NATURAL LANGUAGE PROCESSING*

Jake Vasilakes, MSc[1], Sicheng Zhou, MSc[1], Rui Zhang, PhD[1]

[1]Institute for Health Informatics and College of Pharmacy, University of Minnesota, Minneapolis, MN

**This is a draft of the chapter that appeared in Machine Learning for Cardiovascular Medicine. 2020. Al'Aref, S., Singh, G., Baskaran, L.,Metaxas, D. eds. Elsevier.*

ABSTRACT

Natural Language Processing (NLP) is a subfield of artificial intelligence that is concerned with the automatic understanding of human language by computers. NLP has seen much success in recent years due to increased computing power and the rise of deep learning, and this success has extended into the domains of biomedical and clinical text. NLP has contributed to tasks such as the discovery of drug interactions, the development of clinical decision support systems, and the facilitation of chart review. Part I of this chapter provides an introduction to NLP, some common tasks in the biomedical domain, and the methods for accomplishing these tasks. Part II gives a survey of recent applications of NLP in cardiovascular medicine.

KEYWORDS:

Natural Language Processing, Machine Learning, Artificial Intelligence, Biomedical Informatics, Clinical Informatics, Information Extraction

PART I: INTRODUCTION TO BIOMEDICAL NATURAL LANGUAGE PROCESSING

It is widely assumed that a key component of intelligence is the ability to understand and use natural language. *Natural languages* are methods of communication that have developed naturally within human society without explicit planning. These include languages such as English, Chinese, Pashto, etc. *Natural language processing* (NLP) is a subfield of artificial intelligence whose goal is to program computers to understand and analyze natural language. The main challenge NLP faces in accomplishing this goal is *ambiguity*, which abounds in natural language. Words often have multiple meanings (e.g. "duck" in "We saw her duck."), as do sentences (e.g. "No one does what we do well."), and ungrammatical English is often still quite meaningful (e.g. "Me want cheeseburger" still gets the point across). Natural language thus stands in contrast to *formal language* such as computer programming languages and languages of logic. Formal languages are completely unambiguous. For example, a statement in C++ has exactly one meaning and all statements that are not grammatical C++ have no meaning at all.

To address the ambiguity in natural language, NLP borrows extensively from the fields of linguistics, computer science, and artificial intelligence. Adopting this interdisciplinary approach, NLP has been successful for widely used languages such as English, specifically the "typical" use of English as in news articles and social media posts. However, as the field matures and NLP techniques become more successful, researchers have started applying NLP to various new domains.

In general, the goal of NLP is to understand what a given collection of texts, called a *corpus*, is about. The biomedical domain, which includes journal articles and clinical notes from electronic health records (EHRs), is one of the most exciting domains to which NLP has been applied in recent years. NLP has found much success here, having been used to automatically extract knowledge from the biomedical literature, to improve clinical outcomes via clinical decision support, and to discover new drug interactions. However, the analysis of a biomedical corpus (collection of texts) brings forth an additional set of challenges over traditional NLP, in the form of increased ambiguity and patient data privacy.

Ambiguity is a major challenge in biomedical text. Biomedical text has its own *sublanguage*, that is, sets of words, phrases, and structures that differ significantly from typical language (Friedman, Kra, and Rzhetsky, 2002). It contains a number of words that do not occur elsewhere (e.g. "eIF2α Kinase GCN2") and clinical text uses a shorthand that abandons much of the structure of typical language (e.g. "68yo WM c CHF and DMII admitted for edema and DOE"). As such, many of the NLP systems developed for typical language do not generalize to biomedical text and thus perform poorly. These sublanguages can, furthermore, differ between subdomains and clinical sites making the task even more difficult. Indeed, a major challenge in clinical NLP is generalizability: systems developed using data from one clinical site or specialty (e.g. surgery or cardiovascular medicine) often do not perform well when applied to data from another (Carrell et al., 2017).

Patient data privacy is another critical issue for NLP applied to clinical text. The use of clinical text corpora requires prior approval from institutional review boards (IRBs) and often requires *de-identification*, the process of removing all protected health information (PHI) (e.g. names, dates, and addresses) that could allow one to identify the patient discussed in the note. Obtaining IRB approval and data de-identification are lengthy processes that significantly affect the speed with which new clinical NLP systems can be developed. Even once these processes are complete, sharing data across institutions is difficult, which further hampers the development of NLP systems that perform well across clinical sites.

The above challenges, while specific to biomedical NLP, pose questions that are important for the field of NLP as a whole to answer: *How should we approach sentence structure and meaning for ungrammatical or semi-grammatical text? How can we efficiently learn the meanings of previously unseen words? How can we develop NLP systems that respect personal (patient) data privacy?* We are a long way from definitive answers to these questions and so these are crucial points to consider when deploying NLP systems in the real world.

In the following sections, we discuss (1) The typical tasks when working with biomedical NLP and (2) The methods commonly used to address them. We are careful to illustrate where biomedical NLP differs significantly from traditional NLP, why it does so, and any implications.

TASKS IN BIOMEDICAL NLP

Creating an NLP model that understands a given corpus requires the completion of a number of tasks, which can be broadly categorized into *high-level* and *low-level* tasks. High-level tasks in NLP are often an end in themselves, while low-level tasks are part of the pipeline in addressing a high-level task.

High-Level Tasks

High-level tasks include but are not limited to automatic language translation, summarization, and question answering. The most prevalent high-level task in biomedical NLP, however, is *information extraction* (IE). IE aims to discover information or knowledge that is, in a sense, hidden within the text. IE is therefore crucial for a variety of time- and life-saving clinical processes. For example, IE can help facilitate chart review by extracting information from clinical notes (Wu et al., 2013), it can identify potential drug interactions in the literature (Herrero-zazo et al., 2013; Zhang et al., 2014; Liu et al., 2016) , and it can identify risk factors from clinical text to aid in clinical decision support (Demner-Fushman, Chapman, and McDonald, 2009). IE is comprised of a number of sub-tasks in practice – named-entity recognition, relationship extraction, and entity linking– which are described below. To help illustrate these sub-tasks, we employ the example of extracting adverse drug reactions from clinical reports.

Named-Entity Recognition (NER)

The goal of NER is to find all mentions of certain types of concepts or entities in a given corpus. For example, an NER system might find mentions of symptoms and drugs within clinical notes. That is, it reads through the input text and outputs the entity type (symptom or drug or null) of each word. The main hurdle NER must face is *polysemy*, or words with multiple meanings. In cases of polysemy, NER turns to one of two subtasks: *word-sense disambiguation* (WSD) and *acronym disambiguation*. The goal of WSD is to determine the intended meaning of a word as used in the text. WSD systems leverage context around the target word as well as syntactic information. For example, the word “cold” is polysemous, having different meanings according to the context and its role in the sentence, e.g. “I feel cold” (adjective) vs. “I have a cold” (noun). *Acronym/abbreviation disambiguation* is related to WSD in that it aims to disambiguate acronyms by finding their correct expansion. For example, “MAP” expands to *mitogen-activated protein* but also *mean arterial pressure*.

While polysemy is pervasive in text, many of the entities in biomedical text are much less susceptible to misinterpretation. For example, the word “tamoxifen”, especially in a clinical note, almost certainly refers to the breast cancer drug. In some cases, this means that rather than employ complex context- and syntax-aware methods, biomedical NER can proceed as a simple dictionary lookup. If the given word exists in a list of drugs, then it is annotated as such. For certain entity types, this method works relatively well in practice.

Entity Linking

Biomedical entities often have a large variety of synonymous terms. For example, *vitamin C, L-ascorbic acid,* and *sodium ascorbate* all refer to the same entity. Giving a word or phrase a precise meaning is called *grounding* and it is often accomplished by the task of *entity linking*, which links entity mentions (such as those found by NER) to entries in a terminology or database. Entity linking serves two purposes: (1) it is a means of normalizing all synonymous mentions into a single form (such as a specific database ID) and (2) it enables access to the other information stored about that entity in the database. For example, by linking the term *sodium ascorbate* to its PubChem entry 54670067, we gain access to its chemical structure, safety information, uses, etc. Entity linking also allows the knowledge generated by an IE system to interface with and inform other biomedical information systems.

Entity linking also faces issues with ambiguity. Even if the NER or WSD system correctly identifies a word's meaning, there may be multiple candidate database entries to link to. For example, should "increased mean blood pressure" be linked to the Unified Medical Language System (UMLS) entry `Increased mean arterial pressure` (C0520853) or `mean arterial pressure increased` (C4087413)?

