei Hwu, for the continuously stimulating research discussions and ideas. To my labmates, Jiaqi Mu and Hongyu Gong, thank you for offering the continuous laughs and deep insights.

A big thanks goes to Ihab Nahlus and Peter Kairouz. Besides their precious friendship, Ihab set asail my career in research, and Peter steered me into the exciting field of machine learning.

A special thank you for my friends: Izzat El-Hajj, Adel Ejjeh, Hussein Hazimeh, Hussein Sibai, George Abdallah, Pio Ibrahim, Marie-Joe Noon, Nahla Kreidly, Nabil Ramlawi, Hasan Dbouk, Rabel Rizkallah, Paul Gharzouzi, James Schmidt, Philip Pare, and all my friends in Champaign-Urbana. It was an honor to have shared the campus and all the moments with each and every person.

Finally, I am ever grateful to have met and shared all the great moments with Patrick Birbarah, Pamela Tannous, Ali Kourani, Clara Salame, and Saadeddine Shehab. To love life is the greatest skill, and they were the best teachers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The Health Information Technology for Economic and Clinical Health Act (HITECH) was passed in 2009, creating incentives for the healthcare industry to adopt electronic health records (EHR) [1]. Figure 1.1 demonstrates its significant impact on the adoption of EHRs. As a consequence, the healthcare industry in the US, and worldwide, produces large volumes of digitized text on a daily basis. In these volumes lie knowledge sources untapped due to experts' time limitations, and machines' cognitive limitations. To process these texts for patients' benefit as well as the scientific community, researchers have worked towards equipping machines with cognitive capabilities through the use of natural language processing (NLP) tools [2]. Although NLP methods have rapidly evolved in the past decade, these methods were developed and evaluated mostly on non-biomedical data, relying on large amounts of labeled text. It is unclear to what extent these advances translate to the biomedical domain with significantly constrained labeled data. With the susceptibility of NLP tools to degrade with a shift in domain [3], and the shortage of labeled text in the biomedical domain, this dissertation aims to study the limitation introduced by the shortage in labeled text, and the possible methods to address the shortage.

Different biomedical NLP tasks require different volumes of training data. This requirement depends on the language variability of the domain, the level of semantic understanding and world knowledge needed, and the decision space in the task. Consequently, we answer our overarching question by posing it as a set of pragmatic questions over several downstream biomedical NLP tasks, while varying the difficulty level and data requirement. The first task we approach aims to simplify health text to enhance patients' comprehension of their own health data [4]. The second task streamlines the training and evaluation of student-prepared medical chart notes [5], while the last task offers a conversational agent to assist patients in their knowledge of their

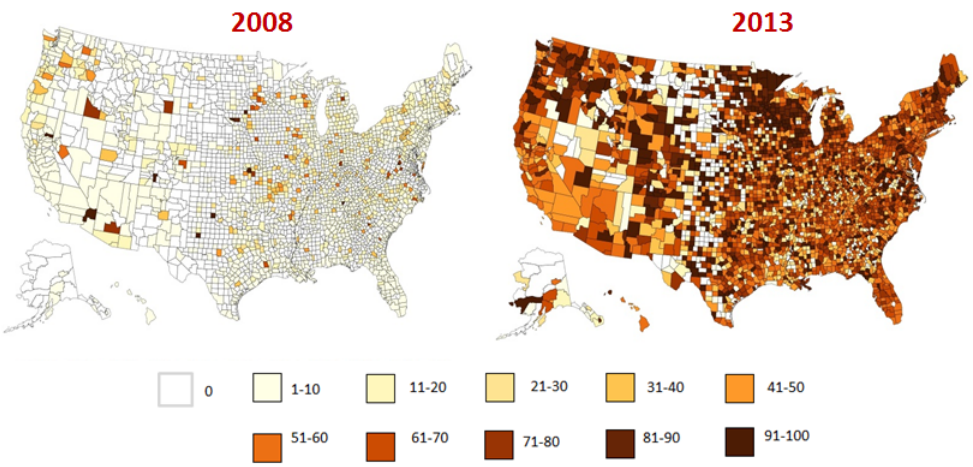

Figure 1.1: Percent of physicians e-prescribing through an electronic health record [7]

medication regimes, and consequently, their medication adherence [6]. The recurring theme in our approach across all the previously mentioned tasks is the infusion of the information in a knowledge base into our system to compensate for the lack of a training data.

Text Simplification of Medication Instructions Biomedical text can be complicated for the target audience to comprehend due to the mismatch in health literacy levels between the authors and the readers [8]. For example, a physician might note "Take 3 tablets PO BID" as a medication instruction, which is incomprehensible to a patient without the intervention of a pharmacist. This motivates automatic text simplification methods for biomedical text.

Text simplification is a well-researched task in NLP, with recent progress borrowing methods from works on neural machine translation [9]. With neural methods hungry for data, and the shortage of the required parallel text (complicated - simple) in the healthcare domain, we pose several questions. First, how directly portable are state-of-the-art neural methods for text simplification in the general domain when used for health text? For that, we construct a suitable, yet limited, parallel corpus of medication instructions and their simplifications, and train/evaluate the aforementioned neural methods. Next, we realize the potential of well-maintained medical ontologies, such as the Unified Medical Language System (UMLS) [10], and study a suggested unsupervised text simplification method that relies on the

UMLS compensating for the linguistic connections made through a parallel corpus.

Entity Linking for Evaluating Medical Chart Notes Automatic short answer grading (ASAG) in the healthcare domain poses challenges unfaced in other domains. The main challenge introduced by biomedical text is its ambiguity level [11]. Healthcare professionals and biomedical scientists deal with an ever-expanding host of terminologies. To simplify communication, abbreviations and qualifier-free rhetoric is heavily used, causing ambiguity in biomedical text and consequent challenges to an NLP system processing such text. For example, it is not straightforward to estimate whether "COLD" in a sentence refers to the feeling of cold, to the common cold sickness, or to chronic obstructive lung disease. A prerequisite to the grading of an answer is the semantic understanding of the answer and the corresponding rubric item. Entity linkers can identify the intended concept for an ambiguous phrase and provide significant semantic value. Although entity linkers are available for the biomedical domain [12], they have been developed for well-written scientific documents and are not readily available for the noisy language of medical chart notes. With the lack of training data for entity linking in medical chart notes, we study the capability of UMLS to assist in entity linking and act as a grounding source for variant phrases.

Conversational Agent for Medical Adherence Another biomedical field with great potential and lacking datasets is the conversational agent space in healthcare. Conversational agents identify the intent of the user, and reply with the appropriate response. Such a technology offers the required presence of a medical consultant despite the limited schedule of healthcare professionals. The underlying technology of conversational agents requires a preset of intents and their possible surface forms [13]. For example, the surface forms "What's my medication's dosage?" and "How much of the medication should I take every time?" can serve as examples for the user's intent to ask about the dosage. Linking back to the previous task, a semantic understanding of the utterance is required to understand the intent of the user. Given the variability of communicating with a health agent, and the variability of health text, we find a similar motivation to identify entities in utterances and automatically produce alternatives for the agent to train. Along the same lines, we continue to explore the utilization of a knowledge base such as the UMLS in automatically producing the dataset required to

enhance the intelligence of the conversational agent.

Ambiguity in Biomedical NLP and Word Sense Disambiguation

As mentioned, the recurring theme in our approach is the infusion of the information in a knowledge base into our system. An identified challenge to this infusion is in resolving the ambiguity of the biomedical phrases, and introducing the right set of information from the knowledge base. Hence, we finally take a step back and explore the potential of BioBERT [14] at advancing entity linking of ambiguous phrases, in biomedical text, to concepts in knowledge bases.

Word sense disambiguation (WSD) is an integral step in entity linking, and relies on the context of an ambiguous phrase to disambiguate it. With BioBERT being a contextualized word representation that has shown great benefits on other biomedical downstream tasks, we study the potential of BioBERT at advancing the state of biomedical WSD. With the ever-expanding terminology of biomedical text, we further explore the potential of automatic generation of WSD datasets, and analyze the cost of relying on noisy automatically generated data for biomedical WSD.

The high-level contributions of this dissertation are:

1. Studying the limitation of general NLP methods in the low-resource setting of healthcare text in three separate tasks.

2. Assessing the capability of knowledge bases such as the UMLS at compensating for the lack of labeled text.

3. Explore the potential of BioBERT at advancing the state of biomedical WSD, which allows for the utilization of knowledge bases in low-resource settings.

The rest of the dissertation is structured as follows. Chapter 2 summarizes previous works that have followed the same approach of utilizing knowledge bases to compensate for the lack of training data in low-resource settings. The next three chapters respectively cover the work performed in each of the aforementioned downstream tasks. Chapter 6 takes a step back and addresses the task of biomedical WSD. Finally, Chapter 7 presents other supporting

work that aids in addressing low-resource settings in general, and Chapter 8 concludes the dissertation.

CHAPTER 2

KNOWLEDGE BASE INTEGRATION IN GENERAL NLP

In the endeavor to create intelligent computer programs, machine learning algorithms have made long strides by learning patterns from annotated data. But to reach human intelligence, our reasoning relies on background knowledge that lies outside the knowledge particular to the task, and accordingly, knowledge outside the realm of the data annotated for the task. For example, when semantically parsing the sentence "The man observes the elephant with his telescope.", it is trivial for human intelligence to realize, even without seeing previous similar examples, that the man is operating the telescope, and not the elephant. But to the machine, if it has not seen a similar example in the annotated data, it is not trivial due to lacking general knowledge such as elephants lacking the means to own a telescope and the intelligence to operate it.

To fill this gap, and progress towards human-level intelligence, NLP researchers have developed and utilized knowledge bases to structurally represent and model background knowledge, or at least attempt it. This resulted in the generation of various types of knowledge bases which can be divided into two categories: (1) databases modeling lexico-semantic aspects of language, and (2) databases of entities and the relations between them.

More particular to our setting, knowledge in the biomedical domain was long motivated to be documented and structured due to the ever-expanding knowledge in this field, and due to the need of medical students to ingest this large amount of knowledge. When the need for biomedical NLP unraveled, along with the benefits of knowledge bases, it was a matter of organizing the documented knowledge into machine-readable knowledge bases. Most notably, the metathesaurus of the UMLS [10] came to unify the information in bibliographic and factual databases, in addition to clinical data. This aspect of biomedical NLP reflects the potential of utilizing knowledge bases in this domain, especially given its shortage of annotated data

In this chapter, we discuss the history of knowledge bases in NLP, and the motivation behind them. We also cover what are the different types of knowledge bases accompanied with example KBs. Next, we address what are current methods of integrating knowledge bases into NLP methods. Finally, we enumerate knowledge bases in the biomedical domain and discuss ways to utilize them to advance NLP algorithms in the field.

2.1 History of Knowledge Bases in NLP

Knowledge bases could be traced back to initial attempts to model and explain how minds and language work [15, 16]. And early NLP systems relied on small-sized handcrafted rules and patterns of morphology and syntax [17], especially in the prevalent environment of rule-based systems in early works of machine learning systems. This was the case up until the creation of one of the earliest and largest knowledge bases of NLP: WordNet, a lexical database containing information on around 155,000 words [18]. This information encodes the different senses a word can have, equivalence between these senses, and relations between words on a sense-level. Later, in 1998, the International Computer Science Institute in Berkeley attempted to model language on another level: FrameNet [19]. What distinguishes FrameNet from WordNet is documenting and abstracting the possible actors in an action (verb), along with examples. On another front, many years later, researchers found the need to extend WordNet beyond the limits of English, coming up with BabelNet [20]. By utilizing the multi-linguality of Wikipedia, and machine translation methods, BabelNet was automatically constructed to extend the acyclic graph of WordNet along another dimension: language.

Besides modeling language, and the characteristics of words forming a respective language, researchers found the need to structure background knowledge on entities, such as famous people, cities of the world, and famous events. What started out as Freebase [21], was later merged into Wikidata [22], which in the spirit of Wikipedia, is a collectively created and maintained knowledge base of information on famous people and entities such as (Barack Obama) being a (President of) the (United States of America), along with metadata such as start and end time. Resources such as WordNet and FrameNet can contribute to more fundamental tasks such as abstract meaning representa-

tion, whereas resources such as Wikidata can contribute to more downstream applications such as question answering.

2.2 Types and Examples of Knowledge Bases

One broad way, although not necessarily exhaustive way, to categorize knowledge bases is based on the type of information it encodes: (1) lexico-semantic information over words, or (2) relational information over world entities.

2.2.1 Word-level Knowledge Bases

The first type of knowledge bases we elaborate on is word-level knowledge bases. These knowledge bases encode the meaning of words of a language and present a structure over these words, understanding the relations and interactions between these words. These resources tend to help NLP applications addressing the lower layers of language such as semantic parsing, and abstract meaning representation. For example, [23] concurrently utilized three word-level knowledge bases to guide semantic parsing: WordNet, VerbNet [24], and FrameNet. Next, we detail several examples of word-level knowledge bases.

WordNet and BabelNet

WordNet organizes and assigns features to a large coverage of English words ranging from nouns, to verbs, to adverbs, to adjectives, etc. As of June 2020, WordNet covers 155,327 English words. WordNet further assigns multiple senses to each word. For example, the word "bank" can exist in WordNet as "bank_1" to represent the financial institution, and "bank_2" to represent the bank of a river. These word-sense pairs, accumulating to 207,016 pairs, are then collapsed, or clustered, to 175,979 synonym sets (synsets). For example, "car_1" and "automobile_1" would be assigned the same cluster as they are synonyms meaning: "a motor vehicle with four wheels; usually propelled by an internal combustion engine". Finally, these synsets are then assigned directed relations among them, portraying interactions between the semantics of the synsets. Relations identify hypernyms, hyponyms, meronyms,

holonyms, antonyms, and entailment, among others. For example, "wheel_2" representing the car wheel sense of the word "wheel", would be connected to "car_1" via the relation "meronym".

Extending WordNet beyond English, BabelNet [20] was created by linking WordNet to Wikipedia, and then using Wikipedia hyperlinks (across languages as well) between words and entities, word-sense-language triplets are added to the aforementioned synsets. For broader coverage, machine translation methods were also utilized to generate more word-sense-language triplets, and estimate relations between them.

Besides utilizing WordNet for definitions and synonym generation, researchers have also relied on the directed acyclic graph of WordNet to assess the semantic similarity between English words using graph-based distance measures [25].

FrameNet

Taking another approach at modeling language and the relations between words of a language, FrameNet instead centers its modeling around verbs, amounting to 3,040 of them. Each frame in FrameNet describes a scenario (for example, "being_born"). A frame can be invoked by several verbs that represent the scenario of this frame (for example, "born"). Each frame (or scenario) also identifies key players (core frame elements: FEs) such as the child in "being_born", and non-core FEs such as the time of birth.

This knowledge base helps in grouping words into scenarios to give an abstract representation of the meaning of discourse. Moreover, it helps identifies actionable relations between words, rather than organizing them on a conceptual level.

2.2.2 Entity-level Knowledge Bases

The other type of knowledge base in our dichotomy is the entity-level knowledge base containing information on famous living and non-living entities and the relations between them. The significance of these knowledge bases is their ability to help downstream NLP applications to reason about the word and provide more accurate answers. For example, in a document describing the life of an elephant, paired with a question "Which mammal is the largest

land animal?", having a knowledge base that identifies elephants as mammals, would help the system provide the accurate answer, even though the system might have not seen anywhere in its training data the word "mammal".

Wikidata

Wikidata structures information on topics, objects, or concepts into documents containing information such as the label, the description, and different properties of these entities [22]. Each document covering one entity is given a unique QID. For example, the different cities of Tripoli, Lebanon (Q168954), and Tripoli, Libya (Q3579), are given the same label but differing unique QIDs. Example properties of Tripoli, Lebanon, are "elevation above sea level" (P2044) by 222 meters. Note that even properties have their own unique ID starting with "P". These properties could also represent relations (links) between entities. For example, Nick Holonyak (Q360445) "educated at" (P69) University of Illinois at Urbana-Champaign (Q457281). By June 8, 2020, Wikidata includes information on 86,942,351 items.

Wikidata is owned by the Wikimedia group, which also owns Wikipedia, and develops Wikidata in the same approach Wikipedia was developed: collaborative curation. This also allows Wikidata to utilize the textual and metadata knowledge of Wikipedia.

DBpedia

DBpedia [26], which precedes Wikidata in time, takes the opposite approach. Instead of collaboratively curating information which might lead to automatically generated infoboxes of Wikipedia, it uses the already present infoboxes to automatically collect information on world entities covered by the Wikipedia project. By 2016, DBpedia had information on around 6 million entities with the following distribution: 1.5M people, 810K locations, 301K species, 275K organization, 135K music albums, 106K movies, 20K video games, and 5K diseases.

NELL: Never-Ending Language Learning

All previously mentioned knowledge bases rely on manual labor either directly (WordNet, FrameNet, Wikidata), or indirectly (BabelNet, DBpedia). NELL [27] on the other hand attempts to emulate the way humans acquire knowledge; namely, starting with a set of known facts, it continuously skims through the internet, everyday learning new facts, and revisiting previously acquired facts. This work, contrary to previously mentioned knowledge bases, falls under automatic population of knowledge bases.

NELL utilizes textual patterns such as knowing that (Cristiano Ronaldo, plays for, Juventus) allows it to look for all sentences including these two entities and learn all the phrasings that represent the meaning "plays for". It can then later use these learned phrasings to estimate new relations of "plays for" for new entities. In less than a year, NELL was able to double its knowledge and learn 440K new relationships that are 87% accurate [27].

2.3 Methods of Integrating Knowledge Base Information

How to best integrate knowledge base information into natural language processing systems depends on the underlying algorithms being used. With the ever-continuous development of NLP algorithms, integration methods need to match the development. Most notably, the shift of NLP algorithms to deep learning methods presents a challenge on ways to integrate this knowledge. In this section we divide the algorithms, and accordingly the way knowledge bases were integrated, into two eras: (1) pre-deep learning era, and (2) post-deep learning era. This separation point is also inspired by the motivation of this work: how to integrate biomedical knowledge bases into biomedical NLP algorithms after the recent shift to deep learning.

2.3.1 Pre-Deep Learning Era

One of the differentiating characteristics of machine learning and NLP before and after deep learning is the amount of feature engineering required by researchers. Pre-deep learning, most of the effort lied in engineering the most

discriminating features to feed into an ML algorithm to perform classification. Naturally, knowledge bases ended up being utilized as a source for instance features.

As a representative example, in [28], when building a system to perform entity-linking, they relied on popularity features from DBpedia when constructing features to be passed to a support vector machine (SVM) [29]. Other examples of features could be similarities of words, POS tags of words, etc.

2.3.2 Post-Deep Learning Era

With the progression of deep learning methods and the takeover of sequence-to-sequence modeling over almost all NLP tasks, integration of knowledge base information adapted to the change. Neural methods of integrating knowledge bases can be divided into: (1) implicit utiliztion, where concepts in a knowledge base are encoded into a fixed-length vector, and used as input into the end-to-end network, or (2) explicit utilization, where the knowledge base directly controls the operation of the end-to-end network.

One example of implicit utilization is the work in [30], where knowledge base information from ConceptNet and Wikipedia was transformed into text via rule-based methods, and then passed through DL-based encoding architectures that learned contextually refined word embeddings. These embeddings were later used as inputs to DL-based architectures for several downstream tasks such as question-answering and recognizing textual entailment. This implicit utilization led to an increase in accuracy of answering questions on the SQuAD dataset from 75.9% to 79.7%, reflecting the significance of added knowledge base information. The challenge in implicit utilization of knowledge bases is how to best encode knowledge into a fixed-length vector. The transformation of a knowledge base to a corpus of text is not directly applicable to any knowledge base, and might differ between domains.

In contrast, other work explicitly integrates information from a knowledge base. Work in [31] explicitly integrates knowledge base information into a machine reading comprehension system that given a passage, and a question, automatically answers that question using information from the passage. In this system, [31] utilize WordNet connections between words to infer whether

two words are related. Then, in the attention mechanism of the neural network, only attention between words that are related in WordNet are allowed to exist, and the rest are masked with a zero. This forces to system to generate its attention-based representations only by attending to words we a priori know are semantically connected. This integration, although it did not push the state-of-the-art results on the SQuAD dataset, was reflected in enhancing performance under adversarial circumstances: (1) limited training dataset size, and (2) injected adversarial sentences intended to confuse the system. Having background knowledge information allowed the system to beat the state-of-the-art system when only 20% of the data was available by approximately 6% absolute, and when adversarial sentences were injected to the data by approximately 9% absolute. The challenge in explicit utilization of knowledge bases is how to minimize the bias introduced by human design which controls how the knowledge base affects the end-to-end system. For example, in this case, the authors assumed that only words related to each other should be considered in the attention mechanism.

2.4 Biomedical Knowledge Bases and Their Potential

Despite the particular obstacles presented in the healthcare domain, resources specific to the healthcare domain present themselves.

The first type of resource is the lexical database, best exemplified by the Unified Medical Language System (UMLS). With the technicality of biomedical terms and the urge for abbreviations, the National Library of Medicine (NLM) realized the added challenge for automated algorithms and addressed that by launching a long-term research project to build the UMLS [10]. Accordingly, NLM quarterly releases an updated version of the UMLS to researchers with information regarding phrasal equivalence of biomedical terms, relations between terms, as well as semantic types of these terms, besides other pieces of information. Moreover, the NLM released a set of off-the-shelf NLP tools called the SPECIALIST NLP Tools [32] to perform infrastructural NLP tasks such as part-of-speech tagging, spell checking, text categorization, etc.

