Title
Learning Robust Representations for Low-resource Information Extraction

Author
Zhou, Yichao

Publication Date
2021

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Learning Robust Representations for Low-resource Information Extraction

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Yichao Zhou

2021

Abstract of the Dissertation

Learning Robust Representations for Low-resource Information Extraction

by

Yichao Zhou

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2021

Professor Wei Wang, Chair

Information extraction (IE) plays a significant role in automating the knowledge acquisition process from unstructured or semi-structured textual sources. Named entity recognition and relation extraction are the major tasks of IE discussed in this thesis. Traditional IE systems rely on high-quality datasets of large scale to learn the semantic and structural relationship between the observations and labels while such datasets are rare especially in the area of low-resource language processing (e.g. figurative language processing and clinical narrative curation). This leads to the problems of inadequate supervision and model over-fitting. In this thesis, we work on the low-resource IE algorithms and applications. We believe incorporating the supervision from domain-specific auxiliary knowledge and learning transferable representations can mitigate the deficiency of low-resource IE. Specifically, we explore pre-training domain-specific deep language models to acquire informative word/sentence embeddings to curate clinical narratives. We experiment with multi-modal learning techniques to recognize humor and to recommend keywords for advertisement designers. We also extract attributes of interest from the semi-structured web data by building transferable knowledge representations across different websites. For more applications of the low-resource IE, we build a COVID-19 surveillance system by inspecting users' daily social media data. Extensive experiments prove that our algorithms and systems outperform the state-of-the-art approaches and are of impressive interpretability as well.

The dissertation of Yichao Zhou is approved.

Cho-Jui Hsieh

Yizhou Sun

Kai-Wei Chang

Wei Wang, Committee Chair

University of California, Los Angeles

2021

To my wife Hanqiu,

for her deep love

To my parents Donghua and Yuling,

for their unconditional support

To my cats, Sheldon, Krissy and Pearl,

for their warm company

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my heartfelt gratitude and respect to my advisor Professor Wei Wang, for her guidance and tremendous support over the past five years. In these years, she has not only polished up my research skills but has also influenced my views towards life. I would also like to thank my PhD thesis committee members: Professor Kai-Wei Chang, Professor Yizhou Sun, and Professor Cho-jui Hsieh. At multiple points during my PhD, I received crucial comments from them which shaped my PhD projects. In particular, I had the fortune of collaborating with Professor Kai-Wei Chang and members of his group on multiple NLP research topics. I sincerely thank Professor Kai-Wei Chang for this fruitful collaboration.

Beyond my committee, I am lucky to have worked with and learned from Dr. Shaunak Mishra, Dr. Jelena Gligorijevic, Dr. Manisha Verma and Dr. Narayan Bhamidipati during my first and second internships at Yahoo! Research, and Dr. Sandeep Tata, Sheng Ying, Dr. Nguyen Vo, Dr. Nick Edmonds at Google AI. They broadened my research horizon and showed me the wide-open world outside the school.

During my PhD years, I had the incredible fortune of working with my friends who are also amazingly talented scholars at UCLA: Dr. Chelsea Ju, Dr. Ruirui Li, Dr. Wenchao Yu, Jyun-yu Jiang, Dr. Cheng Zheng, Xiusi Chen, Zeyu Li, Guangyu Zhou, Junheng Hao, Jieyu Zhao, Dr. Jin Wang, Yunsheng Bai, Yu Yan, Zijie Huang, whose helpful discussions have guided me along the way. I also take this opportunity to thank my collaborators, Professor Peipei Ping, Professor Nanyun Peng, Dr. J Harry Caufield, Dr. David A Liem, Dr. Anders O Garlid, Dr. Giuseppe M Mazzeo, Xinxin Huang, Vincent Kyi, Patrick Tan, Rujun Han, and all their efforts in various interesting projects.

Moving to the United States for graduate studies had a sad but inevitable consequence: distance from my family and friends in China. I would love to thank my parents Dr. Donghua Zhou, Dr. Yuling Zhou, and my wife Hanqiu Xia for their unconditional sacrifice and support throughout my life. Words cannot express how much I appreciate and love them.

Vita

2011-2015	B.S. in Electrical Engineering, Southeast University, Nanjing, China
2015-2017	M.S. in Electrical Engineering, UCLA, Los Angeles, US
2018, 2019	Summer Research Intern, Yahoo Research, Sunnyvale, US
2020	Summer Research Intern, Google AI, Mountain View, US
2016-2021	Graduate Student Researcher & Teaching Fellow, Computer Science Department, UCLA, Los Angeles, US

Publications

Yichao Zhou, Jyun-yu Jiang, Xiusi Chen, Wei Wang. #StayHome or #Marathon? Social Media Enhanced Pandemic Surveillance on Spatial-temporal Graphs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (CIKM 2021).

Yichao Zhou, Yu Yan, Rujun Han, J H Caufield, Kai-Wei Chang, Yizhou Sun, Peipei Ping, Wei Wang. Clinical Temporal Relation Extraction with Probabilistic Soft Logic Regularization and Global Inference. In *Proceedings of 35th AAAI Conference on Artificial Intelligence* (AAAI 2021).

Yichao Zhou, Wei-Ting Chen, Bowen Zhang, D Lee, J H Caufield, Kai-Wei Chang, Yizhou Sun, Peipei Ping, Wei Wang. CREATe: Clinical Report Extraction and Annotation Technology. In *Proceedings of the 37th IEEE International Conference on Data Engineering* (ICDE 2021).

Yichao Zhou, Ying Sheng, Nguyen Vo, Nick Edmonds, Sandeep Tata. Simplified DOM Trees for Transferable Attribute Extraction from the Web. In *Arxiv* 2021. Pre-print.

Rujun Han, **Yichao Zhou**, Nanyun Peng. Domain Knowledge Empowered Structured Neural Net for End-to-End Event Temporal Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2020).

Shaunak Mishra, Manisha Verma, **Yichao Zhou**, Kapil Thadani, Wei Wang. Learning to Create Better Ads: Generation and Ranking Approaches for Ad Creative Refinement. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (CIKM 2020).

Yichao Zhou, Jyun-Yu Jiang, Jieyu Zhao, Kai-Wei Chang, Wei Wang. "The Boating Store Had Its Best Sail Ever": Pronunciation-attentive Contextualized Pun Recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (ACL 2020).

Yichao Zhou, Shaunak Mishra, Manisha Verma, Narayan Bhamidipati, Wei Wang. Recommending Themes for Ad Creative Design via Visual-linguistic Representations. In *Proceedings of The Web Conference 2020* (WWW 2020).

Yichao Zhou*, Jyun-Yu Jiang*, Kai-Wei Chang, Wei Wang. Learning to Discriminate Perturbations for Blocking Adversarial Attacks in Text Classification. 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).

Yichao Zhou, Shaunak Mishra, Jelena Gligorijevic, Tarun Bhatia, Narayan Bhamidipati. Understanding Consumer Journey Using Attention Based Recurrent Neural Networks. In *Proceedings of the 25th International Conference on Knowledge Discovery & Data Mining* (KDD 2019).

J Harry Caufield, **Yichao Zhou**, Yunsheng Bai, David A Liem, Anders O Garlid, Kai-Wei Chang, Yizhou Sun, Peipei Ping, Wei Wang. A Comprehensive Typing System for Information Extraction from Clinical Narratives. In *medRxiv* 2019. Pre-print.

Yichao Zhou, Chelsea Ju, J. Harry Caufield, Kevin Shih, Calvin Chen, Yizhou Sun, Kai-Wei Chang, Peipei Ping, Wei Wang. Clinical Named Entity Recognition using Contextualized Token Representations. In *Arxiv* 2019. Pre-print.

Jieyu Zhao, **Yichao Zhou**, Zeyu Li, Wei Wang, Kai-Wei Chang. Learning Gender-neutral Word Embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2018).

CHAPTER 1

Introduction

1.1 Motivation

Imagine that you are a data analyst with a health organization that recently forecasts the pandemic trends and determines the essential risk factors that may aggravate the infection of Coronavirus. You will at least need to know some prevailing topics that people are discussing on the Internet related to the coming events, famous influencers, or social phenomena. Fortunately, these can be found in the enormous social media data like this one:

Runners of LA marathon were advised to wash their hands before the race and not to shake hands with other participants or the public. Hand sanitizer was also available for runners along the course. #LAMarathon #StaySafe March.14.2020 Los Angeles

where we recognized a coming event *LA marathon* and some activities such as *wash hands* and *shake hands* which may influence the transmission of the virus in the area of *Los Angeles*.

Imagine that you are a cardiovascular doctor looking for related clinical case reports for reference. Specifically, those case reports describing patients who were admitted into the hospital because of fever and cough and then detected tumor cells in a Magnetic Resonance Imaging test are exactly what you are interested in. Ideally, the doctor can quickly find the reference with access to a powerful database recording the chronological diagnosis and treatment of every patient. To build such a magical database, we need to acquire the diseases, diagnostic procedures, laboratory tests, etc., from the large scale of customized clinical reports and as well identify the temporal relationships among these significant entities.

The patient is a *47-year-old women* with long-term use of *glucocorticoids*. She was *confirmed with COVID-19* by tested *positive of antibody* and then *admitted to the hospital*.

The above example demonstrates the information extraction from clinical narratives. In this task, entities (e.g. *confirmed with COVID-19*, *positive of antibody*, *admitted to the hospital*) are extracted and temporal relations *(positive of antibody, BEFORE, admitted to the hospital)* is identified.

We summarize the aforementioned knowledge acquisition processes as **Information Extraction (IE)** – extracting information from *unstructured or semi-structured textual sources* to enable finding *entities* and *relationships* as well as classifying and storing them in a database. IE is essential for increasing the accessibility of knowledge through search engines, conversational AI systems, and medical research tools.

Traditional IE systems rely on a large amount of annotated datasets to build complex but straightforward feature sets by making use of the linguistic attributes such as Part-of-speech (POS) tags of words and syntactic structures of sentences to train supervised machine learning models. The models are used for capturing and memorizing the hidden relationships between the annotated labels and these features. However, we can rarely find datasets of large size or high quality when conducting information extraction tasks in some highly specialized domains such as bioinformatics, clinical science, and social media. It is not always affordable and efficient to clean, annotate, and curate the data in these domains. Extracting knowledge from some languages that are used by only a few people nowadays or from special types of languages such as figurative language (e.g. expressing humor, irony, metaphor, etc.) is an extremely important task but also of dramatic challenge. We summarize the information extraction under these conditions as **Low-resource Information Extraction** and in this thesis, we explore robust and highly interpretable methodologies to overcome the aforementioned problems and aim to build end-to-end information extraction systems for real applications.

1.2 Thesis Statement

This thesis demonstrates that learning with domain-specific auxiliary supervision and word knowledge can overcome the challenges caused by the low-resource circumstances in the information extraction tasks. Moreover, leveraging transfer learning techniques (e.g. few-shot learning, multi-modal learning) to achieve powerful word and sentence representations that incorporate out-of-domain knowledge can also help build generalizable and interpretable information extraction models.

1.3 Contributions

This dissertation advances three approaches to learn semantically and syntactically powerful representations for information extraction tasks: (1) pre-training domain-specific contextualized language models to facilitate down-streaming information extraction applications; (2) learning robust representations by incorporating auxiliary supervision such as the structural relationships among knowledge; (3) learning robust representations with transfer learning techniques such as few-shot learning and multi-modal learning. We detail the contributions as follows.

The first research issue comes from the fact that static language models fail to provide precise word embeddings when the words have multiple semantic meanings in different contexts. Pre-training an informative contextualized language model provides powerful word and sentence embeddings to facilitate the information extraction processes including named entity recognition and relation extraction. In this thesis, we collect millions of clinical case reports and all COVID-19 relevant text corpus including scholarly papers and social media data to pre-train three deep language models, `Clinical-ELMo`, `Clinical-Flair` and `Corona-Bert`. We release the models to facilitate further researches in the biomedical community. We also present experimental results to show the effectiveness of domain-specific language models, compared to the static embeddings and languages models learned on a

general-domain corpus.

The second research direction addresses how to leverage auxiliary supervision to learn robust representations for information extraction tasks. To take figure 2.1 as an example, useful dependency structures exist among multiple temporal relationships, i.e. given the facts that event (a) happens before event (b) and event (b) happens before event (d), we can infer that event (a) also happens before event (d). The dependency structures like this example are the key enabler of classifying the temporal relations. We propose to leverage probabilistic soft logic (PSL) rules to model this auxiliary knowledge. Specifically, we summarize the common transitivity and symmetric dependency patterns of temporal relations as PSL rules and penalize the temporal relation instances that violate any of those rules in the training stage. Different from the traditional approaches using integer linear programming to solve the hard constraints, our solution requires no off-the-shelf solver and conducts the experiments with linear time complexity.

The third research direction focuses on applying transfer learning techniques to learn robust representations for information extraction tasks. We experiment with this approach on three tasks as follows.

- In the task of extracting attributes from semi-structured web documents, we aim to recognize attributes of interest like {title, author, ISBN13, publisher} of a book or {post date, location, company} of a job description from the web pages built with HTML files and DOM trees. We explore two novel few-shot settings of attribution extraction: (i) given a few labeled seed websites from a given domain and we extract the attributes from unseen websites from the same domain; (ii) given a few labeled seed websites from a given domain (say A) and addition labeled websites from a different domain (say B) and we extract attributes from unseen websites in domain A. To tackle the two challenging few-shot scenarios, we learn contextual representations for DOM tree nodes in the web page by leveraging the local tree structures. Extensive experiments show our approach outperforms the state-of-the-art methods largely and the representations are proven robust and transferable within the same domain or across different domains.

- In another work, given the heterogeneous data corpus including ad images and texts associated with the ads, we aim to design a theme (keyphrase) recommendation system for ad creative strategists. We build a cross-modality encoder to train visual-linguistic representations for understanding the relationships between the images and texts. In the experiments, the cross-modal representations show better performance compared to separate image and text representations.

- In the pun detection and location tasks, we aim to detect whether puns exist in a given sentence and which words are the puns. Traditional methods employ word sense disambiguation techniques to identify the equitable intention of words in utterances or make use of external knowledge bases such as WordNet to determining word senses of pun words. However, these approaches cannot tackle heterogeneous puns with distinct word spellings and the knowledge bases often incorporate very limited words. We propose a pronunciation-attentive contextualized pun recognition model to build multi-modal representations for pun detection and location. After considering the contexts and the phonological properties of words, the embeddings can be more beneficial to pun recognition.

1.4 Dissertation Outline

The rest of this dissertation is arranged as follows. Chapter 2 summarizes the related works for some basic tasks of information extraction. chapter 3 describes an end-to-end pipeline of clinical information extraction and demos a clinical case report system. We discuss the effectiveness of utilizing auxiliary knowledge to enhance the temporal relation extraction model. We also show the improvement from our pre-trained contextualized language model. Chapter 4 presents our work on multi-modal representation learning approaches to solve the pun recognition and theme recommendation tasks. Chapter 5 introduces a transferable framework to tackle two few-shot attribute extraction tasks. A simple and effective contextual node representation learning method is proposed. Chapter 6 propose an application

of information extraction: forecasting pandemic trends and detecting risk factors based on the named entities and relationships extracted from the social media corpus. Chapter 7 concludes the thesis with discussions and an outlook for future work.

1.5 Other Publications

During my PhD study, I have published other research works related to the topics of representation learning and information extraction in various domains. I briefly overview the work with reference. These projects either propose state-of-the-art methods to improve the performance of NLP tasks or build end-to-end systems with advanced information extraction techniques.

To extract event temporal relations and address the problem of biased predictions derived from a limited amount of training data, [99] proposes a framework that enhances deep neural network with distributional constraints constructed by probabilistic domain knowledge.

Adversarial attacks against machine learning models have threatened various real-world applications such as spam filtering and sentiment analysis. [272] builds a novel framework to recognize and adjust malicious perturbations, thereby blocking adversarial attacks.

In the ads targeting task, the paths of online users towards a purchase event (conversion) can be very complex but significant. [276] introduces novel attention mechanisms to automatically assign users and activities to different funnel stages. In another work for ad creative refinement task, [182] learns robust multi-modal representations with visual and textual inputs to recommend keyphrases for new ad text and to assign tags for selecting new ad images.

To aid navigation of the analytical tools fragmented across the web, [238, 254] create a novel information extraction-enhanced platform, AZTEC, that empowers users to simultaneously search a diverse array of digital resources including databases, standalone software, web services, publications, and large libraries composed of many interrelated functions.

CHAPTER 2

Background

This chapter provides background information necessary for understanding the rest of this dissertation. This chapter is organized as follows:

Section 2.1 introduces some information extraction tasks including extraction from narrative text and extraction from semi-structured documents. Specifically, I discuss a standard knowledge extraction pipeline from narrative text which is composed of the named entity recognition and the relation extraction tasks. I also overview the attribute extraction task in the semi-structured extraction problem. Section 2.2 introduces the basic machine learning concepts and models for information extraction tasks including statistic models and deep neural networks. I also discuss the challenges of information extraction under low-resource circumstances and advance strategies to tackle the problem including transfer learning techniques and deep language models.

2.1 Information Extraction

An information extraction system responds to a user's information need to identify a subset of information within a document, which is not necessary a summary or gist of the contents of the document. Rather it corresponds to predefined generic types of information of interest and represents specific instances found in the text [192]. For example, the user may be interested in identifying and databasing all the disease-related expressions from doctors' notes to build a structural ontology. The user may want to collect all the book names and the publishers from several book websites. We regard the first scenario as information

extraction from unstructured narrative while the second as extraction from semi-structured documents considering the HTML formats of web pages.

2.1.1 Extraction from Unstructured Narratives

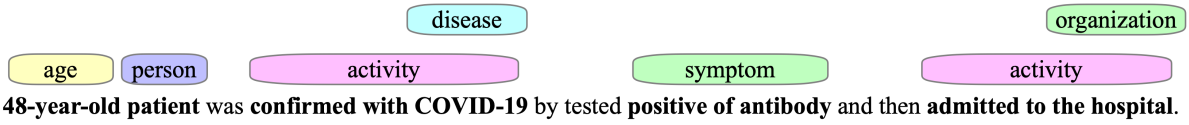


Figure 2.1: An example with extracted named entities.

Named Entity Recognition is one of the important tasks of IE used to extract (i) domain-independent entities such as organization, person, and location; (ii) domain-specific entities such as problem, test, and treatment in the clinical domain [9]. The task is formulated as classifying words or phrases into pre-characterized classes. Named entity recognition usually serves as the first step while an essential step in the information extraction pipeline. In this task, we aim to detect the boundaries of entities in the sentences and to simultaneously recognize the categories of these entities. Figure 2.1 illustrates identifying the entities of interest from an unstructured narrative. Entities can either be a word (e.g. *48-year-old* and *COVID-19*) or phrases composed of a couple of words (e.g. *confirmed with COVID-19*). The nested recognition patterns exist in the results (e.g. *COVID-19* and *confirmed with COVID-19*). The named entity recognition tasks can be applied in different domains such as recognizing the puns in the figurative language, detecting the adversarial examples from attacked data in cybersecurity, and building knowledge bases based on clinical entities and events.

Relation Extraction is the following subtask in the information extraction pipeline, extracting substantial relationships between the extracted named entities. Open relation extraction extracts open-domain relation triples from the sentence, representing a subject, a relation, and the object of the relation. As shown in Figure 2.2, *(patient, ADMITTED TO, hospital)* and *(patient, CONFIRMED WITH, COVID-19)* are two typical examples of

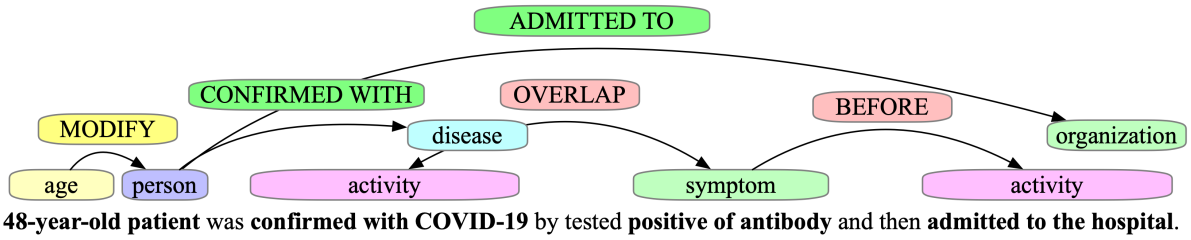


Figure 2.2: An example with identified relations.

open relation extraction. On the other hand, closed relation extraction pre-defines a relation set which not necessarily appear in the sentence. *(48-year-old, MODIFY, patient)* in Figure 2.1 is an example of closed relation extraction and *MODIFY* defines an attributive relationship between a subject and its property. In this thesis, we mainly discuss a special type of closed relation extraction, the temporal relation extraction, determining the chronological relationships between event objects. In this example, we identify two temporal relationships: *(confirmed with COVID-19, OVERLAP, positive of antibody)* and *(positive of antibody, BEFORE, admitted to the hospital)*. Note that hidden dependencies may exist in different temporal relationships, thus we do not need to identify the relationship between any two event objects. In this case, the relation entry *(confirmed with COVID-19, BEFORE, admitted to the hospital)* can be inferred from the previous two relation entries.

2.1.2 Extraction from Semi-structured Documents

Web information extraction processes a vast amount of semi-structured content from the web and has drawn a lot of attention from the data mining research community [53, 153, 80]. Different from the narrative texts, the web page is rendered to display in a browser-based on the source data, a Document Object Model (DOM) tree [94]. Four broad categories of web information extraction tasks can be summarized. They are attribute (entity) extraction, relation extraction, composite extraction, and application-driven extraction. Attribute extraction targets to identify named entity mentions such as book price, phone number, movie title from web documents. Though this task is intuitive to describe, the high-quality corpus annotation requires time-consuming human-crafted rules and dictionaries [148, 102, 46, 197].