Relationship Extraction (RE)

RE aims to discover relationships between entities in the text. The entities are often obtained by running an NER system beforehand. Using the example of discovering adverse drug reactions, an RE system should discern whether (1) the given symptom is an adverse reaction of the drug, (2) the drug treats the symptom, or (3) the symptom and the drug are unrelated in the text. The output of an RE system is thus a set of triples with a <*subject, verb, object*> structure, such as <*tamoxifen, has_adverse_reaction, chest pain*>. RE systems look at the context, such the words between the two entities, and the sentence structure in order to discern whether two entities are related. Both are crucial for an effective RE system as clues in the context can be negated by sentence structure. Take for example the pair of sentences "We reviewed the charts of patients who were administered tamoxifen. It has been reported to cause chest pain but not vomiting." First notice that the drug name "tamoxifen" does not occur in the second sentence, but is rather referred to by "It". Resolving the meaning of pronouns like "It" across sentences is a task called *anaphora resolution*. Second, the context "reported to cause” between the drug (“It”) and the symptom “vomiting” indicates a potential adverse reaction, yet the “but not” indicates the opposite is true. The task of determining the scope of negation words such as this is called *negation detection* and *negation resolution*, respectively.

The set of triples output by an RE system can be considered hypotheses, which can inform real-world clinical outcomes such as those related to pharmacovigilance, precision medicine, and drug repurposing.

Low-Level Tasks

Low level NLP tasks are those which are not an end in themselves, but are rather an integral part of larger NLP systems and crucial to the systems which perform the high-level tasks given above. Some low-level tasks are *tokenization, sentence boundary detection*, *part-of-speech tagging, syntactic parsing,* and *language modeling*. Many modern and very capable systems exist to perform these tasks, even given the challenges of biomedical text, so this section will provide only a brief overview.

Tokenization

Tokenization is the process of splitting a text into its constituent word instances, or *tokens*. While in English most tokens are separated by whitespace or certain punctuation marks, biomedical text often contains exceptional tokens such as “q.i.d.” or “L-ascorbic acid”.

Sentence Boundary Detection

Even in grammatical language, determining where a sentence starts and ends can be difficult. The period “.”, which is used in many languages to denote the end of a sentence, occurs in other places, such as in acronyms (M.D.). This ambiguity is increased in clinical note shorthand, which can leave out punctuation altogether. Systems for sentence segmentation thus employ a variety of contextual clues to determine whether a given character, such as a period, is indeed an end-of-sentence marker.

Part-of-Speech (POS) Tagging

The part-of-speech (POS) of a word indicates the role it plays in a sentence, such as verb, noun, preposition, etc. Knowing a word's POS can thus provide crucial clues about its meaning. Knowing, for example, that "Buffalo" is being used as a proper noun allows an NER system to narrow down the possible labels for it. POS tagging systems assign one of a number of tags, such as those from the Penn Treebank, to each word in the input.

Syntactic Parsing

While POS tagging gives the roles of individual words, syntactic parsing determines how words fit together syntactically. Syntactic parsing can be *shallow* or *deep*. Shallow syntactic parsing groups words into phrases, such a noun or verb phrases. Deep syntactic parsing goes a step further by determining how words and phrases fit together to form a grammatical sentence. In both cases, a syntactic parser starts with the POS tags and combines them according to a set of grammatical rules to form a *parse tree*. A deep syntactic parse of the sentence "The patient reports experiencing chest pain." is given in Figure 1. The structure of a sentence illuminated by a syntactic parse shows how phrases are related to each other and can thus provide crucial clues to an RE system. For example, the parse tree below shows that there is indeed a connection via a verb phrase between "The patient" and "chest pain".

NLP METHODS

A number of methods exist for accomplishing the tasks described in the last section, which can be broadly grouped into *rule-based* methods and *statistical* or *machine-learning* methods. This section describes some common methods in each of these categories.

Rule-Based Methods

As discussed in the introductory section, natural language often breaks the rules of grammaticality. It is therefore impossible to come up with a set of rules that would allow computers to understand every utterance in a given language. Nevertheless, rules can account for a number of linguistic phenomena. While the general NLP community has generally moved away from the use of rule-based models for analyzing text in favor of more complex machine learning models, biomedical NLP continues to benefit from them. Rule-based systems such as MetaMap (National Library of Medicine, 2019) and NegEx (Chapman et al., 2013) are still widely used. Additionally, rule-based models are often straightforward to implement and use, and their decision-making processes are readily interpretable, both of which are vital attributes in the biomedical domain where NLP systems often carry the weight of affecting patients' health outcomes.

One of the most popular and powerful ways to implement rule-based NLP is by using *regular expressions* (regex). Regex is a small programming language designed to match and extract text. A given regex specifies a pattern to search for and is composed of literal characters and metacharacters. Literal characters allow for exact matches, e.g. `cat`. Metacharacters have special meanings that either match multiple characters or which operate on another character. For example, the period "`.`" matches any character except newline and the plus "+" means *match the previous character 1 or more times.* Thus the regex `cat.+` matches the literal string "cat" followed by any number of other characters such as "catamaran", "catch", etc.

There are a number of existing expert-curated biomedical terminologies and ontologies that lend even greater power to pattern matching systems. Resources such as the Unified Medical Language System (UMLS) (Bodenreider, 2004) and the Medical Dictionary of Regulatory Activities (MedDRA) (Maintenance and Support Services Organization, 2019) provide standard naming systems and taxonomies for a large number of biomedical concepts. By augmenting keywords obtained from these resources using regular expressions, an NLP system is often able to handle much of the linguistic variation in biomedical text, such as plural forms.

Statistical and Machine Learning Methods

Despite the continued relevance of rule-based models in biomedical NLP, the rise of machine learning (ML), specifically deep learning, has allowed for unprecedented performance on a variety of biomedical NLP tasks. We here describe a few common ML models for NLP, but in general one can think of an ML model for NLP as taking in a collection of numbers representing the text and outputting a collection of probabilities or *predictions* corresponding to the specified hypotheses (e.g. "at risk of heart attack" vs. "not at risk of heart attack"), which can then be used to make a decision.

Features

As mentioned above, the input to an ML model must be a numerical representation of the input, but what does it mean to represent a word or a sentence numerically? There are a number of specific implementations, but in general the text is transformed into a *vector*, i.e. a list of numbers, that in some way represents its meaning.

Bag-of-words (BOW) is one of the simplest ways to represent text numerically. Each word in the *corpus* (collectively called the *vocabulary*) is assigned a place in a large vector. A "1" is marked down for each word that occurs in a given document. For example, say the vocabulary extracted from a *corpus* is the following:

[*the, patient, arrhythmia, doctor, reports, saw*]

Then the following sentences would be represented as

The patient reports arrhythmia [1, 1, 1, 0, 1, 0]

The doctor saw the patient [1, 0, 0, 1, 0, 1]

Note that BOW does not maintain the order of the words, nor does it indicate how many times a word occurs. Additionally, words such as “the” that provide little insight into a document’s meaning are treated the same as more illuminating words, such as “arrhythmia”. *Term Frequency Inverse Document Frequency (TF-IDF)* accounts for variation in word importance by weighting words according to a combination of their frequency in the document and their frequency in the entire corpus (Manning, Raghavan, and Schütze, 2008). “Important” words occur frequently in a few documents but infrequently across the entire corpus; these will receive a high score. Words that occur frequently throughout the corpus are “unimportant” will receive a low score.

Word Embeddings or *Word Vectors* are one of the most important advances in NLP in recent years. As opposed to BOW and TF-IDF, which represent each word by a single number, word embeddings represent each word as a vector of numbers. Specifically, the numeric vector of a word is computed by a neural network model (discussed later) and is determined by the contexts in which that word occurs in a corpus. The assumption here is that words with similar contexts have similar meanings. In practice this assumption is a powerful one. When estimated using a sufficiently large amount of text data, word embeddings capture phenomena such as synonymy and analogy surprisingly well. Figure 2 provides a geometric interpretation of how word embeddings capture analogous meanings between words.

The invention of word embeddings has led to significant advances in NLP across tasks and domains. Indeed, it was a major motivation for the rise of neural network models in NLP. The original word embedding model, Google’s *Word2Vec* (Mikolov et al., 2013), is still widely used and has been adapted to biomedical text (Pyysalo et al., 2013; Zhang et al., 2019). Other implementations are fasttext (Bojanowski et al., 2016), which allows embeddings to be determined for out of vocabulary words, and GloVe (Pennington, Socher, and Manning, 2014), which estimates the embeddings from a word co-occurrence matrix, rather than directly from the text.