The second type of resource comes in the form of labeled data for several NLP tasks. Foundations such as n2c2 (formerly i2b2) realized the need for

labeled data and accordingly generated datasets and organized challenges around these datasets. Some example challenges are: concept and relation extraction in clinical records [33], extracting temporal relations [34], cohort selection for clinical trials [35], detecting adverse drug events [36], and more recently, evaluating clinical textual similarity.

CHAPTER 3

TEXT SIMPLIFICATION OF MEDICATION INSTRUCTIONS

3.1 Introduction

Healthcare practices have granted patients increased access to their health information to support self-care [37, 38]. But the benefits have been hindered by patients' low comprehension of their own health data [39], as a study shows that readability measures of online health information is significantly higher than patient health literacy abilities [40]. Moreover, older adults, the largest demographic group interacting with the healthcare system, are often the least health-literate [41, 42]. With low levels of health literacy resulting in worse health outcomes [43, 44], there is an urgent need to reduce the gap between the health literacy of patients and the health literacy demands of the US healthcare system.

This mismatch in patient literacy levels and health documents is due in part to the differing language used by healthcare professionals and patients [8]. For example, what professionals refer to as "abdominal pain", patients might refer to as "stomach ache". While previous works have addressed this by performing local word replacement [45], their context-free frameworks lacked the accuracy. In a health document, "Mg" could mean "milligrams" or "Magnesium", and harnessing the contextual information, for example in "Take 50 Mg" or "Mg reacts with", aids accurate simplification.

Our approach is a context-aware medical text simplification system, named Dr. Babel Fish (DBF). We design our system to be independent of the availability of annotated datasets as scarcity of such data is expected due to privacy and proprietary concerns. To compensate for annotated datasets, we instead rely on a structured knowledge base in the form of the Unified Medical Language System (UMLS) [10]. Taking inspiration from the modular and context-aware frameworks of phrase-based statistical machine translation

(PBSMT) systems [46], our system, DBF, first identifies hard (low frequency) words, then collects possible simplifications of these words from the UMLS, and finally chooses the simplification that best reflects patients' preferred medical terms and best fits the context, by relying on a patient language model trained on a suitable monolingual corpus.

Although neural machine translation (NMT) frameworks [47, 48] constitute the state-of-the-art, they suffer in the low-resource settings of the clinical (medical) domains, and we accordingly present our system to complement neural methods in domains lacking the appropriate parallel corpus. Although we take medication instructions as a use case, our system is general enough by construction to handle any medical text. All code and materials associated with this study are released to the public.[1] This chapter:

- studies a knowledge-aware text simplification model that does not rely on parallel text.
- empirically demonstrates the higher precision simplification output of the proposed model compared to previous methods.
- presents a parallel corpus of medication instructions available to foster future research.
- provides a comprehensive and comparative study of NMT models applied to healthcare text simplification, previously impossible due to the lack of a parallel corpus.
- via a human subjects' study, shows the positive impact of DBF on patient comprehension.

3.2 Previous Work

Efforts to improve patient comprehension of health information in the biomedical informatics community can be categorized into: developing standards [49, 50], curating dictionaries [51], annotating text with additional information [52, 53, 54, 55], normalizing terms [56], syntactic simplification [45], and finally, lexical simplification [45, 57, 58]. Our work on biomedical text simplification belongs to the final category.

[1] `http://bit.ly/dbf-ml4health`

One popular previous attempt [45] of health material text simplification relies on the consumer health vocabulary (CHV) [51] for mapping the hard term to its simpler counterpart, disregarding context information. Other word-replacement systems [57, 58] have relied on MetaMap [59] to map medical terms to their simpler counterparts by either utilizing CHV as a thesaurus [57], or relying on an in-house equivalent resource (CoDeMed) [58]. Although, MetaMap performs word sense disambiguation (WSD) by relying on the context, its creators admit its low WSD quality [12]. Therefore, we rely on a language model instead of MetaMap. Nonetheless, since MetaMap followed by a CHV (or another dictionary) is a popular method in previous works, we include it as a baseline in our experiments.

Beyond health materials, lexical simplification is highly researched. A thorough survey of this field is presented in [60], which divides work in this field into four stages of a pipeline: (1) complex word identification, (2) substitution generation, (3) substitution selection, and (4) substitution ranking. We also perform complex word identification as a first stage by relying on word frequencies, which is a popular method among previous work [61, 62, 63, 64]. We also generate substitutions as a second stage by relying on UMLS, similar to how previous work relied on word taxonomies [65, 66]. The last two stages are performed in one shot in DBF, where instead of finding which candidate substitutions fit the context and then selecting the simplest based on a certain metric, we let the language model decide which is the most probable substitution in terms of meaning and simplicity. Our work is the first to combine these stages in a context-aware method tailored for the healthcare domain. Finally, text simplification has been modeled previously as a machine translation task where parallel corpora are available [67, 68]. Accordingly, we compare against these methods in this study to assess their capacity in the low resource setting and their capability to generalize across healthcare domains.

3.3 Materials

Next, we describe the materials used in our study to build and evaluate the various systems.

3.3.1 Parallel Corpus

We collected 4554 unique and de-identified medication instructions from the electronic health records of a collaborating healthcare institution. They were of two types: (1) Structured– automatically populated using three drop-down fields: Dose, Route, Frequency (2) Free-text– manually typed. Free-text instructions tend to have more hard words due to their uncontrolled nature.

Then, for every instruction, a physician, with expertise in standard practices for increasing patient comprehension, annotated each instruction with its accurate simplification. The resulting parallel corpus is essential to the training of the supervised NMT methods, and evaluation of all systems.

3.3.2 Monolingual Corpus

Next, in order to develop a corpus representative of the target language (accessible to patients), we scraped medication-related pages from five medicine-related websites[2] targeted for laypeople. We selected the five websites to be: (1) medication-related, and (2) patient-facing. This corpus C_t ($\approx 11M$ words) was used to: (1) train a language model, and (2) estimate usage frequency of words by DBF's target audience.

3.3.3 Human Subjects Study

Finally, we designed an online human subjects study (via Mechanical Turk) that presents medication instructions to participants and tests their comprehension of the instructions, before and after simplification, using multiple-choice questions.

Accordingly, we randomly choose 100 of the free-text medication instructions of varying levels of hardness (1: 29 instructions, 2: 29 instructions, and 3: 42 instructions) as measured by the number of hard (low frequency) words. Then, we simplify every instruction using DBF, and pair both the original and simplified versions of the instruction with the same multiple-choice question.

[2] `medlineplus.gov`; `nia.nih.gov`; `umm.edu`; `mayoclinic.org`; `medicinenet.com`

3.4 Methods

In this section, we describe our method, DBF, along with the established baselines it was quantitatively evaluated against: MetaMap+CHV, Seq2Seq-w-Attention, and Pointer-Generator.

3.4.1 Dr. Babel Fish

For reproducibility purposes, following is a detailed system description. DBF is designed as a three-stage pipeline. First, hard (and easy) words are identified based on their frequency of usage. Then, in the second stage, candidate simplifications of a given hard word are collected and each given a replacement probability (p_{rm}). In the final stage, every candidate output simplification is assigned a language model score and a replacement model score. We will refer to this system as an "unsupervised" system due its independence of annotated datasets, as well as "knowledge-aware" due to its reliance on a knowledge base in the form of UMLS. The highest scoring simplification is then selected as the output of DBF. We describe the three stages in the following subsections (see Figure 3.1).

Stage 1: Identification of Hard Words

In the first stage, the task is to identify the hard words to be translated from the source sentence and to retain the easy words. Accordingly, we devise a simple statistical model which checks a word's frequency of usage in C_t (see Materials Section). We consider the high usage frequency of a word by patients (or targeted towards patients) to be a strong indicator that it is easy for patients to understand, and vice versa. Thus, if a given word has a frequency lower than a tunable frequency threshold (f_t), DBF labels it as hard.

Stage 2: Candidate Generation

Next, DBF relies on the UMLS to collect all candidate replacements of each hard word, and estimates the probability of each candidate.

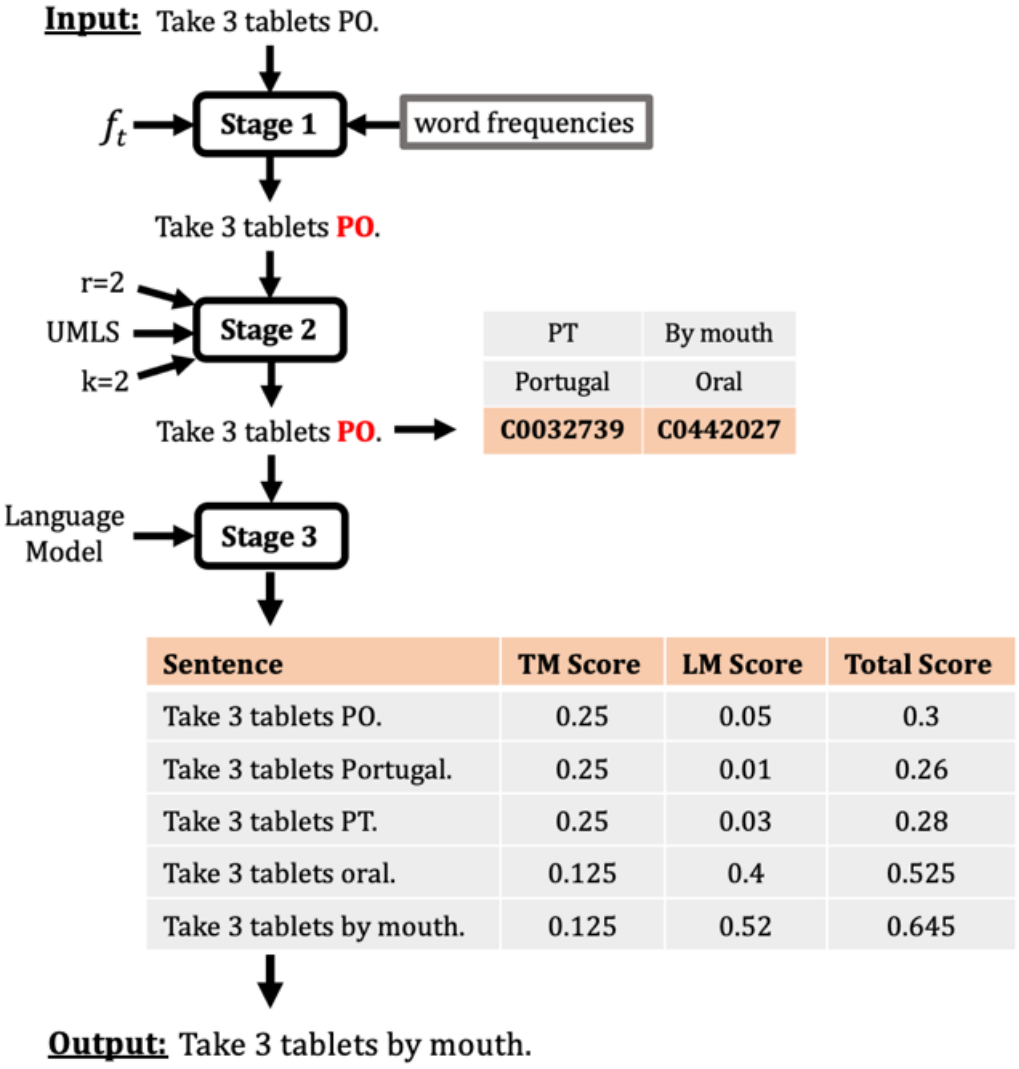

Figure 3.1: Block diagram of DBF for the sample sentence: "Take 3 tablets PO."

Table 3.1: Example UMLS concepts

Oral	**Twice a day**	**Milligram**
PO	BID	Mg
Orally	Twice daily	Milligramos
By mouth	Two times daily	Milligrams

Table 3.2: Example UMLS queries

Query	**PO**	**BID**
1^{st} Result	Portugal	BID Protein
2^{nd} Result	Oral	Twice a day
3^{rd} Result	Positive	BID gene

A salient feature of the UMLS is its groupings of words/phrases into clusters, where each cluster represents one concept. In Table 3.1, we present three example concepts, each headed by its "Preferred Name", followed by three example atoms (the UMLS term for a phrase in a given concept). We note the variability of atoms in a concept in terms of complexity and language.

A second feature of the UMLS is its ability to return an ordered list of concepts to best match a search query. In Table 3.2, we see the top three concepts returned for two example queries: "PO", and "BID". The correct concept for "PO" appears only second in the results, as is the case for "BID". This suggests that just relying on the top result of such a context-insensitive static search of the UMLS is insufficient for accurate simplifications.

Leveraging these two features, DBF uses the hard word from the input sentence as a query to the UMLS search function. Then, all atoms of the top k returned concepts are considered as candidate simplifications, with concepts ranked higher assigned higher probabilities.

Formally, let $\{C_1, C_2, ..., C_k\}$ be the top k concepts returned by the UMLS search feature when using the hard word c as a query. Also, let $C_i = \{a_{i1}, a_{i2}, ..., a_{in}\}$ be all the atoms of the i^{th} concept. Then, the probability of atom a_{ij} being the simpler replacement of c is assigned $p_{rm}(a_{ij}|c) \propto \frac{1}{r^i}$, where $r \geq 1$. Thus, an atom of the i^{th} concept is allocated a probability r times that of an atom of the $(i+1)^{th}$ concept. In this setup, r and k are tunable hyperparameters of the system. To allow for possibly keeping a hard word c unaltered on the output side, we also assign the probability $p_{rm}(c|c)$

equal to that of an atom in the 1^{st} concept. This helps in cases where a word was wrongly identified as hard, or a simpler alternative does not exist for it. For an easy word e, we assign $p(e|e) = 1$ to force retention of easy words.

Stage 3: Decoding via Language Model

Finally, we consider all possible combinations of simplifications and choose that with the highest product of replacement probability and language model probability.

Formally, we identify $T(c) = \{t_1, t_2, ..., t_m\}$ which is the set of possible simplifications for a word c. Using this, the set of possible simplifications of the input sentence becomes $H = T(c_1) \times T(c_2) \times T(c_n)$, where $\times$ refers to the Cartesian product of sets.

Now consider a sentence $t \in H$ and let $t = t_1 t_2 ... t_T$ where t_i is the i^{th} word of t. Then, $P(t|c) \propto \prod_{i=1}^{T} p_{rm}(t_i|c_i) * p_{lm}(t_i|t_{i-1:i-5})$, where $p_{lm}(t_i|t_{i-1:i-5})$ is the probability assigned by the 6-gram language model [69], for the word t_i occurring after the sequence of words $t_{i-5}t_{i-4}t_{i-3}t_{i-2}t_{i-1}$. The 6-gram language model is trained on the patient-friendly corpus to model the target language. Finally, the sentence t with the highest assigned probability $P(t|c)$ is selected as the output simplification of the system.

The significance of the language model is, first, it utilizes the context in which a word like "PO" appears to reward a simplification like "Oral", and penalize a simplification like "Portugal", especially considering that "Portugal" is assigned a higher replacement probability (p_{rm}). Second, it encodes word usage preferences–such as "by mouth" being preferred over "Oral"– even though they both had equal replacement probabilities (p_{rm}).

3.4.2 MetaMap+CHV

To compare DBF to the majority of previously used methods for simplifying health materials, we implement the following baseline. Text is first passed through MetaMap, which maps phrases in the text to their respective UMLS concepts. Then, for every phrase identified, we first check if it includes at least one hard word. If it does, and if that UMLS concept is covered by CHV, we replace it by CHV's most preferred term for that concept; otherwise, we replace it with the UMLS preferred term for that concept. With that

being said, all phrases identified by MetaMap, which are not contiguous, are ignored to avoid errors in sentence structure when performing the phrase replacement.

3.4.3 NMT Baselines

Our last set of baselines are two supervised NMT architectures [47, 70], requiring training data.

One NMT baseline we consider is a Seq2Seq-with-Attention architecture [47]. In this deep learning architecture, a long short-term memory (LSTM) encoder maps the input sentence to a fixed length vector, and generates contextualized representations of the input words. Then, an LSTM decoder generates the output words sequentially based on the fixed length vector and the contextualized representations, while the attention mechanism indicates which input words influence each output decision. For this baseline, we utilize Google's open source implementation [71] with default parameters.

Due to the large overlap in the vocabulary of the source and target sentences, particularly the "easy" words, we consider a second NMT baseline called Pointer-Generator capable of copying words as is from the source sentence [70]. It differs from Seq2Seq-with-Attention in that at every decode step, it estimates a probability g of generating a new word rather than copying a word from the source sentence. If g is low, the model relies more heavily on the estimated attention distribution over the input source words, which increases the chances of copying the word that is most highly weighted by the attention mechanism. To implement the system, we use the author's original open-source implementation [70] with default parameters, except for using the Proximal Adagrad [72] optimization algorithm to maximize performance.

3.5 Results

This section describes the results of two studies. The first study uses automated evaluation metrics to assess DBF's output in comparison to the baselines considered. The second study evaluates the impact of DBF on laypeople comprehension.

One standard measure for machine translation tasks is BLEU score [73],

which measures the overlap in words and phrases between a system's output and a reference output. It is also used frequently in other sequence-to-sequence problems such as text simplification. Nevertheless, BLEU has been shown insufficient for text simplification tasks due to the large overlap between the source and target vocabulary [74]. Therefore, we instead consider the SARI metric, which showed better correlation than BLEU with human judgement on text simplification tasks [74]. SARI, similarly to BLEU, measures the overlap of the system's output with a reference output, but also measures the amount of novelty introduced by the system. The novelty component in the metric rectifies BLEU's shortcoming in measuring the performance of a text simplification system. Moreover, we also use the PINC metric [75] to measure, in isolation, the amount of novelty introduced by a system.

As for the experimental setup, to avoid evaluating systems on a limited dataset size, we perform 5-fold cross validation to utilize the full dataset for evaluation. For every fold, we take 20% of the training data for tuning.

3.5.1 Automated Evaluation:

We present in Table 3.3 the average performance of all systems on the evaluation portion of the dataset for all five folds. We also distinguish between the performance on the full dataset and the more critical subset – free-text instructions, and include the results in Table 3.4. For reference, we also include a baseline system that performs no change.

Table 3.3: Performance of the simplification systems on all medication instructions

Method	**Supervision Type**	**PINC**	**SARI**
No Change	N/A	0.00	32.83
MetaMap+CHV	Knowledge-Aware	25.84	45.64
DBF	Knowledge-Aware	19.61	55.33
Pointer-Generator	Direct Supervision	32.25	54.75
Seq2Seq-w-Att	Direct Supervision	50.81	**79.26**

Table 3.4: Performance of the simplification systems on the free-text subset of the medication instructions

Method	**Supervision Type**	**PINC**	**SARI**
No Change	N/A	0.00	39.29
MetaMap+CHV	Knowledge-Aware	26.32	54.35
DBF	Knowledge-Aware	21.52	**56.51**
Pointer-Generator	Direct Supervision	36.34	40.01
Seq2Seq-w-Att	Direct Supervision	78.35	48.27

We first compare the two knowledge-aware systems: MetaMap+CHV and DBF. First, and confirming our main hypothesis, the context-aware framework of DBF led to higher quality simplifications gaining an absolute 9.7% improvement in SARI scores over MetaMap+CHV, and a 22.5% gain compared to the No-Change case. Even though MetaMap has the added flexibility to operate on a phrase level, we attribute its comparatively lower quality to its poor WSD. Second, and by comparing PINC scores, we notice that DBF is more conservative in its changes, making it less likely to mistakenly alter key information, arguably a desired behavior in a critical domain such as healthcare. This is mainly due to it considering the identity replacement as a possible simplification, and letting the context decide whether to attempt simplification or not. These observations are also consistent on the free-text subset of the evaluation data, though we note that the gap shrinks between the two systems. We hypothesize that this is due to MetaMap+CHV committing consistent errors over one or more highly repeated terms in the structured subset of the medication instructions.

Next, we observe that, including the supervised deep learning methods, Seq2Seq-w-Attention performs significantly better than all systems. The high performance of the Seq2Seq-w-Attention is an expected result, due to the advantage of direct supervision in general, but also because direct supervision would allow it to memorize the annotator's style as well. The poor performance of the Pointer-Generator was unexpected considering its mechanism to pass easy words. Upon further inspection, we noticed two factors that degraded performance. First, the copy mechanism led to meaningless repetition of words as previously noted in the literature [76]. Second, the

copy mechanism led to copying hard words as is.

Finally, we focus our attention on how performance levels are affected when considering free-text instructions only, which are more representative of complicated health material. We notice that all the systems show more activity (higher PINC scores) in their simplifications, as these systems encounter more hard words in the original instructions. This provides further evidence that the free-text instructions constitute a critical component of the evaluation. Second, we notice that the performance of the supervised systems suffers significantly on the free-text instructions (compared to that on All Instructions), while that of the knowledge-aware (utilizing background knowledge such as UMLS and CHV) systems remain comparable, to the extent that DBF becomes the best performing approach on free-text instructions. This reflects the robustness of the knowledge-aware systems in a low-resource setting. In a setting where a sufficient parallel corpus is available, neural machine translation systems are recommended, but in the absence of a sizable in-domain corpus, DBF achieves better performance.