Relation extraction associates pairs of named entities and identifies a pre-defined relationship between them. Composite extraction aims to extract more complex concepts such as reviews, opinions, and sentiment mentions [58, 227, 223], while application-driven extraction includes a broad spectrum of application scenarios such as web representation learning, PDF information extraction using OCR techniques, anomaly detection of web-based attacks and so on [247, 168, 127].

2.2 Machine Learning for Information Extraction

In this section, we explain some common machine learning strategies for two important information extraction tasks: named entity recognition and relation extraction. Overall, a machine learning algorithm aims to learn a function $f(\mathbf{x}, \theta)$ that maps an input space X to an output space Y. The mapping functions are usually parameterized with weights θ. In the supervised learning scenario, given labeled dataset $D = (\mathbf{x}, \mathbf{y})$, we minimize the empirical risk $J(D) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i))$, to find the parameters $\theta^* = \text{argmin}_\theta J(D)$ with optimization techniques such as stochastic gradient descent [203] and Adam stochastic optimization [123]. In the inference stage, output variables can be assigned with maximum a posteriori (MAP) inference which computes $y^* = \text{argmax}\, S(\mathbf{x}, \mathbf{y}, \theta)$, where S is a scoring function.

2.2.1 Task Formulation

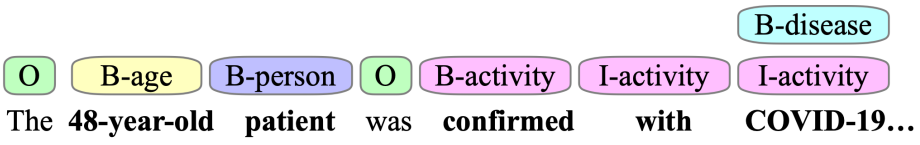


Figure 2.3: An example of extracted entities with *BIO* tagging format.

Named Entity Recognition is formulated into a sequence tagging machine learning task nowadays, i.e. given a sequence of words, the model assigns a label to each word according the word-level and sentence-level features. Two tagging policies are prevailing: (1) *BIO*

tagging [275] where B denotes the beginning of an entity, I representing inside an entity and O means outside an entity; (2) *BIOES* tagging where E additionally marks the end of an entity and S denotes the entity is a single word. Figure 2.3 shows one example of *BIO* tagging format. The single-word entity *48-year-old* is tagged with *B-age* while the middle word *with* of phrasal entity *confirmed with COVID-19* is tagged with *I-activity*. Without loss of generality, we make use of the *BIO* tagging policy in this thesis.

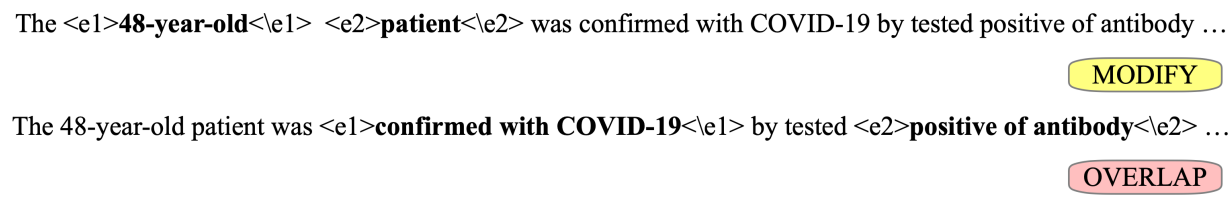


Figure 2.4: An example of relation extraction formulation.

Relation Extraction[*] is formulated into a sentence classification task, i.e. given a sequence of words and special marks of two entities in the sentence, the model assign a label to the sentence. There are multiple ways of marking the entities in the sentence. We adopt the method shown in Figure 2.4 by appending XML tags around the entities [279]. Note that the marking order of two entities, i.e. the order of <e1> and <e2>, matters in the classification such as *(48-year-old, MODIFY, patient)*. Some relations like *OVERLAP* is bidirectional, thus we may swap <e1> and <e2> in the second example. [269] mark the entities using position embeddings, which is an alternative of XML tags.

2.2.2 Prevailing Learning Algorithms

Hidden Markov Model (HMM) [207] and Conditional Random Fields (CRF) [133] are prevailing statistical methods for the boundary detection and named entity recognition tasks. HMM provides a joint distribution over the sentence/tags with an assumption of dependency between adjacent tags. Parameters of the HMM model include an emission matrix A, where $P(X_t = k|Y_t = j) = A_{j,k}, \forall t, k$, a transmission matrix, B, where

[*]In this thesis, we only discuss the closed relation extraction problem.

$P(X_t = k|Y_{t-1} = j) = B_{j,k}, \forall t, k$, and a initial probability, C, where $P(Y_1 = k), \forall k$. HMM models the joint distribution of the sequence by

$$y_0 = \text{START}, p(\mathbf{x}, \mathbf{y}|y_0) = \prod_{t=1}^{T} p(x_t|y_t)p(y_t|y_{t-1}) = \prod_{t=1}^{T} A_{y_t,x_t} B_{y_{t-1},y_t}. \tag{2.1}$$

To learn the parameters in HMM, Maximum Likelihood Estimation (MLE) is used to estimate the matrices $\hat{A}, \hat{B}, \hat{C}$ by

$$\hat{A}, \hat{B}, \hat{C} = \underset{A,B,C}{\text{argmax}} \sum_{i=1}^{N} \log p(x_i, y_i|A, B, C). \tag{2.2}$$

In the inference stage, the marginal inference such as the Forward-Backward Algorithm [146] is applied to compute the marginal distribution for a hidden state, given a sequence of observations while the MAP inference such as Viterbi Algorithm [85] is used to find the most-likely sequence of hidden states.

However, HMM models capture the dependencies between each state and only its corresponding observation and the mismatch between the learning objective function and prediction objective function leads to sub-optimal performances, i.e., HMM, as a generative model, learns the joint distribution of states and observations $P(X, Y)$ while in the prediction task, we seek for the conditional probability $P(Y|X)$. The linear-chain CRF model is similar to an HMM, except that CRF learns a discriminative model and its factors are not necessarily probability distributions,

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{k=1}^{K} \phi_{\text{em}}(y_k, x_k)\phi_{\text{tr}}(y_k, y_{k-1}) = \frac{1}{Z(\mathbf{x})} \prod_{k=1}^{K} \exp(\theta \cdot \mathbf{f}_{em}(y_k, x_k))\exp(\theta \cdot \mathbf{f}_{tr}(y_k, y_{k-1})). \tag{2.3}$$

The Gradient-based Algorithm can be employed to learn the parameters in the CRF model and similarly, the Forward-Backward Algorithm and Viterbi Algorithm are required to compute the partition function and the highest-probability output sequence.

Recurrent Neural Networks [221] is a type of artificial neural network which uses sequential data or time-series data. This machine learning algorithm is commonly used for ordinal or temporal problems, such as named entity recognition and relation extraction.

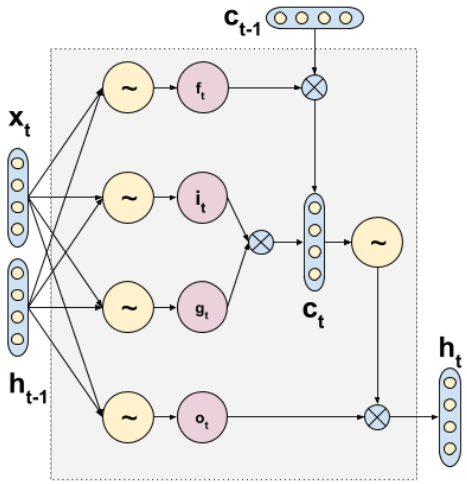


Figure 2.5: A long short term memory unit ($\otimes$:element-wise multiplication, $\sim$:activation function).

RNN is capable of processing variable-length sequences of inputs using its internal memory unit. One problem of the RNN model is known as gradient exploding and vanishing. The gradient may either grow fast or decay to zero if the propagation sequence is too long, which may increase the training difficulty. To control the gradient issue and choose useful information to transit, the long short term memory (LSTM) [105] is taken into consideration for optimizing the model. LSTM was introduced as a solution to the vanishing and exploding gradient problem. Each LSTM cell transit unit state $\mathbf{h}_t$ and cell state $\mathbf{c}_t$ among hidden units. As depicted in Figure 2.5, LSTM blocks contain three *Gates* implemented by using the logistic function to compute a value between 0 and 1, which are intelligent in controlling the information flow into or out of memory. In equation 1-3, when processing the t_{th} word of a sequence, *Input Gate* $\mathbf{i}_t$ controls the extent to which a new value flows into the memory; *Forget Gate* $\mathbf{f}_t$ controls the extent to which a value remains in memory; And *Output Gate* $\mathbf{o}_t$ controls the extent to which the value in memory is used to compute the output activation of the block. A candidate value $\mathbf{g}_t$ is computed for the state of memory cells by equation 2.7. σ denotes a Sigmoid function.

$$\mathbf{i}_t = \sigma(W_i \cdot \mathbf{x}_t + U_i \cdot \mathbf{h}_{t-1} + b_i) \tag{2.4}$$

$$\mathbf{o}_t = \sigma(W_o \cdot \mathbf{x}_t + U_o \cdot \mathbf{h}_{t-1} + b_o) \tag{2.5}$$

$$\mathbf{f}_t = \sigma(W_f \cdot \mathbf{x}_t + U_f \cdot \mathbf{h}_{t-1} + b_f) \tag{2.6}$$

$$\mathbf{g}_t = \tanh(W_g \cdot \mathbf{x}_t + U_g \cdot \mathbf{h}_{t-1} + b_g) \tag{2.7}$$

$$\mathbf{c}_t = \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \mathbf{g}_t \tag{2.8}$$

$$\mathbf{h}_t = \mathbf{o}_t \otimes \tanh(\mathbf{c}_t) \tag{2.9}$$

Current cell state $\mathbf{c}_t$ is the combination of previous cell state $\mathbf{c}_{t-1}$ and candidate value $\mathbf{g}_t$ weighted by *Forget Gate* $\mathbf{f}_t$ and *Input Gate* $\mathbf{i}_t$ (Equation 2.8). Final output of this hidden unit is combined by *Output Gate* $\mathbf{o}_t$ and cell state $\mathbf{c}_t$ through an activation (Equation 2.9).

As an alternative, the Gated Recurrent Unit (GRU) [63] has a slightly simpler architecture. Empirically, GRU is proven more computationally efficient than LSTM while can achieve competitive performances for the machine learning tasks [64]. Equation 2.10 to 2.13 define the information flow over a GRU network.

$$\mathbf{r}_t = \sigma(W_r \cdot \mathbf{x}_t + U_r \cdot \mathbf{h}_{t-1}) \tag{2.10}$$

$$\mathbf{z}_t = \sigma(W_z \cdot \mathbf{x}_t + U_z \cdot \mathbf{h}_{t-1}) \tag{2.11}$$

$$\mathbf{h}_t = \tanh(W_h \cdot \mathbf{x}_t + U_h \cdot \mathbf{h}_{t-1}(\mathbf{r}_t \otimes \mathbf{h}_{t-1})) \tag{2.12}$$

$$\mathbf{h}_t = \mathbf{z}_t \otimes \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \otimes \mathbf{h}_t \tag{2.13}$$

where $\mathbf{r}_t$, $\mathbf{z}_t$ denotes the *Reset Gate* and *Update Gate*, respectively.

2.2.3 Learning under Low-resource Conditions

Deep Language Model. Word embeddings are fixed-length, dense, and distributed representations for words [12], mainly learned with static language models which focus on predicting the next word given the previous words. Word embeddings are capable of encoding accurate syntactic and semantic word properties [176, 199], of which Word2Vec and GloVe are the most influential models. Either neural networks or word-context matrices are leveraged to

encode the relationships between words and their contexts. However, it is sub-optimal to assign each word with one fixed embedding vector and static language models fail to adjust the word embeddings according to different contexts. Contextual embeddings [71, 10, 201], such as ELMo, BERT, and Flair move beyond the static word representations and achieve breaking improvements on a wide range of natural language processing and data mining tasks [157]. The ELMo model extracts context-dependent representations with a bidirectional LSTM-based language model, i.e. a forward LSTM and a backward LSTM are designed to encode both left and right contexts, respectively. Besides, Flair learns contextual string embeddings which are pre-trained without any explicit notion of words and are extremely useful in some character-level tasks such as sequence labeling. Different from building an auto-regressive language model like ELMo and Flair, BERT proposes a masked language modeling objective together with Word-piece tokenization [260] to formulate the model into a transformer-based auto-encoder [248]. BERT also uses the next-sentence-prediction objective to improve the understanding over neighboring sentences, which is essential in the question answering and summarizing tasks. Overall, the contextualized embeddings pre-trained from the deep language models assign each word a representation based on the context, thereby encoding the knowledge across languages. We believe pre-training domain-specific deep language models can facilitate down-streaming information extraction applications under low-resource conditions. Recently, researchers pre-trained a number of deep language models using biomedical literature corpus [138, 275, 31], financial service corpus [18], multi-lingual corpus [147], and so on.

Multi-modal Learning. Modality refers to how something happens or is experienced and obviously, our experience of the world is multi-modal, i.e. we can see the objects, read the texts, hear the sounds, taste the favors, etc. [28]. The purpose of multi-modal learning is to build machine learning models that are capable of gathering and interpreting signals from various modalities together. Usually, we adopt multi-modal learning to learn cross-modality representations to help the modality with less annotated data. Recently, multi-modal learning is drawing the attention of NLP researchers. For instance, image data are

often associated with text explanations where keywords in the text describe the objects in the image. Images are also leveraged to enhance the expression of ideas in an article. In practice, it is challenging to (1) highlight the complementarity and synchrony between modalities; (2) translate data from one mobility to another or transfer the knowledge between modalities; (3) recognize the correspondence between elements from different modalities; and (4) combine multi-modal signals for regression or classification tasks [184]. [237, 114] combine multi-modal embeddings to solve the visual question answering tasks with deep neural networks such as LSTM, modular neural networks, and transformer. [216, 109, 229] improve the performance of machine translation by jointly learning the cross-modality representations from linguistic, acoustic, and visual signals. We believe applying multi-modal learning to tackle the problems in the area of low-resource information extraction is very promising considering cross-modality signals are capable of enhancing the representations learned from the limited mono-modal annotated dataset.

Few-shot Learning. To tackle the problem that information extraction models are hampered when the datasets are small, few-shot learning is proposed as another promising strategy. The few-shot learning is explained as generalizing from a few examples, i.e., using prior knowledge to rapidly generalize to new tasks containing only a few samples with labels [267, 107]. [257] summarized three categories of few-shot learning methods which are (1) prior knowledge enhanced supervised learning; (2) prior knowledge-based hypothesis space reduction; and (3) prior knowledge engaged optimal hypothesis searching. Three categories are in the perspectives of data, model, and algorithm, respectively. Specifically, augmenting data by transforming samples from the training set, other labeled or unlabeled datasets is the common strategy of data-perspective few-shot learning [132, 243, 78]. The model-perspective few-shot learning covers a wide range of advanced approaches such as parameter sharing, learning hybrid embeddings, and learning with external memory [14, 32, 234]. In the algorithm-perspective, researchers refine existing or meta-learned parameters or learns task-specific optimizers [16, 82, 106].

CHAPTER 3

Clinical Information Extraction

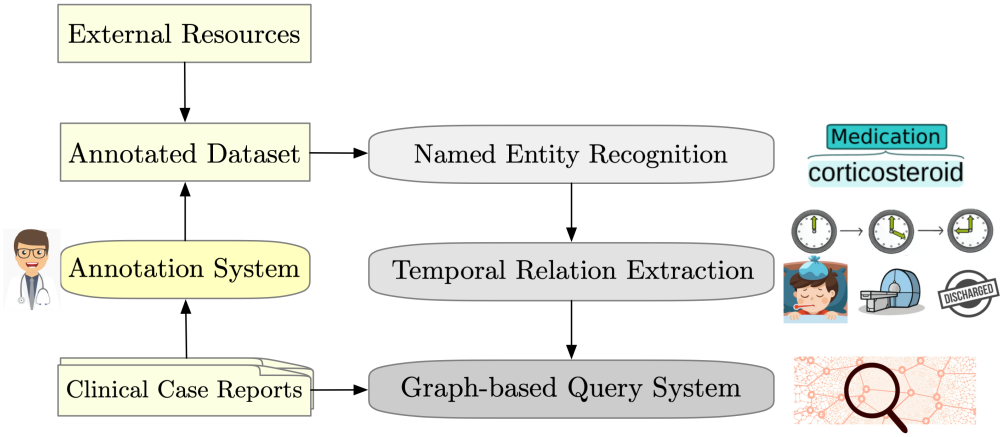


Figure 3.1: The pipeline of clinical information extraction.

Clinical case reports (CCRs) are written descriptions of the unique aspects of a particular clinical case, playing an essential role in sharing clinical experiences about atypical disease phenotypes and new therapies. In this chapter, we introduce our work on building a clinical information extraction system for extracting, indexing, and querying the contents of CCRs. We start by building a comprehensive typing system to facilitate clinical experts to annotate up-to-date case reports and to create new entity and relation types in Section 3.1. We then pre-train two clinical language models to enhance the named entity recognition model for automatically extracting entities of interest from the case reports in Section 3.2. Furthermore, we focus on the temporal relation extraction task, i.e. identifying the chronological orders over the extracted entities and events in Section 3.3. Last but not least in Section 3.4, we demo our `CREATe` search engine which is an end-to-end system incorporating algorithms for extracting, indexing, and querying the contents of CCRs.

3.1 A Comprehensive Clinical Typing System for Information Extraction

3.1.1 Motivation

A diverse set of text documents embodies our expanding knowledge of biological phenomena, including human health and disease. Every type of observation, from the semi-structured results in experimental studies to the detailed narratives in CCRs or electronic health records (EHR), is growing in volume, variety, and complexity. Any single human reader must therefore perform extensive labor when using clinical narratives to comprehensively answer biomedical questions, especially when comparing observations across medical subdomains. The structured data yielded by foundational advances in clinical information extraction (CIE) is of great assistance in addressing this challenge: they can consistently identify concepts and events within specific domains and tasks. However, interpreting clinical text with the greatest accuracy depends upon (sub)domain knowledge: e.g., the word "elongated" may describe different types of clinically concerning but surgically correctable deformities, such as an elongated tricuspid valve leaflet in the heart or an elongated styloid process in the skull (i.e., Eagle's syndrome [27]). Identification of specialized terminology with highly contextual semantics within biological and clinical text remains an open challenge for CIE methods.

Recognizing the need for resources supporting adaptation of CIE to clinical narratives, we seek to standardize the entity, event, and relation types within clinical text with a high degree of granularity. We therefore sought to design a typing system for clinical text capable of reflecting the diverse vocabulary and phenomena described within a clinical document without requiring direct connections to curated concepts or terminology. We believe this approach is ideal for designing practical CIE systems as it primarily reflects the semantics of terminology as it is used rather than an exact correspondence between a set of vocabulary and their expected meaning. A context-driven approach is also intuitive for clinical domain experts.

3.1.2 Methods

3.1.2.1 General design of the ACROBAT clinical typing system

ACROBAT is appropriate for manual annotation, automated labeling (i.e., named entity recognition and relation extraction), or a combination of both. For each document in a corpus, ACROBAT should be used to label all words and phrases in the document corresponding to one or more of the types. We make a distinction between events and entities: events occur during specific points in time (i.e., they may be arranged into a timeline) while entities are other meaningful text spans, often those modifying or describing properties of events. As a general guideline, the smallest span describing a single entity or event is labeled: for example, in the phrase *massive heart attack*, the labeled event is *heart attack*, as "heart attack" refers to a specific condition and *attack* alone is too general. The term *massive* describes is an entity in its own right; the term is a modifier of the Severity type. Words and phrases are labeled even if they do not specifically discuss a patient, e.g., if the authors discuss hypothetical situations or a patient's family members. This increases the total number of annotated instances and therefore the total pool of potential training examples. Labels may also overlap where appropriate or when multiple labels apply.

ACROBAT incorporates relations for semantic purposes, coreference resolution, and temporal order. Semantic relations cover all instances in which an entity modifies or results from another event or entity in any manner. For coreferences, an event/entity and any of its coreferences within a single document are linked through pairwise relations. Temporal order defines events within a continuous time series (e.g., *event 1* $\rightarrow$ *event 2* $\rightarrow$ *event 3*). Directionality is meaningful in ACROBAT and all relations (with the exception of Identity relations) are directed. ACROBAT also supports event properties for indication of changes over time or event negation. In cases where an abbreviation is present (e.g., *optical coherence tomography (OCT)*), the full name (*optical coherence tomography* and the abbreviation (*OCT*) are labeled as separate events and connected with an Identity relation.

Events. Events include words or phrases indicating a discrete activity or occurrence in a

document. In this section, we describe each event category, delineate annotation rules, and provide examples.

Properties. Properties are attributes of a single event. All events may include values for neither, one, or both properties.

Entities. Entities include words or phrases that do not completely constitute a clinical event on their own but generally modify an event or subject.

Relations. Relations are connections between entities or events. There are two general categories of relations: those used to express the temporal order of events (BEFORE, AFTER, and OVERLAP), and those used to define more specific relationships.

Coreferences. Rather than denoting specific events, Coreferences label words or phrases referring to previously defined events or entities (i.e., linguistic anaphora). Annotating a coreference therefore defines a relation but takes the form of an event in our system to accomodate labeling of the corresponding text spans.

3.1.2.2 Annotation of clinical case reports

In order to prepare a deeply annotated resource of clinical text, we sought to annotate clinical case reports using the schema described above. This work follows from creation of our Metadata Acquired from Clinical Case Reports, or MACCR, set [49]: each source document in the new set corresponds to a single entry, and therefore a collection of higher-level metadata, in our MACCR dataset. We therefore refer to our set of deeply annotated CCRs as `MACCROBAT2018`. For each of 200 documents, we obtained open-access text from PubMed Central, limiting the text portion to that comprising the clinical case (i.e., we did not include any other sections including introduction, discussion, figure/table legends, or supplementary materials). The document count was chosen based on manageability; subsequent releases will add more annotated documents. Each document in the set is named based on the respective PubMed identifier of their source document.