N-gram language model

An *n*-gram language model is one that predicts the next word given the *n-1* previous words using a simple count-and-divide method. The predictive model is estimated for a given sequence of *n* words by dividing the number of times that sequence occurs in the corpus by the number of times the previous *n-1* words occur in the corpus. For example, say the 4-gram "She fed the dog" occurs 2 times in a *corpus* and the 3-gram "She fed the" occurs 8 times. Then the probability that the next word is "dog" given "She fed the" is $\frac{2}{8}=0.25$. Language models allow us to compute the overall probability of a series of words, which is useful for a variety of prediction tasks.

Hidden Markov Model (HMM)

HMMs are sequence models. That is, given a sequence of inputs, such as words, an HMM will compute a sequence of outputs of the same length. An HMM model is a graph where nodes are probability distributions over labels and edges give the probability of transitioning from one node to the other. Together, these can be used to compute the probability of a label sequence given the input sequence. Figure 3 illustrates a small HMM for weather given the temperatures over a number of days. According to this model, the temperature is unlikely to change dramatically from one day to the next. That is, it is unlikely to go directly from <0 °C to >20 °C.

HMMs are often used for tagging tasks such as POS tagging and NER where a given label depends not only on the word in question but also the sequence of labels up to that point (e.g. it is unlikely that a verb follows a preposition).

Support Vector Machine (SVM)

The SVM is a model which learns how to separate data points (e.g. words or sentences) into one of two possible classes. As opposed to n-gram models and HMMs, the SVM is a *discriminative* model in that it does not estimate probabilities of belonging to one class or the other but rather it finds a *decision boundary* that separates the data points in geometric space (Cortes, Vapnik, 1995). Furthermore, because there are potentially many boundaries that separate the data points, the SVM finds the boundary which has the maximum distance from the two closest data points of opposite classes (these are the support vectors). Figure 4 shows a decision boundary estimated by an SVM on some toy data.

SVMs are most often used for prediction: after estimating a decision boundary on a set of data points for which we know the labels, the SVM predicts the labels of any new points by computing which side of the boundary they fall on. SVMs work quite well for a variety of NLP problems and, until recently, were used by many state-of-the-art systems for text classification.

Neural Networks

The SVM in its standard form is a *linear classifier*. That is, the decision boundary it finds is always perfectly straight, as shown in Figure 4. However, the classes in many data sets are not separable by a linear boundary. While it is possible to modify the SVM via something called a *kernel function* to allow it to estimate a nonlinear decision boundary, it is still limited in its expressivity by the exact kernel function used. Neural network models (NNs), on the other hand, have the ability to approximate any function whatsoever, meaning the nonlinear decision boundary they find can be as nuanced as desired (Lu et al., 2017). Furthermore, this decision boundary can be found automatically without manual experimentation, as is necessary with SVMs and kernel functions. NNs accomplish this by transforming the data multiple times before computing the output. Each of these intermediary transformations is called a *hidden layer* and an NN that uses multiple hidden layers is called a *deep neural network* (DNN). Figure 5 illustrates a simplistic neural network with a single hidden layer that predicts whether a patient is at risk for heart failure given three risk factors. The three risk factor values are first transformed into two hidden values by multiplying them by a set of weights. The result of this multiplication is in turn multiplied by a second set of weights to obtain the final values, which correspond to the decision.

NNs are realized in a wide variety of architectures. The neural network in Figure 5 is an example of a *feed-forward neural network* (FFNN) as the values of each layer are fed directly into the next layer via a linear transformation. Other common NN architectures are the *convolutional neural network* (CNN) and the *recursive neural network* (RNN). Illustrations of these architectures are given in Figure 6.

CNNs (Figure 6-A) were originally designed to mimic the human vision system in order to improve performance on automatic image recognition (Lu et al., 2017). Rather than apply a linear transformation at each layer as in FFNNs, CNNs use mathematical operations called *convolutions* to filter the input and highlight important information. In the case of images, this filter can, for example, find the edges of the objects in the image. However, it was found that CNNs also achieve very good results on text classification tasks (Lecun et al., 1998). CNNs operate on text data by, in a sense, looking at the entire document at once to distill its overall meaning, and the convolutions, rather than highlighting the edges of an image, highlight important words or phrases. For this reason, CNNs are best for text classification tasks such as determining whether a patient is at risk for heart disease from a clinical note.

RNNs (Figure 6-B) are different from feed-forward neural networks and CNNs in that they operate over time (Rumelhart, Hinton, and Williams, 1986). In other words, the hidden layer at the current time step is composed with the hidden layer from the previous time step. In this way, the RNN remembers what it has previously seen. This makes RNNs well-suited to text data because they are able to reason about the current word given the previous words in the input. While standard RNNs use the same operations as FFNNs to compute the hidden states, two common variations, Long Short-Term Memory (LSTM) (Sepp, Jürgen, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014), replace the standard hidden state with specialized operations that allow the network to selectively remember or forget certain parts of what it has seen before. This allows the network to take long-range context into account when handling a given word. LSTMs accomplish this by using a series of memory cells that selectively remember or forget information as it passes from the hidden layer at the previous time step to the current time step.

Standard RNNs are good for transforming an input sequence into an equal length output sequence, such as biomedical text input to word labels for an NER task. To transform between two sequences of unequal length, on the other hand, an *encoder-decoder* network is used (Figure 7-A), which uses a stack of two RNNs (Sutskever, Vinyals, and Le, 2014). The first RNN (the encoder) transforms the input into a single context vector (sometimes called a thought vector). The context vector is then fed into the second RNN (the decoder), which uses it to generate the required output. Because the input and output can be of unequal length, encoder-decoder networks are often used for tasks such as question answering (where the input is a question and the output is an answer) and translation, for example from English to Chinese or from complex clinical text to a simplified version fit for a lay person.

Nevertheless, the encoder-decoder architecture described above has two issues. First, the entire input sequence is encoded into a single context vector, which is not ideal especially for very long input sequences. Second, the context vector is only made available to the initial state of the decoder, meaning that later decoder states will use it less. To combat these issues, the *attention mechanism* was developed, which provides a robust way to pass information from the encoder to the decoder (Bahdanau, Cho, and Bengio, 2014). Rather than use a single context vector, the attention mechanism computes a new context vector for each decoder output state from a weighted combination of the input states (Figure 7-B). These weights indicate how much "attention" the decoder pays to each input word when generating each output word.

While early uses of attention were in encoder-decoder networks using RNNs, it was later found that the RNNs themselves could be replaced by a modified attention mechanism called *self-attention* (Cheng, Dong, and Lapata, 2016). In self-attention, the hidden state for each input word is computed from a weighted combination of the other input words. By looping back on itself in this way, self-attention is able to compute a hidden state for each word (like RNNs) by looking at all the words surrounding it (like CNNs). The seminal paper, 'Attention is All You Need' (Vaswani et al., 2017), showed that an encoder-decoder network using only attention and self-attention (an architecture called a *transformer*) siginificantly outperforms RNN-based encoder-decoder networks on a number of different NLP tasks.

The use of attention and self-attention culminated with the development of Google's BERT model, short for *Bidirectional Encoder Representations from Transformers* (Devlin et al., 2018). By applying the model architecture from 'Attention is All You Need' to a language modeling task, BERT is able to learn contextualized representations of input words. These contextualized representations are similar to word embeddings, but use the surrounding words to determine the sense of the target word and adjust its embedding accordingly. This eliminates the issue with standard word embeddings in which multiple meanings (for example, river *bank* and money *bank*) are collapsed into a single vector. Furthermore, BERT's contextualized representations can be used as a starting point for training task-specific NLP models in a process called *fine-tuning*. The BERT paper showed that by using a BERT model trained on a sufficiently large amount of text data, fine-tuning can achieve state-of-the-art performance on a variety of NLP tasks using a fraction of the training data of its competitors.

The combination of BERT's performance and the relative ease of fine-tuning it for specific tasks shot it into the spotlight, motivated a variety of modified versions, such as AlBERT (Lan et al, 2019), RoBERTA (Liu et al., 2019), and BART (Lewis et al., 2019), and inspired researchers to apply it in various disciplines. The base BERT model has been retrained on both biomedical (Lee et al., 2019) and clinical (Kexin, Jaan, and Rajesh, 2019) text, leading to significant improvements over the state-of-the-art on a number of tasks in these domains..

BIOMEDICAL NLP RESOURCES AND SYSTEMS

The field of biomedical NLP is growing rapidly. As such, many new tools and resources are being developed for performing a variety of NLP tasks on biomedical and clinical text. This section briefly describes a few often used, open-source resources and systems.