3.5.2 Simplification Effects on Patient Comprehension:

We also investigated whether DBF's simplification efficacy helped improve laypeople comprehension, by measuring their ability to answer multiple choice questions (percent correct) on medication instructions before and after simplification (see Figure 3.2). Participants, on Amazon Mechanical Turk, were 160 adults diverse in age, cultural and academic background, and gender. 100 instructions were randomly selected from the free-text subsample of our original set of medication instructions (see Materials Section), along with their DBF simplifications and their respective multiple choice questions. Each participant read 50 instructions and answered the corresponding questions. A counterbalancing scheme ensured that each participant read 25 original instructions (as written by the physician) and 25 instructions simplified by DBF. No participant encountered both the original and the simplified version of the same instruction. Also, hardness levels of medication instructions were balanced for each participant.

The key result of this experiment was that the participants understood the simplified instructions 24.4% better than the original instructions ($F(1, 7973) =$

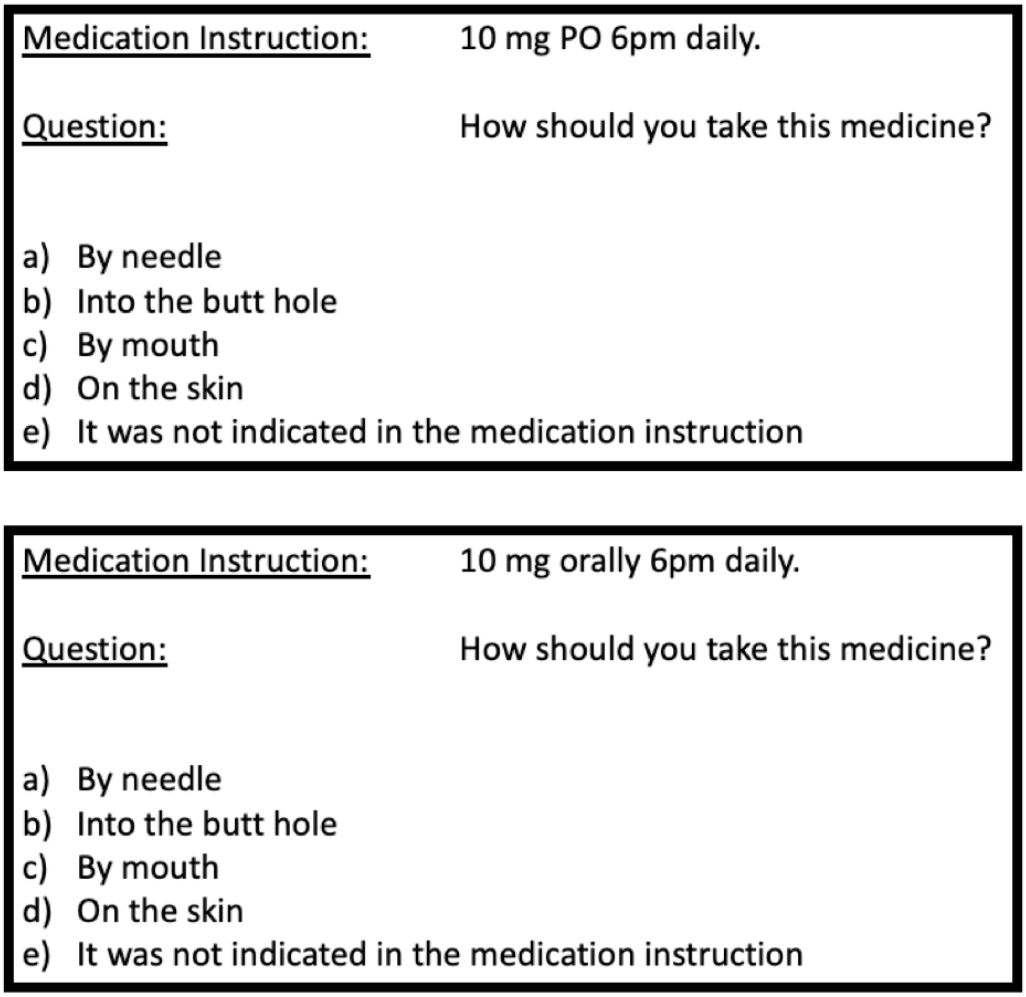

Figure 3.2: Example questions from the online human subjects study

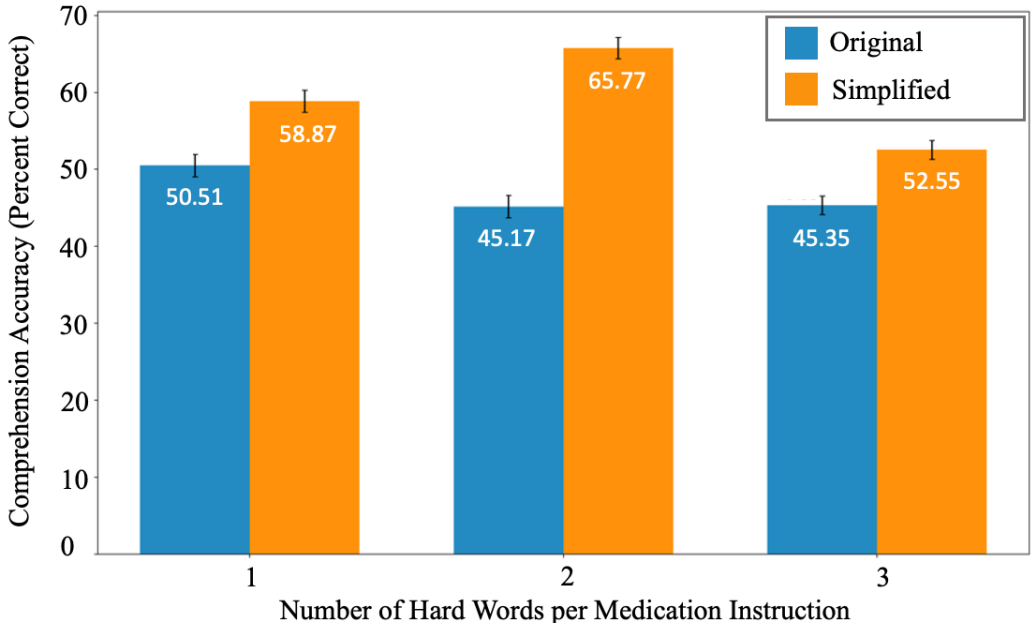

Figure 3.3: Impact of DBF on the different hardness levels of medication instructions

112.3, $p < .0001$, 58.23% vs 46.80%). Hardness level also influenced comprehension ($F(2, 7973) = 14.6$, $p < .0001$; see Figure 3.3). The simplification benefit was largest when there were two difficult words (45.63% relative), rather than one (16.55% relative) or three difficult words (15.87% relative). It is possible that having two rather than one difficult word gave more potential for DBF's simplification to increase comprehension. However, when the simplification process involved three words, the propagation of error led to a decrease in the quality of the simplification, and this may have negatively impacted comprehension.

3.6 Discussion

To better understand the functioning of the knowledge-aware systems, we study the effect of f_t on their first stage of identifying hard words. Upon tuning the systems on the validation dataset, f_t was set to 672 for both systems coincidentally. Based on Figure 3.4, we deduce that around 18% of words in the original instructions were attempted for translation, reflecting a high recall of hard words.

Along the same lines, we check the effect of f_t on DBF and MetaMap+CHV

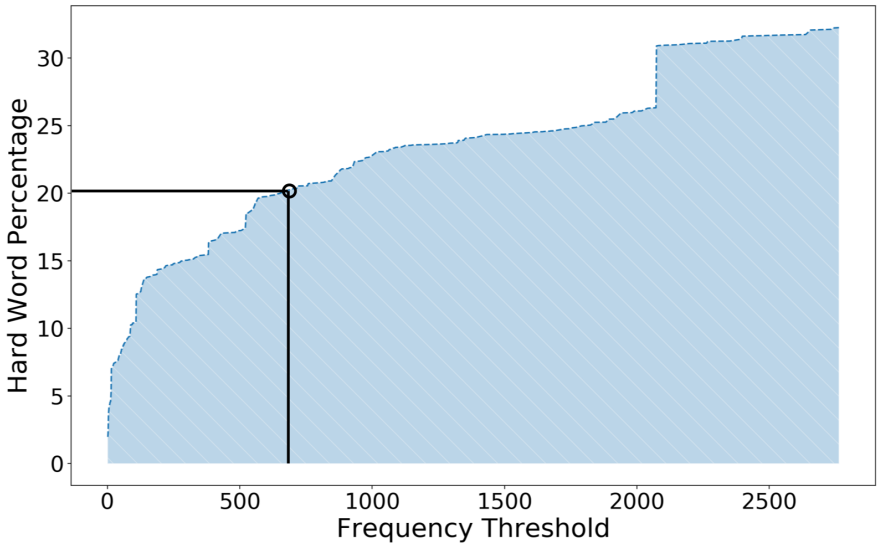

Figure 3.4: Percentage of words considered hard, and accordingly considered for lexical replacement, as we vary the frequency threshold

in terms of the two evaluation scores (see Figure 3.5). In terms of PINC scores, we observe an expected pattern of increase as we increase f_t for both systems. As f_t increases, both systems attempt to modify more of the original sentence (including easy words) leading to a lower overlap with reference sentences. MetaMap+CHV introduces more novelty as we increase f_t since DBF can retain easy words even if they were identified as hard, unlike MetaMap+CHV. As for SARI scores, we observe the significance of tuning the first stage, where too low of an f_t results in reduced performance due to lack of attempted translations (low PINC scores), and too high of an f_t results in reduced performance due to translating easy words. Moreover, we observe consistent enhances in performance for DBF over MetaMap+CHV for all f_t considered.

Moving our attention to the effect of k and r on the performance of DBF, we show in Figure 3.6, DBF's SARI score when varying k and r from 1 to 5, and fixing f_t to 672. Our first observation is a positive trend as we increase k, particularly for $r = 1$. This shows the aptitude of the language model at selecting the best translation even when faced with a plethora of options given equal translation probabilities. As for r, we notice reduced performances for any r value different from 1. We thus conclude that the model we use

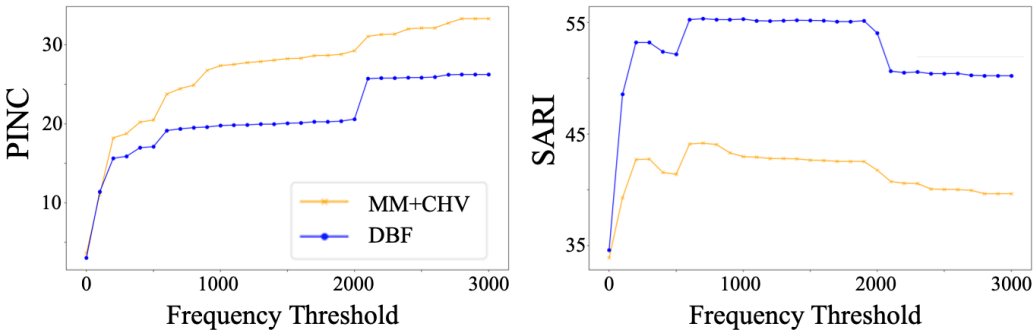

Figure 3.5: Effect of the frequency threshold on the performance of DBF, and MetaMap+CHV

for estimating translation probabilities is not benefiting translation quality. Moreover, the ranking of the concepts returned by the UMLS search function has insignificant value, when an appropriate language model is present.

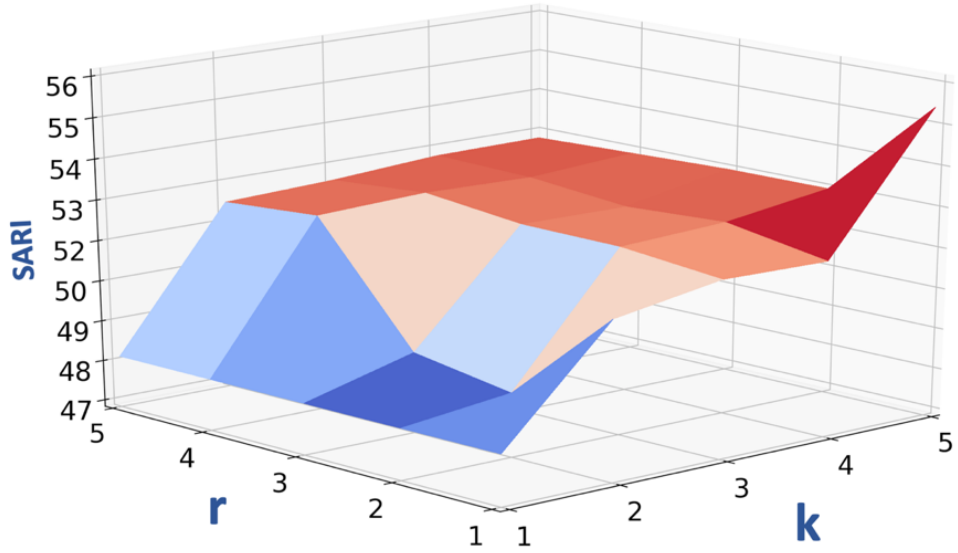

Figure 3.6: Effect of the hyperparameters k and r on the performance of DBF

Finally, we analyze several example simplifications from the various systems in Table 3.5. The first example shows the incapability of Seq2Seq-w-Att to recover from a wrongly generated first word (Wheeled). Moreover, we notice Pointer-Generator's tendency to even pass hard words. In the second example, we notice how MetaMap+CHV retains "PRN" despite being a hard word, due to MetaMap not mapping it to any UMLS concept. Additionally, we see Seq2Seq-w-Attention's mishandling of numbers since it does

Table 3.5: Sample output simplifications from the different systems considered

Source:	Total 90 mg QAM.
Gold:	Total 90 milligrams every morning.
DBF:	Total 90 mg every morning.
MetaMap+CHV:	Total 90 mg every morning.
Seq2Seq-w-Att:	Wheeled systolic blood sugar test result is between 301 and 180,
Pointer-Generator:	Total 90 mg QAM.
Source:	Every 4-6 hours PRN thoracic back pain.
Gold:	Every 4 up to 6 hours as needed for chest back pain.
DBF:	Every 4-6 hours as needed thoracic back pain.
MetaMap+CHV:	Every 4-6 hours PRN thoracic back pain.
Seq2Seq-w-Att:	Every 6 hours as needed for back pain.
Pointer-Generator:	Every 4-6 hours PRN back pain.
Source:	Take 15 g by mouth 2 times daily as needed.
Gold:	Take 15 grams by mouth 2 times daily as needed.
DBF:	Take 15 grams by mouth 2 times daily as needed.
MetaMap+CHV:	Take 15 gram per deciliter by mouth 2 times daily as needed.
Seq2Seq-w-Att:	Take 15 grams by mouth 2 times daily as needed.
Pointer-Generator:	Take 15 grams by mouth 2 times daily as needed.
Source:	For better hearing with the ear, avoid cleaning your cerumen.
Gold:	For better hearing with the ear, avoid cleaning your earwax.
DBF:	For better hearing with the ear, avoid cleaning your wax.
MetaMap+CHV:	For better hearing with the ear, avoid cleaning your earwax.
Seq2Seq-w-Att:	Provide syringes dressings with the month, and Sunday more Lantus.
Pointer-Generator:	For UNK UNK with the UNK UNK

not have a mechanism for passing easy words. We also notice how Pointer-Generator wrongly eliminates words (thoracic) essential to the meaning of the sentence. On the other hand, the next example shows the shortcomings of MetaMap+CHV's disambiguation algorithms, while DBF was able to accurately map "g" to "grams". Whereas both deep learning methods get the full mark on this example since it is a structured medication instruction.

The last point we would like to address is the last example in Table 3.5. This example, contrary to the previous ones, was not taken from the medication instruction dataset, but rather created by us to portray a complicated sentence from another medical domain, in this case: online health tips. As can be seen from the systems' outputs, the robustness of knowledge-aware systems is evident in comparison to the supervised deep learning methods, which are completely off the mark.

CHAPTER 4

ENTITY LINKING FOR AUTOMATIC SHORT ANSWER GRADING FOR MEDICAL TRAINING

4.1 Introduction

Clinical text is known to be highly ambiguous in nature, posing a challenge for downstream NLP applications [77]. As shown in Figure 4.1, among the terms appearing in UMLS [10], a majority of them have more than one sense, reflecting a high level of ambiguity. A term like "CP" could refer to "Chest Pain", "Cerebral Palsy", "CP gene", and many more concepts. Accordingly, downstream clinical natural language processing (NLP) applications, such as named entity recognition [78], syntactic parsing [79], or relation extraction [80, 81], require the resolution of these ambiguities as part of their algorithm. The complexity of the task is showcased in Figure 4.2. Although "CP" might refer to "Chest Pain" in text, (1) it can map to multiple concepts in an anthology, and (2) it might not map to the intended concept. This motivates the task of entity linking clinical text to an ontology of clinical and biomedical concepts.

Targeting clinical text, several entity linking systems have been proposed: MedLEE [82] relies on traditional parsing techniques such as Definite Clause Grammer (DCG), MetaMap [59, 12] which has been shown to struggle with WSD [12], cTAKES [83] which builds on the SPECIALIST Lexical Tools by NLM [84], and others. These systems have been shown to underperform with F-scores between 0.17 and 0.60 on the task of abbreviation identification and resolution as mentioned in a recent comparative study of clinical text processing systems [85]. Moreover, these systems required intensive training or development, and are susceptible to the nature of the domain language they were developed on. For example, with a shift in the target medical case from a chest pain, to cerebral palsy, abbreviations such as "CP" would also shift in the sense profile while the aforementioned systems have shown to

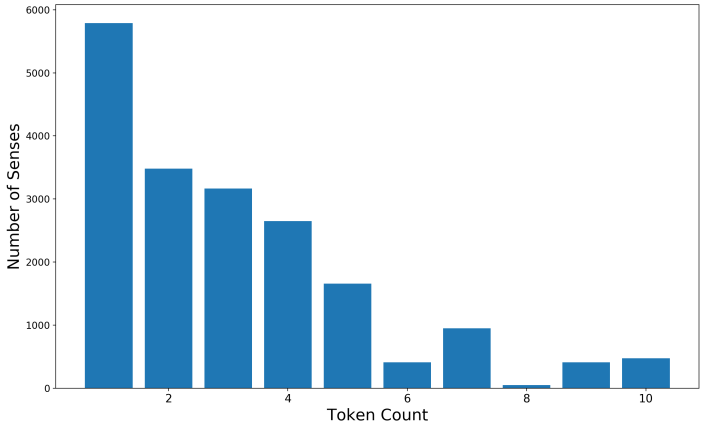

Figure 4.1: Ambiguity of terms in a dataset of clinical chart notes. The senses of a term are counted by the number of UMLS concepts a term participates in.

Chart Notes:
Patient suffering from CP for 2 days. Factors ...
UMLS(Chest Pain):
chest pain, thoracic pain, thorax pain, ...
UMLS concepts with CP as an atom:
Cerebral Palsy, CP Protocol, Centipoise, etc.

Figure 4.2: "CP", which refers to chest pain in a chart note on chest pain, does not take part in the UMLS concept for Chest Pain, while being part of several other concepts.

be non-robust to such shifts [86]. Other systems such as CARD [86] have also been proposed, but require undesired manual intervention. Accordingly, we present an unsupervised, medical case-sensitive entity linking algorithm. It can handle ambiguous full terms as well as abbreviations, while being sensitive to the language characteristics of different medical cases.

To perform entity linking, we rely on three sources of information: (1) distributed representations of words trained on clinical text, (2) unannotated clinical text, and (3) an ontology of clinical technical terms. Given an ambiguous term such as "CP", we assume the presence of a nonambiguous surface form alternative of it such as "Chest Pain" in one of the similar case notes from the given dataset, and use regular expressions to look for such candidates. Then, we rely on Positive Pointwise Mutual Information

(PPMI) as our distributed representations, trained on the aforementioned chart notes, to filter out candidates based on semantic information. We then use candidate results as queries to search for exact matches in an ontology of clinical technical terms for candidate technical concepts. We finally choose the concept whose various surface forms appeared the majority number of times in the text.

In this chapter, we (1) offer a medical case-adaptable, unsupervised entity linking method for clinical text, and (2) showcase its benefits on the task of automatically grading chart notes in comparison to the well-established entity linking system of MetaMap.

4.2 Related Works

As previously mentioned and demonstrated in Figure 4.1, clinical text is rife with ambiguity. The issue is also present in the sister domain of biomedical text. Accordingly, several works have realized the importance of entity normalization (and consequently, entity linking) in biomedical and clinical text. Several studies have been proposed crossing rule-based [87, 88, 89], neural-based [90, 91], and machine learning-based methods [92, 93]. For example, one system [89] focused on disease normalization by identifying five different types of rules affecting the surface form of disease names. Another system [88], also focusing on disease and disorder normalization, automatically learned, by relying on the training data present, transformations done on disease names to generate the different synonyms. Also, several other learning to rank-based systems were proposed, by relying on either a linear-RankSVM [94] or a convolutional neural network [90] to accurately rank the pairs of surface form and normalized form.