All documents were annotated by at least one of six annotators. All annotators had

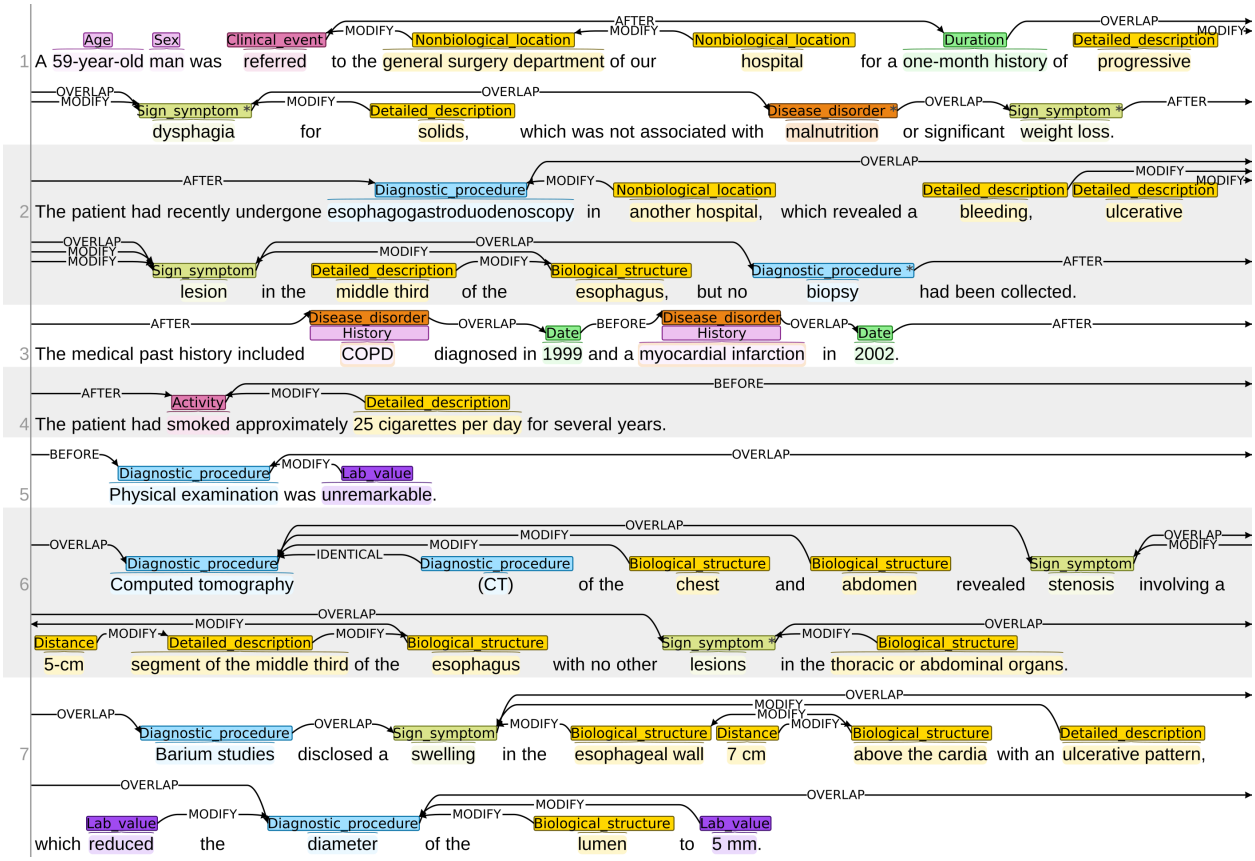


Figure 3.2: Example visualization of a selection of an ACROBAT annotated document.

previous experience reading biomedical and clinical language. Annotations were checked for format and type consistency upon completion. Annotation was performed through an implementation of the *brat* tool [231]; all annotation visualizations presented in this manuscript were created with *brat*. An example of annotation of a CCR is provided in Figure 3.2.

3.1.3 Results

The MACCROBAT2018 is intended to serve multiple purposes. This collection of annotations serves as both a demonstration of the ACROBAT scheme: each document is annotated with all appropriate event and entity types as well as relations, thereby providing numerous contextual examples of the typing scheme's implementation. The set is also ideal for training/testing CIE methods as it covers a variety of disease presentations and corresponding

vocabulary: the 200 documents in the set contain an average of 22.7 sentences, or 4,541 sentences in total, with sentences containing an average of 21.6 single-word tokens and 98,038 tokens in total. Because the text is annotated with multiple label and relation types, it may be used for the initial training of joint models (e.g., a tagger for both diagnostic procedure events and their results).

The MACCROBAT2018 set contains a total of 3,652 sentences and 59,164 annotations of any kind, including event/entity labels and relations.* Out of all categories and all 200 CCRs, Diagnostic_procedure events occur most frequently, with an average of more than 45 occurrences per document. The set also includes more than 6,700 annotations of signs/symptoms and more than 3,500 lab values.

As compared to other annotated corpora in biomedicine, MACCROBAT includes far more entity and relation types, as well as explicitly defined types for integration with knowledgebases. Sets with similar or greater numbers of coreferences (e.g., the GENIA [122], 2011 i2b2 Coreference Challenge [244], ODIE [217], or CRAFT [66] corpora) have been completed but do not exclusively focus on clinical language and, in some cases, are freely available in their entirety. As coreferences are not the primary focus of our annotations, it may be appropriate to use our set along with other corpora for applications in training coreference resolution models.

3.1.4 Discussion

A consistent set of concept types is a valuable resource for clinical informatics in both philosophy and practice. We see ACROBAT and MACCROBAT2018 as a way to manually or computationally enforce structure upon biomedical language, and in doing so, produce resources for training and developing systems for better understanding the concepts within clinical documents and publications.

*The MACCROBAT2018 set can be downloaded at Figshare: https://doi.org/10.6084/m9.figshare.c.4652765.

3.2 Clinical Named Entity Recognition using Contextualized Token Representations

3.2.1 Motivation

Clinical named entity recognition (CNER) is an important text mining task in the domain of Clinical Information Extraction (CIE) [275]. It aims to identify clinical entities and events from the case reports. For example, in the sentence "CT of the maxillofacial area showed no facial bone fracture." "CT of the maxillofacial area" is a "diagnostic procedure" and "facial bone fracture" belongs to the "disease and disorder" category. As with documents describing experimental procedures and results—often the focus of general biomedical annotated corpora such as PubTator [258] – CCRs include a large variety of entity types and potential orders of events [51]. Methods to better enable biomedical and clinical NLP at scale, across numerous entity types, and with generalizable approaches across topics are necessary, as single-task or single-entity type methods provide insufficient detail for comprehensive CNER. Fine-grained CNER supports development of precision medicine's hope to leverage advanced computer technologies to deeply digitize, curate and understand medical records and case reports [208, 29].

Biomedical NER (BioNER), of which CNER is a subtask, has been a focus of intense, groundbreaking research for decades but has recently undergone a methodological shift. Its foundational methods are largely rule-based (e.g., Text Detective [236]), dictionary-based (e.g., BioThesaurus [154] or MetaMap [20]), and basic statistical approaches (e.g., the C-value / NC-value method [86]). Source entities for NER are sourced from extensive knowledgebases such as UMLS [41] and UniProtKB [241]. Readily applicable model-based BioNER methods, including those built upon non-contextualized word embeddings such as Word2Vec and GloVe [177, 200] now promise to more fully address the challenges particular to the biomedical domain: concepts may have numerous names, abbreviated forms, modifiers, and variants. Furthermore, biomedical and clinical text assumes readers have extensive domain knowledge. Its documents follow no single structure across sources or topics, rendering their

content difficult to predict.

These models neither avoid time-consuming feature engineering, nor make full use of semantic and syntactic information from each token's context. Context can thoroughly change an individual word's meaning, e.g., an "infarction" in the heart is a heart attack but the same event in the brain constitutes a stroke. Context is crucial for understanding abbreviations as well: "MR" may represent the medical imaging technique *magnetic resonance*, the heart condition *mitral regurgitation*, the concept of a *medical record*, or simply the honorific *Mister*. Non-contextualized word embeddings exacerbate the challenge of understanding distinct biomedical meanings as they contain only one representation per word. The most frequent semantic meaning within the training corpus becomes the standard representation.

Inspired by the recent development of contextualized token representations [201, 71, 10] supporting identification of how the meaning of words changes based on surrounding context, we refresh the technology of CNER to better extract clinical entities from unstructured clinical text. The deep contextualized token representations are pre-trained with a large corpus using a language model (LM) objective. ELMo [201] takes word tokens as input and pre-trains them with a bidirectional language model (biLM). Flair [10] proposes a pre-trained character-level language model by passing sentences as sequences of characters into a bidirectional LSTM to generate word-level embeddings. Following recent work demonstrating impressive performance and accuracy of pre-training word representations with domain-specific documents [225], we collected domain-specific documents all related to CCRs, roughly a thousandth of PMC documents, and pre-trained two deep language models, `Clinical-ELMo` and `Clinical-Flair`. In this study, we focus on the CNER task and evaluate the two language models across three datasets. Our two pre-trained language models can support applications beyond CNER, such as clinical relation extraction or question answering.

3.2.2 Contributions

- To the best of our knowledge, we are the first to build a framework for solving clinical natural language processing tasks using deep contextualized token representations.
- We pre-train two contextualized language models, `Clinical-ELMo` and `Clinical-Flair` for public use[†]. We evaluate our models on three CNER benchmark datasets, MACROBAT2018, i2b2-2010, NCBI-disease, and achieve dramatic improvements of 10.31%, 7.50%, and 6.94%, respectively.
- We show that pre-training a language model with a light domain-specific corpus can result in better performance in the downstream CNER application, compared with domain-generic embeddings.

3.2.3 Methods

3.2.3.1 Contextualized Language Models

ELMo. ELMo is a language model that produces contextualized embeddings for words. It is pre-trained with a two-layered bidirectional language model (biLM) with character convolutions on a large corpus. The left lower part in Figure 3.3 is the high level architecture of ELMo, where R($\cdot$) means the representation of a word.

ELMo takes a sequence of words $(w_1, w_2, ..., w_N)$ as input and generates context-independent token representations using a character-level CNN. Then ELMo feeds the sequence of tokens $(t_1, t_2, ..., t_N)$ into the biLM which is a bidirectional Recurrent Neural Network (RNN). The forward-LM computes the probability of each sequence by:

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2, ..., t_{k-1}). \tag{3.1}$$

[†]The pre-trained model can be downloaded at `https://drive.google.com/drive/folders/1b8PQyzTc_HUa5NRDqI6tQXz1mFXpJbMw?usp=sharing`.

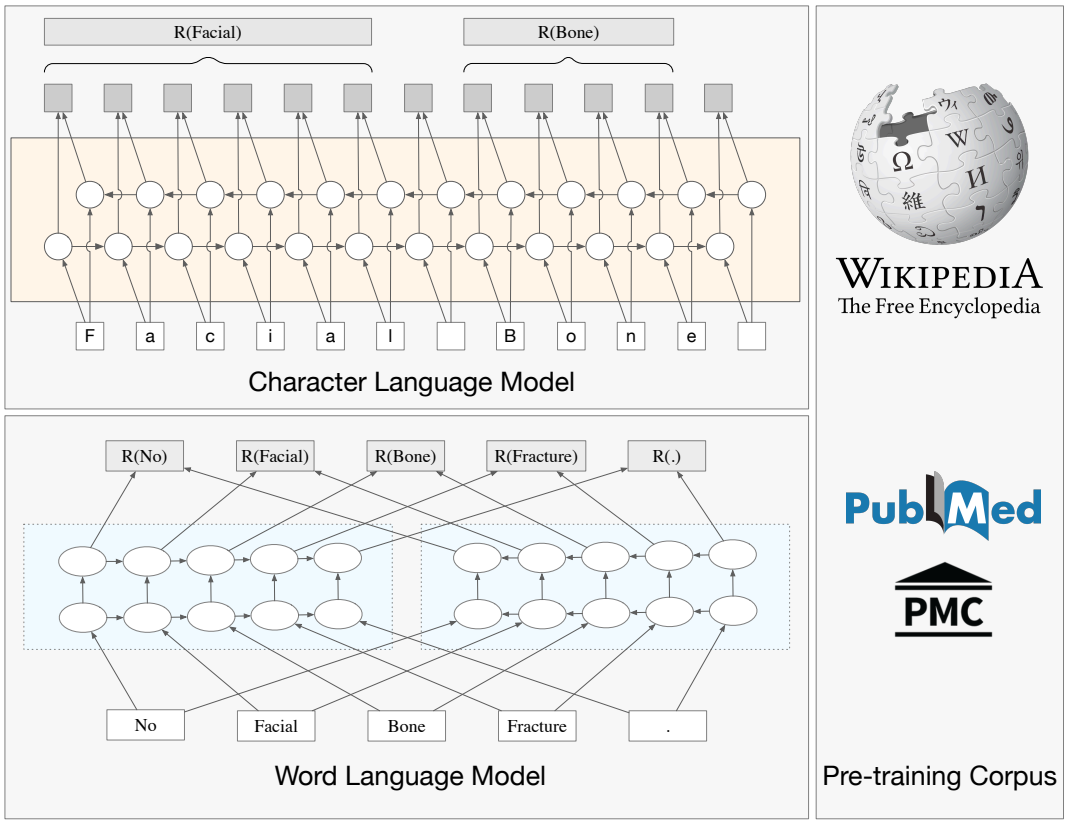


Figure 3.3: Pre-training Character and Word Language models.

Flair. Flair is a character-level word representation model that also uses RNN as the language modeling structure. Different from ELMo, Flair treats the text as a sequence of characters. The goal of most language models is to estimate a good distribution $p(t_0, t_2, ..., t_T)$ where $t_0, t_1, ..., t_n$ is a sequence of words. Instead of computing the distribution of words, Flair aims to estimate the probability $p(x_0, x_1, ...x_T)$, where $x_0, x_1, ..., x_T$ is a sequence of characters. The joint distribution over the entire sentence can then be represented as follows:

$$p(x_0, x_1, ..., x_T) = \prod_{t=0}^{T} p(x_t | x_1, x_2, ..., x_{t-1}) \tag{3.2}$$

where $p(x_t|x_0, ..., x_{t-1})$ is approximated by the network output h_t from one RNN layer. The details are illustrated in the left upper part in Figure 3.3.

3.2.3.2 CNER Model

We used a well-established BiLSTM-CRF sequence tagging model [110, 256, 95] to address the downstream sequence labeling tasks.

First, it passes sentences to a user-defined token embedding model, which converts a sequence of tokens into word embeddings: $r_0, r_1, r_2, ..., r_n$. We may concatenate embedding vectors from different sources to form a new word vector. For example, the concatenated embeddings of GloVe and Flair is represented as:

$$r_i = r_i^{GloVe} \oplus r_i^{Flair} \tag{3.3}$$

Then, the concatenated embeddings are passed to the BiLSTM-CRF sequence labeling model to extract the entity types.

3.2.4 Experiments

3.2.4.1 Datasets

Table 3.1: Number of tokens and types in each CNER dataset.

Dataset Name	Train	Dev	Test	# of Entity Types
MACCROBAT2018	64,879	862	7,955	24
i2b2-2010	134,586	14,954	267,250	3
NCBI-disease	135,701	23,969	24,497	1

MACCROBAT2018. MACCROBAT2018 contains 3,100 curated CCRs spanning 15 disease groups and more than 750 reports of rare diseases. We randomly selected 10% case reports as development set and 10% as test set. The remaining documents are used to train the CNER model. Detailed description is shown in 3.1.

i2b2-2010. The i2b2-2010 dataset provides "layered" linguistic annotation over a set of clinical notes. The dataset contains three entity types which are "test", "problem", "treatment". We followed [245] to split the dataset into train/development/test sets.

NCBI-disease. The NCBI-disease [77] dataset is fully annotated at the mention and concept level. The dataset contains 793 PubMed abstracts with 6,892 disease mentions which leads to 790 unique disease concepts. Therefore, the dataset only has one types which is "disease".

3.2.4.2 Pre-training Corpus

To pre-train the two language models, we obtained articles through the PubMed Central (PMC) FTP server[‡], and in total picked 47,990 documents that are related to clinical case reports. We indexed these documents with some keyword including "case report" and "clinical report".

3.2.4.3 Pre-trained Language Model

We proposed `Clinical-ELMo` and `Clinical-Flair`, which are respectively a pre-trained ELMo and a pre-trained Flair with the domain-specific corpus. To fairly compare the two models, we do not initialize `Clinical-ELMo` and `Clinical-Flair` with any pre-trained ELMo and Flair, and pre-train them on the same clinical case report corpus described in Section 3.2.4.2. Moreover, we tried to set both models' parameter sizes to a similar scale. Since Flair's parameter size is 20M when it performs at its best (hidden size of 2048), we chose the medium size ELMo model correspondingly, which has 25M parameters according to AllenNLP [201]. All models were pre-trained on one NVIDIA Tesla V100 (16GB), with each requiring roughly one week to complete.

For `Clinical-Flair`, we followed the default settings of Flair, a hidden size of 2048, a sequence length of 250, and a mini-batch size of 100. The initial learning rate is 20, and the annealing factor is 4. For `Clinical-ELMo`, we chose the medium-size model among all configurations, which has a hidden size of 2048 and projection dimension of 256. For the convolutional neural network token embeddings, the maximum length of a word is 50 and

[‡]ftp://ftp.ncbi.nlm.nih.gov/pub/pmc

Table 3.2: The comparison of F1-scores (%) on three datasets among different types of embeddings.

Embeddings	MACCROBAT2018	i2b2-2010	NCBI-disease
GloVe	59.63	81.35	82.18
ELMo	61.69	84.61	84.50
Flair	57.25	81.65	84.23
GloVe+ELMo	63.09	84.82	85.37
GloVe+Flair	62.63	81.21	85.58
GloVe+ELMoPubMed	64.56	86.50	87.04
GloVe+`Clinical-ELMo`	**65.75**	87.29	**87.88**
GloVe+`Clinical-Flair`	64.18	**87.45**	86.60

the embedding dimension is 16.

3.2.4.4 Results

To fairly compare the performance of each model, we pre-trained `Clinical-Flair` and `Clinical-ELMo` on the same subset of PubMed Central (PMC) documents. We then applied the BiLSTM-CRF model [110] to evaluate the downstream sequence labeling tasks. The results of our experiments are shown in Table 3.2. Note that "Embeddings" in Table 3.2 denotes the stacking embeddings which can be the concatenation of different word embedding vectors. We used the pre-trained GloVe embeddings of 100 dimensions. The Flair embeddings are pre-trained with a 1-billion word corpus [54]. ELMo denotes the pre-trained medium-size ELMo on the same 1-billion word corpus and ELMoPubMed denotes the pre-trained ELMo model with the full PubMed and PMC corpus. We used the micro F1-score as the evaluation metric.

Domain-specific v.s. Domain-generic corpus. From Table 3.2, we can observe that the models pre-trained on the selected case report corpus outperformed all the other language

Table 3.3: The performance of three baseline methods and our best model on three datasets.

Models	MACCROBAT2018	i2b2-2010	NCBI-disease
Our best model	**65.75**	**87.45**	87.88
CNN [166]	60.13	81.41	82.62
Cross-type [256]	63.10	84.97	86.14
BioBERT [138]	64.38	86.46	**89.36**

models pre-trained on the domain-generic corpus. The concatenated embedding of GloVe and `Clinical-ELMo` performs the best on MACCROBAT2018 and NCBI-disease datasets, while GloVe plus `Clinical-Flair` achieved the best performance on i2b2-2010. We can conclude that pre-training the language models with a small domain-specific corpus can be more efficient and effective for improving the performance of some downstream tasks. The domain-specific knowledge can alter the distribution and the proximity among words, thus contributing a better understanding of the relationship between word and entity types in our task.

Contextualized v.s. Non-contextualized embeddings. We also used the static word embeddings, GloVe itself, to represent the tokens in the sequence labeling task. The results in Table 3.2 show that the stacking contextualized embeddings dramatically boosted the F1-score on three different datasets by 10.31%, 7.50%, and 6.94%. It proves that the deep language models absorb more intensive semantic and syntactic knowledge from the contexts. We noticed that the F1-score of Flair on MACCROBAT2018 dataset was surprisingly low. It showed that the performance of a purely character-level language model may be not as robust as the word-level models.

Compared with other baseline models. [166] proposed a bi-directional LSTM-CNNs-CRF model to make use of both word- and character-level representations. [256] leveraged multi-task learning and attention mechanisms to improve the performance of biomedical sequence labeling task. Compared with these two state-of-the-art models, as shown in

Table 3.3, our methods perform consistently better. We suppose that with the help of pre-trained contextualized embeddings, even a light-loaded downstream model can achieve extraordinary performances.

The BioBERT proposed in [138] was pre-trained using a language model with around 110M parameters and using a large number of computational resources (8 NVIDIA V100 32GB GPUs). However, this contextualized language model only gets better performance in the simplest dataset (NCBI-disease) with only one entity type. On MACCROBAT2018 and i2b2-2010, we improved the performance by 2.13% and 1.15%. This shows that good experimental results can be achieved by making rational use of limited resources.

3.2.4.5 Case Study and Analysis

We analyze the `Clinical-Flair` and `Clinical-ELMo` on specific categories for the MACCROBAT2018 dataset. We look into the F1-scores of 10 different entity types. All these types appear more than 50 times in the dataset.