Biomedical Terminologies and Ontologies

The Unified Medical Language System (UMLS)

The UMLS (Bodenreider, 2004), developed by the U.S. National Library of Medicine (NLM), collects and integrates a large number of biomedical terminologies and ontologies into a single system. The UMLS encompasses the Metathesaurus, the Semantic Network, and the SPECIALIST Suite of tools. The Metathesaurus is a collection of biomedical terms from over 200 existing vocabularies such as MeSH, RxNorm, and SNOMED-CT. The Semantic Network consists of a set of broad subject categories (i.e. semantic types) and relationships (i.e. semantic relations) between these semantic types.. The SPECIALIST suite provides a lexicon and lexical tools for working with biomedical text.

RxNorm

RxNorm (Liu et al., 2005) is a system for normalizing drug names and supporting interoperability between health systems. It normalizes names by assigning types to each component of the name. For example, the drug name “Fluoxetine 4 MG/ML Oral Solution” is comprised of an ingredient (Fluoxetine), a strength (4 MG/ML), and a dose form (Oral Solution).

Biomedical NLP Systems

MetaMap and SemRep

MetaMap (National Library of Medicine, 2019) is a tool developed by the U.S. NLM for linking entities to the UMLS. The SemRep (Rindflesch, Fiszman, 2003) tool extracts relationships between UMLS concepts identified by MetMap from biomedical literature. Both tools use a rule-based approach. The NLM regularly runs MetaMap and SemRep on the entirety of PubMed abstracts to produce a database of extracted relationships, called SemMedDB (Kilicoglu et al., 2012) .

cTAKES

Apache cTAKES (Savova et al., 2010) is a free information extraction system for biomedical and clinical text originally developed at the Mayo Clinic. It is able to extract mentions, perform entity linking, mark negated or uncertain expressions, and identify expressions of time (e.g. “follow up in 1-2 weeks”). cTAKES is built using the Apache Unstructured Information Management Architecture (UIMA) (Ferrucci et al., 2009) , which means its pipelines are specified using modular components. In addition to modularity, UIMA allows for interoperability between systems built according to its framework.

The BioMedical Information Collection and Understanding System (BioMedICUS)

BioMedICUS (BioMedICUS, 2019) is a free and open-source system for large-scale biomedical and clinical text analysis developed at the University of Minnesota Institute for Health Informatics. It contains a number of components for doing low-level tasks such as tokenization, sentence boundary detection, and POS, tagging as well as prespecified pipelines for processing clinical documents. These pipelines focus on areas such as social history, time, and measures.

CLAMP (Clinical Language Annotation, Modeling, and Processing Toolkit)

CLAMP (Soysal et al., 2018) is developed at The University of Texas Health Science Center at Houston. It incorporates state-of-the-art NLP modules that can be combined into customized pipelines and is built upon the UIMA framework, meaning it is compatible with other systems such as cTAKES. CLAMP also provides user-friendly interfaces for end-users to leverage NLP components to build customized NLP pipelines. CLAMP allows users to combine both machine learning, deep learning with rule-based approaches for performance optimization.

SUMMARY

Natural language processing seeks to address a variety of problems, from word tokenization to syntactic parsing to question answering. In the biomedical domain, however, information extraction (IE) is the major focus, and is comprised of subtasks such as named-entity recognition (NER), entity linking, and relationship extraction. In general, the NLP methods used for these tasks can be divided into two categories. The first are rule-based, which employ regular expressions, keyword lists, and other manually defined logic to process text. While, rule-based methods have largely been discarded by the NLP community in favor of modern machine learning approaches, they still find a home in the biomedical domain, as they are easy to implement, use, and interpret, and in many cases perform competitively with modern approaches. Some notable rule-based systems are MetaMap and SemRep for entity linking and relationship extraction, and NegEx for negation detection. The second category are machine learning methods, which automatically learn to perform NLP tasks by using statistical modeling techniques. Most notably, the explosion of deep neural networks in recent years has motivated a number of powerful models for NLP, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers, the last of which have shown enormous potential across many tasks and domains. Nevertheless, earlier models such as hidden-markov models (HMMs) and support vector machines (SVMs) often perform well in practice.

A number of existing tools exist for NLP in the biomedical domain. The Unified Medical Language System (UMLS) provides an invaluable and expansive biomedical terminology. The UMLS is used by the aforementioned MetaMap and SemRep to perform entity linking and relationship extraction.

Finally, cTAKES, CLAMP, and BioMedICUS are toolkits for biomedical NLP that include functionality for a variety of tasks, from tokenization to relationship extraction.

PART 2: APPLICATIONS OF NLP IN CARDIOVASCULAR RESEARCH

It is a rising trend to apply NLP in research related to cardiovascular diseases. The American College of Cardiology released the "Roadmap for Innovation" in 2017 that emphasizes the importance of NLP and machine learning in the era of digital health (Bhavnani et al., 2017). The rapid development and adoption of EHR systems documenting vast amounts of real-world patient data has created many research opportunities. Valuable clinical information (e.g., symptoms, lab values, disease histories) for cardiovascular research is found documented in free text in EHR systems, making it impractical to manually identify the information. Thus, one important topic is to accurately extract the value information from numbers of clinical notes in an automatic way. In this section, we summarize relevant studies to demonstrate the role of NLP in current research related to cardiovascular diseases.

NLP for Information Extraction

Extracting information from text data is one of the primary tasks of NLP in the clinical area. The unstructured part of EHRs mainly contain three types of data, real-time data, retrospective data, and clinically irrelevant data (Bhavnani et al., 2017) . The real-time data often contains information about the current encounter (e.g. diagnosis, test values, symptoms of diseases). The retrospective data provides information about past encounters (e.g. past diagnosis, treatments applied). The clinical irrelevant data includes information about administration information, etc. Research regarding NLP applications in clinical settings has provided practical ways to extract targeted information from the clinical texts in the face of irrelevant information.

Rule-based NLP applications in cardiovascular research

Researchers have been using NLP for information extraction tasks in clinical settings since the 2000s. Friedlin et al. developed an NLP tool called REgenstrief data eXtraction tool (REX), with the object to extract a list of targeted congestive heart failure (CHF) related concepts (e.g. CHF, cardiomegaly) from x-ray reports (Friedlin, McDonald, 2006) . The REX uses regular expressions and a set of rules to detect the targeted concepts and analyze their contexts. They found that the performance of REX was better than the human annotators in the evaluation, with an average sensitivity of 0.98. The shortcomings of REX include that it could only extract simple concepts and it is not good at dealing with spelling errors and variations, acronyms disambiguation, and contextual information (e.g. negation). Khalifa et al. developed an NLP application based on the Apache UIMA framework by integrating two existing NLP resources, cTAKES and Textractor, to identify cardiovascular risk factors (e.g. obesity, smoking status, diabetes) from clinical notes (Khalifa, Meystre, 2015). Their NLP pipeline is shown in Figure 7. The raw texts are pre-processed (e.g. sentence detection, tokenization, POS tagging), which are then fed into the rule-based modules in cTAKES and Textractor to extract different risk factors.

This study applies dictionary look-up, machine learning, and regular expression methods in the pipeline to identify different variables. For instance, it uses a machine learning module to identify the smoking status of patients with features like words of the sentence and their POS tags. Medications and laboratory results are detected using the rule-based pattern matching modules, while disease and risk factor terms are identified using the UMLS Metathesaurus lookup module from Textractor. This application was evaluated using the 2014 i2b2 challenge dataset and achieved an overall micro-averaged F1-measure of 87.47% and a macro-averaged F1-measure of 86.99%. This study shows that it is practical to reuse and integrate existing NLP tools to solve similar tasks with minor modifications.

NLP has also been used for cardiovascular disease phenotyping in the EHR. Ye et al. developed a pipeline that identifies atherosclerotic cardiovascular disease (AVD) phenotypes from both the EHR and biorepository data of patients which includes both structured data and unstructured data (Ye et al., 2013). Rule-based NLP methods were used to identify risk factors and medication usage of the AVD, such as smoking status, diabetes, hypoglycemic agents, and anti-hypertensive medications. The output of the system was further combined with structured data (e.g. demographics, laboratory tests, diagnosis codes) to finally determine the phenotypes of patients.

Machine learning and deep learning with NLP applications in research

NLP applications in clinical settings have been evolving from rule-based methods to machine learning and more advanced deep learning methods. Weng et al. constructed an NLP pipeline based on cTAKES to identify the subtype of the clinical notes, such as cardiology reports (Weng et al., 2017). Both shallow machine learning and deep learning methods were investigated in the study. Two types of features were used. The first is lexical features of words (e.g. bag-of-word and UMLS concepts) obtained by the NLP pipeline. The other type of feature is the distributed word (word2vec) and document representations (distributed memory model of paragraph vectors). The classifiers used include the Naïve Bayes algorithm, multinomial logistic regression, regularized SVM with linear kernel, and two ensemble algorithms, random forest and adaptive boosting. They also evaluated two deep learning methods: CNN and LSTM models. The best performance was obtained by the LSTM model with fastText word embeddings, with an AUC score of 0.98.