More recently, and with the advancement of large pre-trained contextualized language models [95, 96, 97], and biomedical [14], as well as, clinical [98] versions of them, authors in [99] advanced the state-of-the-art in biomedical entity normalization by 1.17%. Our system differs from the previously mentioned learning-based systems in that it does not require training in a setting where chart notes for a particular medical case are not present. Moreover, it differs from rule-based methods in not having to create the ever-expanding rules of variations of clinical term among the different subdomains of clinical

text, as well as among the different physicians/students.

4.3 Method

Linking terms in clinical text to entities is unique in two aspects: (1) Given a medical case, a term tends to be consistent in the sense it takes, and (2) terms shift sense profiles across medical cases. So, for example, a term like "CP" is almost always surely to have the sense "Chest Pain" in a case about chest pain of a patient, and almost always surely to have the sense "Cerebral Palsy" in a case about a patient suffering from "Cerebral Palsy". Accordingly, we design a method which is case-adaptable, but also static per case, or better referred to as one sense per discourse [100], as it has been shown that majority sense methods work well in the absence of a domain shift.

Our method is designed as a three-stage pipeline (Figure 4.3), described later in detail: (1) Candidate Expansion Collection, (2) Candidate Expansion Selection, (3) UMLS Concept Selection. The first two stages are essential to handle the abundance of abbreviations, standard and non-standard, in clinical text, while the last stage operates on all technical terms.

4.3.1 Candidate Expansion Collection

Abbreviations compose a large portion of technical terms, and coverage of abbreviations in the thesaurus we use, UMLS, is low. That is even more aggravated by non-standard abbreviations. Hence, the need to identify the expansion of the abbreviation first.

We first overgenerate candidate expansions of abbreviations using regular expressions. Accordingly, given an abbreviation $C_1C_2...C_n$, where C_i is the i^{th} character of the abbreviation, we look for either n-grams where the i^{th} word starts with the i^{th} character of the abbreviation, or unigrams that start with C_1 and maintain the order of the order of the remaining characters.

For example, an abbreviation like "CP" would possibly return "Chest Pain", and "computer" as candidate expansions among others.

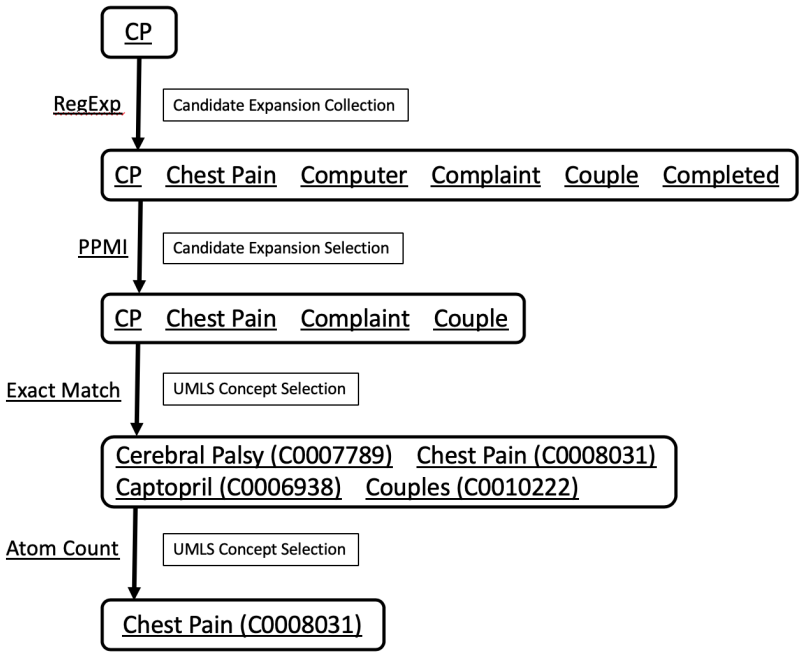

Figure 4.3: Block diagram of entity linking algorithm. Case example: "CP".

4.3.2 Candidate Expansion Selection

Due to the overgeneration of the previous stage, we found it necessary to narrow down the candidates. To perform this operation we rely on the assumption that the abbreviation and the expansion carry the same semantic meaning and thus will be accompanied by similar context. One method to model the semantics via the context associated with terms is relying on Positive Pointwise Mutual Information (PPMI) [101]. In PPMI, word collocations are measured as the positive logarithm of the probability of two words appearing in the same context window, normalized by the product of the probability of the words appearing independently. More formally, $PPMI(x, y) = max(0, log(\frac{p(x,y)}{p(x)p(y)}))$, where x and y are two words in the vocabulary. After we have a measure of the collocation of two words, a word gets a distributed representation of its collocation level with every other word in the vocabulary. Accordingly, we use PPMI on the corpus of chart notes of the case of interest to generate a distributed representation $\vec{w}$ of every word w in our corpus. Assuming $\vec{a}$ is the abbreviation of interest, and $\{\vec{e}_1, \vec{e}_2, ..., \vec{e}_m\}$ is the set of candidate expansions from the previous stage, we check for semantic similarity of every $\vec{e}_i$ with $\vec{a}$ using cosine similarity. More formally, we choose the top candidates that maximize $cos(\vec{a}, \vec{e}_i)$. We found it best,

qualitatively and computationally, to keep only the top three candidates for the next stage.

4.3.3 UMLS Concept Selection

The final stage maps the term of interest to the representative UMLS concept. Given the top three candidate expansions (if any) from the previous stage, and given the term itself, all UMLS concepts with an atom matching one of the expansions or the term are retrieved and considered candidate UMLS concepts $\{c_1, c_2, ..., c_k\}$.

Based on the assumption that different nonambiguous surface forms of the same concept are expected to appear in the corpus of chart notes of a medical case, we score a UMLS concept based on the count of its atoms that appear in the corpus C. Accordingly, $score(c_i) = \sum_{atom \in c_i} \mathbb{1}\{atom \in C\}$. The technical term is then mapped to the UMLS concept with the highest score, finalizing the entity linking process.

4.4 Experiments

We extrinsically evaluate our system and compare against MetaMap on the task of automatic grading of chart notes.

4.4.1 Automatic Grading of Chart Notes

To assess the impact of our entity linking algorithm on clinical NLP downstream applications, we choose the task of automatic grading of chart notes. Medical students interview standardized patient actors (SPs) and are required to document the visit on an electrical chart note. Faculty physicians are then required to manually grade each chart note based on a set of rubric items. This manual labor wastes high in-demand physician time, and delays feedback for students, hindering learning.

Baseline System

Operating with a scope of one medical case at a time, chest pain for example, our baseline system takes as input: (1) rubric, (2) student chart note, (3) case description. The rubric includes items that are expected to be noted in the chart note, and includes items of six categories: (1) Pertinent Positives, (2) Pertinent Negatives, (3) Pertinent Physical Exam Positives, (4) Pertinent Physical Exam Negatives, (5) Diagnoses, and (6) Tests Ordered. The chart note is also divided into four sections: (1) Patient History, (2) Physical Exam, (3) Diagnosis, (4) Tests Ordered, with a correspondence between the rubric item and the chart note sections. The system is required to check, for each rubric item, if it was covered by the corresponding section in the chart note. If yes, a credit of 1 point is assigned to the rubric item, otherwise 0. The case description helps only as better data for PPMI and RegExp stages in the entity linking algorithm.

Accordingly, we devise the following pipelined baseline. First, all rubric items and all chart note sections undergo standard preprocessing methods, particularly: (1) sentence tokenization, (2) word tokenization, (3) stopwords removal, (4) lowercasing, and (5) removal of non-alpha words. Handling counts is out of the scope of the baseline, although not ideal. Also, preprocessing methods are implemented using NLTK [102].

Operating on a word level, our baseline checks, for every word in the rubric item, if there is a semantically equivalent word for it in the corresponding chart note section. The algorithm relies on distributed word representations for a more flexible comparison than 1-hot vector representations. Let the rubric item be denoted as $R = r_1 r_2 ... r_k$, and corresponding chart note section be denoted as $c_1 c_2 ... c_m$, where k is the number of rubric item words, and m is the number of words in the corresponding chart note section. Then, every rubric item is given a score $score(R) = \sum_{i \leq k, j \leq m} \mathbb{1}\{cos(\vec{r_i}, \vec{c_j}) \geq v_t\}$, where $\vec{r_i}$ and $\vec{c_j}$ are the vector word representations of i^{th} and j^{th} word of the rubric item and chart note section respectively. We use off-the-shelf clinical word vector representations that were trained using FastText [103] on PubMed and MIMIC-III [104] with a dimension of 200. The term v_t is a tunable cosine similarity threshold above which two words are considered to be semantically close enough. Finally, a rubric item is given credit if its score crosses a tunable threshold r_t.

Table 4.1: Dataset statistics

Case Name	Chest Pain	Back Pain	Headache
Students	55	55	55
Rubric Items	36	30	42
Avg. Rubric Len	4.14	4.93	4.71
Avg. Note Sec. Len	218.99	231.49	213.98

One issue with the above vanilla system is that some words might be mistakenly missed such as "ECG" in the rubric item and "EKG" in the chart note. This is where entity linking helps. And so to assess the impact of our entity linking algorithm, after normalizing to a UMLS concept, we not only compare the word vectors of the original surface form, but we check if any of the atoms of the UMLS concept selected on the rubric side matches with any of the atoms of the UMLS concept selected at the chart note side. For entity linking, we use both our system and MetaMap, and compare in terms of effect on accuracy.

Data

We collect data on three medical cases from a collaborating healthcare institution. The cases were on: (1) Chest Pain, (2) Back Pain, and (3) Headache. The diversity of cases also helps in assessing the generalization of the method across cases. This is essential as the three cases at hand are not exhaustive to the list of medical cases to be delivered for grading in the future. Establishing good performance on all three cases ensures the generalization of methods. To have a better understanding of the dataset, we show statistics of the dataset in Table 4.1.

4.4.2 Results

To check the impact of our entity linking algorithm on the system's performance, we run the system on all three cases while manipulating the entity linking component into three conditions: no entity linking (Baseline), entity linking via MetaMap, and entity linking using our algorithm (Case-adaptive). We measure performance based on accuracy on a rubric item level. We choose

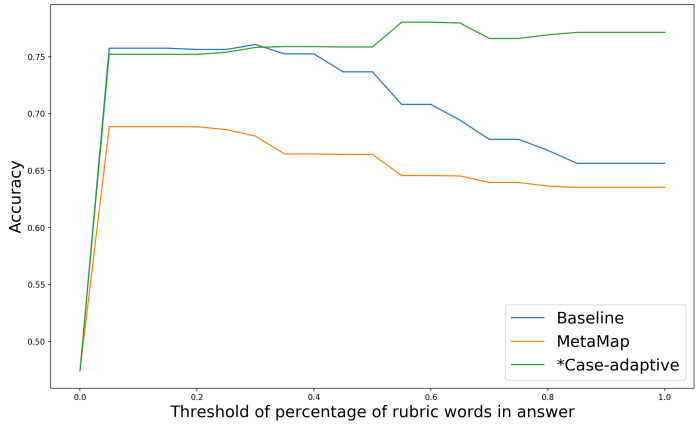

Figure 4.4: Effect of entity linking and UMLS integration on the Chest Pain case

Table 4.2: Examples of entity resolution by both systems on the Chest Pain case. The output of each system is indicated by the preferred name of the UMLS entity.

Ambiguous term	**UMLS**	**Non-ambiguous synonym**	**MetaMap**	**Case-Adaptive***
URI	Yes	Upper Respiratory Infections	N/A	Upper Respiratory Infections
MI	Yes	Myocardial Infarction	Myocardial Infarction ECG Assessment	Myocardial Infarction
CP	No	Chest Pain	Captopril	Chest Pain
SOB	Yes	Shortness of Breath	Dyspnea	Dyspnea
h/o	No	History of	History of	N/A
F	Yes	Female	Fluorides	Females
C	No	Cold	N/A	N/A
PMH	Yes	Past Medical History	Medical History	Medical History
WNL	No	Within Normal Limits	N/A	N/A
ACS	Yes	Acute Chest Syndrome	Acute Chest Syndrome	Activities

the best performing v_t, and check performance for all r_t values on all cases in Figures 4.4, 4.5, and 4.6. Our system significantly outperforms MetaMap on all three cases given its poor WSD system in a highly ambiguous setting, with MetaMap being non-adaptive to the different medical cases, but designed once for general use. We also notice across all three cases that the optimum performance with UMLS integration, using our system, is always at a higher r_t than without. This is because with UMLS integration, the chances for a random hit increase, and thus the need for a stricter r_t. In terms of absolute percentage increase in performance, our case-adaptive system increases performance over the vanilla baseline by 1.96%, 0.42%, and 2.44%, and over the MetaMap baseline by 9.18%, 8.02%, and 9.56% on Chest Pain, Back Pain, and Headache case respectively.

Since the datasets are not balanced across the binary labels 0/1, we also

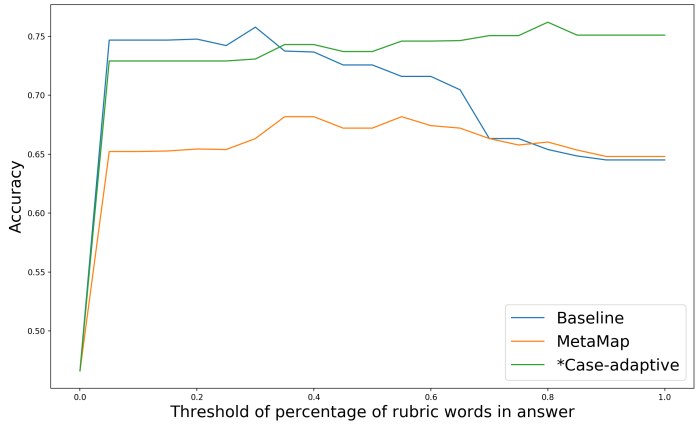

Figure 4.5: Effect of entity linking and UMLS integration on the Back Pain case

Table 4.3: Examples of entity resolution by both systems on the Back Pain case. The output of each system is indicated by the preferred name of the UMLS entity.

Ambiguous term	**UMLS**	**Non-ambiguous synonym**	**MetaMap**	**Case-Adaptive***
wk	Yes	Week	Week	Week
LE	No	Lower Extremity	LE, Rat Strain	LE, Rat Strain
SLR	No	Straight Leg Raise	N/A	Straight Leg Raise Test Response
PT	Yes	Patient	Physical Therapy	Present
MRI	Yes	Magnetic Resonance Imaging	Magnetic Resonance Imaging	Magnetic Resonance Imaging
FADIR	No	Flexion Adduction Internal Rotation	N/A	N/A
FABER	No	Flexion Abduction External Rotation	N/A	N/A
VS	No	Vital Signs	N/A	Patient Visit
CVA	No	Costovertebral Angle	Renal Angle Tenderness	Cyclophosphamide ...
CTA	Yes	Computed Tomography Angiography	PCYT1A wt Allele	Cancer/testis antigen

measure our system's performance for F-1 score and note a performance of 79.32%, 76.12%, and 70.52% on the Chest Pain, Back Pain, and Headache case respectively. This reflects that the F1 score and the accuracy measure are comparable for all cases except for a large drop for the headache case, showing that the system is not biased towards one label. Although this also shows that the headache case could be highly imbalanced towards positive instances.

Finally, to understand the performance of our case-adaptive system in comparison to MetaMap, we consider several examples of ambiguous abbreviations in Tables 4.2, 4.3, and 4.4. For every example, we indicate the ambiguous term, whether that term appears in the right concept in UMLS, the nonambiguous synonym of that term, the preferred name of the concept that MetaMap resolved the term to, and our system's output. In general,

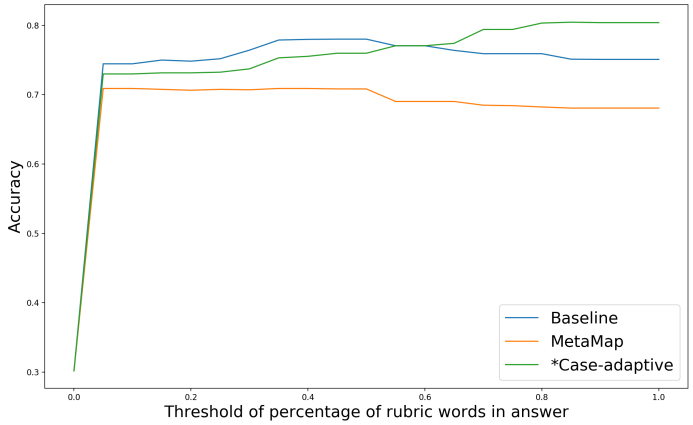

Figure 4.6: Effect of entity linking and UMLS integration on the Headache case

Table 4.4: Examples of entity resolution by both systems on the Headache case. The output of each system is indicated by the preferred name of the UMLS entity.

Ambiguous term	**UMLS**	**Non-ambiguous synonym**	**MetaMap**	**Case-Adaptive***
PMH	Yes	Past Medical History	Medical History	Medical History
HTN	Yes	Hypertension	Hypertensive Disease	Hypertensive Disease
R	Yes	Right	Right	Roentgen
SOB	Yes	Shortness of Breath	Dyspnea	Dyspnea
PPD	No	Packets Per Day	Purified protein derivative ...	Menstruation
FH	Yes	Family History	N/A	CFH wt Allele
MI	Yes	Myocardial Infarction	Myocardial Infarction ECG Assessment	Morning
DM	Yes	Diabetes Mellitus	Dextromethorphan	Diabetes Mellitus
UE	No	Upper Extremity	N/A	Upper Extremity
LE	No	Lower Extremity	LE, Rat Strain	HPS4 Gene

our system correctly resolves more examples (14) in comparison to MetaMap (10). More particularly, MetaMap does well on standardized abbreviations such as PMH, HTN, MRI, and SOB, but it cannot handle case-specific, non-standardized abbreviations such as CP, SLR, and UE.

Experiments conclude that an unsupervised entity linking algorithm is capable of adapting with the change of the medical case and not requiring any training data: two well-known obstacles in previous works. The entity linking system benefits from the positive impact of the algorithm on the downstream task of automatic grading of medical student chart notes, a task well motivated educationally and operationally.

CHAPTER 5

CONVERSATIONAL AGENT FOR MEDICAL ADHERENCE

5.1 Introduction

Another biomedical NLP field lacking the required training data is the field of medical conversational agents (CA). For CAs to hold an engaging conversation with the user, the CA needs to first understand the intent of the user. The training data in this setup would be examples of phrases to express a certain intent [13]. For example, for a conversational agent to understand that a user wants to hear the weather report, multiple example ways of asking for the weather need to be supplied to the conversational agent, such as: "How is the weather?", "What's the weather like today?", "Is it sunny today?", and ideally, many more. Such datasets, in general, are not available naturally, and are harder to find in the biomedical domain. Consequently, we pose the question, Can biomedical knowledge bases assist the process of collecting training data for CAs?

The particular biomedical application we consider for this CA is increasing medical adherence through the process of teachback [105]. One of the causes of low medical adherence, and consequently, low health levels, is a lack of understanding of the medical prescription itself [106]. Many conditions can prevent a patient from fully comprehending their medical prescription such as mental status at the time, cognitive overload with multiple medications to learn about, limited physician-patient time [107]. For that purpose, a recommended practice is to use the teachback method when teaching a patient about their medication at discharge time [108]. The teachback method requires the physician, or the nurse, to inform the patient of details about their medication in stages, and requires the patient to repeat the information back to the physician. The purpose of that repetition is to: (1) enhance retention, and (2) ensure accuracy of understanding. Due to resource constraints, this

teachback process tends to be overlooked, and the use of a CA is promising for this resource-constrained scenario. Motivated by this issue, we take on the task of developing a CA to assist in medical adherence by delivering a patient's medical adherence using the teachback method. Along the course of development, we study the potential of knowledge bases to alleviate the dataset sparsity issue, and assist in the training of the CA.

Owing to the requirements of teachback, the CA asks the patient a question to test their comprehension of an aspect of their medical prescription. One challenge in the development of the CA is how to use a technology intended to be user-initiated and utilize it for a CA-initiated conversation. Current uses of CAs tend to be a user asking a question, and the CA responding. To tackle this challenge, we design our own dialogue management logic, and rely on the present CA technology solely for assessing whether the patient uttered a paraphrase of the expected answer. In this setup, intents become answers, and intent training phrases become possible paraphrases of the required answer. Essentially, we utilize the CA technology solely for its paraphrase detection capabilities.

For the CA technology, we utilize Google's Dialogflow [13], which in turn requires a small set (recommended ≈ 10) of paraphrases for every intent/answer. Hence, we first collect a seed dataset of paraphrases for every intent. This allows for the development of a benchmark CA, and then we study the potential of a knowledge base to augment the initial dataset into a larger, more comprehensive dataset. Data augmentation is performed by first identifying UMLS concepts in the training phrases, retrieving their synonyms from UMLS, and augmenting the dataset by considering all the combinations of synonyms that could be used in the full sentence. For example, for a training phrase such as "I should take my medication orally", "medication" and "orally" can be identified, and replaced with their synonyms: "drug", "by mouth" respectively. We further use the seed dataset to evaluate the CA at correctly assessing the patients' answers before, and after data augmentation.

5.2 Previous Work

A recent interest in conversational agents for healthcare has resulted in exploring the potential conversational agents can have on a user's health. ran-

domized controlled trials (RCTs) showed that embodied agents can have a positive impact on, including but not limited to, a user's physical activity, dietary habits, comprehension of health data [109, 110, 111, 112]. Despite the apparent potential, this field is far from mature. Most work identified and mentioned here dates after 2010, with most lacking RCTs [113].