From Table 3.4, we can see that the character-level language model `Clinical-Flair` shows an advantage in the type "Dosage". We find that this entity type has a number of entities that do not appear in the word-level vocabulary, such as "60 mg/m2", "0.5 mg", and "3g/d". On the other hand, `Clinical-ELMo` has a better performance in the type "Severity", which contains words like "extensive", "complete", "significant", and "evident". `Clinical-ELMo` also extensively outperforms `Clinical-Flair` in "Detailed Description". The representations of tokens rely more on the word-level context in these types. Therefore, `Clinical-ELMo` shows better power of capturing the relationship between the word-level contextual features with the entity types.

We noticed in Table 3.4, "Disease Disorder" achieved around 50% F1-score with both models. Though they performed well on NCBI-disease dataset, it is hard for them to correctly recognize complex phrase-level disease entities on MACCROBAT2018, such as "Scheuer stage 3", and "feeding difficulties".

Table 3.4: The comparison of F1-scores (%) between `Clinical-ELMo` and `Clinical-Flair` on different entity types of MACCROBAT2018.

Entity	GloVe+`Clinical-ELMo`	GloVe+`Clinical-Flair`
Biological Structure	63.94	**64.88**
Detailed Description	**45.81**	40.00
Diagnostic Procedure	**74.93**	74.71
Disease Disorder	**50.84**	50.83
Dosage	77.42	**80.00**
Lab Value	**74.48**	72.31
Medication	**76.34**	72.13
Non-biological Location	**80.77**	76.00
Severity	**72.41**	61.81
Sign Symptom	**62.27**	60.64

3.2.5 Conclusion

In our study, we showed that contextual embeddings show a sizable advantage against non-contextual embeddings for clinical NER. In addition, pre-training a language model with a domain-specific corpus results in better performance in the downstream CNER task, compared to the off-the-shelf corpus.

3.3 Clinical Temporal Relation Extraction with Probabilistic Soft Logic Regularization and Global Inference

3.3.1 Motivation

There is a perennial need to automatically and precisely curate the clinical case reports into structured knowledge, i.e. extract important clinical named entities and relationships from the narratives [21, 218, 228, 50, 11, 279]. This would greatly enable both doctors and patients to retrieve related case reports for reference and provide a certain degree of technical support for resolving public health crises like the recent COVID-19 pandemic. Clinical reports describe chronicle events, elucidating a chain of clinical observations and reasoning [235, 55]. Extracting temporal relations between clinical events is essential for the case report retrieval over the patient chronologies. Besides, medical question answering systems require the precise ordering of clinical events in a time series within each document.

In this study, we tackle the temporal relation extraction problem in clinical case reports. Figure 3.4 illustrates a paragraph from a typical CCR document with three common types of temporal relations, "Before", "After", and "Overlap". *Glucocortocoids* was described as the medicine history of this patient, which happened before *confirmed with COVID-19* and *positive of antibody*. An "Overlap" temporal relation exists between *nasal congestion* and *a mild cough*. We consider the aforementioned clinical concepts as events, while regarding *a day later* as a time expression. A temporal relation may exist between event and event (`E-E`), event and time (`E-T`) or time and time (`T-T`).

There is a consensus within the clinical community regarding the difficulty of temporal information extraction, due to the high demand for domain knowledge and high complexity of clinical language representations [87]. [172, 137] apply machine learning models with lexical, syntactic features, or pre-trained word representations to tackle the problem but neglect the strong dependencies between narrative containment and temporal order, thus predicting inconsistent labels and garbled time-lines [141].

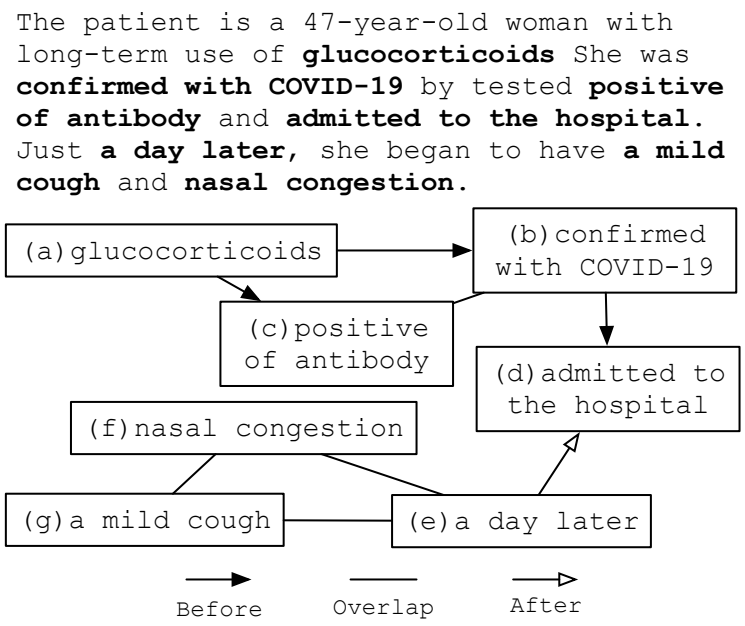


Figure 3.4: An illustration of a clinical case report with its partial temporal graph where transitivity dependencies exist.

The dependency is the key enabler of classifying the temporal relations. For instance in Figure 3.4, given that `b` happened before `d`, `e` happened after `d` and `e` happened simultaneously with `f` , we can infer according to the temporal transitivity rule that `b` was before `f`. Some recent studies [141, 190, 99] convert the task to a structured prediction problem and solve it with Maximum a posteriori Inference. Integer Linear Programming (ILP) with hard constraints is deployed for optimization, which however needs an off-the-shelf solver to tackle the NP-hard optimization problem and can only approximate the optimum via relaxation. Besides, globally inferring the relations at the document level would also be intractable for them due to the high complexity and low scalability [24].

Recently, some researchers [70, 59, 108] have explored Probabilistic Soft Logic (PSL) [24] to tackle the structured prediction problem. Inspired by them, we propose to leverage the PSL rules to model relation extraction more flexibly and efficiently. In specific, we summarize common transitivity and symmetry patterns of temporal relations as PSL rules and penalize the training instances that violate any of those rules. Different from ILP solutions, no off-

the-shelf solver is required and the algorithm conducts the training process with linear time complexity. Besides, logical propositions in PSL can be interpreted not just as $true$ or $false$, but as continuously valued in the $[0, 1]$ interval. We also propose a simple but effective time-anchored global temporal inference algorithm to classify the relations at the document level. With such a mechanism, we can easily verify some relations, such as the relation between `b` and `f`, with long-term dependencies which are intractable with existing approaches.

3.3.2 Contributions

- To the best of our knowledge, this is the first work to formulate the probabilistic soft logic rules of temporal dependencies as a regularization term to jointly learn a relation classification model,
- We show the efficacy of globally inferring the temporal relations with the time graphs,
- We release the codes[§] to facilitate further developments by the research community.

3.3.3 Clinical Temporal Relation Extraction

Corpora. Different from the datasets in the news domain [206, 91], the corpora in the clinical domain require rich domain knowledge for annotating the temporal relations. I2b2-2012 [235] and Clinical TempEval [35, 36, 37] are some great efforts of building clinical datasets with extensive annotations including labels of clinical events and temporal relations, the second of which was not tested in our work due to lack of access to the data.

Models. Some early efforts to solve the clinical relation extraction problem leverage conventional machine learning methods [161, 235, 262, 239, 137, 61] such as SVMs, MaxEnt and CRFs, and neural network based methods [150, 149, 74, 242, 151, 92, 152, 88]. They either require expensive feature engineering or fail to consider the dependencies among temporal relations within a document. [141, 97, 98, 190] formulate the problem as a structured pre-

[§]The codes are available at `https://github.com/yuyanislearning/CTRL-PG`.

diction problem to model the dependencies but can not globally predict temporal relations. Instead, our method can infer the temporal relations at document level.

3.3.4 Preliminaries for Probabilistic Soft Logic

3.3.4.1 Problem Statement

Document D contains sequences $[s_1, s_2, ..., s_M]$ and named entities $x_i \in \mathcal{E} \bigcup \mathcal{T}, 1 \leq i \leq N$, where M, N are the total number of sequences and entities in D. $\mathcal{E}$ and $\mathcal{T}$ represent the set of events and time expressions, respectively. There is a potential temporal relation between any pair of annotated named entities (x_j, x_k), where $1 \leq j, k \leq N$. Formally, the task is modeled as a classification problem with a set of temporal relation types $\mathcal{Y}$. Given a sequence s_i together with two named entities $x_{i,1}, x_{i,2}$ included, we predict the temporal relation $y_i \in \mathcal{Y}$ from $x_{i,1}$ to $x_{i,2}$. In practice, we create a triplet with three pairs of entities to be one training instance $\mathcal{I}$, to enable the PSL rule grounding, as explained in the following section.

3.3.4.2 Probabilistic Soft Logic and Temporal Dependencies in Clinical Narratives

Here, we introduce some concepts and notations for the language PSL and illustrate how PSL is applicable to define templates for temporal dependencies and to help jointly learn a relation classifier.

Definition 1. *A **predicate** $\tilde{p}$ is a relation defined by a unique identifier and an **atom** $\tilde{l}$ is a predicate combined with a sequence of terms of length equal to the predicate's argument number. Atoms in PSL take on continuous values in the unit interval* $[0, 1]$.

Example 1. Before/2 indicates a predicate taking two arguments, and the atom Before(A, B) represents whether A happens before B.

Definition 2. *A **PSL rule** $\tilde{r}$ is a disjunctive clause of atoms or negative atoms:*

$$\eta_r : T_1 \wedge T_2 \wedge ... \wedge T_m \rightarrow H_1 \vee H_2 \vee ... \vee H_n, \tag{3.4}$$

Table 3.5: Temporal transitivity and symmetry PSL rules $\mathcal{R}$. A, B, C are three terms representing either events or time expressions.

Abbrev.	PSL rules
Transitivity Dependencies	
BBB	$\text{Before}(A,B) \wedge \text{Before}(B,C) \rightarrow \text{Before}(A,C)$
BOB	$\text{Before}(A,B) \wedge \text{Overlap}(B,C) \rightarrow \text{Before}(A,C)$
OBB	$\text{Overlap}(A,B) \wedge \text{Before}(B,C) \rightarrow \text{Before}(A,C)$
OOO	$\text{Overlap}(A,B) \wedge \text{Overlap}(B,C) \rightarrow \text{Overlap}(A,C)$
AAA	$\text{After}(A,B) \wedge \text{After}(B,C) \rightarrow \text{After}(A,C)$
AOA	$\text{After}(A,B) \wedge \text{Overlap}(B,C) \rightarrow \text{After}(A,C)$
OAA	$\text{Overlap}(A,B) \wedge \text{After}(B,C) \rightarrow \text{After}(A,C)$
Symmetry Dependencies	
BA	$\text{Before}(A,B) \rightarrow \text{After}(B,A)$
AB	$\text{After}(A,B) \rightarrow \text{Before}(B,A)$
OO	$\text{Overlap}(A,B) \rightarrow \text{Overlap}(B,A)$

where $T_1, T_2, ..., T_m, H_1, H_2, ..., H_n$ are atoms or negative atoms.

We name $T_1, T_2, ..., T_m$ as r_{body} and $H_1, H_2, ..., H_n$ as r_{head}. $\eta_r \in [0, 1]$ is the weight of the rule r, denoting the prior confidence of this rule. To the opposite, an unweighted PSL rule is to describe a constraint that is always true. The unweighted logical clauses in Table 3.5 describe the common temporal transitivity and symmetry dependencies we summarize from the clinical narratives.

Definition 3. *The **ground atom** l and **ground rule** r are particular variable instantiation of some atom $\tilde{l}$ and rule $\tilde{r}$, respectively.*

Example 2. That Overlap (e, f) $\wedge$ Overlap (f, g) $\rightarrow$ Overlap (e, g) from Figure 3.4 is a ground rule composed of three ground atoms, denoted as l_1, l_2, and l_3, respectively. It is grounded from the OOO rule, as shown in Table 3.5.

Definition 4. *The interpretation $I(l)$ denotes the soft truth value of an atom l.*

Definition 5. *Łukasiewicz t-norm [126] is used to define the basic logical operations in PSL, including logical conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$):*

$$I(l_1 \wedge l_2) = \max\{I(l_1) + I(l_2) - 1, 0\} \tag{3.5}$$

$$I(l_1 \vee l_2) = \min\{I(l_1) + I(l_2), 1\} \tag{3.6}$$

$$I(\neg l_1) = 1 - I(l_1) \tag{3.7}$$

The PSL rule in Definition 2 can also be represented as:

$$I(r_{body} \rightarrow r_{head}) = I(\neg r_{body} \vee r_{head}),$$

so we can induce the distance to satisfaction for rule r.

Definition 6. *The **distance to satisfaction** $d_r(I)$ of rule r under an interpretation I is defined as:*

$$d_r(I) = \max\{0, I(r_{body}) - I(r_{head})\} \tag{3.8}$$

PSL program determines a rule r as satisfied when the truth value of $I(r_{head}) - I(r_{body}) \geq 0$.

Example 3. Given that $I(l_1) = 0.7, I(l_2) = 0.8$, and $I(l_3) = 0.3$, we can compute the distance according to Equation (3.5)-(3.8):

$$\begin{aligned} d_r &= \max\{0, I(l_1 \wedge l_2) - I(l_3)\} \\ &= \max\{0, 0.7 + 0.8 - 1 - I(l_3)\} \\ &= \max\{0, 0.5 - 0.3\} \\ &= 0.2 \end{aligned}$$

This equation indicates that the ground rule in Example 2 is completely satisfied when $I(l_3)$ is above 0.5. Otherwise, a penalty factor will be raised (0.2 in this case). When $I(l_3)$ is

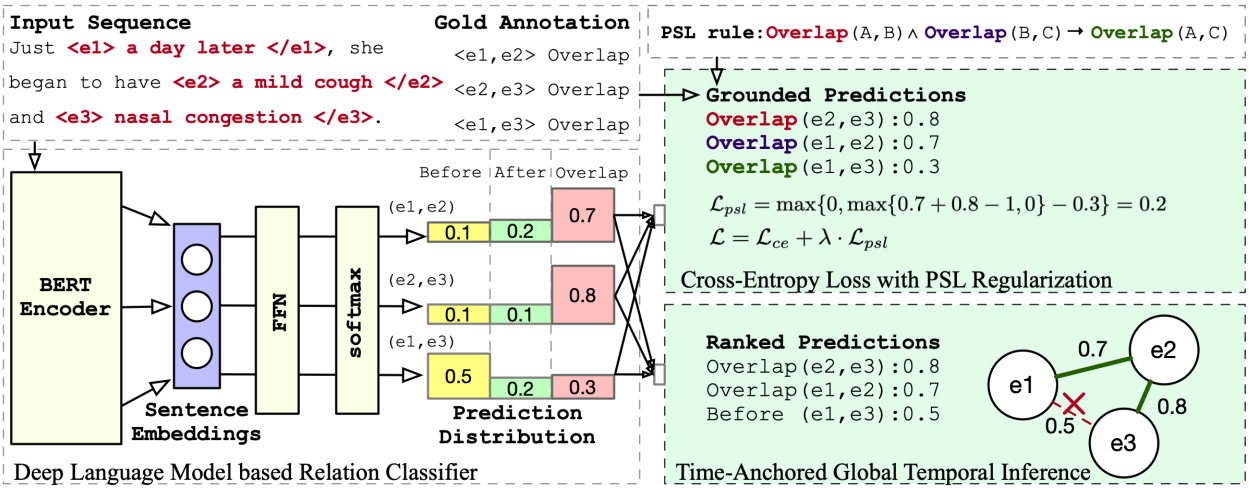


Figure 3.5: The overall architecture of CTRL-PG.

under 0.5, the smaller $I(l_3)$ is, the larger penalty we have. In short, we compute the distance to satisfaction for each ground rule as a loss regularization term to jointly learn a relation classification model. We finally use the smallest one as the penalty because we only need one of the rules to be satisfied.

3.3.5 Methodologies for Temporal Relation Extraction

Figure 3.5 shows the overall framework of the proposed CTRL-PG model. The framework consists of three components, (i) a temporal relation classifier composed of a deep language encoder and a Feed-Forward Network (FFN), (ii) a Cross-Entropy loss function with PSL regularization, and (iii) a time-anchored global temporal inference module. We will introduce the details of the three modules in the following subsections.

3.3.5.1 Temporal Relation Classifier

The context is essential for capturing the syntactic and semantic features of each word in a sequence. Hence, we propose to apply the contextualized language model, BERT [73], to derive the sentence representation v_i of d_s-dimension to encode the input sequence s_i including two marked named entities $x_{i,1}, x_{i,2}$ from the instance $\mathcal{I}$, where $i \in \{1, 2, 3\}$. We group

three sequences together to facilitate the computation of regularization term introduced in the next subsection.

By feeding the sentence embedding v_i to a layer of FFN, we can predict the relation type $\hat{y}_i$ with the softmax function:

$$\hat{y}_i = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \ \mathbb{P}(y|s_i) \tag{3.9}$$

$$\mathbb{P}(y|s_i) = \text{softmax}(W_f \cdot v_i + b_f), \tag{3.10}$$

where W_f and b_f are the weights and bias in the FFN layer.

To learn the relation classification model, we first compute a loss with the Cross-Entropy objective for each instance $\mathcal{I}$:

$$\mathcal{L}_{ce} = -\sum_{i \in \{1,2,3\}} \sum_{y \in \mathcal{Y}} y \log \mathbb{P}(y|s_i) \tag{3.11}$$

3.3.5.2 Learning with Probabilistic Soft Logic Regularization

We also aim to minimize the distance to rule satisfaction for each instance. We compute the distance with function $\mathcal{F}(\cdot, \cdot)$, as described in Algorithm 1, by finding the minimum of all possible PSL rule grounding results, i.e., when one PSL rule is satisfied, $\mathcal{F}(\cdot, \cdot)$ should return 0. In specific, we first ground the three relation predictions $\hat{y}_i$ with potential PSL rules. We then incorporate Equation (3.5)-(3.8) for distance computation. The prediction probabilities are regarded as the interpretation of the ground atoms l_i. If none of the rules can be grounded, the distance will be set as 0. Then, we formulate the distance to satisfaction as a regularization term to penalize the predictions that violate any PSL rule:

$$\mathcal{L}_{psl} = \mathcal{F}(\mathcal{R}; \{(\mathbb{P}(y|s_i), \hat{y}_i)\}), i = \{1, 2, 3\} \tag{3.12}$$

and finalize the loss function by summing up (3.11) and (3.12):

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \cdot \mathcal{L}_{psl}, \tag{3.13}$$

where λ is a hyperparameter as the weight for PSL regularization term. We apply gradient descent to minimize the loss function (3.13) and to update the parameters of our model.

Algorithm 1: Function $\mathcal{F}$ for PSL Rule Grounding and Distance Calculation.

Input: PSL Rules $\mathcal{R}$, Prediction $\hat{y}_i$, and Probability $\mathbb{P}(y|s_i)$, $i = \{1, 2, 3\}$;

Output: Distance d_r;

Set $d_r = 1$; $d_t = 0$; IsGround = $false$;

for *each* $l_1 \wedge l_2 \rightarrow l_3 \in \mathcal{R}$ **do**

  if $\hat{y}_1$ *matches* l_1 *and* $\hat{y}_2$ *matches* l_2 **then**

    Determine $\bar{y}_3$ with l_3;

    $d_t \leftarrow \max\{\mathbb{P}(y = \hat{y}_1|s_1) + \mathbb{P}(y = \hat{y}_2|s_2) - 1, 0\}$;

    $d_t \leftarrow \max\{d_t - \mathbb{P}(y = \bar{y}_3|s_3), 0\}$;

    $d_r \leftarrow \min\{d_r, d_t\}$;

    IsGround $\leftarrow true$;

end

if *IsGround* == $false$ **then**

  $d_r \leftarrow 0$;

3.3.5.3 Global Temporal Inference

In the inference stage, we leverage the Timegraph algorithm [179] to resolve the conflicts in the temporal relation predictions $\hat{\boldsymbol{y}}$. Timegraph is a widely used algorithm of time complexity $\mathcal{O}(v+e)$ for deriving the temporal relation for any two nodes in a connected graph, where v and e denote the numbers of nodes and edges. Nodes and edges represent the named entities and temporal relations, respectively. Our goal is to construct a conflict-free time graph $\mathcal{G}$ for each document D through a greedy Check-And-Add process, described as 4 steps in Algorithm 2. Intuitively, we want to rely on some trustworthy edges to resolve the conflicts in the time graph with the transitivity and symmetry dependencies listed in Table 3.5. As illustrated in Figure 3.5, the probabilities of predictions Overlap(e1, e2) and Overlap(e2, e3) are 0.7 and 0.8, which are higher than that of Overlap(e1, e3). When we trust the first two predictions, the third prediction could be neglected considering the relation between e1 and e3 can already be inferred with the transitivity dependency. In this way, the predicting

Algorithm 2: Check-And-Add Process for Constructing a Conflict-free Time Graph $\mathcal{G}$

Step 1: Predict temporal relations P_1 on pairs of the time expressions `T-T`;

Step 2: Construct a time graph $\mathcal{G}$ with P_1;

Step 3: Rank all other predictions P_2 on the relations of type `E-E` and `E-T` according to the predicting probabilities in decreasing order, naming P_2^{ranked};

Step 4:

for *each* p *in* P_2^{ranked} **do**

    Apply Timegraph algorithm to check the conflict between p and $\mathcal{G}$;

    if *there exists a conflict* **then**

        Drop p;

    else

        Add the edge p to $\mathcal{G}$;

    end

end

mistakes with low confidence scores can be ruled out, leading to better model performance in the closure evaluation.

We believe that the relations between time expressions are the easiest ones to predict. For example, the ground atom Before (*06-15-91*, *July 1st 1991*) is obviously *true*. Therefore, we try to build up a base time graph on top of the relations of type `T-T`. Next, we rank the rest of the predictions according to their probabilities in decreasing order and then check whether each of the predictions is inconsistent with the current time graph iteratively. The relation will be dropped if it raises a conflict, otherwise added to the graph as a new edge.