Zhang et al. developed both rule-based and machine learning-based NLP methods to identify the New York Heart Association (NYHA) class of patients from their clinical notes, and compared their performance (Zhang et al., 2018) . The NYHA is used as a measure of a patient's response to Cardiac Resynchronization Therapy (CRT), which may inform a better understanding of the progression of heart failure to assess CRT effectiveness. For the machine learning based method, the texts were pre-processed (e.g. remove stop words, normalize lexical variants) and bag-of-words and *n*-grams were extracted as features. Using these features SVM, logistic regression, and random forest models were developed and evaluated. The random forest model with *n*-gram features obtained the best performance with a F-1 score of 93.78%, which surpassed the performance of rule-based method. The study indicates that traditional machine learning algorithms with features created by NLP methods can be effective in clinical text classification tasks.

Chokwijitkul et al. evaluated two deep learning architectures, CNN and RNN as well as three RNN variants, including LSTM, bidirectional long short-term memory (Bi-LSTM), and gated recurrent unit (GRU), for extracting cardiac risk factors from EHRs (Chokwijitkul et al., 2018). The dataset used in the study comes from the i2b2/UTHealth shared task. The risk factors to be extracted include clinical concepts related to CAD, diabetes, hypertension, hyperlipidemia and their relevant medications. The CNN model uses the window approach to classify each individual word at a time instead of the entire sentence for the NER task. This approach assumes that the label of a word is dependent on its neighboring words within the window size. The clinical text was transformed into pre-trained word embeddings as input of neural network models. Evaluation showed that RNN-type networks outperformed the CNN in their task. Among different RNN models, Bi-LSTM achieved the best performance with F-measure of 0.908 on the test data. Their study shows that the deep learning approaches were comparable to highly feature-engineered hybrid systems, and could obtain promising results without the help of any knowledge-driven methods. Thus, lots of human labor could be saved by the deep learning approaches.

Viani et al. developed a novel application that used an RNN to extract clinical events (e.g., problem, test, treatment) from cardiology reports written in Italian (Viani et al., 2019). Traditional machine learning methods often rely on complicated linguistic features (e.g., lexical, morphological and syntactic features), which usually need manually feature engineering. For neural networks, it could learn the representations starting from tokens or characters. A GRU model with pre-trained word embeddings and POS tags as features was compared with a dictionary look-up method, a standard SVM model, and a set of existing Bi-LSTM-CRF models. The GRU model obtained an F-1 score of 87.4% and after integration with the dictionary look-up method, the F-1 score increased to 90.1%, indicating that combining different methods may increase the performance at low cost.

NLP for Clinical Decision Support

Clinical decision support (CDS) aims to aid clinicians by providing them with easily accessible health-related information at the point of potential actionable advice. NLP is instrumental in extracting valuable information from unstructured data to represent clinical knowledge and drive CDS interventions which would improve the quality of healthcare (Demner-Fushman, Chapman, and McDonald, 2009).

Moon et al. developed and deployed NLP algorithms to automatically extract sudden cardiac death (SCD) factors such as syncope, family history of SCD, and hypertrophic cardiomyopathy from clinical narratives (Moon et al., 2019) . They developed regular expression and rule-based NLP algorithms using MedTagger, which has been adopted enterprise-wide by the Mayo Clinic. The keywords for different clinical concepts used for the developed algorithms were obtained by manually reviewing and searching lexical variations and synonyms from the UMLS Metathesaurus. The assertion and negation detection module in MedTagger is integrated in the NLP pipeline to improve the results. For evaluation, they compared the NLP pipeline with the standard method of using the registry, billing codes, and patient surveys to extract the syncope, family history of SCD, and hypertrophic cardiomyopathy. The NLP pipeline showed superior performance to the standard method for this task. The extracted SCD risk factors are expected to be fed into clinical decision support systems to increase efficiency of the HCM patients' management workflow for providers to improve the quality of care.

Meystre et al. developed a new CHF treatment performance measure information extraction system based on the UIMA framework called CHIEF (Meystre et al., 2016). The system uses a combination of rules, dictionaries, and machine learning methods to extract left ventricular function mentions and values, CHF medications, and documented reasons for a patient not receiving these medications. For instance, to extract the left ventricular ejection fraction (LVEF) mentions and corresponding values, it uses both regular expressions and machine learning with morphological (e.g. prefixes and suffixes), lexical (e.g. words themselves and *n*-grams), syntactic (e.g. POS tags), and semantic features (e.g. output of regular expression). The output information was integrated to classify the CHF treatment quality of patients. If a patient is identified with low LVEF measurements (<40%) and does not take the required medications, it indicates poor treatment quality. It may achieve the fast and scalable detection of CHF patients not receiving recommended treatment, which could help clinicians decide further treatment for patients at the point of care.

NLP for Predictive Models Related to Cardiovascular Disease

The combination of NLP and machine learning methods is promising in the development of models for predicting cardiovascular diseases. Choi et al. explored using RNNs to predict the initial diagnosis of heart failure (HF) (Choi et al., 2016). Compared to traditional machine learning methods, RNNs (LSTM and GRU specifically) are more suited to identify patterns in longitudinal data. The GRU model was developed to capture the relations among events (e.g., diagnosis of diseases, medication orders and procedure orders) with time stamps. They processed the clinical events in both structured and unstructured data using a standard NLP pipeline and encoded the events using a set of one-hot vectors as the input of the GRUs. The output is the risk of HF for a patient. When comparing with logistic regression, SVM, and multilayer perceptron, the GRU model obtained an AUC of 0.883, which is significantly higher than other models. This study could help the early detection of HF, which open new opportunities for delaying or preventing progression to diagnosis of HF and reduce cost. Maragatham and Devi performed a similar study, developing an LSTM model that used the temporal information of patients' hospital encounters to predict their risk of HF (Maragatham, Devi, 2019). They applied skipgram to encode the procedure, medicine, and diagnosis events as vectors for LSTM model training. They evaluated the LSTM model with different observation windows (3-12 months), and compared its performance with several baseline models (e.g., SVM, multilayer perceptron, logistic regression). The LSTM model obtained the best performance with an AUC of 0.7969, using a 12 month observation window. Using NLP and deep learning methods to predict risk of disease is promising and has the potential to be integrated into CDS systems to improve the quality of care.

NLP for Cohort Identification of Cardiovascular Diseases

Cohort identification is another important task in the field of clinical research. It is a vital step for other tasks such as clinical trial screening and recruitment and public health studies. Traditional cohort identification uses structured data such as the International Classification of Diseases (ICD) codes and billing codes. However, using NLP methods on unstructured clinical texts has shown to better improve cohort identification tasks. Wang et al. developed an NLP-based algorithm to find CHF cases from the EHR (Wang et al., 2015). In total, there are 32 CHF discriminant features included in the final pipeline. The features were extracted from both structured (e.g. demographics, vital signs) and unstructured data (e.g. ICD code of 'CHF', terms like 'heart failure' and 'congestive heart') in the EHR. A binary classification algorithm was then developed based on the features to judge if the patient has CHF. In evaluation, the NLP-based algorithm obtained a F-measure of 0.789. Also, the algorithm was integrated into the Health Information Exchange population exploration system in the state of Maine to conduct a real-time CHF identification task. The algorithm identified over a thousand patients that have not been coded as CHF patients previously. Geva et al. developed a computable algorithm to improve the cohort identification of pulmonary hypertension (PH) (Geva et al., 2017). In order to include text data in the EHR as features for the computable algorithm development, they applied an NLP tool, Narrative Information Linear Extraction (NILE) package, to identify clinical concepts (e.g. medication mentions, symptoms of the diseases) relevant to PH in patients' clinical notes. Based on the extracted features, they fitted an adaptive least absolute shrinkage and selection operator (LASSO) penalized logistic regression model to identify if the patient has PH. The algorithm was evaluated and obtained the best performance with AUC of 0.900. The developed algorithm is promising to recruit the largest cohort of pediatric PH patients to date for further research on this disease.

Conclusion

In summary, NLP applications in cardiovascular disease are developed for various tasks in processing and analyzing clinical texts, and a few of them have been applied in real practice. Currently, the major goals of NLP applications in clinical research are to extract and classify the target information of patients. NLP methods have evolved from rule-based methods to conventional machine learning based methods. Most recently, advanced deep learning techniques, especially transformers such as BERT, have led to unprecedented advances in NLP. These the potential to radically change how we approach NLP problems by removing the need for vast amounts of task-specific training data. There are still challenges and opportunities of NLP tasks in cardiovascular diseases. In the future, NLP will continue to play vital roles in both clinical research and clinical practice.