As mentioned earlier, the majority of previous work has found value in an embodied agent [114, 115, 116, 117, 118], rather than a chat bot [119, 120]. Among the embodied agents, four of them [114, 115, 116, 118] focus on a specific task, and control the dialogue in a rule based mechanism, similar to our work, while the other uses a frame-based dialogue management system, and does not focus on a specific task.

The first embodied agent [114] delivers social skills training to people with autism spectrum disorders. To assess the performance of the patients and perform feedback, it relies on predefined acoustic, linguistic, and visual cues, and does not rely on a labeled training set, as ours. The second embodied agent [115] diagnoses patients whether they suffer from major depressive disorders (MDD) or not. The agent in this case solicits input but offers no feedback. The input is then used to assess whether the patient suffers from MDD according to predefined diagnostic criteria. Thirdly, the work in [116] developed an agent to diagnose excessive daytime sleepiness. To perform the diagnosis, the agent asks a series of rating questions thus severely limiting the user's possible utterances. The answers are then used to generate a predefined score for the patient. Finally, the work in [118] utilizes conversational agents to allow for anonymous self-reporting of symptoms that patients would rather not disclose otherwise. The example in this study was veterans finding more comfort in reporting to a virtual agent rather than an actual physician. In this study, although an agent is used to solicit input from war veterans, the agent did no NLP processing and offered no instantaneous feedback, but rather used human coders to analyze the input of the user post-study. Despite the potential of conversational agents in healthcare, most agents do not rely on supervised NLP possibly due to lack of training data.

5.3 System Setup

The communication efficiency of a CA does not solely rely on the quality of the underlying NLP technology. For a user to smoothly communicate with a CA, they need to feel that the conversation is as natural as possible. For that reason, users, and especially older patients, prefer vocal communication over a textual communication, and prefer communicating with an embodied agent rather than just a sound [121].

To achieve such a vocalized embodied agent, an NLP component communicates with a speech component and a visual component to deliver the teachback process to the patient. The flow of communication, from the patient to the submodules and back to the patient, is depicted in Figure 5.1. Step (1) has the CA vocally deliver a piece of information (referred to as a frame) about the prescribed medicine, followed by a question to the patient about the delivered frame. Table 5.1 enumerates, with examples, all the frames that the CA delivers. In step (2), the patient answers the question soliciting a repetition of the frame. In (3) the speech recognition module transcribes the utterance of the patient into text and feeds it into the NLP component, which assesses the patient's response, and generates the CA's textual response (4). The CA's response is either an affirmation of the patient's answer and a delivery of the next frame, or an identification a wrong answer, delivering the frame again, and repeating the question. The speech generation module takes the textual response and generates the audio response (5). The visual renderer takes the audio response as an input, and generates the embodied response, animating the agent's body and lips, in an appropriate background.

The embodied agent is depicted in Figure 5.2. Following the recommendation of a human subjects study [121], the agent is embodied as a female rather than a male, and an older female rather than a younger female. The combination of an older female established the most trust with the human subjects, which is an important aspect of health-related communication. Furthermore, the agent is placed in a professional setup to communicate credibility.

Off-the-shelf tools are adapted to our task at hand for all 3 modules: NLP, speech, and visual. For the visual module, we use the Virtual Human Toolkit [122]. For speech, we use the Kaldi toolkit [123]. And for NLP, we utilize Google Dialogflow [13], Google's engine for conversational agents. Focusing

Table 5.1: Medication information segmented to frames of information

Frame	Text
Name	Your medication is called Metformin.
Purpose	It is used to treat type 2 diabetes by helping to control the amount of sugar in your blood.
Benefits	If you take Metformin as directed, you will have a better blood sugar. A better blood sugar helps to keep your heart, eyes, kidneys, and blood vessels healthy.
Warnings	Be sure to follow all exercise and diet recommendations from your doctor or dietitian. It is important to eat a healthy diet.
Dose	Take one tablet of Metformin by mouth.
Frequency	Take your medicine two times a day, with breakfast and with dinner. Swallow the table whole.
Duration	Continue to take Metformin even if you feel well. Do not stop taking it without talking to your doctor.
Missed Dose	If you forget to take your Metformin on time, take it as soon as you can. However, if it is almost time for the next dose, skip the missed dose and continue your regular schedule. Do not double dose to make up for a missed one.
Side Effects	Some side effects are expected, but talk to your doctor if you notice diarrhea or metallic taste. Other side effects can be serious. If you experience any of these symptoms, call your doctor immediately or get emergency treatment: itching or hives, swelling in face, hands, or mouth, stomach pain, or trouble breathing.

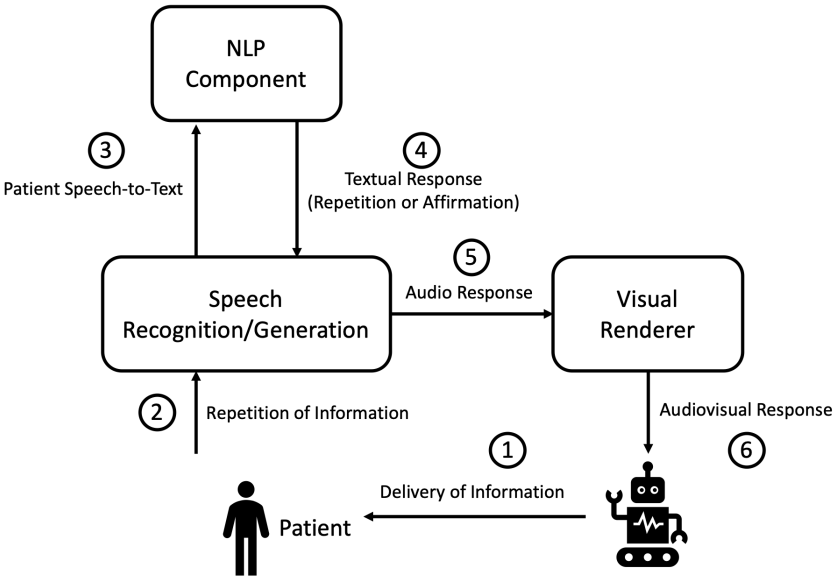

Figure 5.1: Block diagram of Health EdVisor pipeline

Figure 5.2: Appearance and setup of Edna

on the NLP component, we detail next the training procedure of the NLP module, and the data collection process.

5.3.1 NLP Component

The NLP component is responsible for intelligently assessing the quality of patients' answers by assigning them a binary label $\in \{correct, wrong\}$. To perform this assessment, we model it as a paraphrase detection problem, in which we have a set of reference answers and we assess whether the input utterance by the user is a paraphrase of the reference answers or not. In this setup, utterances are assessed by their semantic similarity to a set of correct answers. Nevertheless, a wrong answer (such as missing a negation) can still have a high semantic similarity with the reference answers. To avoid such false positives, we further provide sample wrong answers. In this updated setup, an input answer is assessed by comparing it to a reference of correct and wrong answers simultaneously. An answer is predicted as wrong if it did not match any of the classes, or if it matched the class of wrong answers. The wrong answers are designed to be highly adversarial by including a significant amount of lexical overlap with the correct answers for the CA to be able to identify wrong answers with high overlap.

Accordingly, for every frame, a set of correct and wrong answers is collected and fed to Google Dialogflow for training. A separate paraphrase detection module is trained for every frame since the agent, at runtime, is aware which frame the patient is answering about, and thus reduces the hypothesis set to the correct and wrong label of the respective frame.

Data Collection

An essential component of this supervised setup is the phrases for each class $\{correct, wrong\}$ of each frame. Accordingly, for each frame, seven annotators were given: (1) the information that would be given to the patient, and (2) the question that would be asked as a followup to check the comprehension of the patient. Based on this information, each annotator was asked to provide 10 examples of correct answers, and 10 examples of wrong answers. Furthermore, the annotators were asked to divide their 10 examples

Table 5.2: Example frame (purpose) along with its training phrases for every class.

Frame	It is used to treat type 2 diabetes by helping to control the amount of sugar in your blood.
Question	How does your medication treat diabetes.
Correct	Maintains sugar level.
Wrong-Adversarial	It increases the amount of sugar in my blood.
Wrong-Easy	It controls something.

of wrong answers to 5 wrong answers with high lexical overlap to the delivered information, and 5 wrong examples with low overlap to the delivered information. The purpose of this distinction is to understand the behavior of the system under different levels of adversary, as well as provide the system with a diverse set of wrong answers.

The final output of the collection process was 140 examples per frame (70 correct, 35 wrong with high lexical overlap, 35 wrong with low lexical overlap). This was done for all 9 frames. After the collection process, the data was divided equally into training (development) and a held-out dataset for final testing. The division of the data was ensured to be balanced between annotators, frames, and classes.

The distinction between wrong instances with high overlap and wrong instances with low overlap is only of significance when sampling the training data. Such a distinction helps ensure a balance for the *wrong* class in terms of examples high and low in lexical overlap. Table 5.2 includes an example frame, purpose in this case, the associated question, and an example training phrase for every class of aforementioned answers.

UMLS Integration

The collected dataset is limited in size, and its coverage of the different surface forms an answer could take could be enhanced. We propose the use of a biomedical ontology, such as the UMLS, to generate extra data points using the seed set of examples. The method is to retrieve biomedical phrases in the original seed set, link them to a concept in UMLS, retrieve the synonyms (atoms) of the concept, and replace them in the original example to generate new examples. For example, for an answer such as "Maintains sugar level." the word sugar is linked to the "glucose" concept in UMLS since

sugar is an atom in that concept. Then, "glucose" is identified as a synonym of "sugar", and a new data point "Maintains glucose level." is added to the original dataset of correct answers for the Purpose frame.

Let $C = \{C_1, C_2, ..., C_M\}$ be the set of M concepts in UMLS, and let $A_i = \{a_{i1}, a_{i2}, ..., a_{iN}\}$ be the set of atoms (synonym set) for concept i. Then, for every training phrase $= \{w_1, w_2, ..., w_P\}$, every word w_k in the phrase of length P is checked against the UMLS concepts C. A word w_k is said to be part of a concept C_i if $w_k \in A_i$. Then, assuming an input phrase $\{w_1, ..., w_k, ..., w_P\}$, new training phrases $\{w_1, ..., a_{ij}, ..., w_P\}, \forall a_{ij} \in A_i$ are generated and added to the original seed set of training phrases. Note that one word could be mapped to multiple UMLS concepts, and this word-level definition of the UMLS-integration is extended to phrases by considering n-grams up to size 3.

5.4 Results

The experiments are designed to answer two main questions: (1) Does UMLS integration lead to a more accurate system? and (2) Is the impact of UMLS integration magnified under a low-resource setting? Accordingly, the experiments evaluate the accuracy, precision, and recall of the system while varying two components: (1) whether UMLS is used to augment the data or not, and (2) the size of the initial training phrases (20% or 100%).

Examining the results in Table 5.3, we conclude that when using 100% of the training phrases as a seed set, using UMLS for data augmentation did not have a positive impact on the F1 score, or accuracy. Suspecting that possibly the seed set is exhaustive enough, leaving little space for UMLS to improve, we check the impact of UMLS integration when only 20% of the initial dataset is used. Despite the evident drop in performance, and increased potential for improvement, an insignificant impact of UMLS-integration is observed, confirming that the designed UMLS integration does not boost CA accuracy. Examples detailed later answer why UMLS-integration had no impact on performance in this case.

Next, we examine the performance of the NLP module in more detail through the confusion matrix in Figure 5.3. We observe that adversarial examples with high lexical overlap are mislabeled more often (59 times) than

Table 5.3: Performance of Health EdVisor at assessing accuracy of answers after varying percentage of data used (20%, 100%) as well as whether UMLS was injected or not

UMLS-Augmented	Data	Acc	P	R	F1
Yes	20%	65.94	63.26	60.58	61.89
No	20%	65.70	62.91	60.58	61.73
Yes	100%	79.23	74.64	82.54	78.39
No	100%	80.19	76.62	81.48	78.97

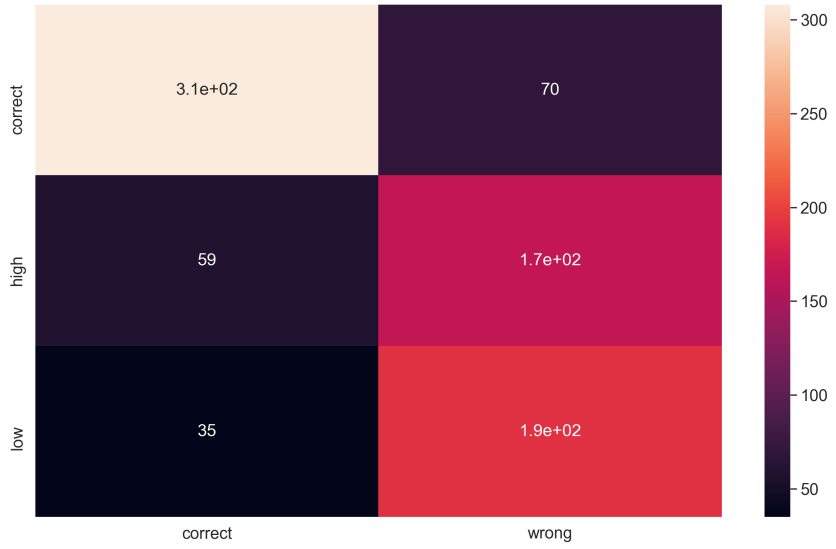

Figure 5.3: Confusion matrix for Health EdVisor assessment

their low lexical overlap counterpart (35 times), as expected by design.

Next, looking at specific examples, in Table 5.4, of different type of errors and accurate predictions, we can understand better the functionality of the system. The first four examples pertain to the "Benefits" frame where the user is told that "If you take Metformin as directed, you will have a better blood sugar. A better blood sugar helps to keep your heart, eyes, kidneys, and blood vessels healthy". In the first example, the user flips the direction of two words, "helps" to "prevents" and "keep" to "damage" resulting in the same intended meaning. The CA is still capable of assessing as it as a correct answer due to the training data containing similar example paraphrases. The second example shows how the system is not capable of handling negation

in the case of high overlap of words. The third example shows some of the characteristics of the data at hand. The input utterance in the third example is labeled originally as "wrong" although one can argue that the inaccuracy in the statement comes from the name of the medication and not from the frame being tested: "Benefits". This example shows the high adversary level of the dataset where only word is changed, and the rest overlap with the input utterance. The fourth example illustrates good handling of wrong answers of high overlap with given information, again just due to presence of similar adversarial examples in the negative class. This reflects the importance of collecting negative instances for training. Finally, the last example shows the incapability of the system in handling numbers. In conclusion, most of the errors originate from incapability of handling negation and numbers, rather than synonymy. This explains the lack of impact of UMLS on the system.

Table 5.4: Error analysis of Health EdVisor

Gold Label	Correct
Predicted Label	Correct
Input Utterance	It prevents the damage of my heart of my kidneys.
Trigger	My heart, lungs, and kidneys might fail otherwise.
Gold Label	Correct
Predicted Label	Wrong
Input Utterance	My metformin makes my eyes healthy.
Trigger	The medication will not keep my heart, eyes, kidneys and blood vessels healthy.
Gold Label	Wrong
Predicted Label	Correct
Input Utterance	If you take Metamorphosis as directed, you will have a better blood sugar. A better blood sugar helps to keep your heart, eyes, kidneys, and blood vessels healthy.
Trigger	Metformin results in better blood sugar. A better blood sugar helps to keep your heart, eyes, kidneys, and blood vessels healthy.
Gold Label	Wrong
Predicted Label	Wrong
Input Utterance	It helps to keep my eyes, blood vessels, heart and kidneys healthy by increasing my blood sugar.
Trigger	The medication will not keep my heart, eyes, kidneys and blood vessels healthy.
Gold Label	Correct
Predicted Label	Wrong
Input Utterance	One by mouth.
Trigger	Two tablets by mouth.

CHAPTER 6

AMBIGUITY IN BIOMEDICAL NLP AND WORD SENSE DISAMBIGUATION

While experimenting with integrating knowledge bases into NLP biomedical algorithms, one recurrent issue is the ambiguity of the biomedical text. To retrieve knowledge from a knowledge base pertaining to a word or phrase in the text, a system needs to first correctly identify the concept it links to in the knowledge base. This would have been trivial were it not for ambiguity of words and language in general. Given the high level of ambiguity of biomedical text, we take a step back and explore the potential of current contextualized word representations in assisting Word Sense Disambiguation and Entity Linking.

6.1 Introduction

Biomedical text tends to be highly ambiguous in nature [124]. For example, a term such as AA could refer either to the concept of Amino Acids or to the concept of Alcoholics Anonymous. To quantify the level of ambiguity in biomedical text, we take machine reading comprehension (MRC) as an example task and examine the ambiguity level in one of its datasets: BioASQ [125]. As shown in Figure 6.1, terms in BioASQ tend to belong to more than one concept in the Unified Medical Language System (UMLS) [10]. This presents a challenge not only for MRC, but other downstream natural language processing applications operating on biomedical text, such as relation extraction [126], and so on. To address the challenge, several systems have been proposed ranging from rule-based methods [127] to neural methods [128, 129]. With advances in contextualized pre-trained word representations [95, 97], and the impact they have had on downstream applications, we explore the utility of a recent contextualized biomedical word representation, namely BioBERT [14], for the task of biomedical word sense

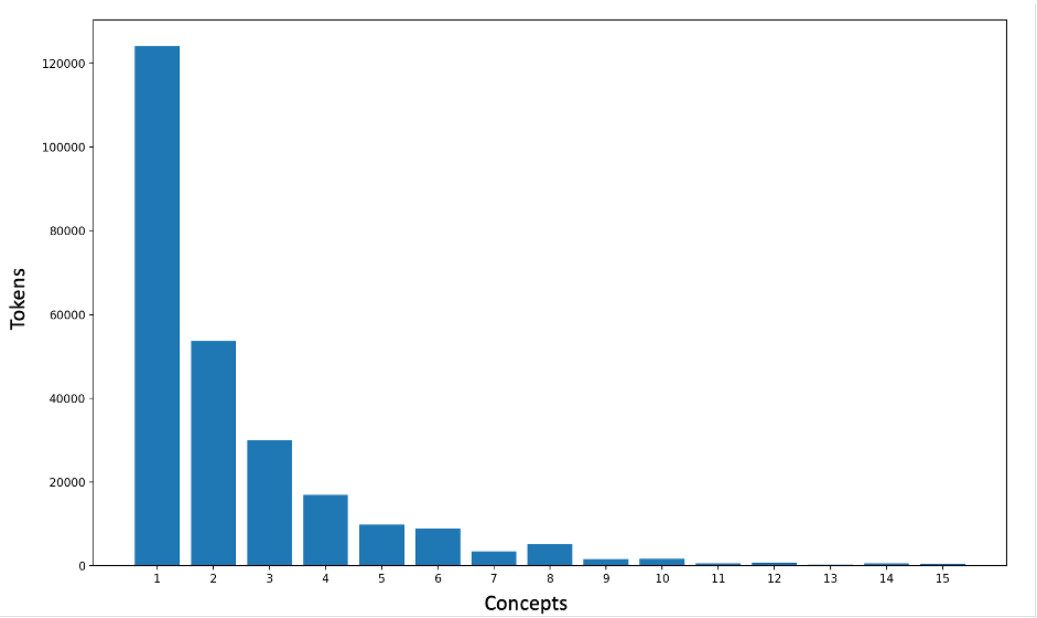

Figure 6.1: Distribution of tokens based on number of UMLS concepts they belong to

disambiguation (WSD).

Although one biomedical term could be ambiguous and refer to multiple concepts, humans utilize the context an ambiguous term appears in to disambiguate it to the correct concept. Similarly, previous WSD methods, most notably deepBioWSD [129], resort to modeling the context to perform predictions. Nonetheless, these methods rely on static word representations, which cannot model the change in semantics of a word based on the sense it takes in that particular context. With recent advances in contextualized word representations, and more particularly, biomedical ones represented by BioBERT, the first question we pose is understanding the capacity of BioBERT at capturing the semantics of the different senses of an ambiguous phrase, and accordingly aid WSD. The advancements of contextualized word representations and their ability to model the semantics of a term given its context, positions BioBERT favorably for the task of word sense disambiguation in general, and biomedical word sense disambiguation in particular.

Another obstacle facing biomedical WSD is availability of training data given the ever-expanding list of ambiguous technical terms, and the expensive manual labor [130, 131]. For biomedical WSD to become practical, it is essential to automate the process of data collection, assuming supervised

Table 6.1: Context characteristics of ambiguous term based on positional order

Occurrence	Example Context
First	... prefeeding on prececal amino acid (AA) digestibility of ...
Second	... and digested amounts of AA were determined ...

methods. Self-supervision has been found to be a more realistic and scalable avenue for machine learning applications, rather than relying on human-annotated corpora. Accordingly, we follow methods suggested by previous works, and utilize UMLS and PubMed to automatically create a sense-tagged corpus, and study its quality as a training set [129].