3.3.6 Experiments

In this section, we develop experiments on two benchmark datasets to prove the effectiveness of both PSL regularization and global temporal inference. We also discuss the limitation

Table 3.6: I2B2-2012 and TB-Dense Dataset Statistics.

Dataset		Train	Dev	Test
I2B2-2012	# doc	181	9	120
	# relation	29,736	1,165	24,971
TB-Dense	# doc	22	5	9
	# relation	4,032	629	1,427

and perform error analyses for `CTRL-PG`.

3.3.6.1 Datasets

Experiments are conducted on I2B2-2012 and TB-Dense datasets and an overview of the data statistics is shown in Table 3.6. The datasets have diverse annotation densities and instance numbers.

I2B2-2012. The I2B2-2012 challenge corpus [235] consists of 310 discharge summaries. Two categories of temporal relations, `E-T` and `E-E`, were annotated in each document. Three temporal relations, Before, After, and Overlap, were used. I2B2-2012 has a relatively low annotation density[¶], which is 0.21.

TB-Dense. To prove that our PSL regularization is a generic algorithm and can be easily adapted to other domains, we also test it on the TB-dense [48] dataset, which is based on TimeBank News Corpus [206]. Annotators were required to label all pairs of events/times in a given window to address the sparse annotation issue in the original data. Thus the annotation density is 1. This dataset has six relation types, Simultaneous, Before, After, Includes, Is_Include, and Vague.

[¶]Annotation density denotes the percentage of annotated pairs of event/time expressions.

3.3.6.2 Baseline Models

We employ different baseline models for the two datasets to compare our method with the SOTA models in both clinical and news domains.

I2B2-2012. (1) Feature-engineering based models from I2B2-2012 challenge, `MaxEnt-SVM` [262] incorporating Maximum Entropy with Support Vector Machine (SVM), `CRF-SVM` [239] using Conditional Random Fields and SVM, `RULE-SVM` [189] relying on rule-based algorithms; (2) Neural network based model, `RNN-ATT` [158], which applies Recurrent Neural Network plus attention mechanism; (3) Structured Prediction method, `SP-ILP` [97, 141] leveraging the ILP optimization; (4) Basic version of our model, `CTRL`, which only fine-tunes a BERT-BASE [73] language model with one layer of FFN, similar to the implementations in [151, 92].

TB-Dense. (1) `CAEVO` [52] with a cascade of rule-based classifiers; (2) `LSTM-DP` [60] using LSTM-based network and cross-sentence dependency paths; (3) `GCL` [172] incorporating LSTM-based network with discourse-level contexts; (4) `SP-ILP` and `CTRL`, same as the baselines for I2B2-2012. Note that the results of `CAEVO`, `LSTM-DP`, `GCL`, and `SP-ILP` are collected from [97].

3.3.6.3 Evaluation Metrics

To be consistent with previous work for a fair comparison, we adopt two different evaluation metrics. For TB-Dense dataset, we compute the Precision, Recall, and Micro-average F1 scores. Following [98, 172], we only predict the `E-E` relations and exclude all other relations from evaluation. Note that Micro-averaging in a multi-class setting will lead to the same value for Precision, Recall, and F1. For I2B2-2012, we leverage the TempEval evaluation metrics used by the official challenge [235], which also calculates the Precision, Recall, and Micro-average F1 scores. This evaluation metrics differ from the standard F1 used for TB-Dense in a way that it computes the Precision by verifying each prediction in the closure of the ground truths and computes the Recall by verifying each ground truth in the closure of the predictions. We explore all types of temporal relations in I2B2-2012 dataset.

Table 3.7: Performance of temporal relation extraction on I2B2-2012 datasets.

Model	P	R	F1
RULE-SVM	71.09	58.39	64.12
MaxEnt-SVM	74.99	64.31	69.24
CRF-SVM	72.27	66.81	69.43
RNN-ATT	71.96	69.15	70.53
SP-ILP	78.15	**78.29**	78.22
CTRL	84.88	73.28	78.65
CTRL-PG	**86.80**	74.53	**80.20**

3.3.6.4 Implementation Details

In the framework of CTRL-PG, any contextualized word embedding method, such as BERT [73], ELMo [202], and RoBERTa [159], can be utilized. We choose BERT to derive contextualized sentence embeddings without loss of generality. BERT adds a special token [CLS] at the beginning of each tokenized sequence and learns an embedding vector for it. We follow the experimental settings in [73] to use 12 Transformer layers and attention heads and set the embedding size d_s as 768. The CTRL-PG is implemented in PyTorch and we use the fused Adam optimizer [124] to optimize the parameters. We follow the experimental settings in [73] to set the dropout rate, and batch size as 10^{-1} and 8. We perform grid search for the initial learning rate from a range of $\{1 \times 10^{-5}, 2 \times 10^{-5}, 4 \times 10^{-5}, 8 \times 10^{-5}\}$ and finally select 2×10^{-5} for both datasets. We train 10 epochs for each experiment on two datasets, which can all be completed within 2 hours on single DGX1 Nvidia GPU.

We search the PSL regularization term λ from $\{0.1, 0.5, 1, 2, 5, 10\}$. For I2B2-2012 and TB-Dense datasets, we set λ as 5 and 0.5, respectively. The hyperparameters are selected by observing the best F1 performance on the validation set.

Table 3.8: Ablation study on I2B2-2012 dataset. GTI denotes the global temporal inference.

Feature	P	R	F1	Lift
Best	86.80	74.53	80.20	-
w/o PSL	85.78	73.31	79.06	**1.44**%
w/o GTI	85.08	73.31	78.76	**1.83**%

Table 3.9: Comparison of different ranking methods applied in the global inference on I2B2-2012 dataset.

Strategy	P	R	F1	Lift
Random	85.08	73.93	79.21	-
Confidence	86.07	73.76	79.44	**0.29%**
Confidence + Time Anchor	86.80	74.53	80.20	**1.25**%

3.3.6.5 Experimental Results

Table 3.7 and Table 3.10 contains our main results. As we observe, our `CTRL-PG` enhanced by PSL regularization and global inference achieve the best relation extraction performances per F1 score. Compared with the baseline models, the F1 score improvements are 2.0% and 2.5% on I2B2-2012 and TB-Dense data respectively, which are all statistically significant[‖].

I2B2-2012. As shown in Table 3.7, our model `CTRL-PG` outperforms the best baseline method `CTRL` by 2% and outperforms the structured prediction method `SP-ILP` by 2.5% per F1 score. `SP-ILP` gets the highest Recall score, but sacrifice the predicting precision instead. We also observe that by simply fine-tuning the BERT to generate the sentence embeddings and then feeding them into one layer of FFN for classification, `CTRL` can achieve

[‖]All improvements of `CTRL-PG` over baseline methods are statistically significant at a 99% confidence level in paired t-tests.

Table 3.10: Performance of temporal relation extraction on TB-dense datasets.

	SP-ILP			CTRL-PG		
	P	R	F1	P	R	F1
Before	71.1	58.9	64.4	52.6	74.8	61.7
After	75.0	55.6	63.5	69.0	72.5	70.7
Includes	24.6	4.2	6.9	60.9	29.8	40.0
Is_Include	57.9	5.7	10.2	34.7	27.7	30.8
Simultaneous	-	-	-	-	-	-
Vague	58.3	81.2	67.8	72.8	64.8	68.6
Micro-average			63.2			**65.2**
CAEVO						49.4
LSTM-DP						52.9
GCL						57.0
CTRL						63.6

an impressive F1 score of 78.65%. This proves the advantage of contextualized embeddings over static embeddings used by other baseline models. Besides, CTRL-PG outperforms the feature-based systems, CRF-SVM and MaxEnt-SVM, by over 10% per F1 score.

We develop an ablation study to test different features, as shown in Table 3.8. We see that PSL regularization and global temporal inference modules lift the performance by 1.44% and 1.83% separately. Both Precision and Recall performances are improved. We can clearly conclude that learning the relations with the proposed algorithms improves our model significantly (also at a 99% level in paired t-tests).

We also show the comparisons among different ranking strategies for the global inference module in Table 3.9. Random denotes that we randomly add a new prediction to the time graph and resolve the conflict. Confidence denotes we rank the predictions per the prediction probabilities and then add them to the graph in decreasing order. Time Anchor represents

that we first construct the time graph based on the predictions for temporal relations of type T-T. In the results, we see a 0.29% improvement per F1 score when switching from the Random to the Confidence strategy. After adding the Time Anchor method, we observe a 1.25% performance lift, compared to Random strategy. This proves the effectiveness of the time-anchored global temporal inference module.

TB-Dense. We show the experimental results on TB-Dense dataset in Table 3.10. Our model outperforms the best baseline model CTRL by 2.5% and outperforms the structured prediction method SP-ILP by 3.2% per Micro-average F1 score. We observe that in the performance breakdown for each relation class, CTRL-PG obtains similar scores on Before, After, and Vague as SP-ILP and gets much better performances on Is_Include and Includes. These two types only occupy 5.7% and 4.5% of all the instances. CTRL-PG and SP-ILP both fail to label any instance as Simultaneous because of its even fewer instances (1.5%) for training.

Besides, we observe CTRL-PG achieves higher Recall values in all the categories of temporal relations, which prove that incorporating the dependency rules into model training can dramatically lift the coverage of predictions.

3.3.6.6 Case Study and Error Analysis

Table 3.11 shows the results of a case study with the outputs of CTRL and CTRL-PG. In the first case, the temporal relation between *Her acute bradycardic event* and *the beta blocker* is hard to predict due to the noise brought by the long context. CTRL predicts it as Overlap, while CTRL-PG corrects it to After according to the potential PSL rule that can be matched with the first two correct predictions. In some cases, however, CTRL-PG will make new mistakes. For example in case 2, if our model initially predicts the relation between *started* and *a beta Elmore* wrong, a potential PSL rule sometimes will lead to an extra mistake when predicting the relation between *started* and *Maxine ACE*. In the case 3, *antibiotics* treated the *attacks*

Table 3.11: Case study and error analysis of the model predictions on I2B2-2012 Dataset.

1	Text	**Her acute bradycardic event** was felt likely secondary to **her new beta blocker** in conjunction with a vagal response. It was determined to stop **the beta blocker**, and atropine was placed at the bedside.		
	(e1, e2)	(Her...event, her...blocker)	(her...blocker, the beta blocker)	(Her...event, the beta blocker)
	True Label	After	Overlap	After
	`CRTL`	After	Overlap	Overlap
	`CTRL-PG`	After	Overlap	After
	Rule	$\text{After}(A, B) \wedge \text{Overlap}(B, C) \rightarrow \text{After}(A, C)$		
2	Text	The patient was given an aspirin and Plavix and in addition **started** on **a beta Elmore**, **Maxine ACE inhibitor**, and these were titrated up as her blood pressure tolerated.		
	(e1, e2)	(started, a beta Elmore)	(a beta Elmore, Maxine ACE)	(started, Maxine ACE)
	True Label	Before	Overlap	Before
	`CRTL`	After	Overlap	Before
	`CTRL-PG`	After	Overlap	After
	Rule	$\text{After}(A, B) \wedge \text{Overlap}(B, C) \rightarrow \text{After}(A, C)$		
3	Text	She has had attacks treated with **antibiotics** in the past notably in **12/96** and **08/97**.		
	(e1, e2)	(antibiotics, 12/96)	(12/96, 08/97)	(antibiotics, 08/97)
	True Label	Overlap	Before	Overlap
	`CRTL`	Overlap	Before	Overlap
	`CTRL-PG`	Overlap	Before	Before
	Rule	$\text{Overlap}(A, B) \wedge \text{Before}(B, C) \rightarrow \text{Before}(A, C)$		

twice in both 12/96 and 08/97, where the PSL rule is no longer valid since the *antibiotics* in fact denote two occurrences of this event. In such special cases with invalid rules, `CTRL-PG` may make a mistake.

3.3.7 Conclusion

In this work, we propose `CTRL-PG` that leverages the PSL rules to model the temporal dependencies as a regularization term to jointly learn a relation classification model. Extensive experiments show the efficacy of the PSL regularization and global temporal inference with time graphs.

3.4 CREATe: Clinical Report Extraction and Annotation Technology

3.4.1 Motivation

Case reports are a time-honored means of sharing observations and insights about novel patient cases [44, 50, 271]. As of 2020, at least 160 case report journals were in existence, with over 90% having open access policies and almost half indexed by PubMed [170]. The narrative of a case report details the symptoms, diagnosis, treatment, and outcome of an individual, describing observations made over the course of clinical care.

Often these case reports contain exceptionally valuable clinical data, addressing unusual disease situations. To our knowledge, there has been no attempt to annotate, index, or otherwise curate these reports. Unlike other types of medical literature, there are no organizational frameworks or methodical review articles for case reports, and no metadata standards exist for their curation. A significant challenge exist for this rapidly growing corpus. Focusing our efforts, we address the domain of cardiovascular disease, an important area for which a range of research and clinical questions occur frequently.

In this paper, we demonstrate an end-to-end systems incorporating algorithms for extracting, indexing, and querying the contents of clinical case reports.** Our proposed system, Clinical Report Extraction and Annotation Technology (CREATe), automates generation of metadata about case reports, unlocking this important data resource through a searchable portal.

To make case reports findable and accessible, our primary innovation is a graph-based representation of each case report, where nodes signify concepts described in the narrative (e.g. sign/symptoms, diagnosis, etc). To build a case report's graph, we start by employing named entity recognition techniques to identify concepts (nodes), which are then standardized against existing biomedical ontology [50] to make the metadata interoperable. Concepts

**An online video of the demonstration can be viewed at https://youtu.be/Q8owBQYTjDc.

are then connected (edges) by detecting described temporal relationships in the text, facilitating retrieval over the patient chronologies that are hallmarks of case report descriptions. Collectively, `CREATe` framework will be used to share metadata around published case reports, making it reusable by others via public APIs.

3.4.2 System Properties

- As of October 2020, it offers rich resources of over 10k reports for cardiovascular disease with depositions from a wide range of sources such as Scientific Literature and Authorized User Submissions, as shown in Figure 3.7.

- It provides a PDF submission service, based on Grobid [164], which is able to convert the publications in PDF format into well organized XML format. Metadata such as title, author, affiliation information can be automatically extracted for users by text mining technology.

- It is powered by `CREATe-IR`, a relation-based information retrieval system for clinical case reports, which outperforms solr [226]. Instead of simple keyword match, `CREATe-IR` embeds advanced deep learning algorithms to extract significant named entities and relations from the narrative. Relative case reports are retrieved from the database based on these structured knowledge. This also enables a temporal reasoning on the user queries and provides better search results.

- It provides a user-friendly interface for entity and relation annotation. A graph visualization of temporal order of the clinical events is generated for each document.

3.4.3 System Architecture and Design

`CREATe` is a cloud-based application and its service is mainly hosted on the Amazon ECS (Fargate) with a CI/CD pipeline as illustrated in Figure 3.6.

As shown in Figure 3.7, the main feature of `CREATe` is to allow users perform `CREATe-IR`

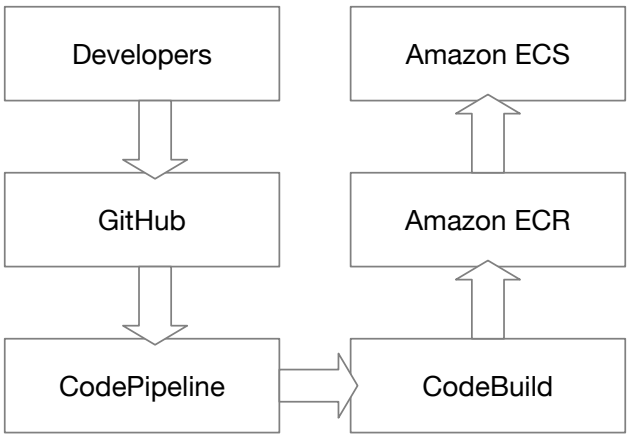


Figure 3.6: The Development Pipeline of `CREATe`.

search, which will be elaborated in later sections. The two sources of data are case reports collected from PubMed and user-input reports. The frontend of `CREATe` is a single page application developed using React, which allows users to communicate with the RESTful API in the backend built with Express. The application is served by Nginx, a light-weight software for web serving. The majority of data for `CREATe`, is stored in the MongoDB server for persistency. The data in MongoDB server is queried via the Express backend to ensure security and consistency. This is the same for both Neo4j server and ElasticSearch server, which allow the application to perform complex search as explained in later section.

3.4.4 `CREATe-IR`

3.4.4.1 Data Source and Preprocessing

Identification of data source begins with a query to PubMed using the publication type and MeSH term filters to locate cardiovascular (CVD) case reports. A query of PubMed in six areas of CVD (cardiomyopathy, ischemic heart disease, cerebrovascular accidents, arrhythmias, congenital heart disease, and valve disease) returns around 118,000 case reports. From these results, a web crawler (e.g. Apache Nutch) is used to locate the associated case reports and publication metadata. The contents can be captured in XML or online PDFs. The XML and PDF documents will be parsed into plain text to facilitate subsequent analysis,

organized into case report sections and sentences.

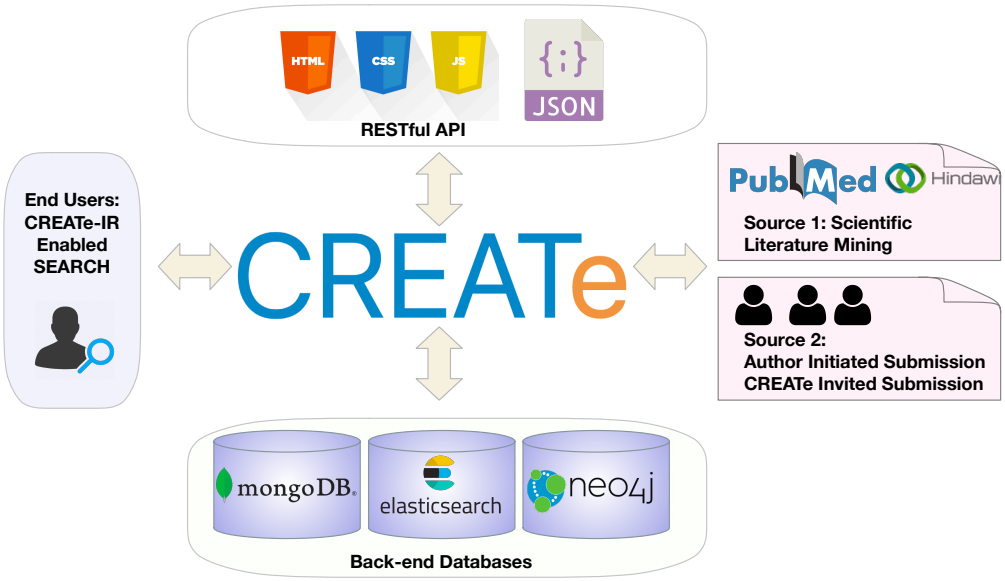


Figure 3.7: The Architecture of CREATe.

3.4.4.2 Data Annotations

As described in Section 3.4, to enable a supervised learning process for the automatic extraction of events/entities and relations from the clinical case reports, we build a comprehensive typing schema for information extraction from clinical narratives [50] and invite several medical experts to annotate hundreds of case reports based on this schema.

3.4.4.3 Information Extraction from User Queries

Once we receive a user query like "A patient was admitted to the hospital because of fever and cough.", we apply two machine learning modules to parse the query and extract important knowledge from them (i.e. hospital (*Non-biological Location*), fever (*Sign/Symptom*), cough (*Sign/Symptom*), and the temporal relation between cough and fever (OVERLAP)).

Named Entity Recognition. We develop a named entity recognizer, as described in Section 3.2, using contextualized token representations to locate and classify clinical termi-

nologies into predefined categories, such as diagnostic procedure, disease disorder, severity, medication, medication dosage, and sign symptom.

Temporal Relation Reasoning. We continue to predict the temporal relations among the extracted named entities, as introduced in Section 3.3. We notice the dependencies among events in one clinical document are key enabler of classifying the temporal relations. In our system, we build a temporal relation extraction module [279] based on common dependencies such as transitivity and symmetry patterns.

3.4.4.4 Search Approach

We provides multiple search functions in our search engine, including search by keywords, search by entities and search by relations. ElasticSearch and Neo4j are utilized to build up the search component, where ElasticSearch mainly handles keyword search and Neo4j handles entity&relation search. For independent resource occupation purposes, a collection of case reports are indexed separately on each search engine. Figure 3.8 illustrates the overall search flow. By default, Neo4j is the primary search engine in `CREATe-IR` system. The results returned by Neo4j will be placed on top, followed by results from ElasticSearch.

Neo4j. In Neo4j, data is saved as a graph of nodes and edges. Therefore, indexing each case report into Neo4j requires a transformation from texts to nodes and edges. A particular node will contain a *nodeId*, a *label* and a *entityType*. Property *label* keeps a natural language description for the node and property *entityType* represents the classification of this node. An edge will contain a *source*, a *target* and a *label*, which records the source nodeId and destination nodeId, as well as the relation type. Then, all nodes and edges are put into Neo4j via cypher query.