FIGURE LEGENDS

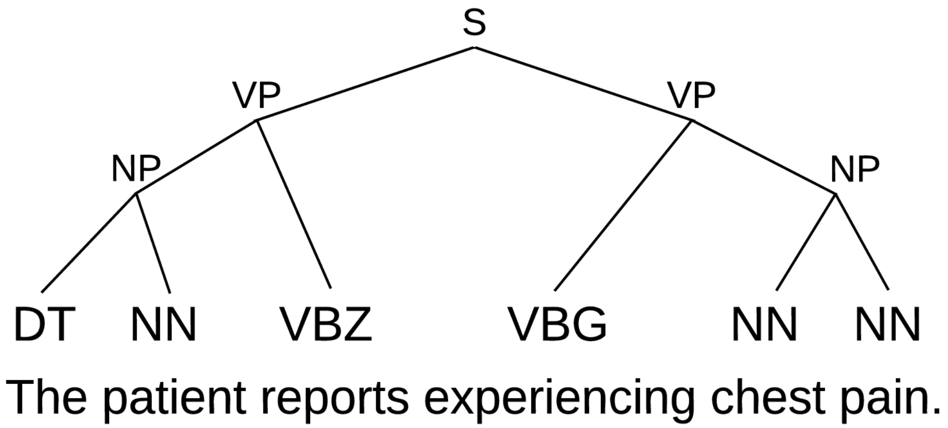

Figure 1: The full syntactic parse for the sentence "The patient reports experiencing chest pain.". The parser recursively combines the POS tags according to predefined grammatical rules until it obtains a full parse. The abbreviations are defined as follows. DT: determiner, NN: singular noun, VBZ: third-person singular verb, VBG: Gerund, NP: noun phrase, VP: verb phrase, S: sentence.

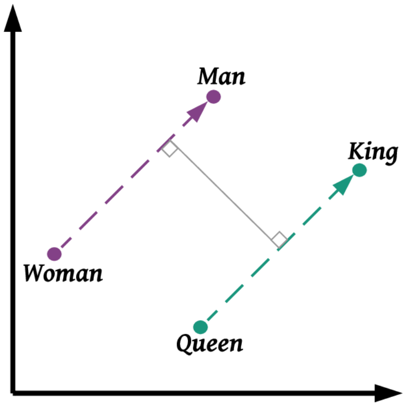

Figure 2: An illustration of how word embeddings capture similar meanings between concepts. The word embeddings for "Woman", "Man", "Queen", and "King" can be visualized as points in space. When estimated on sufficient data, the difference vectors between the embeddings of the word pairs are parallel and of the same magnitude, showing that the embeddings capture the distinction between woman and man.

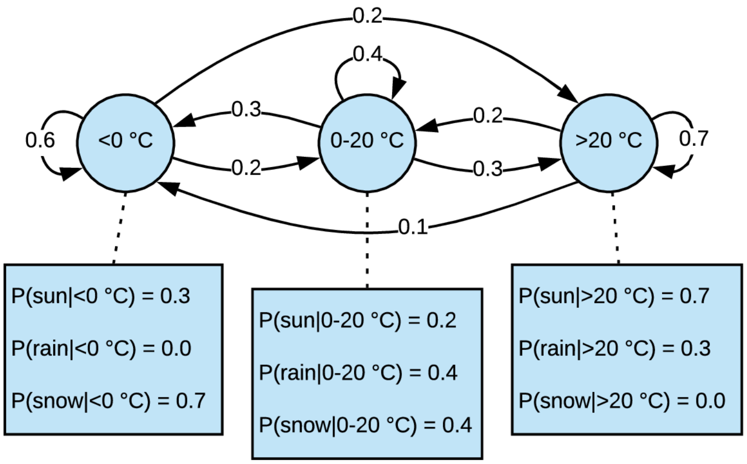

Figure 3: A simple hidden markov model (HMM) for the weather over a number of days. Together, the transition probabilities between the states of the HMM and the emission probabilities at each state encode the probability of a sequence of temperatures and conditions. Such a model could be used to predict the next day's weather given the current weather.

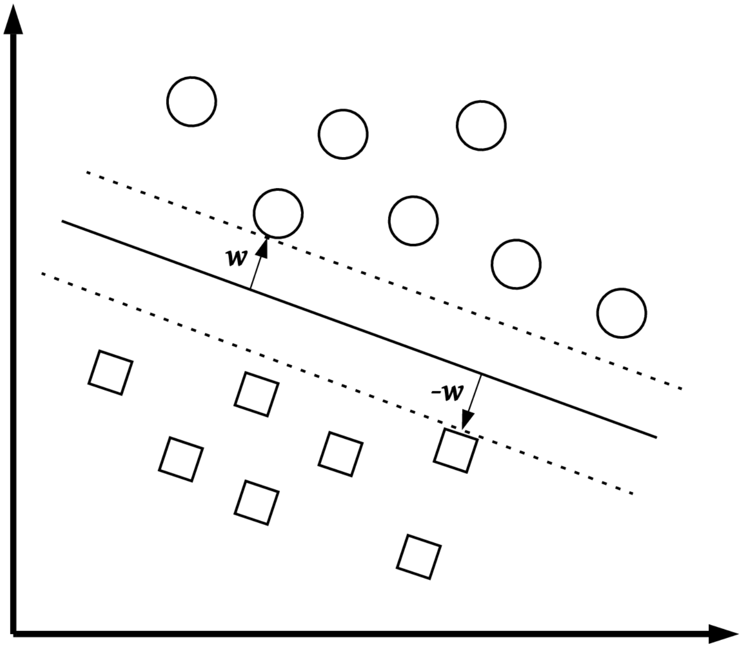

Figure 4: An illustration of the decision boundary estimated by an SVM on toy data. The data points are arranged in 2 dimensions and belong to one of two classes (the circles or squares). The optimal decision boundary estimated by the SVM is given by the solid line. The support vector is ***w*** (***-w*** in the opposite direction), which defines the distance between the boundary and the closest examples of opposite classes.

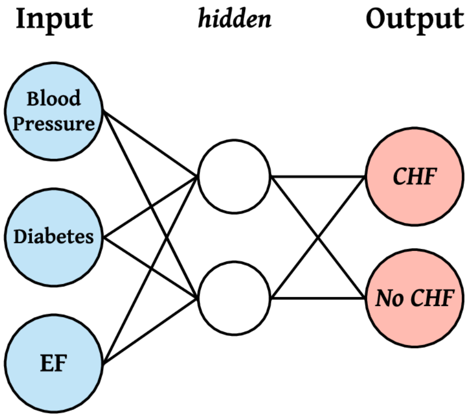

Figure 5: A feed-forward neural network with a single hidden layer for predicting whether a patient is as risk for CHF given the values of their risk factors.