Our study also explores the influence of the positional order of an ambiguous term on the challenge of disambiguating an ambiguous term. As authors delve deeper into their document, they assume a higher level of reader's understanding of the context, and exhibit less explicit context in their writing. For example, when an ambiguous acronym is first introduced, authors tend to explicitly precede it with its expanded unambiguous form, and omit that later on, as in Table 6.1. We hypothesize that WSD models should take that into account when training, as well as evaluation. One of the popular biomedical WSD datasets does not account for that aspect and labels only the first occurrences of an ambiguous term in abstract. We claim that that leads to mismatch in modeling for real world applications on terms that appear later in the text. We also claim that evaluating on these less challenging occurrences overestimates the performance of biomedical WSD algorithms. We further recommend alternative training/evaluation setups.

Finally, we distinguish between two types of ambiguous terms: (1) with related candidate senses, and (2) with unrelated candidate senses. For example, "Alcoholics Anonymous" and "Amino Acids" are two unrelated senses of the ambiguous term "AA", and thus, "AA" belongs to the second type. In contrast, a term such as "Yellow Fever" belongs to the first type since it can either mean the disease or its vaccine: two related senses. This distinction is important as a mistake on the first type is expected to have less of a negative effect on a downstream task, and so measuring the isolated performance of a WSD system on each type can give a better sense of the impact of the WSD system on downstream tasks.

In summary, in this chapter, we answer the following questions:

1. What is the potential of BioBERT in biomedical WSD?
2. What is the efficacy of self-supervised methods at generating training data?
3. What is the sensitivity of the WSD systems to the positional order of the ambiguous term?
4. How does the relatedness of candidate concepts affect the performance of biomedical WSD systems?

6.2 Previous Work

Several approaches to Biomedical WSD have been proposed and evaluated in the past. These methods primarily differ in the amount and nature of supervision.

One set of approaches, especially in earlier years, utilized knowledge-bases such as UMLS and MEDLINE to perform biomedical WSD [132, 133]. In [132], for example, authors also explore the potential of automatically created sense-tagged corpora using a thesaurus (UMLS) and a corpus of abstracts (MEDLINE). Given the automatically collected supervised data, naïve Bayes was used to perform sense classification. We find it important to revisit the potential of automatically curated datasets given the advances in machine learning algorithms and representations.

Another set of approaches relies on semi-supervised learning [124, 134, 135]. For example, in [134], authors first use unsupervised methods to cluster different occurrences of an ambiguous term into unlabeled clusters representing the different senses. This represents the profile of an ambiguous term. Then a human annotator would annotate and verify the different clusters. These clusters are then used for supervised methods, thus requiring less human input. Along the same lines, active learning approaches have been proposed to minimize human input [131].

Finally, the last set of approaches belong to the supervised learning framework [136, 137, 138, 129], with the closest work to ours being that of deepBioWSD [129]. Not only do they rely on modeling word representations from

unannotated text, they further enrich their representations with information from the UMLS. On top of these representations they developed a deep bi-directional LSTM network to perform WSD. We offer a simpler, yet competitive, method and architecture to perform WSD by utilizing the context modeled in the biomedical pre-trained word representations of BioBERT, which has not been explored in the aforementioned previous work [136, 137, 138, 129].

6.3 Materials

Following are the materials used for this study. We use pre-trained BioBERT embeddings as a basic building block of our WSD framework. We utilize the MSH WSD data set [139] for training and evaluation. Finally, for the unsupervised setup, we describe our automatically collected training dataset.

6.3.1 BioBERT

Distributed word representations have shown their capabilities at capturing the semantics of words and phrases, but their static versions would assign equal representations to different senses of the same phrase [140, 141]. Contextualized word representations, on the other hand, were developed to address this issue, and carry sense information by encoding the context in the word representation. Recently, BERT has been found useful in a variety of NLU tasks, and an essential component in our WSD framework is a contextualized representation of biomedical terms.

Building on the success of BERT, authors in [14] trained and released a biomedical version of BERT representation: BioBERT. They utilized the architecture, as well as the pretrained representations, of BERT, and resumed its training on biomedical scientific articles: Pubmed (4.5B words), and PMC (13.5B words). With a simple feedforward layer on top of these representations, they were able to advance the state-of-the-art on several downstream biomedical NLP applications: named entity recognition [142], relation extraction [143], and question answering [125]. This reflects the richness of BioBERT representation in encoding the semantics of the word and its context. BioBERT representations were made available here: https://github.com/dmis-

Table 6.2: MSH WSD dataset statistics

Count of ambiguous terms	203
Abbreviations	102
Abbreviation-Word combination	13
Non-abbreviated words	88
Average count of senses per term	2.08
Average count of abstracts per sense	89.57
Average count of words per abstract	200.38
Avgerage % of majority sense	54.2%

lab/biobert, and we use version BioBERT-Base v1.1 (+ PubMed 1M).

6.3.2 MSH WSD Data set

To train supervised WSD frameworks, we need a corpus of biomedical text with annotations of polysemous biomedical phrases being linked to their respective senses. A common practice in biomedical NLP is to assign the sense as a concept in UMLS. For example, AA in the sense of amino acids would be assigned the UMLS concept ID: C0002520, and AA in the alcoholics anonymous sense would be assigned the UMLS ID: C0001972.

One popular available dataset, due to its size and term diversity, is the MSH WSD Data Set. This dataset provides training instances for 203 ambiguous terms frequently occurring in biomedical text. The text is collected from the title and abstract of 37,090 MEDLINE citations. For every term, 2 (most cases) to 5 candidate senses are pre-determined, and for every sense, a maximum of 100 instances are provided. In each instance, only the first occurrence of the ambiguous word is annotated with the accurate sense (UMLS ID). For further statistics on the dataset, the reader is referred to Table 6.2. The count and size of abstracts were found sufficient for training purposes. Table 6.2 also reflects the balance in the dataset with the majority sense covering only 54.2% of the cases. Also, the dataset covers a variety of ambiguous terms between those that are ambiguous due to abbreviations overlapping with other abbreviations (102), abbreviations overlapping with full words (13), or full words that are inherently ambiguous (88).

6.4 Methods

The experiments we perform are targeted towards answering the individual research questions.

6.4.1 BioBERT for Biomedical WSD

With BioBERT's capacity at modeling contextual information, and comparing to previous complicated architectures for biomedical WSD [129], we take a simple approach following the success of BERT and BioBERT, which found that a single hidden layer feedforward neural network is sufficient to perform classification on top of the rich contextualized representations. Both works of BERT and BioBERT have only utilized such a simple prediction layer and achieved significant boosts in performance. The underlying transformer network and the masked language model training allow BioBERT to explicitly encode semantic information of the context on both sides of the term into the distributed representation of the center word. Moreover, its parallelizable architecture allows for training on larger amounts of unannotated biomedical text. Accordingly, a simple single hidden layer neural network is capable of performing word sense disambiguation given the contextual information. Another motivation behind the simple classification layer is scalability. This simplifies the process of learning to disambiguate new ambiguous words, as training a 1-layer neural network for that word is computationally cheap and fast. A separate multi-class classification layer is trained for every ambiguous term.

Contrary to the more general setting of entity linking [144], in WSD, the goal is to provide a sense for an ambiguous term with identified boundaries. Hence, what remains is how to aggregate the BioBERT's subword representations of the ambiguous phrase at hand to perform predictions. We follow the most straightforward approach of averaging the subword representations to form the phrase representations. BioBERT's operation on a subword level helps address the issue of unseen and rare words. We further explore alternative methods of aggregating context by considering the full representation of the sentence an ambiguous phrase appears in, and even the full abstract. Towards that end, we consider three setups before feeding the representations into the neural network: (1) aggregating only the subwords of the ambiguous

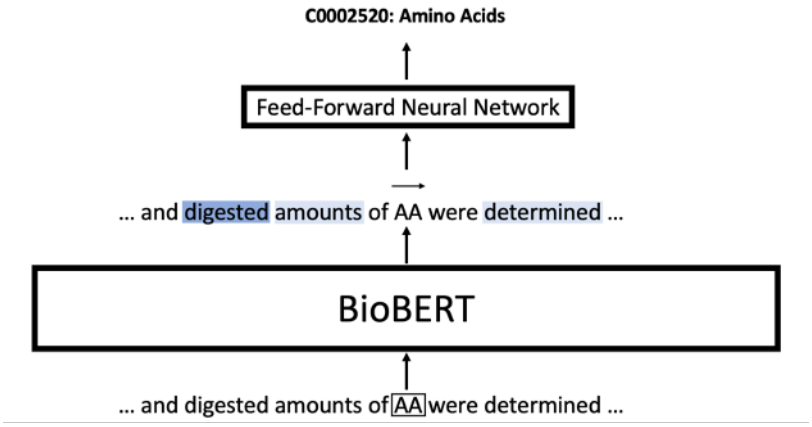

Figure 6.2: Block diagram of WSD framework. The highlighting reflects the contribution of context words to the representation of the center word.

phrase, (2) aggregating the subwords of the full sentence, and (3) aggregating the subwords of all the abstract. The first setup assumes that all required information to perform disambiguation is present in the representations of the phrase itself. In other words, it assumes a perfect operation of BioBERT of encoding the context information into the local representation. The second setup considers the possibility of semantic information present in the rest of the sentence not encoded in the local representation itself. Finally, and since the attention of BioBERT representations is limited to the context of the sentence, the third setup posits that helpful semantic information could be present outside the sentence of the ambiguous phrase, and includes that in the final representation to be fed to the neural network. Figure 6.2 illustrates the different building blocks of the WSD framework in its phrasal setup.

6.4.2 Self-Supervision

As previously mentioned in the materials section, the MSH WSD dataset is limited to 203 phrases, and any system trained on it can only perform WSD on these 203 phrases. This is a strong limitation, which we attempt to address, given the highly ambiguous nature of biomedical phrases as shown in the introduction. Another limitation of the above dataset, although we do not address it in this chapter, is the non-exhaustive pre-determined set of candidate senses. Accordingly, we follow in the footsteps of [129] and study the feasibility of automatically collecting a WSD dataset.

The automatic collection process assumes that for every occurrence of an ambiguous word, another surface form (polyonomy), which is unambiguous, of the same concept appears in the text as well. For example, if AA in the amino acids sense appeared in a scientific article, this method assumes that the phrase "amino acids" would also appear in the same text. This is an acceptable assumption to have in the scientific domain given the standard of using the full form of a concept before truncating it, yet this assumption does not always hold, as we detail later in the results section. One issue with the method described in [129] is that they look for the nonambiguous synonym and replace that synonym with the ambiguous phrase. This could create unnatural text such as "... amino acids (AA) ..." being mapped to "... AA (AA) ...", and thus a better alternative is to look for the ambiguous phrase instead of synthetically inserting it.

6.4.3 Sensitivity to Positional Order

We observed that in the MSH WSD data set the annotated ambiguous phrase is always the first instance of that phrase in the abstract. More importantly, that ambiguous phrase was highly likely to appear in the close vicinity of the nonambiguous form of the sense, as is the standard in scientific writing. For example, for an abstract about amino acids, it was highly likely that "AA" appeared for the first time in parentheses following the expanded form "amino acids". We hypothesize that to train and evaluate WSD systems on such trivial examples, first, overestimates the optimal performance of the WSD system in general, and second, would lead to deteriorated performance levels once tested on real examples where that triviality is not guaranteed, and even more, not expected.

6.4.4 Effect of Relatedness

Another aspect we noticed of the MSH WSD data set, and of WSD in general, is the range in difficulty of resolving ambiguous biomedical phrases. More particularly, we expect that ambiguous phrases where the candidate senses are related, are harder to resolve than ambiguous phrases where the candidate senses are unrelated. This stems from the fact that related senses

will appear in closer semantics of the context, and thus less distinct to disambiguate trivially. For example, a term such as AA should be easy to disambiguate since the candidate senses "alcoholics anonymous" and "amino acids" are semantically distant, or in other words unrelated. In contrast, an ambiguous phrase such as "Yellow Fever" with the candidate senses "Yellow fever disease" and "Yellow fever vaccine" is harder to disambiguate due to the high relation between the two candidate sense, one being the vaccine to the other.

6.5 Experimental Setup

6.5.1 BioBERT for Biomedical WSD

To build our system, we utilize Python's scikit-learn package [145], and more particularly, the multi-layer perceptron with a single layer of 200 hidden units and a ReLU activation function [146]. Increasing the number of hidden units beyond 200 did not benefit performance. Phrase-level, sentence-level, and abstract level representations were pre-computed for all experiments using the BioBERT source code. The WSD framework is evaluated in two settings. In the first setting, we take a supervised approach and randomly split the instances from all senses of all ambiguous phrases into an 70-10-20 split, with 70% used for training, 10% used for development, and 20% used for evaluation. After the dev set guided architecture and hyperparameter decisions, a separate neural layer is trained for every ambiguous phrase using the 80% (train + dev) split of each sense belonging to that phrase, and then evaluated on the rest of the instances. Given the computational simplicity of the classification layer, it is inexpensive to train separate layers, and offers modularity for training on newly introduced ambiguous terms. In the second setting, which is unsupervised, the automatically collected dataset is used solely for training and all of the MSH WSD data set is used for evaluation. These two settings are compared against the results of deepBioWSD as a baseline [129].

6.5.2 Self-Supervision

To perform the automatic collection of the training data, for every sense of every ambiguous phrase, we use UMLS to look for a non-ambiguous synonym. A non-ambiguous synonym is defined as a phrase of a UMLS concept that does not participate in any other concept. Using that non-ambiguous synonym, we look for Pubmed abstracts via the Entrez-Direct tool [147]. Then, in every abstract returned as a result of a query of the non-ambiguous form, we look for an exact match of the ambiguous form in the text, and annotate its sense accordingly. We limit the number of retrieved abstracts to 500 per sense.

6.5.3 Sensitivity to Positional Order

To test our hypothesis on the effect of positional order, we trained two systems: one on the first occurrences of an ambiguous phrase, and another on the second occurrences of an ambiguous phrase. We also had two settings for evaluation, one evaluating disambiguation on the first occurrences of the ambiguous phrase, and the other evaluating on the second occurrences. We claim that the second setting of evaluation is more representative of the real world setting where the expanded form of the ambiguous phrase is not expected to be in the close vicinity of the ambiguous phrase. We also expect that when the training setup matches the evaluation setup, which in this case translates to training being on the second occurrences when evaluating on second occurrences, the performance would increase.

6.5.4 Effect of Relatedness

To validate the impact of relatedness, we divided the MSH WSD data set phrases into ambiguous phrases with related candidates (at least 2 related), and phrases with unrelated candidate. To perform this division of phrases into these two categories, we relied on UMLS, that identifies whether a relation exists between two concepts (senses), or not. Given the scope of this work being limited to word sense disambiguation, and assuming the word boundaries of the ambiguous term to be given, we rely on accuracy as a metric.

Table 6.3: Performance of WSD framework with respect to different aggregation levels of BioBERT representations

Aggregation	**Phrase-Level**	**Sentence-Level**	**Abstract-Level**
Accuracy	93.82%	91.06%	91.07%

6.6 Results and Discussion

6.6.1 BioBERT for Biomedical WSD

Considering first the comparative performance of our WSD framework according to the different levels of aggregation, we notice in Table 6.3 that aggregation on the phrasal level performs the best by a 2.75% margin. This reflects BioBERT's ability at capturing contextual information into the phrase at hand, without requiring the inclusion of features beyond the phrase boundaries. Moreover, these positive results of BioBERT add to the successes of BioBERT on other BioNLP tasks such as Relation Extraction and Question Answering.

6.6.2 Self-Supervision

Next, we analyze the performance of our system in the unsupervised setting, and compare its performance in both the supervised and the supervised setting to the recent work of deepBioWSD in Table 6.4. In terms of comparing BioBERT to deepBioWSD, we notice a 3% gap in performance, but taking into account the simplicity of the approach and the complexity of the deepBioWSD network, a comparable performance by BioBERT reflects the quality of the contextual information present in the BioBERT representations. When switching to the unsupervised setting, we notice a significant drop in performance. Since BioBERT representations have already proven capable of encoding contextual information, and the neural layer capable of performing predictions, it can only be that the dataset collected is noisy, and we explore that next. Nevertheless, the results are promising for building a generalizable biomedical WSD system that goes beyond the human labeled data.

Upon further investigation, we detected the source of noise in the auto-

Table 6.4: Performance of the different systems on the MSH WSD dataset

System	Accuracy
BioBERT - Self Supervised	84.02%
BioBERT	93.82%
deepBioWSD	96.82%

Table 6.5: Effect of order of occurrence on training and evaluation

Training Setup	Evaluation Setup	Accuracy
First occurrences	First occurrences	93.82%
First occurrences	Second occurrences	88.41%
Second occurrences	Second occurrences	91.71%

matically collected dataset back to the initial assumptions not always holding. One of the assumptions is that if the ambiguous synonym appears in the same abstract as the nonambiguous synonym, then both have the same sense. It turns out that does not always hold true. For example, the ambiguous phrase "CH", which in the MSH WSD data set could either mean China or Switzerland, appears in the sense of "Methylene" in our automatically collected corpus. In other words, several examples included studies performed in China or Switzerland and included CH (Methylene) as a substance, and thus CH was labeled wrongly in these abstracts as either China or Switzerland.

6.6.3 Sensitivity to Positional Order

As shown in Table 6.5, when comparing the first row of results to the second, the decrease in performance when evaluating systems trained on the first occurrences on second occurrences instead of first, reflects the overestimation of the performance of the systems trained and evaluated on the MSH WSD data set. As for comparing the second row to the third row, the jump in performance when adjusting the training setup to match the real world emulating evaluation shows what is better recommended as training procedure when training WSD systems for real world examples.

Hence, for future research on WSD in general, and the MSH WSD data set in particular, we recommend having the annotated instances to be the occurrences later than the first to match what is expected to be encountered

by WSD systems in downstream tasks.

6.6.4 Effect of Relatedness

After dividing the ambiguous terms into those that have related candidate senses and those that have unrelated candidate senses, the performance of our system on the phrases with related candidates was 78.45%, whereas the performance on those with unrelated candidates was 96.34%. This large disparity in performance first reflects the intuition that those with unrelated candidate senses should be easier to resolve due to the highly distinct contexts the candidates appear in. More importantly, this result emphasizes the capacity of the WSD systems to perform significantly better on the cases that matter more. Mistaking a sense for a related sense is expected to have less impact on downstream tasks than mistaking a sense for an unrelated concept.

We conclude from our experimental results that, first, BioBERT representations contain a high level of semantic contextual information that can significantly aid biomedical word sense disambiguation. Second, self-supervised methods to automatically create training date are feasible, yet noisy. Third, and most importantly, we identify that WSD systems are highly sensitive to positional order, and recommend training and evaluation on second occurrences of ambiguous terms. Finally, we showcase the disparity in performance on words with related candidate concepts versus those with unrelated candidate concepts, reflecting higher performance on the type of ambiguous words that have higher impact on downstream tasks. Reflecting on our work, we identify two possible avenues for future research. First, one purpose of this work was to showcase BioBERT's WSD capacity, yet it does not optimize performance. More advanced classification architectures with attention mechanisms could guide classification by attending to the more relevant words of an ambiguous phrase, or even the more relevant words of the full sentence if fed in full. Second, and extending from WSD to entity linking, it would be interesting to study BioBERT's capacity at not only disambiguating phrases, but also identifying biomedical phrases to begin with, and disambiguating those that are ambiguous.

CHAPTER 7

SUPPORTING WORK

Finally, we summarize two of our previous NLP systems that tackle data shortage in general, and can assist low-resource biomedical application. The first system addresses semantically-aware morpheme segmentation, while the second addresses domain extraction: the curation of a large in-domain monolingual corpus given a seed corpus.

7.1 Morpheme Segmentation

One of the standard preprocessing steps in NLP, including biomedical NLP, is morpheme segmentation. In morpheme segmentation, a word like "doctors" is segmented into its meaningful morphemes "doctor" + "s". This kind of segmentation is helpful for NLP algorithms to reduce vocabulary size and consequently reduce sparsity in the training data. Example NLP applications that utilize morpheme segmentation are information retrieval (IR) [148, 149], automatic speech recognition (ASR) [150, 151], and machine translation (MT) [152, 153]. Morpheme segmentation becomes even more essential in the context of limited datasets such as in the case of the biomedical domain.

Although recent advances in deep learning frameworks have been less reliant on semantic segmentation and sufficing with orthographic segmentation such as Byte-Pair Encoding [154], meaningful segmentation remains of importance for applications requiring semantic-based segmentation.

7.1.1 Drawback of Previous Methods

The majority of systems prior to ours [155] relied solely on the surface form of a word without giving attention to the underlying semantics of the word

[156, 157, 158, 159, 160]. Relying only on surface form signals led to over-segmenting words, where a change in the surface form was a necessary but insufficient indication of a morphological change. For example, although appending "man" to "police" to form "policeman" is a valid morphological transformation, the addition of "man" to "fresh" is not a valid morphological transformation resulting in "freshman", since a freshman is not a fresh man.

To compensate for the drawback of previous methods we develop the system, named MORSE [155], which performs morpheme segmentation using both orthographic features and semantic features estimated from distributed representations of words.

7.1.2 Our Contribution

One of the basic building blocks to our algorithm is the geometric shapes apparent in word embeddings. Particularly, we rely on the fact that $v(doctor) - v(doctors) \approx v(patient) - v(patients)$, where $v(word)$ is the embedding of a word.