ElasticSearch. To better cater to the keyword search demand in `CREATe-IR` system, we build the document index with customized analyzer. In ElasticSearch, an analyzer can be divided into three sub-components: token filters, tokenizers and character filters. For token filter, we choose *asciifolding*, *lowercase*, *snowball*, *stop* and *stemmer*. For tokenizer,

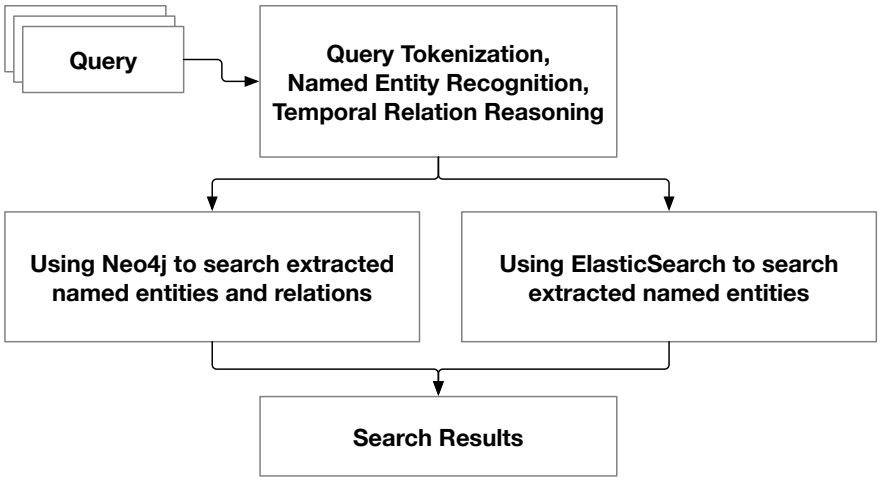


Figure 3.8: The search workflow of CREATe-IR.

considering that some of the symptoms or medications may have longer names, we select N-gram tokenizer and customize it with min_gram=3 and max_gram=25.

3.4.4.5 Event Visualization in Temporal Order

Annotations break the text structure down to event anchors and the temporal relations between them. These representations emphasize the atomic units and relational structures that are crucial for an understanding of the progression of a clinical case. CREATe-IR represents these underlying structures through network graph visualizations, which show the interconnections between sets of entities and events, giving focus to the semantic roles played by these fine-grained elements over the course of a clinical narrative. An example of such a visualization is shown in Figure 3.9. This visualization component is rendered using scalable vector graphics under a force-directed algorithm, which distributes nodes and clusters in space to minimize their repulsive energies and crossing edges. This component also possesses functionality for recognizing standard user interface gestures. The layout of nodes can be reconfigured by selecting a node with the mouse and dragging it to a different location in space. Similarly, the visualization window can be adjusted by zooming and panning using mouse wheel and drag gestures, respectively.

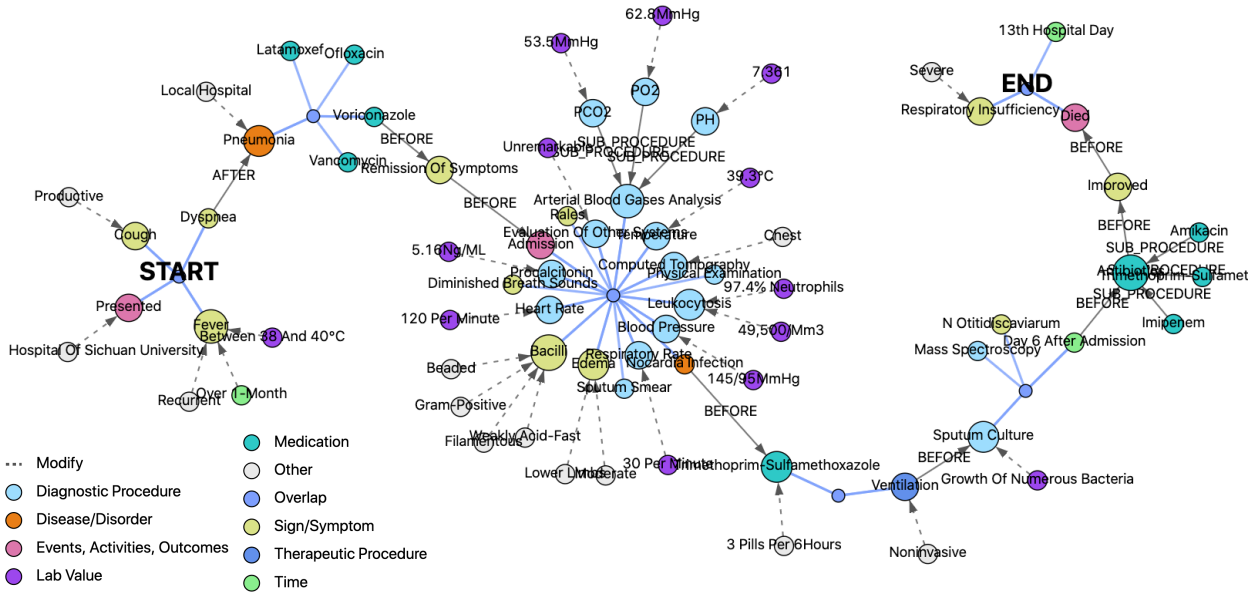


Figure 3.9: Example network graph visualization representing a clinical case matching the query: "A patient was admitted to the hospital because of fever and cough".

3.4.5 Conclusion

In this work, we demonstrate `CREATe`, the first end-to-end system for annotating, indexing and curating the clinical case reports. `CREATe` is powered by `CREATe-IR`, which includes two state-of-the-art information extraction techniques to parse the important clinical events and temporal relationships for retrieving related clinical documents.

3.5 Acknowledgment

The work was supported by NSF DBI-1565137, DGE-1829071, NIH R35-HL135772, NIH U54-GM114833, NSF III-1705169, NSF CAREER Award 1741634, NSF #1937599, DARPA HR00112090027, Okawa Foundation Grant, and Amazon Research Award. Section 3.1, 3.2, 3.3, 3.4 are versions of [50, 275, 279, 271], respectively. I want to thank my co-authors for their contributions to the publications.

CHAPTER 4

Multi-modal Representation Learning for Information Extraction

In this chapter, we introduce two explorations on the multi-modal representation learning algorithms for information extraction tasks: (1) theme (keyphrase) extraction and (2) pun recognition. In the first work, we design a cross-modality representation learning framework to ingest ad images as well as textual information while in the second work, we leverage the self-attention mechanism to learn a joint representation for both text and pronunciation embeddings.

4.1 Recommending Themes for Ads Design via Visual-Linguistic Representations

4.1.1 Motivation

With the widespread usage of online advertising to promote brands (advertisers), there has been a steady need to innovate upon ad formats, and associated ad creatives [2, 277]. The image and text comprising the ad creative can have a significant influence on online users, and their thoughtful design has been the focus of creative strategy teams assisting brands, advertising platforms, and third party marketing agencies. Numerous recent studies have indicated the emergence of a phenomenon called *ad fatigue*, where online users get tired of repeatedly seeing the same ad each time they visit a particular website (*e.g.*, their personalized news stream) [220, 268]. Such an effect is common even in native ad formats

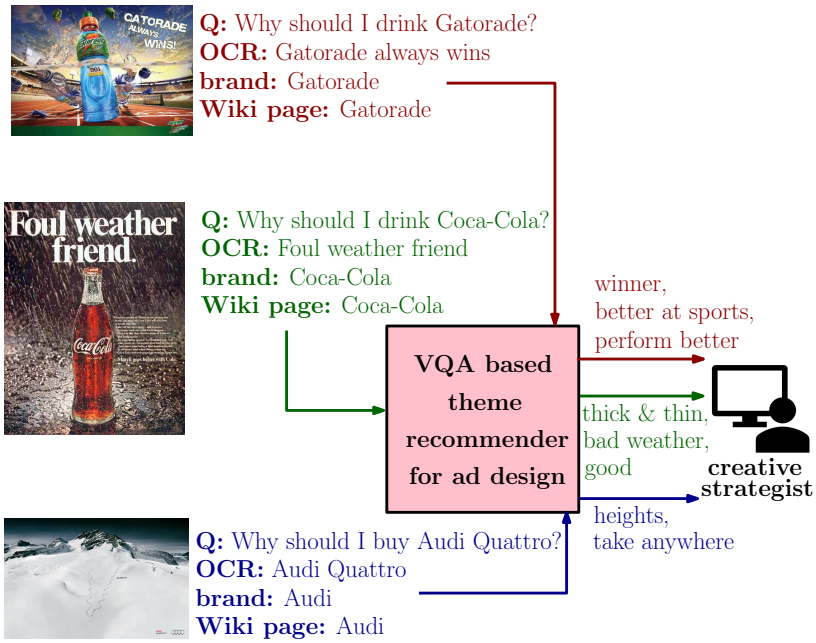


Figure 4.1: Ad (creative) theme recommender based on a VQA approach.

where the ad creatives are supposed to be in line with the content feed they appear in [2, 268]. In this context, frequently refreshing ad creatives is emerging as an effective way to reduce ad fatigue [3, 4].

From a creative strategist's view, coming up with new themes and translating them into ad images and text is a time taking task which inherently requires human creativity. Numerous online tools have emerged to help strategists in translating raw ideas (themes) into actual images and text, *e.g.*, via querying stock image libraries [6], and by offering generic insights on the attributes of successful ad images and text [7]. In a similar spirit, there is room to further assist strategists by automatically recommending brand specific themes which can be used with downstream tools similar to the ones described above. In the absence of human creativity, inferring such brand specific themes using the multi-modal (images and text) data associated with successful past ad campaigns (spanning multiple brands) is the focus of this study.

A key enabler in pursuing the above data driven approach for inferring themes is that of a dataset of ad creatives spanning multiple advertisers. Such a dataset [1] spanning $64,000$

ad images was recently introduced in [113], and also used in the followup work [266]. The collective focus in the above works [113, 266] was on understanding ad creatives in terms of sentiment, symbolic references and VQA. In particular, no connection was made with the brands inferred in creatives, and the associated world knowledge on the inferred brands. As the first work in connecting the above dataset [1] with brands, [181] formulated a *keyword* ranking problem for a brand (represented via its Wikipedia page), and such keywords could be subsequently used as themes for ad creative design. However, the ad images were not used in [181], and recommended themes were restricted to single words (keywords) as opposed to longer keyphrases which could be more relevant. For instance, in Figure 4.1, the phrase *take anywhere* has much more relevance for Audi than the constituent words in isolation.

In this study, we primarily focus on addressing both the above mentioned shortcomings by (i) ingesting ad images as well as textual information, *i.e.*, Wikipedia pages of the brands and text in ad images (OCR), and (ii) we consider keyphrases (themes) as opposed to keywords. Due to the multi-modal nature of our setup, we propose a VQA formulation as exemplified in Figure 4.1, where the questions are around the advertised product (as in [266, 113]) and the answers are in the form of keyphrases (derived from answers in [1]). Brand specific keyphrase recommendations can be subsequently collected from the predicted outputs of brand-related VQA instances. Compared to prior VQA works involving questions around an image, the difference in our setup lies in the use of Wikipedia pages for brands, and OCR features; both of these inputs are considered to assist the task of recommending ad themes.

4.1.2 Contributions

- We study two formulations for VQA based ad theme recommendation (classification and ranking) while using multi-modal sources of information (ad image, OCR, and Wikipedia),
- We show the efficacy of transformer based visual-linguistic representations for our task, with significant performance lifts versus using separate visual and text representations,

- We show that using multi-modal information (images and text) for our task is significantly better than using only visual or textual information, and
- We report selected ad insights from the public dataset [1].

4.1.3 Visual-linguistic representations and VQA

With an increasing interest in joint vision-language tasks like visual question answering (VQA) [17], and image captioning [224], there has been lot of recent work on visual-linguistic representations which are key enablers in the above mentioned tasks. In particular, there has been a surge of proposed methods using transformers [72], and we cover some of them below.

In LXMERT [237], the authors proposed a transformer based model that encodes different relationships between text and visual inputs trained using five different pre-training tasks. More specifically, they use encoders that model text, objects in images and relationship between text and images using (image,sentence) pairs as training data. They evaluate the model on two VQA datasets. More recently ViLBERT [165] was proposed, where BERT [72] architecture was extended to generate multi-modal embeddings by processing both visual and textual inputs in separate streams which interact through co-attentional transformer layers. The co-attentional transformer layers ensure that the model learns to embed the interactions between both modalities. Other similar works include VisualBERT [145], VLBERT [232], and Unicoder-VL [144].

In this study, our goal is to focus on leveraging visual-linguistic representations to solve an ads specific VQA task formulated to infer brand specific ad creative themes. In addition, VQA tasks on ad creatives tend to relatively challenging (*e.g.*, compared to image captioning) due to the subjective nature and hidden symbolism frequently found in ads [266]. Another difference between our work and existing VQA literature is that our task is not limited to understanding the objects in the image but also the emotions the ad creative would evoke in the reader. Our primary task is to predict different themes and sentiments that an ad

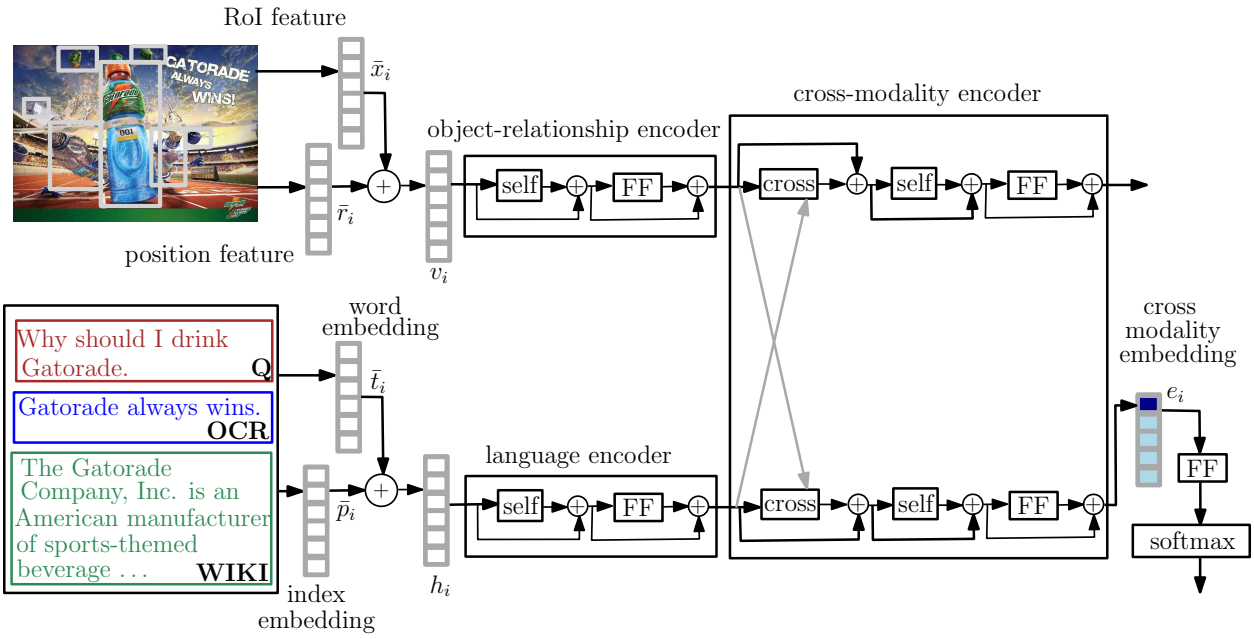


Figure 4.2: Cross modality encoder architecture, and subsequent feed forward (FF) network with softmax layer for the classification objective.

creative image can invoke in its reader, and use such brand specific understanding to help creative strategists in developing new ad creatives.

4.1.4 Methods

4.1.4.1 Theme recommendation: classification formulation

In our setup, we are given an ad image X_i (indexed by i), and associated text denoted by S_i. Text S_i is sourced from: (i) text in ad image (OCR), (ii) questions around the ad, and (iii) Wikipedia page of the brand in the ad. Given X_i, we represent the image as a sequence of objects $x_i = \{x_{i,1}, x_{i,2}, ..., x_{i,n}\}$ together with their corresponding regions in the image $r_i = \{r_{i,1}, r_{i,2}, ..., r_{i,n}\}$ (details in Section 4.1.4.2). The sentence S_i is represented as a sequence of words $w_i = \{w_{i,1}, w_{i,2}, ..., w_{i,m}\}$. Given the three sequences x_i, r_i, w_i, the objective is to recommend a keyphrase $\hat{k}_i \in \mathcal{K}$, where $\mathcal{K}$ is a pre-determined vocabulary of keyphrases. In other words, for $k \in \mathcal{K}$, the goal is to estimate the probability $\mathbb{P}(k|x_i, r_i, w_i)$,

and then the top keyphrase $\hat{k}_i$ for instance i can be selected as:

$$\hat{k}_i = \underset{k \in \mathcal{K}}{\operatorname{argmax}} \ \mathbb{P}(k|x_i, r_i, w_i). \tag{4.1}$$

The above classification formulation is similar to that for VQA in [266]; the difference is in the multi-modal features explained below.

4.1.4.2 Text and image embeddings

Text embedding. We first use WordPiece Tokenizer [260] to convert a sentence w_i into a sequence of tokens $t_i = \{t_{i,1}, t_{i,2}, ..., t_{i,l}\}$. Then, the tokens are projected to vectors in the embedding layer leading to $\bar{t}_i$ (as shown in (4.2)). Their corresponding positions p_i are also projected to vectors leading to $\bar{p}_i$ (as shown in (4.3)). Then, $\bar{t}_i$ and $\bar{p}_i$ are added to form h_i as shown in (4.4) below:

$$\bar{t}_i = E_t * t_i, \tag{4.2}$$

$$\bar{p}_i = E_p * p_i, \tag{4.3}$$

$$h_i = 0.5 * (\bar{t}_i + \bar{p}_i), \tag{4.4}$$

where $E_t \in \mathbb{R}^{|V_t| \times D_t}$ and $E_p \in \mathbb{R}^{|V_p| \times D_p}$ are the embedding matrices. $|V_t|$ and $|V_p|$ are the vocabulary size of tokens, and token positions. D_t and D_p are the dimensions of token and position embeddings.

Image embedding. We use bounding boxes and their region-of-interest (RoI) features to represent an image. Same as [165, 237], we leverage Faster R-CNN [209] to generate the bounding boxes and RoI features. Faster R-CNN is an object detection tool which identifies instances of objects belonging to certain classes, and then localizes them with bounding boxes. Though image regions lack a natural ordering compared to token sequences, the spatial locations can be encoded (*e.g.*, as demonstrated in [237]). The image embedding layer takes in the RoI features x_i and spatial features r_i and outputs a position-aware image

embedding v_i as shown below:

$$\bar{x}_i = W_x * x_i + b_x,$$
$$\bar{r}_i = W_r * r_i + b_r,$$
$$v_i = 0.5 * (\bar{x}_i + \bar{r}_i), \tag{4.5}$$

where W_x and W_r are weights, and b_x and b_r are biases.

4.1.4.3 Transformer-based cross-modality encoder

We apply a transformer-based cross-modality encoder to learn a joint embedding from both visual and textual features. Here, without loss of generality, we follow the LXMERT architecture from [237] to encode the cross-modal features. As shown in Figure 4.2, the token embedding h_i is first fed into a language encoder while the image embedding v_i goes through an object-relationship encoder. The cross-modality encoder contains two unidirectional cross-attention sub-layers which attend the visual and textual embeddings to each other. We use the cross-attention sub-layers to align the entities from two modalities and to learn a joint embedding e_i of dimension D_e. We follow [72] to add a special token [CLS] to the front of the token sequence. The embedding vector learned for this special token is regarded as the cross-modal embedding*. In terms of query (Q), key (K), and value (V), the visual *cross-attention*$(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V$ where Q, K, and V are linguistic features, visual features, and visual features, respectively; d represents the dimension of linguistic features [237]. Textual cross-attention is similar with visual and linguistic features swapped.

4.1.4.4 Learning and optimization.

Based on the joint embedding for each image and sentence pair, the keyphrase recommendation task can now be tackled with a fully connected layer. Given the cross-modal embedding, the probability distribution over all the candidate keyphrases is calculated by a

*Recently proposed ViLBERT [165], and VisualBERT [145] can serve as alternatives.

fully-connected layer and the softmax function as shown below:

$$\hat{\mathbb{P}}(k|x_i, r_i, w_i) = \text{softmax}(W_f \cdot e_i + b_f) \tag{4.6}$$

where W_f and b_f are the weight and bias of a fully connected layer, and e_i is the cross modal embedding.

4.1.4.5 Theme recommendation: ranking formulation

We also consider solving the theme recommendation problem via a ranking model, where the model outputs a list of keyphrases in decreasing order of relevance for a given *(image, sentence)* pair, *i.e.*, (X_i, S_i). We use the state-of-the-art pairwise deep relevance matching model (DRMM) [93] whose architecture for our theme recommendation setup is shown in Figure 4.3. It is worth noting that our pairwise ranking formulation can be changed to accommodate other multi-objective or list-based loss-functions. We chose the DRMM model since it is not restricted by the length of input, as most ranking models are, but relies on capturing local interactions between query and document terms with fixed length matching histograms. Given an *(image, sentence, phrase)* combination, the model first computes fixed length matching histogram between cross-modal embedding and the phrase embedding. Each matching histogram is passed through a multi-layer perceptron (MLP), and the overall score is aggregated with a query term gate which is a softmax function over all terms in that query.