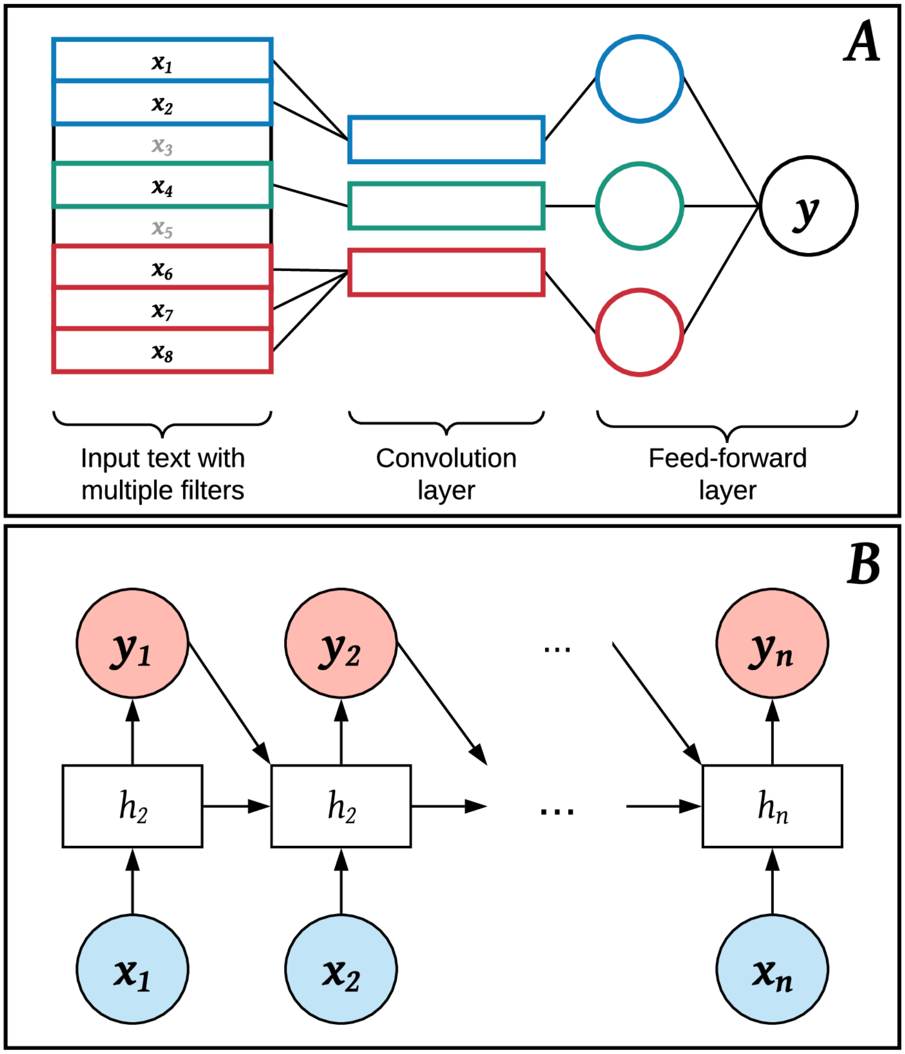

Figure 6: A) A convolutional neural network (CNN) architecture. Each word of a given input text is represented as a vector (usually a word embedding). A series of filters (i.e. convolutions) of various sizes (the blue, green, and red boxes) is applied to the input to obtain an intermediate representation, which is then passed to a feed-forward neural network for prediction. **B)** A recursive neural network (RNN) architecture. Here $\boldsymbol{x}_t$, $\boldsymbol{h}_t$, and $\boldsymbol{y}_t$ refer to the inputs, hidden units, and outputs, respectively. The values of the hidden unit at a given time t are composed with the values from the previous time step and thus the previous input is used to predict the current output. Furthermore, these hidden units can be swapped out for Long-Short Term Memory units (LSTM) or Gated Recurrent Units (GRU) to improve processing of long-range dependencies in the input.

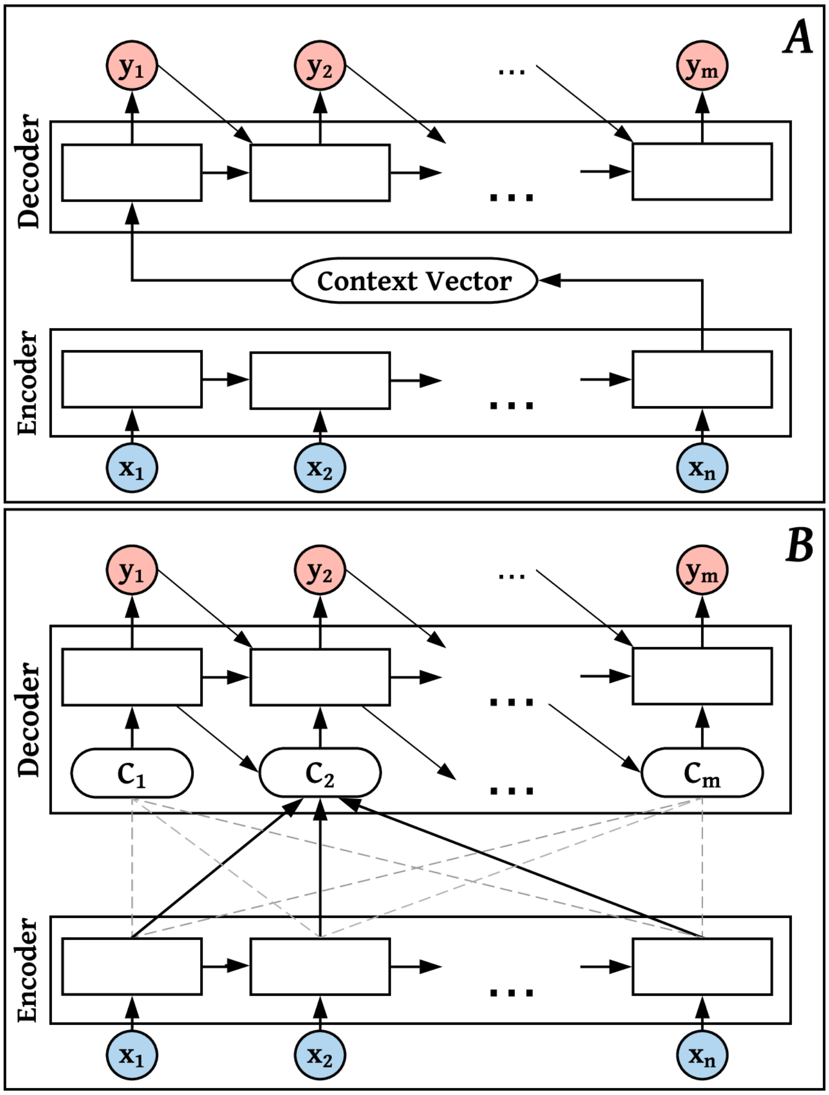

Figure 7: A) A standard encoder-decoder network. The encoder uses an RNN to compress the input into a single context vector, which is then fed into the first layer of the decoder, which is also an RNN. The lengths of the input and output need not be the same length. **B)** An encoder-decoder network with attention. As opposed to a single context vector, attention allows the network to use all the encoder hidden states when computing the output of each decoder hidden states. The encoder hidden states are weighted by parameters learned during model training.

References

Bahdanau, D., Cho, K. and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint,1409.0473.

Bhavnani, S.P., Parakh, K., Atreja, A., Druz, R., Graham, G.N., Hayek, S.S., Krumholz, H.M., Maddox, T.M., Majmudar, M.D., Rumsfeld, J.S., and Shah, B.R. (2017). '2017 Roadmap for Innovation—ACC Health Policy Statement on Healthcare Transformation in the Era of Digital Health, Big Data, and Precision Health: A Report of the American College of Cardiology Task Force on Health Policy Statements and Systems of Care'. Journal of the American College of Cardiology. 70(21), pp.2696--718.

Bodenreider, O. (2004). 'The Unified Medical Language System (UMLS): integrating biomedical terminology'. Nucleic Acids Res, 32(Database issue), pp.D267--70.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). 'Enriching Word Vectors with Subword Information'. arXiv preprint, 1607.04606.

Carrell, D.S., Schoen, R.E., Leffler, D.A., Morris, M., Rose, S., Baer, A., Crockett, S.D., Gourevitch, R.A., Dean, K.M., and Mehrotra, A. (2017). 'Challenges in adapting existing clinical natural language processing systems to multiple, diverse health care settings'. J Am Med Inform Assoc. 24(5), pp.986--991.

Chapman, W.W., Hilert, D., Velupillai, S., Kvist, M., Skeppstedt, M., Chapman, B.E., Conway, M., Tharp, M., Mowery, D.L., and Deleger, L. (2013). 'Extending the NegEx lexicon for multiple languages'. Stud Health Technol Inform, 192, pp.677--681.

Chen, T., Wu, M., and Li, H. (2019). 'A general approach for improving deep learning-based medical relation extraction using a pre-trained model and fine-tuning'. Chen, T., Wu, M., & Li, H. (2019). A general approach for improving deep learning-based medical relation extraction using a pre-trained model and fine-tuning. *Database : the journal of biological databases and curation*, baz116.

Cheng, J., Dong, L., and Lapata, M. (2016). ‘Long short-term memory-networks for machine Reading’. arXiv preprint, 1601.06733.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). ‘Learning Phrase Representations using {RNN} Encoder{--}Decoder for Statistical Machine Translation’. In Proc EMNLP, 2014, pp.1724--1734.

Choi, E., Schuetz, A., Stewart, W.F., and Sun, J. (2016). ‘Using recurrent neural network models for early detection of heart failure onset’. Journal of the American Medical Informatics Association, 24(2), pp.361--70.

Chokwijitkul, T., Nguyen, A., Hassanzadeh, H., and Perez, S. (2018). ‘Identifying Risk Factors For Heart Disease in Electronic Medical Records: A Deep Learning Approach’. In Proceedings of the BioNLP 2018 workshop, pp. 18--27.

Cortes, C., Vapnik, V. (1995). ‘Support-Vector Networks’. Machine Learning, pp.273-297.

Demner-Fushman, D., Chapman, W.W., and McDonald, C.J.. (2009). ‘What can natural language processing do for clinical decision support?’. J Biomed Inform, 42(5), pp.760--72.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’. arXiv preprint, 1810.04805.