Accordingly, our algorithm first relies on orthographic signals to generate a candidate list of morphological rules. It first clusters pairs of words with equivalent change of affix. For example, it clusters pairs of words differing only in the suffix "s" together, such as ("doctor", "doctors"), and ("patient", "patients"). Then, it checks the consistency of the difference vector of pairs in a cluster. For example, it checks the similarity between $v(doctor) - v(doctors)$ and $v(patient) - v(patients)$. The closer the similarity between the difference vectors, and the larger the cluster is, the larger is the evidence for it being a valid morphological rule. Finally, our system measures the consistency of one pair of words with all other pairs in the cluster. This is for the purpose for invalidating false positives in a valid rule. For example, this would invalidate the pair ("on", "only") in the valid rule of adding an "ly".

Finally, for the purpose of segmentation, given a word, we sequentially choose the rule that maximizes the scores of the various signals, and segment accordingly. For a more detailed explanation, please check [155].

7.1.3 Experiments and Results

We compare our system, "MORSE", to a popular system, "Morfessor", which relies on orthographic features only. We perform the comparison over the standard dataset of Morpho Challenge on three languages of varying levels of morphology: English, Turkish, and Finnish. As shown in Table 7.1, MORSE performs significantly better than Morfessor on English, while the gap reduces as the level of morphology increases in the language considered, until Morfessor performs better than MORSE on Finnish. We hypothesize that the high level of morphology in a language increases the sparsity in the training data, which reduces the quality of the word embeddings learned, and consequently hurts the performance of MORSE.

Table 7.1: Scores of MORSE and Morfessor on the Morpho Challenge dataset

	English			**Turkish**			**Finnish**		
	P	R	F1	P	R	F1	P	R	F1
Morfessor	74.46	56.66	64.35	40.81	25.00	31.01	**43.09**	**28.16**	**34.06**
MORSE	**81.98**	**61.57**	**70.32**	**49.90**	**30.78**	**38.07**	36.26	9.44	14.98

7.2 Domain Extraction

Building NLP applications for the healthcare domain, as for all other specific domains, often requires the existence of a large in-domain corpus. For example, in our project for simplifying healthcare text, it was of the essence to have available not only a corpus of healthcare documents, but also articles that are patient-friendly. The process to collect such a corpus manually is time and labor-consuming. Other examples of utilizing monolingual corpora are training word embeddings [140], and training document embeddings [161]. Such resources are also central to downstream applications such as automatic speech recognition [162], machine translation [46], and text categorization [163].

Although possible, it would be suboptimal, and detrimental to performance, to use out-of-domain corpora regardless of the size [164]. Accordingly, we build a system, named Dexter, to automatically extract a large in-domain

corpus from a large multi-domain corpus (such as Wikipedia) given a small set of seed documents representative of the desired domain [165].

7.2.1 Drawback of Previous Methods

To the best of our knowledge, there is one previous system with a highly similar functionality: BootCaT [166]. This system utilizes a set of user-supplied keywords and the world-wide web to automatically scrape pages returned by queries of combinations of the keywords using a search engine of the user's choice. Despite having access to a larger set of documents than Dexter, which is limited to Wikipedia or a multi-domain corpus of the user's choice, we hypothesize that the unconstrained access of BootCaT would introduce noise due to the imperfect information retrieval algorithms, and due to the imperfect automated scraping algorithms.

7.2.2 Our Contribution

Dexter's algorithm instead relies on the assumption that within the multi-domain corpus, articles covering similar topics would be assigned embeddings close by in the vector space. To inspect such an assumption, we estimate an embedding space over Wikipedia's articles using Doc2Avg,[1] then map these articles into 2D space using t-SNE [167]. We see in Figure 7.1 that, for example, scientific articles (which are a superset of medical articles) cluster well together. This validates our initial assumption.

Following this observation, we build Dexter as follows. Dexter takes, as an input, seed articles representing the target domain (e.g. medicine) then ranks articles in Wikipedia based on their distance to the seed set in the embedding space, and finally returns the closest k (set by the user) articles from Wikipedia.

7.2.3 Experiments and Results

To assess the quality of the output corpus and compare against BootCaT, we evaluate the output corpus in terms of two purposes: training word em-

[1] Every document is assigned the average of its words' embeddings.

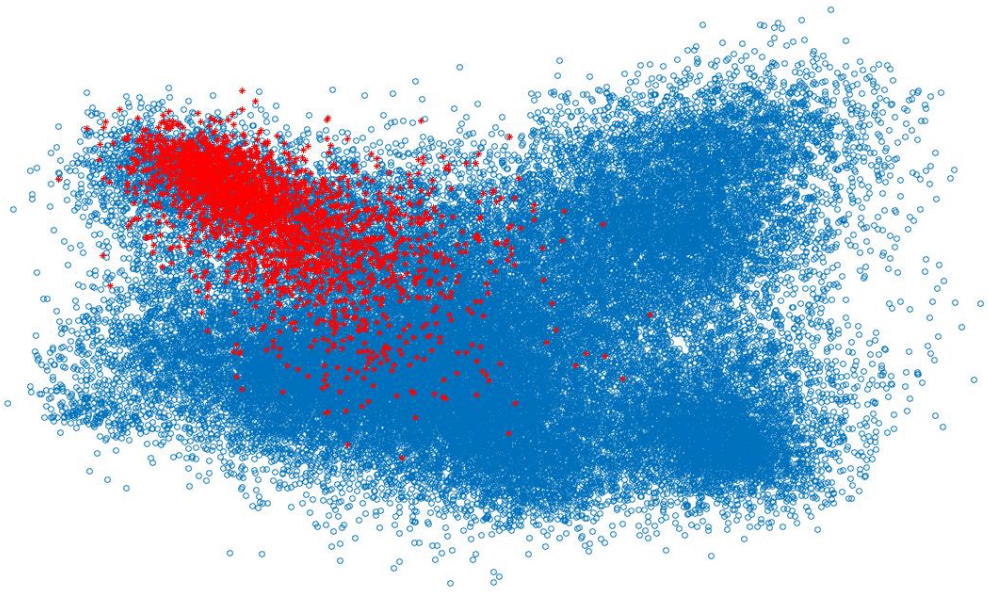

Figure 7.1: Mapping of Wikipedia articles into 2D space: Scientific articles in red, all other articles in blue

beddings, and estimating word representations.

Word Embeddings To evaluate the output corpus' quality for training word embeddings, we estimate word embeddings based on different corpora using the FastText algorithm [103]. For this experiment, we use the science domain as a target domain. Moreover, to assess the intrinsic quality of word embeddings, we assess its extrinsic ability at distractor generation for multiple choice questions (check [165] for more details).

As shown in the second column of Table 7.2, Dexter's output leads to the highest recall, higher than BootCaT's output. More surprisingly, Dexter's output results in a recall higher than the corpus extracted based on Wikipedia's manually constructed taxonomy.

Even qualitatively, one can see the quality of in-domain embeddings by checking neighbors of polysemous words in Table 7.3. For example, "Field" had a sports sense when trained on all of Wikipedia, and had a scientific sense when trained on Dexter's output.

Language Modeling Repeating the previous experiment, but for language modeling, returned similar results as shown in Table 7.2 with Dexter

Table 7.2: Distractor recall@100 for word embeddings (middle) and perplexity of language models (right) while varying training corpora. C is all of Wikipedia, C_D is taxonomy-based extracted corpus, BootCaT-KE is corpus constructed by BootCaT given seed set of documents, BootCaT-M is corpus constructed by BootCaT given set of keyphrases, Dexter is output of our system, while Dexter-Downsampled is after downsampling to the size of BootCaT outputs.

Corpus	**Recall**	**Perplexity**
C	17.43%	431.78
C_{silver}	N/A	334.57
C_D	20.47%	N/A
BootCaT-KE	15.28%	3199.30
BootCaT-M	13.82%	4586.80
Dexter-Downsampled	18.86%	1117.34
Dexter	22.71%	294.20

Table 7.3: Neighbors of scientific words when embeddings were estimated on all of Wikipedia (left), on C_D (top right), and on Dexter's output (bottom right).

Word	**Neighbors (General)**			**Neighbors (Science)**		
Force	Forces	Troops	Army	Deflection	Torque	Gravity
Digest	Review	Guide	Supplement	Digested	Extract	Metabolize
Matter	Matters	Subject	Debate	Particles	Materials	Universe
Field	Fields	Football	Professional-sized	Fields	Magnetobiology	Ambipolar
Rock	Punk	Pop	Indie	Rocks	Shoegazing	Screamo
Cellular	Cell	Signalling	Apoptosis	Cell	Organelle	Automata

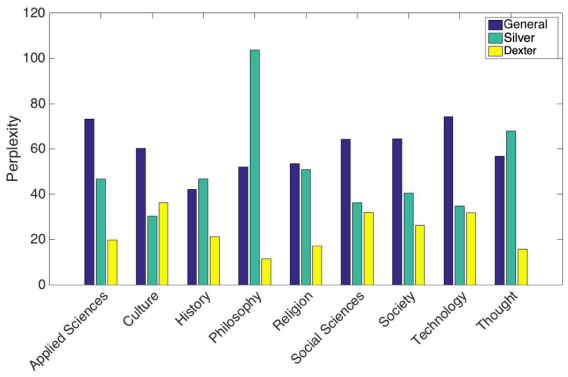

Figure 7.2: Performance of language models in terms of perplexity after training on General, Silver, and Dexter corpora on several Wikipedia domains

exceeding the performance of other corpora. Moreover, repeating the language modeling experiment shows competitive performance by Dexter over several domains, showcasing its ability to generalize to other domains as shown in Figure 7.2. This ability to generalize reflects the possible impact for healthcare.

CHAPTER 8

CONCLUSION

Biomedical NLP applications struggle with shortage of data. This work studies the potential of knowledge bases in compensating for the lack of data. Due to the nature of the technical biomedical field, knowledge is stored in highly curated ontologies such as the UMLS, and can serve as the aforementioned knowledge bases. To study the impact of knowledge bases, we take three biomedical applications as use cases: (1) text simplification of medication instructions, (2) entity linking for automatic short answer grading for medical training, and (3) training a conversational agent for medical adherence.

To build a text simplification system, the standard practice is to resort to Seq2Seq neural models that require a large training data consisting of a parallel corpus. Given the unavailability of a parallel corpus to automatically learn word correspondences between complicated terms and simpler ones, we rely on UMLS instead. Paired with a language model, we design and build an unsupervised text simplification system for medication instructions which outperforms neural methods. Moreover, the text simplification system positively impacted the human subjects' comprehension of health data.

As for automatically grading medical chart notes written by students in learning, and given the small size of the dataset, we rely on UMLS to both handle entity linking and to handle synonymy between an answer and a rubric item. The entity linker was designed to be unsupervised and can generalize across medical cases. The entity linker and the use of UMLS to extend a word beyond its surface forms increased the accuracy of the grading system.

For the last application of building a CA for medical adherence, our CA relies on a training data of multiple paraphrases of a patients' answer. The UMLS was used to augment the limited seed training data. Synonyms of words in the training data, retrieved from the UMLS, replaced the original wording to generate extra examples. This integration of UMLS did not lead to the CA more accurately assessing the patient's answer. The CA's biggest

sources of errors were counts in the text and incapability of handling negation, two issues not handled by synonymy extension using UMLS.

Finally, one recurring challenge in integrating knowledge bases into biomedical NLP applications was disambiguating the words of interest into a concept in UMLS given the high ambiguity of biomedical text. Accordingly, we took a step back and explored the potential of BioBERT at increasing the accuracy of WSD and entity linking in biomedical text. A simple feed-forward layer network showed results competitive with those of the SOTA deepBioWSD system. Moreover, we studied the impact of the position of the ambiguous word on the training and evaluation of a WSD system. We recommend a WSD system to be trained and tested on any word beyond the first occurrence which tends to be trivial.

8.1 Limitations and Future Work

This work showcases the potential of BioBERT for biomedical WSD, but does not optimize a WSD system architecture around the potential of BioBERT. The experiments presented in this work were limited to a single feed-forward neural layer, and more complex architectures have the potential to enhance accuracy of such a critical task. Future work can focus on devising neural architectures to best cultivate the signals in BioBERT representations for a more accurate BioWSD system.

With that being said, a stand-alone WSD system cannot resolve the ambiguity issue when integrating knowledge bases. In our setting, the trained WSD network had the mention boundaries, and the candidate concepts, given. In the real setting, this information is not provided, and for such a utility to be of use, further work needs to be done on how to detect the mention boundaries, and how to narrow down candidate concepts.

Future studies could examine whether BioBERT could also assist in detecting mentions of a span by modeling the problem as a supervised Seq2Seq task with BioBERT vectors as input. For a more domain-robust, generalizable setting, one can explore detecting biomedical mentions through surface level matching to atoms in UMLS itself with a biomedical concept type. As for removing the limitation on assuming candidate concepts are given, further studies can explore whether overgenerating candidates by checking exact

match in UMLS concepts is feasible. Addressing these two limitations would result in an end-to-end entity linking system, which can, in turn, significantly contribute to the integration of knowledge bases.

Finally, across all three downstream tasks considered, UMLS utilization was limited to synonymy. UMLS contains information beyond synonymy to include types of concepts and relations between them. Furthermore, it provides definitions for every concept. Future work can explore how to best utilize UMLS beyond synonymy.

REFERENCES

[1] Index for Excerpts from the American Recovery and Reinvestment Act of 2009, "Health Information Technology (HITECH) Act 2009," pp. 112–64.

[2] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1589–1604, 2017.

[3] H. Elsahar and M. Gallé, "To annotate or not? Predicting performance drop under domain shift," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 2163–2173.

[4] R. J. Adams, "Improving health outcomes with better patient understanding and education," *Risk Management and Healthcare Policy*, vol. 3, p. 61, 2010.

[5] W.-w. Yim, A. Mills, H. Chun, T. Hashiguchi, J. Yew, and B. Lu, "Automatic rubric-based content grading for clinical notes," in *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, 2019, pp. 126–135.

[6] T. W. Bickmore, K. Puskar, E. A. Schlenk, L. M. Pfeifer, and S. M. Sereika, "Maintaining reality: Relational agents for antipsychotic medication adherence," *Interacting with Computers*, vol. 22, no. 4, pp. 276–288, 2010.

[7] Office of the National Coordinator for Health Information Technology, "Percent of physicians e-prescribing through an electronic health record," dashboard.healthit.gov/quickstats/pages/FIG-Percent-Physicians-eRx-through-EHR.php, February 2014.

[8] A. Rotegard, L. Slaughter, and C. M. Ruland, "Mapping nurses' natural language to oncology patients' symptom expressions," *Studies in Health Technology and Informatics*, vol. 122, p. 987, 2006.

[9] S. Nisioi, S. Štajner, S. P. Ponzetto, and L. P. Dinu, "Exploring neural text simplification models," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (volume 2: Short Papers)*, 2017, pp. 85–91.

[10] D. A. Lindberg, B. L. Humphreys, and A. T. McCray, "The unified medical language system," *Methods of Information in Medicine*, vol. 32, no. 4, p. 281, 1993.

[11] M. Stevenson and Y. Guo, "Disambiguation in the biomedical domain: The role of ambiguity type," *Journal of Biomedical Informatics*, vol. 43, no. 6, pp. 972–981, 2010.

[12] A. R. Aronson and F.-M. Lang, "An overview of metamap: Historical perspective and recent advances," *Journal of the American Medical Informatics Association*, vol. 17, no. 3, pp. 229–236, 2010.

[13] N. Sabharwal and A. Agrawal, "Introduction to Google Dialogflow," in *Cognitive Virtual Assistants using Google Dialogflow*. Springer, 2020, pp. 13–54.

[14] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[15] C. J. Fillmore et al., "Frame semantics and the nature of language," in *Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech*, vol. 280, no. 1, 1976, pp. 20–32.

[16] M. Minsky, *Society of Mind*. Simon and Schuster, 1988.

[17] B. Yang and T. Mitchell, "Leveraging knowledge bases in LSTMs for improving machine reading," *arXiv preprint arXiv:1902.09091*, 2019.

[18] G. A. Miller, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[19] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet project," in *Proceedings of the 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 1998, pp. 86–90.

[20] R. Navigli and S. P. Ponzetto, "BabelNet: Building a very large multilingual semantic network," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 216–225.

[21] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 2008, pp. 1247–1250.

[22] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[23] L. Shi and R. Mihalcea, "Putting pieces together: Combining Framenet, VerbNet and WordNet for robust semantic parsing," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2005, pp. 100–111.

[24] K. K. Schuler, "VerbNet: A broad-coverage, comprehensive verb lexicon," Ph.D. dissertation, University of Pennsylvania, 2005.

[25] T. Pedersen, S. Patwardhan, J. Michelizzi et al., "WordNet: Similarity-measuring the relatedness of concepts." in *AAAI*, vol. 4, 2004, pp. 25–29.

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

[27] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[28] K. Nebhi, "Named entity disambiguation using freebase and syntactic parsing," in *Proceedings of the First International Workshop on Linked Data for Information Extraction (LD4IE 2013) co-located with the 12th International Semantic Web Conference (ISWC 2013)*. Gentile, AL; Zhang, Z.; d'Amato, C. & Paulheim, H., 2013.

[29] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[30] D. Weissenborn, T. Kočiskỳ, and C. Dyer, "Dynamic integration of background knowledge in neural NLU systems," *arXiv preprint arXiv:1706.02596*, 2017.

[31] C. Wang and H. Jiang, "Explicit utilization of general knowledge in machine reading comprehension," *arXiv preprint arXiv:1809.03449*, 2018.

[32] A. C. Browne, A. T. McCray, and S. Srinivasan, "The specialist lexicon," *National Library of Medicine Technical Reports*, pp. 18–21, 2000.

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

[34] W. Sun, A. Rumshisky, and O. Uzuner, "Evaluating temporal relations in clinical text: 2012 i2b2 challenge," *Journal of the American Medical Informatics Association*, vol. 20, no. 5, pp. 806–813, 2013.

[35] Y. Xiong, X. Shi, S. Chen, D. Jiang, B. Tang, X. Wang, Q. Chen, and J. Yan, "Cohort selection for clinical trials using hierarchical neural network," *Journal of the American Medical Informatics Association*, 2019.

[36] T. Miller, A. Geva, and D. Dligach, "Extracting adverse drug event information with minimal engineering," in *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, 2019, pp. 22–27.

[37] K. Davis, S. C. Schoenbaum, and A.-M. Audet, "A 2020 vision of patient-centered primary care," *Journal of General Internal Medicine*, vol. 20, no. 10, pp. 953–957, 2005.

[38] D. Detmer, M. Bloomrosen, B. Raymond, and P. Tang, "Integrated personal health records: Transformative tools for consumer-centric care," *BMC Medical Informatics and Decision Making*, vol. 8, no. 1, p. 45, 2008.

[39] T. Irizarry, A. D. Dabbs, and C. R. Curran, "Patient portals and patient engagement: A state of the science review," *Journal of Medical Internet Research*, vol. 17, no. 6, p. e148, 2015.

[40] N. Mcinnes and B. J. Haglund, "Readability of online health information: implications for health literacy," *Informatics for Health and Social Care*, vol. 36, no. 4, pp. 173–189, 2011.

[41] M. Kutner, E. Greenburg, Y. Jin, and C. Paulsen, "The health literacy of America's adults: Results from the 2003 national assessment of adult literacy. NCES 2006-483." *National Center for Education Statistics*, 2006.

[42] R. P. Kessels, "Patients' memory for medical information," *Journal of the Royal Society of Medicine*, vol. 96, no. 5, pp. 219–222, 2003.

[43] D. A. Kindig, A. M. Panzer, L. Nielsen-Bohlman et al., *Health literacy: a prescription to end confusion*. National Academies Press, 2004.

[44] J. F. Ha and N. Longnecker, "Doctor-patient communication: A review," *Ochsner Journal*, vol. 10, no. 1, pp. 38–43, 2010.

[45] S. Kandula, D. Curtis, and Q. Zeng-Treitler, "A semantic and syntactic text simplification tool for health content," in *AMIA Annual Symposium Proceedings*, vol. 2010. American Medical Informatics Association, 2010, p. 366.

[46] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 48–54.

[47] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[48] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[49] M. S. Wolf, L. M. Curtis, K. Waite, S. C. Bailey, L. A. Hedlund, T. C. Davis, W. H. Shrank, R. M. Parker, and A. J. Wood, "Helping patients simplify and safely use complex prescription regimens," *Archives of Internal Medicine*, vol. 171, no. 4, pp. 300–305, 2011.

[50] A. Atreja, N. Bellam, and S. R. Levy, "Strategies to enhance patient adherence: Making it simple," *Medscape General Medicine*, vol. 7, no. 1, p. 4, 2005.

[51] Q. T. Zeng and T. Tse, "Exploring and developing consumer health vocabularies," *Journal of the American Medical Informatics Association*, vol. 13, no. 1, pp. 24–29, 2006.

[52] A. V. Mohan, M. B. Riley, D. R. Boyington, and S. Kripalani, "Illustrated medication instructions as a strategy to improve medication management among Latinos: A qualitative analysis," *Journal of Health Psychology*, vol. 18, no. 2, pp. 187–197, 2013.

[53] J. Zheng and H. Yu, "Methods for linking EHR notes to education materials," *Information Retrieval Journal*, vol. 19, no. 1-2, pp. 174–188, 2016.

[54] A. Martin-Hammond and J. E. Gilbert, "Examining the effect of automated health explanations on older adults' attitudes toward medication information," in *Proceedings of the 10th EAI International Conference on Pervasive Computing Technologies for Healthcare*, 2016, pp. 186–193.