The ranking segment of the model takes two inputs: (i) cross-modal embedding for (X_i, S_i) (as explained in Section 4.1.4.3), and (ii) the phrase embedding. It then learns to predict the relevance of the given phrase with respect to the query *(image, sentence)* pair. Given that our input documents (*i.e.*, keyphrases) are short, we select top θ interactions in matching histogram between the cross-modal embedding, and the keyphrase embedding. Mathematically, we denote the *(image, sentence)* pair by just the *imgq* below. Given a triple ($imgq$, p^+, p^-) where p^+ is ranked higher than p^- with respect to image-question, the loss function is defined as:

$$\mathcal{L}(imgq, p^+, p^-; \theta) = max(0, 1 - s(imgq, p^+) + s(imgq, p^-)), \tag{4.7}$$

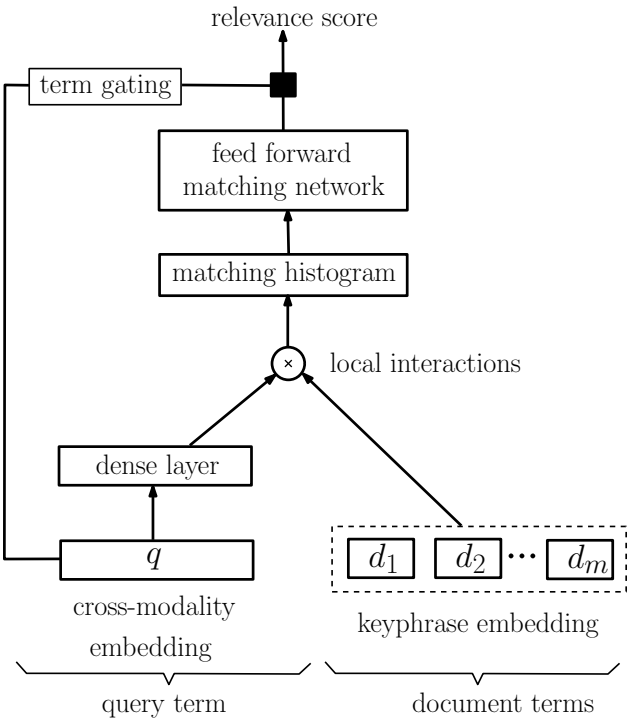


Figure 4.3: DRMM for the keyphrase ranking objective.

where $s(imgq, p)$ denotes the predicted matching score for phrase p, and the query image-question pair.

4.1.5 Experiments

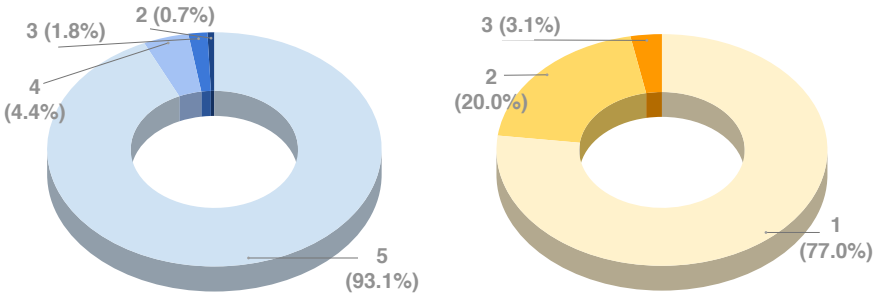
In this section, we go over the public dataset used in our experiments, classification and ranking results, and inferred insights.

4.1.5.1 Dataset

We rely on a publicly available data set [113, 1] that consists of 64,000 advertisement creatives, spanning 700 brands across 39 categories, among which 80% is training set and 20% is test set. We select 10% data from the training set for validation. Crowdsourcing was used to gather following labels for each creative: (i) topics (39 types), (ii) questions and answers as reasons for buying from the brand depicted in the creative ($\sim$3 per creative). In

addition to the existing annotations, we add the following annotations: (i) brand present in a creative, (ii) Wikipedia page relevant to the brand-category pair in a creative, and (iii) the set of target themes (keyphrases) associated with each image. In particular, for (i) and (ii) we follow the method in [181], and for (iii) the keyphrases (labels) were extracted from the answers using the position-rank method [42, 83] for each image. The number of keyphrases was limited to at most 5 (based on the top keyphrase scores returned by position-rank). We define a score for each keyphrase. All five keyphrases have scores of $1.0, 0.9, 0.8, 0.7$, and 0.6 in order [†].

The minimum, mean and maximum number of images associated with a brand are 1, 19 and 282 respectively. The top three categories of advertisements are *clothing*, *cars* and *beauty products* with 7798, 6496 and 5317 images respectively. Least number of advertisements are associated with *gambling* (32), *pet food* (37) and *security and safety services* (47) respectively. Additional statistics around the dataset (*i.e.*, keyphrase lengths, images per category, and unique keyphrases per category) are shown in Figures 4.4, 4.5, and 4.6.



(a) #keyphrases per instance (b) #words per keyphrase

Figure 4.4: Frequency and length distribution of keyphrases.

4.1.5.2 Evaluation Metrics

We use different evaluation metrics to measure performance of our classification and ranking models. We use three different metrics to evaluate the performance of each model. [Ja)

[†]The annotated ads dataset can be found at `https://github.com/joey1993/ad-themes`.

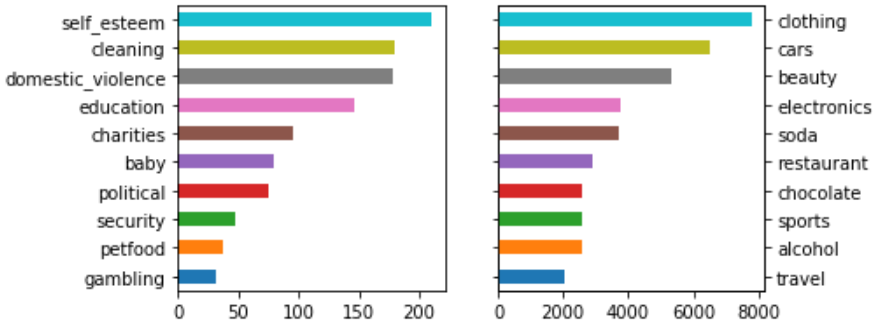


Figure 4.5: Distribution of images per category.

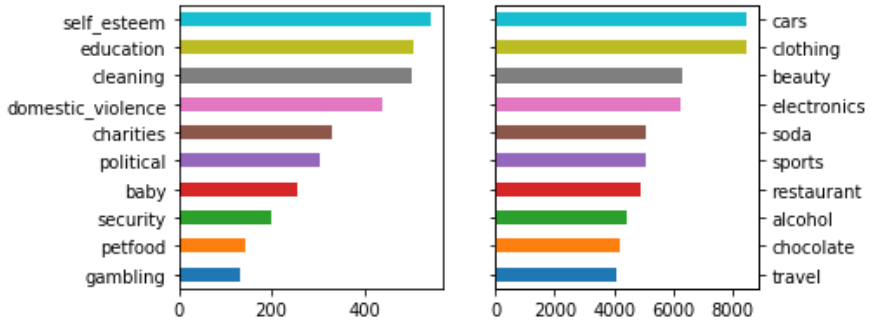


Figure 4.6: Distribution of unique phrases per category.

accuracy, similarity, VQA recall (as defined below) to evaluate our proposed method to evaluation classification and *b)* precision (P@K), recall (R@K), and normalized cumulative discounted gain [118] (NDCG@K) to evaluate our model with ranking formulation.

Classification metrics. We rely on accuracy, text similarity and set-intersection based recall to evaluate model performance.

- **Accuracy.** We predict the keyphrase with the highest probability for each image and match it with the labels (ground truth keyphrases) for the image. We use the score of the matched phrase as the accuracy. If no labels match for a sample, the accuracy is 0. We average the accuracy scores over all the test instances to report test accuracy.

- **Similarity:** Accuracy neglects the semantic similarity between the predicted phrase and the labels. For example, a predicted keyphrase "a great offer" is similar to one of the labels, "great sale", but will gain 0 for accuracy. So we calculate the cosine

similarity [96] between the embeddings of the predicted keyphrase and each label. Then we multiply the similarity scores with each label's score and keep the maximum as the final similarity score for the sample.

- **VQA Recall@3:** we use Recall at 3 (R_{VQA}@3) as an evaluation metric for the classification task (essentially like the VQA formulation in [266]). For each test instance i, the ground truth is limited to top 3 keyphrases leading to set $\mathcal{K}_i^*$. From the classification model's predictions the top 3 keyphrases are chosen leading to set $\hat{\mathcal{K}}_i$. R_{VQA}@3 is simply $\frac{|\hat{\mathcal{K}}_i \cap \mathcal{K}_i^*|}{3}$.

Ranking metrics. We use the same evaluation metrics from prior work [181], mainly precision (P@K), recall (R@K), and NDCG@K [118] to evaluate the proposed ranking model. It is worth noting that recall is computed differently for evaluating ranking and classification models proposed in this work. Formally, given a set of queries $\mathcal{Q} = \{q_1 \cdots q_n\}$, set of phrases $\mathcal{D}_i$ labeled *relevant* for each query q_i and the set of relevant phrases $\mathcal{D}_{ik}$ retrieved by the model for q_i at position k, we define $R@K = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{|\mathcal{D}_{ik}|}{|\mathcal{D}_i|}$.

4.1.5.3 Implementation details

For the classification model, we set the number of object-relationship, language, and cross-modality layers as 5, 9, 5, and leverage pre-trained parameters from [237]. We fine-tune the encoders with our dataset for 4 epochs. The learning rate is $5e^{-5}$ (adam optimizer), and the batch size is 32. We also set D_t, D_p, and D_e equal to 768. For the similarity evaluation, we average the GloVe [200] embeddings of all the words in a phrase to calculate the phrase embedding. For the DRMM model (ranking formulation), we used the MatchZoo implementation [5], with 300 for batch size, 10 training epochs, 10 as the last layer's size, and learning rate of 1.0 (adadelta optimizer). We combine textual data from different sources in a consistent order separated by a [SEP] symbol before feeding them into the encoder.

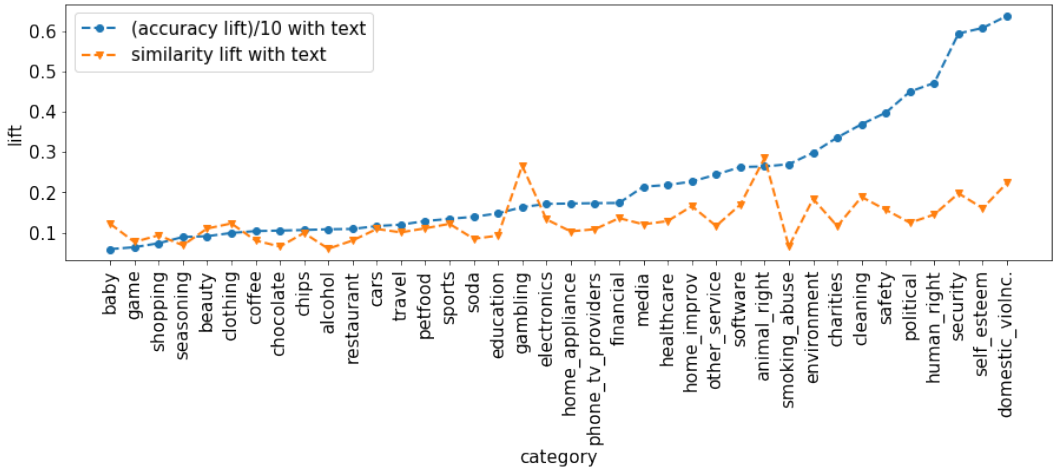


Figure 4.7: Performance lifts across different categories after using text features.

4.1.5.4 Results

For different sets of multi-modal features, the performance results are reported in Table 4.1 (for classification) and Table 4.2 (for ranking) respectively. The presence of Wikipedia and OCR text gives a significant lift over using only the image. Both classification and ranking metrics show the same trend in terms of feature sets.

Table 4.1 shows that linguistic features dramatically lift the performance by 103%, 5%, and 6% in accuracy, similarity, and R_{VQA}@3, compared to the performance of the model trained *only* with visual features while only using linguistic features (Q+W+O) causes a big drop in all the performances. It occurs that the OCR features bring more performance lift, compared to the Wiki. We think knowing more about the brand with the Wikipedia pages is beneficial to recommend themes to designers [181] while the written texts on the images (OCR) are sometimes more straightforward for recommendations. In addition, as reported in Table 4.1 (non cross-modal), using separate text and image embeddings (obtained from the model in Figure 4.2) is inferior in performance compared to the cross-modal embeddings. We notice that the accuracy scores are comparatively low; this reflects the difficult nature of understanding visual ads [113].

In Table 4.2, we observe very similar patterns: OCR features bring more benefits to

Table 4.1: Classification performance with different features.

features	accuracy (%)	similarity (%)	R_{VQA}@3
I	10.05	58.05	0.447
I×Q	12.18	58.26	0.450
I×(Q+W)	19.01	60.12	0.467
I×(Q+O)	19.50	60.34	0.470
I×(Q+W+O)	**20.40**	**60.95**	**0.473**
Q+W+O	13.39	60.13	0.450
non cross-modal	18.65	60.68	0.460

Table 4.2: Ranking performance with different features.

Features	Precision		Recall		NDCG	
	@5	@10	@5	@10	@5	@10
I	0.150	0.126	0.161	0.248	0.158	0.217
I×Q	0.152	0.124	0.158	0.259	0.162	0.227
I×(Q+W)	0.154	0.130	0.160	0.271	0.161	0.234
I×(Q+O)	0.174	0.137	0.182	0.287	0.185	0.254
I×(Q+W+O)	**0.183**	**0.141**	**0.191**	**0.294**	**0.198**	**0.265**

ranking than Wikipedia pages. We notice that in P@10 and R@5, only using image feature (I) achieves a better score compared to adding the question features. This may indicate that the local interactions in DRMM are not effective with short questions, but favor longer textual inputs such as OCR and Wikipedia pages.

4.1.5.5 Insights

Figure 4.7 shows the performance lifts in accuracy and similarity metrics per category (where lift is defined as ratio of improvement to baseline result without using text features in the

Figure 4.8: The ad image on the left is a sample in the public dataset [1], and the ground truth keyphrases with scores are as shown.

classification task). As shown, multiple categories, *e.g.*, public service announcement (PSA) ads around domestic violence and animal rights benefit from the presence of text features; this may be related to the hidden symbolism [266] common in PSAs, where the text can help clarify the context even for humans. Also, similarity and accuracy metrics do not have the same trend in general. Along the lines of inferring themes from past ad campaigns, and assisting strategists towards designing new creatives, we show an example based on our classification model in Figure 4.8. In general, a strategist can aggregate recommended keyphrases across a brand or product category, and use them to design new creatives.

4.1.6 Conclusion

In this study, we make progress towards automating the inference of themes (keyphrases) from past ad campaigns using multi-modal information (*i.e.*, images and text). The proposed method can increase diversity in ad campaigns (and potentially reduce ad fatigue), reduce end-to-end design time, and enable faster exploratory learnings from online ad campaigns by providing multiple themes per brand (and multiple images per theme via stock image libraries).

4.2 Pronunciation-attentive Contextualized Pun Recognition

4.2.1 Motivation

During the last decades, social media has promoted the creation of a vast amount of humorous web contents [188]. Automatic recognition of humor has become an important task in the area of figurative language processing, which can benefit various downstream NLP applications such as dialogue systems, sentiment analysis, and machine translation [171, 22, 90, 33, 40, 274]. However, humor is one of the most complicated behaviors in natural language semantics and sometimes it is even difficult for humans to interpret. In most cases, understanding humor requires adequate background knowledge and a rich context.

Puns are a form of humorous approaches using the different meanings of identical words or words with similar pronunciations to explain texts or utterances. There are two main types of puns. Homographic puns rely on multiple interpretations of the same word. As shown in Table 4.3, the phrase *all right* means *good condition* or *opposite to left*; the word *reaction* means *chemical change* or *action*. The two meanings of the same expression are consistent with its context, which creates a humorous pun in both sentences when there is a clear contrast between two meanings. On the other hand, heterographic puns take advantage of phonologically same or similar words. For example, the word pairs *sale* and *sail*, *weak* and *week* in Table 4.3 have the same or similar pronunciations. The sentences are funny because both words fit the same context. Understanding puns is a big fish to fry for deep comprehension of complex semantics.

These two forms of puns have been studied in literature from different angles. To recognize puns in a sentence, word sense disambiguation techniques (WSD) [187] have been employed to identify the equitable intention of words in utterances [198]. External knowledge bases such as WordNet [178] have been applied in determining word senses of pun words [191]. However, these methods cannot tackle heterographic puns with distinct word spellings and knowledge bases that only contain a limited vocabulary. To resolve the issues

	Homographic Puns
1.	Did you hear about the guy whose whole left side was cut off? He's **all right** now.
2.	I'd tell you a chemistry joke but I know I wouldn't get a **reaction**.
	Heterographic Puns
1.	The boating store had its best **sail (sale)** ever.
2.	I lift weights only on Saturday and Sunday because Monday to Friday are **weak (week)** days.

Table 4.3: Examples of homographic and heterographic puns.

of sparseness and heterographics, the word embedding techniques [176, 199] provide flexible representations to model puns [112, 115, 45]. However, a word may have different meanings regarding its contexts. Especially, an infrequent meaning of the word might be utilized for creating a pun. Therefore, static word embeddings are insufficient to represent words. replacing a word with another word with the same or similar pronunciation as examples shown in Table 4.3. Therefore, to recognize puns, it is essential to model the association between words in the sentence and the pronunciation of words. Despite existing approaches attempt to leverage phonological structures to understand puns [76, 117], there is a lack of a general framework to model these two types of signals in a whole.

In this work, we propose Pronunciation-attentive Contextualized Pun Recognition (PCPR) to jointly model the contextualized word embeddings and phonological word representations for pun recognition. To capture the phonological structures of words, we break each word into a sequence of phonemes as its pronunciation so that homophones can have similar phoneme sets. For instance, the phonemes of the word *pun* are {P, AH, N}. In PCPR, we construct a pronunciation attentive module to identify important phonemes of each word, which can be applied in other tasks related to phonology. We jointly encode the contextual and phonological features into a self-attentive embedding to tackle both pun detection and location tasks.

4.2.2 Contributions

- To the best of our knowledge, PCPR is the first work to jointly model contextualized word embeddings and pronunciation embeddings to recognize puns. Both contexts and phonological properties are beneficial to pun recognition.
- Extensive experiments are conducted on two benchmark datasets. PCPR significantly outperforms existing methods in both pun detection and pun location. In-depth analyses also verify the effectiveness and robustness of PCPR.
- We release our implementations and pre-trained phoneme embeddings[‡] to facilitate future research.

4.2.3 Methods

4.2.3.1 Problem Statement

Suppose the input text consists of a sequence of N words $\{w_1, w_2, \cdots, w_N\}$. For each word w_i with M_i phonemes in its pronunciation, the phonemes are denoted as $R(w_i) = \{r_{i,1}, r_{i,2}, \cdots, r_{i,M_i}\}$, where $r_{i,j}$ is the j-th phoneme in the pronunciation of w_i. These phonemes are given by a dictionary. In this work, we aim to recognize potential puns in the text with two tasks, including pun detection and pun location, as described in the following.

Task 1: Pun Detection. The pun detection task identifies whether a sentence contains a pun. Formally, the task is modeled as a classification problem with binary label y^D.

Task 2: Pun Location. Given a sentence containing at least a pun, the pun location task aims to unearth the pun word. More precisely, for each word w_i, we would like to predict a binary label y_i^L that indicates if w_i is a pun word.

In addition to independently solving the above two tasks, the ultimate goal of pun recognition is to build a pipeline from scratch to detect and then locate the puns in texts. Hence,

[‡]Codes can be found at https://github.com/joey1993/pun-recognition.

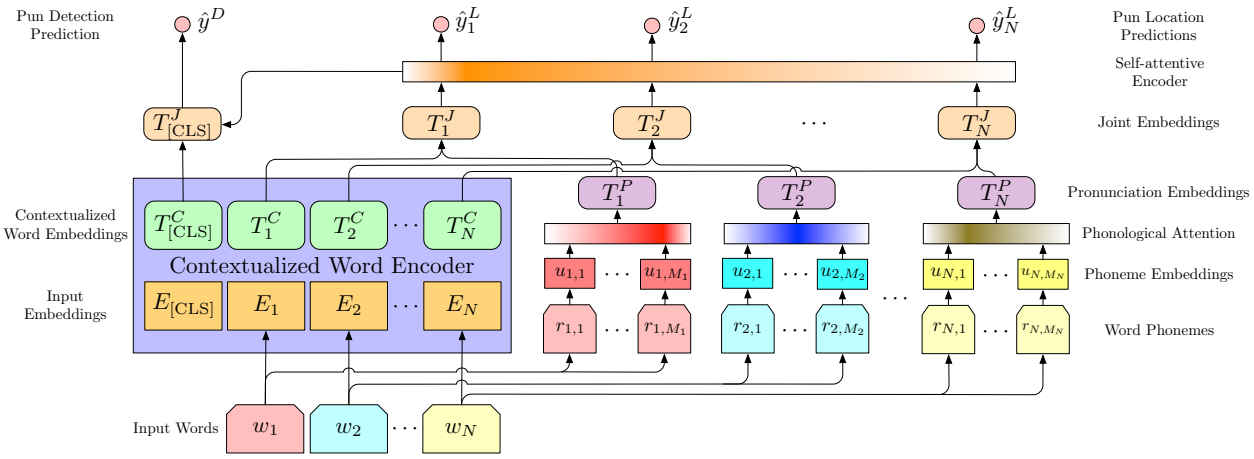


Figure 4.9: The overall framework of PCPR.

we also evaluate the end-to-end performance by aggregating the solutions for two tasks.

4.2.3.2 Framework Overview

Figure 4.9 shows the overall framework of the proposed Pronunciation-attentive Contextualized Pun Recognition (PCPR). For each word in the input text, we first derive two continuous vectors, including contextualized word embedding and pronunciation embedding, as representations in different aspects. Contextualized word embeddings derive appropriate word representations with consideration of context words and capture the accurate semantics in the text. To learn the phonological characteristics, each word is divided into phonemes while each phoneme is projected to a phoneme embedding space, thereby obtaining pronunciation embeddings with the attention mechanism [25]. Finally, a self-attentive encoder blends contextualized word embeddings and pronunciation embeddings to capture the overall semantics for both pun detection and location.

4.2.3.3 Contextualized Word Embeddings

The context is essential for interpreting a word in the text. Hence, we propose to apply contextualized word embeddings to derive word representations. In the framework of PCPR, any contextualized word embedding method, such as BERT [72], ELMo [202], and XLNet [265],

can be utilized. Here, we choose BERT to derive contextualized word embeddings without loss of generality.