Ferrucci, D., Lally, A., Verspoor, K., and Nyberg, E. (2009). Unstructured Information Management Architecture (UIMA) Version 1.0. [online] Available at: https://docs.oasis-open.org/uima/v1.0/uima-v1.0.pdf. [Accessed October 2019].

Friedlin, J., McDonald, C.J. (2006). 'A natural language processing system to extract and code concepts relating to congestive heart failure from chest radiology reports'. In Proc AMIA annual sym proc, 2006, p. 269.

Friedman, C., Kra, P., and Rzhetsky, A. (2002). 'Two biomedical sublanguages: a description based on the theories of Zellig Harris'. J Biomed Inform, 35(4), pp.222--35.

Geva, A., Gronsbell, J.L., Cai, T., Murphy, S.N., Lyons, J.C., Heinz, M.M., Natter, M.D., Patibandla, N., Bickel, J., and Mullen, M.P. (2017). 'A computable phenotype improves cohort ascertainment in a pediatric pulmonary hypertension registry'. The Journal of pediatrics, 188, pp.224--31.

Herrero-zazo, M., Segura-Bedmar, I., Martínez, P., and Declerck, T. (2013) 'The DDI corpus: an annotated corpus with pharmacological substances and drug-drug interactions'. J Biomed Inform, 46(5), pp.914--20.

Kexin, H., Jaan, A., and Rajesh, R. (2019). 'ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission'. arXiv preprint, 1904.05342.

Khalifa, A., Meystre, S. (2015). 'Adapting existing natural language processing resources for cardiovascular risk factors identification in clinical notes'. Journal of biomedical informatics, 58, pp.128--32.

Kilicoglu, H., Shin, D., Fiszman, M., Rosemblat, G., and Rindflesch, T.C. (2012). 'SemMedDB: a PubMed-scale repository of biomedical semantic predications'. Bioinformatics, 28(23), pp.3158--60.

Kim, Y. (2014). ‘Convolutional Neural Networks for Sentence Classification’. In Proc EMNLP, 1746--1751.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. and Soricut, R. (2019). ‘Albert: A lite bert for self-supervised learning of language representations’. arXiv preprint, 1909.11942.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). ‘Gradient-based learning applied to document recognition’. In Proc IEEE, 86(11), pp.2278--2324.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., and Kang, J. (2019). ‘BioBERT: a pre-trained biomedical language representation model for biomedical text mining’. Bioinformatics, 36(4), pp.1234--1240.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L. (2019). ‘Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension’. arXiv preprint, 1910.13461.

Liu, S., Ma, W., Moore, R., Ganesan, V., and Nelson, S. (2005). ‘RxNorm: prescription for electronic drug information exchange’. IT Professional, 7(5), pp.17--23.

Liu, S., Tang, B., Chen, Q., and Wang, X. (2016). ‘Drug-Drug Interaction Extraction via Convolutional Neural Networks’. Comput Math Methods Med, 2016, p.6918381.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). ‘Roberta: A robustly optimized bert pretraining approach’. arXiv preprint, 1907.11692.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). ‘The Expressive Power of Neural Networks: A View from the Width’. Advances in Neural Information Processing Systems, 30, pp.6231-6239.

Maintenance and Support Services Organization. (2019). ‘Medical Dictionary for Regulatory Activities (MedDRA)’. [online] Available at: https://www.meddra.org [Accessed 11 Oct. 2019].

Manning, C.D., Raghavan, P., and Schütze, H. (2008). ‘Scoring, term weighting, and the vector space model. In: Introduction to Information Retrieval’. Cambridge University Press, pp.100--123.

Maragatham, G., Devi, S. (2019). ‘LSTM Model for Prediction of Heart Failure in Big Data’. Journal of medical systems, 43(5), p.111.

Meystre, S.M., Kim, Y., Gobbel, G.T., Matheny, M.E., Redd, A., Bray, B.E., and Garvin, J.H. (2016). ‘Congestive heart failure information extraction framework for automated treatment performance measures assessment’. Journal of the American Medical Informatics Association, 24(1), pp.40--6.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). ‘Efficient estimation of word representations in vector space’. arXiv preprint, 1301.3781.

Moon, S., Liu, S., Scott, C.G., Samudrala, S., Abidian, M.M., Geske, J.B., Noseworthy, P.A., Shellum, J.L., Chaudhry, R., Ommen, S.R., and Nishimura, R.A. (2019). ‘Automated extraction of sudden cardiac death risk factors in hypertrophic cardiomyopathy patients by natural language processing’. International journal of medical informatics, 128, pp.32--8.

National Library of Medicine. (2019). ‘MetaMap - A Tool For Recognizing UMLS Concepts in Text’. [online] Available at: https://metamap.nlm.nih.gov/ [Accessed 11 Oct. 2019].

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

Pyysalo, S., Ginter, F., Moen, H., and Salakoski, T., and Ananiadou, S. (2013). ‘Distributional Semantics Resources for Biomedical Text Processing’. Proc LBM, pp.39--44.

Rindflesch, T.C., Fiszman, M. (2003). ‘The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text’. J Biomed Inform, 36(6), pp.462–77.

Rumelhart, D., Hinton, G., and Williams, R. (1986). ‘Learning representations by back-propagating errors’. Nature, 323(6088), pp.533–536.

Savova, G.K., Masanz, J.J., Ogren, P.V., Zheng, J., Sohn, S., Kipper-Schuler, K.C., and Chute, C.G. (2010). ‘Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications’. J Am Med Inform Assoc. 17(5), pp.507–13.

Sepp, H., Jürgen, S. (1997). ‘Long Short-Term Memory’. Neural Computation, 9(8), pp.1735–1780.

Soysal, E., Wang, J., Jiang, M., Wu, Y., Pakhomov, S., Liu, H., and Xu, H. (2018). ‘CLAMP - a toolkit for efficiently building customized clinical natural language processing pipelines’. J Am Med Inform Assoc, 25(3), pp.331–6.

Sutskever, I., Vinyals, O., and Le, Q.V. (2014). ‘Sequence to sequence learning with neural networks’. In Proc NeurIPS, 2014, pp. 3104—3112.

The BioMedical Information Collection and Understanding System (BioMedICUS). (2019). [online]. Available at: https://nlpie.github.io/biomedicus/. [Accessed October 11, 2019].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. (2017) 'Attention is All you Need'. In Proc NeurIPS, 2017, pp. 5998--6008.

Viani, N., Miller, T.A., Napolitano, C., Priori, S.G., Savova, G.K., Bellazzi, R., and Sacchi, L. (2019). 'Supervised methods to extract clinical events from cardiology reports in Italian'. Journal of biomedical informatics, 103219.

Wang, Y., Luo, J., Hao, S., Xu, H., Shin, A.Y., Jin, B., Liu, R., Deng, X., Wang, L., Zheng, L., and Zhao, Y. (2015). 'NLP based congestive heart failure case finding: A prospective analysis on statewide electronic medical records'. International journal of medical informatics, 84(12), pp.1039--47.

Weng, W.H., Wagholikar, K.B., McCray, A.T., Szolovits, P., and Chueh, H.C. (2017). 'Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach'. BMC medical informatics and decision making, 17(1), p.155.

Wu, S.T., Sohn, S., Ravikumar, K.E., Wagholikar, K., Jonnalagadda, S.R., Liu, H., and Juhn, Y.J. (2013). 'Automated chart review for asthma cohort identification using natural language processing: an exploratory study'. Ann Allergy Asthma Immunol, 111(5), pp.364--9.

Ye, Z., Kalloo, F.S., Dalenberg, A.K., and Kullo, I.J. (2013). 'An electronic medical record-linked biorepository to identify novel biomarkers for atherosclerotic cardiovascular disease'. Global Cardiology Science and Practice, 2013(1), p.10.

Zhang, R., Cairelli, M.J., Fiszman, M., Rosemblat, G., Kilicoglu, H., Rindflesch, T.C., Pakhomov, S.V., and Melton, G.B. (2014). 'Using semantic predications to uncover drug-drug interactions in clinical data'. J Biomed Inform, 49, pp.134--47.

Zhang, R., Ma, S., Shanahan, L., Munroe, J., Horn, S., and Speedie, S. (2018). ‘Discovering and identifying New York heart association classification from electronic health records’. BMC medical informatics and decision making. 18(2), p.48.

Zhang, Y., Chen, Q., Yang, Z., Lin, H., and Lu, Z. (2019). ‘BioWordVec, improving biomedical word embeddings with subword information and MeSH’. Scientific Data, 6(1), p.52.