[55] J. R. Tupper, "Plain language thesaurus for health communications," Ph.D. dissertation, University of Southern Maine, Portland, ME, 2008.

[56] D. L. Mowery, B. R. South, L. Christensen, J. Leng, L.-M. Peltonen, S. Salanterä, H. Suominen, D. Martinez, S. Velupillai, N. Elhadad et al., "Normalizing acronyms and abbreviations to aid patient understanding of clinical texts: ShARe/CLEF eHealth Challenge 2013, Task 2," *Journal of Biomedical Semantics*, vol. 7, no. 1, p. 43, 2016.

[57] B. Qenam, T. Y. Kim, M. J. Carroll, and M. Hogarth, "Text simplification using consumer health vocabulary to generate patient-centered radiology reporting: Translation and evaluation," *Journal of Medical Internet Research*, vol. 19, no. 12, p. e417, 2017.

[58] J. Chen, E. Druhl, B. P. Ramesh, T. K. Houston, C. A. Brandt, D. M. Zulman, V. G. Vimalananda, S. Malkani, and H. Yu, "A natural language processing system that links medical terms in electronic health record notes to lay definitions: system development using physician reviews," *Journal of Medical Internet Research*, vol. 20, no. 1, p. e26, 2018.

[59] A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program." in *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 2001, p. 17.

[60] G. H. Paetzold and L. Specia, "A survey on lexical simplification," *Journal of Artificial Intelligence Research*, vol. 60, pp. 549–593, 2017.

[61] S. Bott, L. Rello, B. Drndarevic, and H. Saggion, "Can Spanish be simpler? LexSiS: Lexical simplification for Spanish," *Proceedings of COLING 2012*, pp. 357–374, 2012.

[62] G. Leroy, J. E. Endicott, D. Kauchak, O. Mouradi, and M. Just, "User evaluation of the effects of a text simplification algorithm using term familiarity on perception, understanding, learning, and information retention," *Journal of Medical Internet Research*, vol. 15, no. 7, p. e144, 2013.

[63] M. Shardlow, "The CW corpus: A new resource for evaluating the identification of complex words," in *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, 2013, pp. 69–77.

[64] K. Wróbel, "PLUJAGH at SemEval-2016 Task 11: Simple system for complex word identification," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016, pp. 953–957.

[65] S. Devlin, "The use of a psycholinguistic database in the simplification of text for aphasic readers," *Linguistic Databases*, 1998.

[66] J. Carroll, G. Minnen, Y. Canning, S. Devlin, and J. Tait, "Practical simplification of English newspaper text to assist aphasic readers," in *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, 1998, pp. 7–10.

[67] S. Wubben, A. Van Den Bosch, and E. Krahmer, "Sentence simplification by monolingual machine translation," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 1015–1024.

[68] T. Wang, P. Chen, J. Rochford, and J. Qiang, "Text simplification using neural machine translation," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[69] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[70] H. Jhamtani, V. Gangal, E. Hovy, and E. Nyberg, "Shakespearizing modern language using copy-enriched sequence-to-sequence models," *EMNLP 2017*, vol. 6, p. 10, 2017.

[71] TensorFlow Developers, "TensorFlow neural machine translation tutorial," 2017.

[72] Y. Singer and J. C. Duchi, "Efficient learning using forward-backward splitting," in *Advances in Neural Information Processing Systems*, 2009, pp. 495–503.

[73] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[74] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, "Optimizing statistical machine translation for text simplification," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 401–415, 2016.

[75] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 190–200.

[76] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017.

[77] R. Kavuluru, A. Rios, and Y. Lu, "An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records," *Artificial Intelligence in Medicine*, vol. 65, no. 2, pp. 155–166, 2015.

[78] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, "Deep learning with word embeddings improves biomedical named entity recognition," *Bioinformatics*, vol. 33, no. 14, pp. i37–i48, 2017.

[79] S. Garg, A. Galstyan, U. Hermjakob, and D. Marcu, "Extracting biomolecular interactions using semantic parsing of biomedical text," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[80] K. Lee, S. Lee, S. Park, S. Kim, S. Kim, K. Choi, A. C. Tan, and J. Kang, "BRONCO: Biomedical entity relation oncology corpus for extracting gene-variant-disease-drug relations," *Database*, vol. 2016, 2016.

[81] Y. Luo, Ö. Uzuner, and P. Szolovits, "Bridging semantics and syntax with graph algorithms—state-of-the-art of extracting biomedical relations," *Briefings in Bioinformatics*, vol. 18, no. 1, pp. 160–178, 2017.

[82] C. Friedman, P. O. Alderson, J. H. Austin, J. J. Cimino, and S. B. Johnson, "A general natural-language text processor for clinical radiology," *Journal of the American Medical Informatics Association*, vol. 1, no. 2, pp. 161–174, 1994.

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

[84] A. C. Browne, G. Divita, A. R. Aronson, and A. T. McCray, "UMLS language and vocabulary tools: Amia 2003 open source expo," in *AMIA Annual Symposium Proceedings*, vol. 2003. American Medical Informatics Association, 2003, p. 798.

[85] Y. Wu, J. C. Denny, S. T. Rosenbloom, R. A. Miller, D. A. Giuse, and H. Xu, "A comparative study of current clinical natural language processing systems on handling abbreviations in discharge summaries," in *AMIA Annual Symposium Proceedings*, vol. 2012. American Medical Informatics Association, 2012, p. 997.

[86] Y. Wu, J. C. Denny, S. Trent Rosenbloom, R. A. Miller, D. A. Giuse, L. Wang, C. Blanquicett, E. Soysal, J. Xu, and H. Xu, "A long journey to short abbreviations: Developing an open-source framework for clinical abbreviation recognition and disambiguation (card)," *Journal of the American Medical Informatics Association*, vol. 24, no. e1, pp. e79–e86, 2017.

[87] J. D'Souza and V. Ng, "Sieve-based entity linking for the biomedical domain," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, pp. 297–302.

[88] O. Ghiasvand and R. J. Kate, "UWM: Disorder mention extraction from clinical text using CRFs and normalization using learned edit distance patterns," in *In: Proc. SemEval 2014*. Citeseer, 2014.

[89] N. Kang, B. Singh, Z. Afzal, E. M. van Mulligen, and J. A. Kors, "Using rule-based natural language processing to improve disease normalization in biomedical text," *Journal of the American Medical Informatics Association*, vol. 20, no. 5, pp. 876–881, 2013.

[90] H. Li, Q. Chen, B. Tang, X. Wang, H. Xu, B. Wang, and D. Huang, "CNN-based ranking for biomedical entity normalization," *BMC Bioinformatics*, vol. 18, no. 11, pp. 79–86, 2017.

[91] Y. Luo, G. Song, P. Li, and Z. Qi, "Multi-task medical concept normalization using multi-view convolutional neural network," in *AAAI*, 2018, pp. 5868–5875.

[92] J. Xu, H.-J. Lee, Z. Ji, J. Wang, Q. Wei, and H. Xu, "UTH_CCB system for adverse drug reaction extraction from drug labels at TAC-ADR 2017." in *TAC*, 2017.

[93] R. Leaman, R. Islamaj Doğan, and Z. Lu, "DNorm: disease name normalization with pairwise learning to rank," *Bioinformatics*, vol. 29, no. 22, pp. 2909–2917, 2013.

[94] C.-P. Lee and C.-J. Lin, "Large-scale linear RankSVM," *Neural computation*, vol. 26, no. 4, pp. 781–817, 2014.

[95] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of NAACL-HLT*, 2018, pp. 2227–2237.

[96] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding with unsupervised learning," OpenAI, Tech. Rep., 2018.

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

[98] K. Huang, J. Altosaar, and R. Ranganath, "ClinicalBERT: Modeling clinical notes and predicting hospital readmission," *arXiv preprint arXiv:1904.05342*, 2019.

[99] Z. Ji, Q. Wei, and H. Xu, "BERT-based ranking for biomedical entity normalization," *AMIA Summits on Translational Science Proceedings*, vol. 2020, p. 269, 2020.

[100] W. A. Gale, K. Church, and D. Yarowsky, "One sense per discourse," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[101] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," *Proceedings of GSCL*, pp. 31–40, 2009.

[102] E. Loper and S. Bird, "NLTK: The natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[103] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[104] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, no. 1, pp. 1–9, 2016.

[105] J. D. Karpicke and J. R. Blunt, "Retrieval practice produces more learning than elaborative studying with concept mapping," *Science*, vol. 331, no. 6018, pp. 772–775, 2011.

[106] E. H. Wagner, B. T. Austin, C. Davis, M. Hindmarsh, J. Schaefer, and A. Bonomi, "Improving chronic illness care: translating evidence into action," *Health Affairs*, vol. 20, no. 6, pp. 64–78, 2001.

[107] L. I. Horwitz, J. P. Moriarty, C. Chen, R. L. Fogerty, U. C. Brewster, S. Kanade, B. Ziaeian, G. Y. Jenq, and H. M. Krumholz, "Quality of discharge practices and patient understanding at an academic medical center," *JAMA Internal Medicine*, vol. 173, no. 18, pp. 1715–1722, 2013.

[108] C. Kornburger, C. Gibson, S. Sadowski, K. Maletta, and C. Klingbeil, "Using "teach-back" to promote a safe transition from hospital to home: An evidence-based approach to improving the discharge process," *Journal of Pediatric Nursing*, vol. 28, no. 3, pp. 282–291, 2013.

[109] T. W. Bickmore, R. A. Silliman, K. Nelson, D. M. Cheng, M. Winter, L. Henault, and M. K. Paasche-Orlow, "A randomized controlled trial of an automated exercise coach for older adults," *Journal of the American Geriatrics Society*, vol. 61, no. 10, pp. 1676–1683, 2013.

[110] T. W. Bickmore, D. Schulman, and C. Sidner, "Automated interventions for multiple health behaviors using conversational agents," *Patient Education and Counseling*, vol. 92, no. 2, pp. 142–148, 2013.

[111] A. Watson, T. Bickmore, A. Cange, A. Kulshreshtha, and J. Kvedar, "An internet-based virtual coach to promote physical activity adherence in overweight adults: Randomized controlled trial," *Journal of Medical Internet Research*, vol. 14, no. 1, p. e1, 2012.

[112] R. A. Edwards, T. Bickmore, L. Jenkins, M. Foley, and J. Manjourides, "Use of an interactive computer agent to support breastfeeding," *Maternal and Child Health Journal*, vol. 17, no. 10, pp. 1961–1968, 2013.

[113] L. Laranjo, A. G. Dunn, H. L. Tong, A. B. Kocaballi, J. Chen, R. Bashir, D. Surian, B. Gallego, F. Magrabi, A. Y. Lau et al., "Conversational agents in healthcare: A systematic review," *Journal of the American Medical Informatics Association*, vol. 25, no. 9, pp. 1248–1258, 2018.

[114] H. Tanaka, H. Negoro, H. Iwasaka, and S. Nakamura, "Embodied conversational agents for multimodal automated social skills training in people with autism spectrum disorders," *PloS One*, vol. 12, no. 8, p. e0182151, 2017.

[115] P. Philip, J.-A. Micoulaud-Franchi, P. Sagaspe, E. De Sevin, J. Olive, S. Bioulac, and A. Sauteraud, "Virtual human as a new diagnostic tool, a proof of concept study in the field of major depressive disorders," *Scientific Reports*, vol. 7, no. 1, pp. 1–7, 2017.

[116] P. Philip, S. Bioulac, A. Sauteraud, C. Chaufton, and J. Olive, "Could a virtual human be used to explore excessive daytime sleepiness in patients?" *Presence: Teleoperators and Virtual Environments*, vol. 23, no. 4, pp. 369–376, 2014.

[117] E. Hudlicka, "Virtual training and coaching of health behavior: Example from mindfulness meditation training," *Patient Education and Counseling*, vol. 92, no. 2, pp. 160–166, 2013.

[118] G. M. Lucas, A. Rizzo, J. Gratch, S. Scherer, G. Stratou, J. Boberg, and L.-P. Morency, "Reporting mental health symptoms: Breaking down barriers to care with virtual human interviewers," *Frontiers in Robotics and AI*, vol. 4, p. 51, 2017.

[119] D. Ireland, C. Atay, J. J. Liddle, D. Bradford, H. Lee, O. Rushin, T. Mullins, D. Angus, J. Wiles, S. McBride et al., "Hello Harlie: Enabling speech monitoring through chat-bot conversations," *Studies in Health Technology and Informatics*, vol. 227, pp. 55–60, 2016.

[120] R. Crutzen, G.-J. Y. Peters, S. D. Portugal, E. M. Fisser, and J. J. Grolleman, "An artificially intelligent chat agent that answers adolescents' questions related to sex, drugs, and alcohol: An exploratory study," *Journal of Adolescent Health*, vol. 48, no. 5, pp. 514–519, 2011.

[121] R. F. Azevedo, D. Morrow, J. Graumlich, A. Willemsen-Dunlap, M. Hasegawa-Johnson, T. S. Huang, K. Gu, S. Bhat, T. Sakakini, V. Sadauskas et al., "Using conversational agents to explain medication instructions to older adults," in *AMIA Annual Symposium Proceedings*, vol. 2018. American Medical Informatics Association, 2018, p. 185.

[122] J. Gratch, A. Hartholt, M. Dehghani, and S. Marsella, "Virtual humans: A new toolkit for cognitive science research," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 35, no. 35, 2013.

[123] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[124] H. Liu, Y. A. Lussier, and C. Friedman, "Disambiguating ambiguous biomedical terms in biomedical narrative text: An unsupervised method," *Journal of Biomedical Informatics*, vol. 34, no. 4, pp. 249–261, 2001.

[125] G. Tsatsaronis, M. Schroeder, G. Paliouras, Y. Almirantis, I. Androutsopoulos, E. Gaussier, P. Gallinari, T. Artieres, M. R. Alvers, M. Zschunke et al., "BioASQ: A challenge on large-scale biomedical semantic indexing and question answering," in *AAAI Fall Symposium: Information Retrieval and Knowledge Discovery in Biomedical Text*. Citeseer, 2012.

[126] Y. Zhang, H. Lin, Z. Yang, J. Wang, S. Zhang, Y. Sun, and L. Yang, "A hybrid model based on neural networks for biomedical relation extraction," *Journal of Biomedical Informatics*, vol. 81, pp. 83–92, 2018.

[127] D. Widdows, S. Peters, S. Cederberg, C.-K. Chan, D. Steffen, and P. Buitelaar, "Unsupervised monolingual and bilingual word-sense disambiguation of medical documents using UMLS," in *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, 2003, pp. 9–16.

[128] M. J. Schuemie, J. A. Kors, and B. Mons, "Word sense disambiguation in the biomedical domain: An overview," *Journal of Computational Biology*, vol. 12, no. 5, pp. 554–565, 2005.

[129] A. Pesaranghader, S. Matwin, M. Sokolova, and A. Pesaranghader, "DeepBioWSD: Effective deep neural word sense disambiguation of biomedical text data," *Journal of the American Medical Informatics Association*, vol. 26, no. 5, pp. 438–446, 2019.

[130] M. Weeber, J. G. Mork, and A. R. Aronson, "Developing a test collection for biomedical word sense disambiguation," in *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 2001, p. 746.

[131] Y. Wang, K. Zheng, H. Xu, and Q. Mei, "Interactive medical word sense disambiguation through informed learning," *Journal of the American Medical Informatics Association*, vol. 25, no. 7, pp. 800–808, 2018.

[132] H. Liu, S. B. Johnson, and C. Friedman, "Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS," *Journal of the American Medical Informatics Association*, vol. 9, no. 6, pp. 621–636, 2002.

[133] H. Yu, W. Kim, V. Hatzivassiloglou, and W. J. Wilbur, "Using MEDLINE as a knowledge source for disambiguating abbreviations and acronyms in full-text biomedical journal articles," *Journal of Biomedical Informatics*, vol. 40, no. 2, pp. 150–159, 2007.

[134] H. Xu, P. D. Stetson, and C. Friedman, "Combining corpus-derived sense profiles with estimated frequency information to disambiguate clinical abbreviations," in *AMIA Annual Symposium Proceedings*, vol. 2012. American Medical Informatics Association, 2012, p. 1004.

[135] G. P. Finley, S. V. Pakhomov, R. McEwan, and G. B. Melton, "Towards comprehensive clinical abbreviation disambiguation using machine-labeled training data," in *AMIA Annual Symposium Proceedings*, vol. 2016. American Medical Informatics Association, 2016, p. 560.

[136] H. Liu, V. Teller, and C. Friedman, "A multi-aspect comparison study of supervised word sense disambiguation," *Journal of the American Medical Informatics Association*, vol. 11, no. 4, pp. 320–331, 2004.

[137] H. Xu, M. Markatou, R. Dimova, H. Liu, and C. Friedman, "Machine learning and word sense disambiguation in the biomedical domain: Design and evaluation issues," *BMC Bioinformatics*, vol. 7, no. 1, p. 334, 2006.

[138] Y. Wu, J. Xu, Y. Zhang, and H. Xu, "Clinical abbreviation disambiguation using neural word embeddings," in *Proceedings of BioNLP 15*, 2015, pp. 171–176.

[139] A. J. Jimeno-Yepes, B. T. McInnes, and A. R. Aronson, "Exploiting MeSH indexing in MEDLINE to generate a data set for word sense disambiguation," *BMC Bioinformatics*, vol. 12, no. 1, p. 223, 2011.

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

[141] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[142] R. I. Doğan, R. Leaman, and Z. Lu, "NCBI disease corpus: A resource for disease name recognition and concept normalization," *Journal of Biomedical Informatics*, vol. 47, pp. 1–10, 2014.

[143] À. Bravo, J. Piñero, N. Queralt-Rosinach, M. Rautschka, and L. I. Furlong, "Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research," *BMC bioinformatics*, vol. 16, no. 1, p. 55, 2015.

[144] J. G. Zheng, D. Howsmon, B. Zhang, J. Hahn, D. McGuinness, J. Hendler, and H. Ji, "Entity linking for biomedical literature," *BMC Medical Informatics and Decision Making*, vol. 15, no. S1, p. S4, 2015.

[145] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., "Scikit-learn: Machine learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[146] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *ICML*, 2010.

[147] J. Kans, "Entrez direct: E-utilities on the UNIX command line," in *Entrez Programming Utilities Help [Internet]*. National Center for Biotechnology Information (US), 2020.

[148] Y. L. Zieman and H. L. Bleich, "Conceptual mapping of user's queries to medical subject headings," in *Proceedings of the AMIA Annual Fall Symposium*. American Medical Informatics Association, 1997, p. 519.

[149] M. Kurimo, M. Creutz, and V. T. Turunen, "Unsupervised morpheme analysis evaluation by IR experiments-Morpho Challenge 2007," in *CLEF (Working Notes)*, 2007.

[150] J. A. Bilmes and K. Kirchhoff, "Factored language models and generalized parallel backoff," in *Companion Volume of the Proceedings of HLT-NAACL 2003–Short Papers*. Association for Computational Linguistics, 2003, pp. 4–6.

[151] M. Kurimo, M. Creutz, M. Varjokallio, E. Arisoy, and M. Saraçlar, "Unsupervised segmentation of words into morphemes–challenge 2005: An introduction and evaluation report," in *Proceedings of the PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, 2006.

[152] Y.-S. Lee, "Morphological analysis for statistical machine translation," in *Proceedings of HLT-NAACL 2004: Short Papers*. Association for Computational Linguistics, 2004, pp. 57–60.

[153] S. Virpioja, J. J. Väyrynen, M. Creutz, and M. Sadeniemi, "Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner," *Machine Translation Summit XI*, vol. 2007, pp. 491–498, 2007.

[154] Y. Shibata, T. Kida, S. Fukamachi, M. Takeda, A. Shinohara, T. Shinohara, and S. Arikawa, "Byte pair encoding: A text compression scheme that accelerates pattern matching," DOI-TR-161, Department of Informatics, Kyushu University, Tech. Rep., 1999.

[155] T. Sakakini, S. P. Bhat, and P. Viswanath, "Morse: Semantic-ally drive-n morpheme segment-er," in *55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*. Association for Computational Linguistics (ACL), 2017, pp. 552–561.

[156] Z. S. Harris, "From phoneme to morpheme," in *Papers in Structural and Transformational Linguistics*. Springer, 1970, pp. 32–67.

[157] J. Goldsmith, "Linguistica: An automatic morphological analyzer," in *Proceedings of 36th Meeting of the Chicago Linguistic Society*, 2000.

[158] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning-Volume 6*. Association for Computational Linguistics, 2002, pp. 21–30.

[159] M. Creutz and K. Lagus, "Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0," Helsinki University of Technology, 2005.

[160] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 4, no. 1, p. 3, 2007.

[161] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.

[162] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 400–401, 1987.

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

[164] D. McClosky, "Any domain parsing: automatic domain adaptation for natural language parsing," Ph.D. dissertation, Brown University, 2010.

[165] T. Sakakini, H. Gong, J. Y. Lee, R. Schloss, J. Xiong, and S. Bhat, "Equipping educational applications with domain knowledge," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2019, pp. 472–477.

[166] M. Baroni and S. Bernardini, "BootCaT: Bootstrapping corpora and terms from the web," in *LREC*, 2004, p. 1313.

[167] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.