BERT deploys a multi-layer bidirectional encoder based on transformers with multi-head self-attention [248] to model words in the text after integrating both word and position embeddings [233]. As a result, for each word, a representative contextualized embedding is derived by considering both the specific word and all contexts in the document. Here we denote T_i^C as the d_C-dimensional contextualized word embedding for the word w_i. In addition, BERT contains a special token `[CLS]` with an embedding vector in BERT to represent the semantics of the whole input text.

4.2.3.4 Pronunciation Embeddings

To learn the phonological characteristics of words, `PCPR` models the word phonemes. For each phoneme $r_{i,j}$ of the word w_i, we project $r_{i,j}$ to a d_P-dimensional embedding space as a trainable vector $u_{i,j}$ to represent its phonological properties.

Based on the phoneme embeddings of a word, we apply the attention mechanism [25] to simultaneously identify important phonemes and derive the pronunciation embedding T_i^P. Specifically, the phoneme embeddings are transformed by a fully-connected hidden layer to measure the importance scores α_i^P as follows:

$$v_{i,j} = \tanh(\mathcal{F}_P(u_{i,j})),$$

$$\alpha_{i,j}^P = \frac{v_{i,j}^\intercal v_s}{\sum_k v_{i,k}^\intercal v_s},$$

where $\mathcal{F}_P(\cdot)$ is a fully-connected layer with d_A outputs and d_A is the attention size; v_s is a d_A-dimensional context vector that estimates the importance score of each pronunciation embedding. Finally, the pronunciation embeddings T_i^P can be represented as the weighted combination of phoneme embeddings as follows:

$$T_i^P = \sum_j \alpha_{i,j} u_{i,j}.$$

Moreover, we can further derive the joint embedding T_i^J to indicate both word semantics and phonological knowledge for the word w_i by concatenating two different embeddings as follows:

$$T_i^J = \left[T_i^C; T_i^P\right].$$

Note that the joint embeddings are d_J-dimensional vectors, where $d_J = d_C + d_P$.

4.2.3.5 Pronunciation-attentive Contextualized Embedding with Self-attention

For the task of pun detection, understanding the meaning of input text is essential. Due to its advantages of interpretability over convolutional neural network [136] and recurrent neural network [221], we deploy the self-attention mechanism [248] to capture the overall semantics represented in the joint embeddings. For each word w_i, the self-attention mechanism estimates an importance vector α_i^S:

$$\mathcal{F}_S(T) = \text{Softmax}(\frac{TT^\intercal}{\sqrt{d}})T,$$

$$\alpha_i^S = \frac{\exp(\mathcal{F}_S(T_i^J))}{\sum_j \exp(\mathcal{F}_S(T_j^J))},$$

where $\mathcal{F}_S(\cdot)$ is the function to estimate the attention for queries, and d is a scaling factor to avoid extremely small gradients. Hence, the self-attentive embedding vector is computed by aggregating joint embeddings:

$$T_{\texttt{[ATT]}}^J = \sum_i \alpha_i^S \cdot T_i^J.$$

Note that the knowledge of pronunciations is considered by the self-attentive encoder but not the contextualized word encoder. Finally, the pronunciation-attentive contextualized representation for the whole input text can be derived by concatenating the overall contextualized embedding and the self-attentive embedding:

$$T_{\texttt{[CLS]}}^J = \left[T_{\texttt{[CLS]}}^C; T_{\texttt{[ATT]}}^J\right].$$

Moreover, each word w_i is benefited from the self-attentive encoder and is represented by a joint embedding:

$$T_{\texttt{i,[ATT]}}^J = \alpha_i^S \cdot T_i^J.$$

4.2.3.6 Inference and Optimization

Based on the joint embedding for each word and the pronunciation-attentive contextualized embedding for the whole input text, both tasks can be tackled with simple fully-connected layers.

Pun Detection. Pun detection is modeled as a binary classification task. Given the overall embedding for the input text $T^J_{\texttt{[CLS]}}$, the prediction $\hat{y}^D$ is generated by a fully-connected layer and the softmax function:

$$\hat{y}^D = \underset{k \in \{0,1\}}{\operatorname{argmax}} \mathcal{F}_D(T^J_{\texttt{[CLS]}})_k,$$

where $\mathcal{F}_D(\cdot)$ derives the logits of two classes in binary classification.

Pun Location. For each word w_i, the corresponding self-attentive joint embedding $T^J_{\texttt{i,[ATT]}}$ is applied as features for pun location. Similar to pun detection, the prediction $\hat{y}^L_i$ is generated by:

$$\hat{y}^L_i = \underset{k \in \{0,1\}}{\operatorname{argmax}} \mathcal{F}_L(T^J_{\texttt{i,[ATT]}})_k,$$

where $\mathcal{F}_L(\cdot)$ derives two logits for classifying if a word is a pun word.

Since both tasks focus on binary classification, we optimize the model with cross-entropy loss.

4.2.4 Experiments

In this section, we describe our experimental settings and explain the results and interpretations. We will verify some basic assumptions of this work: (1) the contextualized word embeddings and pronunciation embeddings are both beneficial to the pun detection and location tasks; (2) the attention mechanism can improve the performance.

Table 4.4: Homographic and Heterographic Pun Data statistics.

Dataset	SemEval		PTD
	Homo	Hetero	
Examples w/ Puns	1,607	1,271	2,423
Examples w/o Puns	643	509	2,403
Total Examples	2,250	1,780	4,826

4.2.4.1 Experiment settings

Experimental Datasets. We conducted experiments on the SemEval 2017 shared task 7 dataset[§] (SemEval) [180] and the Pun of The Day dataset (PTD) [264]. For pun detection, the SemEval dataset consists of $4,030$ and $2,878$ examples for pun detection and location while each example with a pun can be a homographic or heterographic pun. In contrast, the PTD dataset contains $4,826$ examples without labels of pun types. Table 4.4 further shows the data statistics. The two experimental datasets are the largest publicly available benchmarks that are used in the existing studies. SemEval-2017 dataset contains punning and non-punning jokes, aphorisms, and other short texts composed by professional humorists and online collections. Hence, we assume the genres of positive and negative examples should be identical or extremely similar.

Evaluation Metrics. We adopt precision (P), recall (R), and F_1-score [222, 204] to compare the performance of `PCPR` with previous studies in both pun detection and location. More specifically, we apply 10-fold cross-validation to conduct evaluation. For each fold, we randomly select 10% of the instances from the training set for development. To conduct fair comparisons, we strictly follow the experimental settings in previous studies [281, 45] and include their reported numbers in the comparisons.

Implementation Details. For data pre-processing, all of the numbers and punctuation marks are removed. The phonemes of each word are derived by the CMU Pronouncing Dictio-

[§]`http://alt.qcri.org/semeval2017/task7/`

nary[¶]. We initialize the phoneme embeddings by using the *fastText* word embedding [175] trained on Wikipedia articles[‖] crawled in December, 2017. The `PCPR` is implemented in PyTorch while the fused Adam optimizer [123] optimizes the parameters with an initial learning rate of 5×10^{-5}. The dropout and batch size are set as 10^{-1} and 32. We follow BERT (BASE) [72] to use 12 Transformer layers and self-attention heads. To clarify, in `PCPR`, tokens and phonemes are independently processed, so the tokens processed with WordPiece tokenizer [260] in BERT are not required to line up with phonemes for computations. To deal with the out-of-vocabulary words, we use the output embeddings of the first WordPiece tokens as the representatives, which is consistent with many state-of-the-art named entity recognition approaches [72, 139]. We also create a variant of `PCPR` called `CPR` by exploiting only the contextualized word encoder without considering phonemes to demonstrate the effectiveness of pronunciation embeddings.

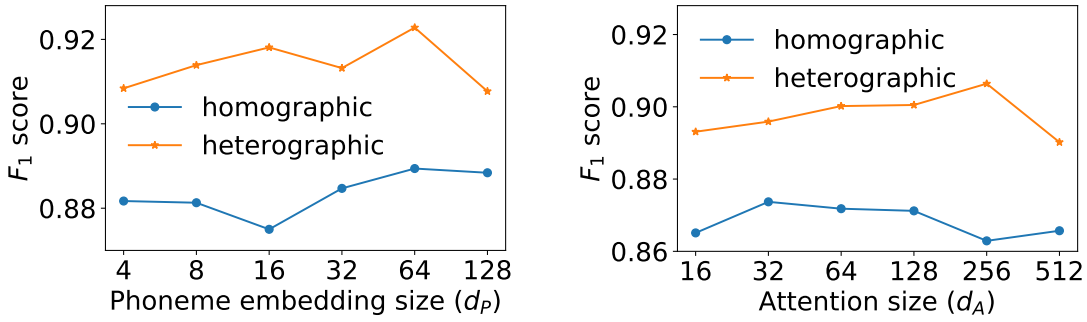
To tune the hyperparameters, we search the phoneme embedding size d_P and the attention size d_A from $\{8, 16, 32, 64, 128, 256, 512\}$ as shown in Figure 4.10. For the SemEval dataset, the best setting is $(d_P = 64, d_A = 256)$ for the homographic puns while heterographic puns favor $(d_P = 64, d_A = 32)$. For the PTD dataset, $(d_P = 64, d_A = 32)$ can reach the best performance.

Baseline Methods. For the SemEval dataset, nine baseline methods are compared in the experiments, including `Duluth` [198], `JU_CES_NLP` [205], `PunFields` [174], `UWAV` [246], `Fermi` [115], and `UWaterloo` [249]. While most of them extract complicated linguistic features to train rule based and machine learning based classifiers. In addition to task participants, `Sense` [45] incorporates word sense representations into RNNs to tackle the homographic pun location task. The `CRF` [281] captures linguistic features such as POS tags, n-grams, and word suffix to model puns. Moreover, the `Joint` [281] jointly models two tasks with RNNs and a CRF tagger.

For the PTD dataset, four baseline methods with reported performance are selected for

[¶]`http://svn.code.sf.net/p/cmusphinx/code/trunk/cmudict/`

[‖]`https://dumps.wikimedia.org/enwiki/latest/`



(a) Phoneme emb. size d_P

(b) Attention size d_A

Figure 4.10: Pun location performance over different phoneme embedding sizes d_P and attention sizes d_A on the SemEval dataset.

comparisons. `MCL` [173] exploits word representations with multiple stylistic features while `HAE` [264] applies a random forest model with Word2Vec and human-centric features. `PAL` [56] trains a convolutional neural network (CNN) to learn essential feature automatically. Based on existing CNN models, `HUR` [57] improves the performance by adjusting the filter size and adding a highway layer.

4.2.4.2 Experimental Results

Pun Detection. Table 4.5 presents the pun detection performance of methods for both homographic and heterographic puns on the SemEval dataset while Table 4.6 shows the detection performance on the PTD dataset. For the SemEval dataset, compared to the nine baseline models, `PCPR` achieves the highest performance with 3.0% and 6.1% improvements of F_1 against the best among the baselines (i.e. `Joint`) for the homographic and heterographic datasets, respectively. For the PTD dataset, `PCPR` improves against `HUR` by 9.6%. Moreover, the variant `CPR` beats all of the baseline methods and shows the effectiveness of contextualized word embeddings. In addition, `PCPR` further improves the performances by 2.3% and 1.1% with the attentive pronunciation feature for detecting homographic and heterographic puns, respectively. An interesting observation is that pronunciation embeddings also facilitate

Table 4.5: Performance of detecting and locating puns on the SemEval dataset.

Model	Homographic Puns						Heterographic Puns					
	Pun Detection			Pun Location			Pun Detection			Pun Location		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Duluth	78.32	87.24	82.54	44.00	44.00	44.00	73.99	86.62	68.71	-	-	-
JU_CSE_NLP	72.51	90.79	68.84	33.48	33.48	33.48	73.67	94.02	71.74	37.92	37.92	37.92
PunFields	79.93	73.37	67.82	32.79	32.79	32.79	75.80	59.40	57.47	35.01	35.01	35.01
UWAV	68.38	47.23	46.71	34.10	34.10	34.10	65.23	41.78	42.53	42.80	42.80	42.80
Fermi	90.24	89.70	85.33	52.15	52.15	52.15	-	-	-	-	-	-
UWaterloo	-	-	-	65.26	65.21	65.23	-	-	-	79.73	79.54	79.64
Sense	-	-	-	81.50	74.70	78.00	-	-	-	-	-	-
CRF	87.21	64.09	73.89	86.31	55.32	67.43	89.56	70.94	79.17	88.46	62.76	73.42
Joint	91.25	93.28	92.19	83.55	77.10	80.19	86.67	93.08	89.76	81.41	77.50	79.40
CPR	91.42	94.21	92.79	88.80	85.65	87.20	93.35	95.04	94.19	92.31	88.24	90.23
PCPR	**94.18**	**95.70**	**94.94**	**90.43**	**87.50**	**88.94**	**94.84**	**95.59**	**95.22**	**94.23**	**90.41**	**92.28**

homographic pun detection, implying the potential of pronunciation for enhancing general language modeling. All improvements of PCPR and CPR over baseline methods are statistically significant at a 95% confidence level in paired t-tests.

Pun Location. Table 4.5 shows that the proposed PCPR model achieves highest F_1-scores on both homographic and heterographic pun location tasks with 10.9% and 15.9% incredible increment against the best baseline method. The improvement is much larger than that on pun detection task. We posit the reason is that predicting pun locations relies much more on the comparative relations among different tokens in one sentence. As a result, contextualized word embeddings acquire an enormous advantage. By applying the pronunciation-attentive representations, different words with similar pronunciations are linked, leading to a much better pinpoint of pun word for the heterographic dataset. We notice that some of the baseline models such as UWaterloo, UWAV and PunFields have poor performances. These methods consider the word position in a sentence or calculate the inverse document frequency of words. We suppose such rule-based recognition techniques can hardly capture the deep

Table 4.6: Performance of pun detection on the PTD dataset.

Model	P	R	F_1
MCL	83.80	65.50	73.50
HAE	83.40	88.80	85.90
PAL	86.40	85.40	85.70
HUR	86.60	94.00	90.10
CPR	98.12	**99.34**	98.73
PCPR	**98.44**	99.13	**98.79**

Table 4.7: Performance of pipeline recognition in the SemEval dastaset.

Model	Homographic Puns			Heterographic Puns		
	P	R	F_1	P	R	F_1
Joint	67.70	67.70	67.70	68.84	68.84	68.84
PCPR	**87.21**	**81.72**	**84.38**	**85.16**	**80.15**	**82.58**

semantic and syntactic properties of words.

Pipeline Recognition. The ultimate goal of pun recognition is to establish a pipeline to detect and then locate puns. Table 4.7 shows the pipeline performances of PCPR and Joint, which is the only baseline with reported pipeline performance for recognizing the homographic and heterographic puns in the SemEval dataset. Joint achieves suboptimal performance and the authors of Joint attribute the performance drop to error propagation. In contrast, PCPR improves the F_1-scores against Joint by 24.6% and 20.0% on two pun types.

4.2.4.3 Ablation Study and Analysis

Ablation Study. To better understand the effectiveness of each component in PCPR, we conduct an ablation study on the homographic puns of the SemEval dataset. Table 4.8

Table 4.8: Ablation study on different features of PCPR for homographic pun detection on the SemEval dataset.

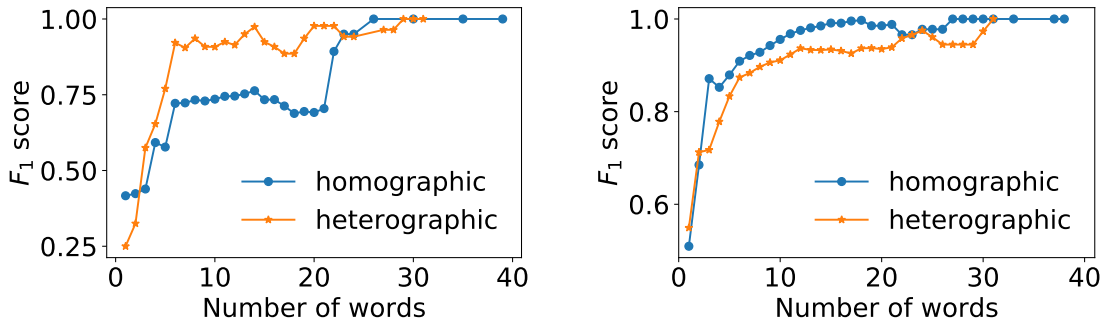
Model	P	R	F_1
PCPR	**90.43**	**87.50**	**88.94**
w/o Pre-trained Phoneme Emb.	89.37	85.65	87.47
w/o Self-attention Encoder	89.17	86.42	87.70
w/o Phonological Attention	89.56	87.35	88.44

Table 4.9: A case study of the model predictions for the pun location task of SemEval 2017.

Sentence	Pun	CPR	PCPR
In the dark? Follow the son.	son	-	son
He stole an invention and then told patent lies.	patent	patent	lies
A thief who stole a calendar got twelve months.	got	-	-

shows the results on taking out different features of PCPR, including pre-trained phoneme embeddings, the self-attentive encoder, and phonological attention. Note that we use the average pooling as an alternative when we remove the phonological attention module. As a result, we can see the drop after removing each of the three features. It shows that all these components are essential for PCPR to recognize puns.

Attentive Weights Interpretation. Figure 4.12 illustrates the self-attention weights α_i^S of three examples from heterographic puns in the SemEval dataset. The word highlighted in the upper sentence (marked in pink) is a pun while we also color each word of the lower sentence in blue according to the magnitude of its attention weights. The deeper colors indicate higher attention weights. In the first example, *busy* has the largest weight because it has the most similar semantic meaning as *harried*. *The barber* also has relatively high weights. We suppose it is related to *hairy* which should be the other word of this double entendre. Similar, *the zoo* is corresponded to *lion* while *phone* and *busy* indicate *line* for the pun. Moreover, *boating*



(a) Pun Detection

(b) Pun Location

Figure 4.11: Pun recognition performance over different text lengths for homographic and heterographic puns on the SemEval dataset.

confirms *sail* while *store* supports *sale*. Interpreting the weights out of our self-attentive encoder explains the significance of each token when the model detects the pun in the context. The phonemes are essential in these cases because they strengthen the relationship among words with distant semantic meanings but similar phonological expressions.

Sensitivity to Text Lengths. Figure 4.11 shows the performance of pun detection and location over different text lengths for homographic and heterographic puns in the SemEval dataset. For both tasks, the performance gets higher when the text lengths are longer because the context information is richer. Especially in the pun detection task, we observe that our model requires longer contexts (more than 20 words) to detect the homographic puns. However, shorter contexts (less than 10 words) are adequate for heterographic pun detection, which indicates the contribution from phonological features. In short, the results verify the importance of contextualized embeddings and pronunciation representations for pun recognition.

Case Study and Error Analysis. Table 6.2 shows the results of a case study with the outputs of `CPR` and `PCPR`. In the first case, the heterographic pun comes from the words *son* and *sun*. `CPR` fails to recognize the pun word with limited context information while the phonological attention in `PCPR` helps to locate it. However, the pronunciation features in

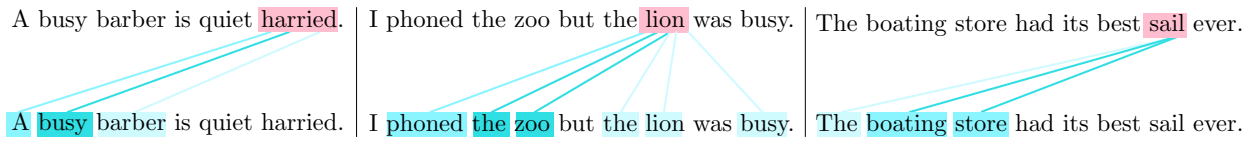


Figure 4.12: Visualization of attention weights of each pun word (marked in pink) in the sentences. A deeper color indicates a higher attention weight.

some cases can mislead the model to make wrong predictions. For example, *patent* in the second sentence is a homographic pun word and has several meanings, which can be found with the contextual features. Besides, the phonemes in *lies* are ubiquitous in many other words like *laws*, thereby confusing the model. In the last case, *got* is a widely used causative with dozens of meanings so that the word is hard to be recognized as a pun word with its contextual and phonological features.

4.2.5 Conclusions

In this work, we propose a novel approach, `PCPR`, for pun detection and location by leveraging a contextualized word encoder and modeling phonemes as word pronunciations. Moreover, we would love to apply the proposed model to other problems, such as general humor recognition, irony discovery, and sarcasm detection, as the future work.

4.3 Acknowledgment

Section 4.1 and Section 4.2 are versions of [277] and [274]. Section 4.1 introduces a work during my summer internship at Yahoo Research. I want to thank my colleagues for contributing to the papers.

CHAPTER 5

Few-shot Attribute Extraction from Semi-structured Web Documents

In this chapter, we introduce our work on few-shot attribution extraction: given a web page, extracting an object along with various attributes of interest (e.g. price, publisher, author, and genre for a book). We propose a novel method, Simplified DOM Trees for Attribute Extraction (`SimpDOM`) to model the problem as a tree node tagging task. The key insight is to learn a contextual representation for each node in the DOM tree where the context explicitly takes into account the tree structure of the neighborhood around the node.

5.1 Motivation

The World Wide Web contains vast amounts of information in a semi-structured format. Translating this information into structured knowledge has long been an important research goal [53, 102, 278]. Extracting structured objects with relevant attributes from semi-structured HTML can power applications including large-scale knowledge base/graph construction [75, 259], e-commerce product search [38, 102], and personalized recommendation [253]. Attribute extraction from web pages is complicated by the semi-structured data format, noisy page contents, complex formatting, and imperfect alignment of the source and visual representations. Whereas unstructured texts can easily be modeled as a sequence [162], web pages demand more sophisticated techniques.

In this work, we focus on the problem of extracting structured objects with a given target schema (like a book, such as in Figure 5.1, to extract attributes of interest like {*title,*

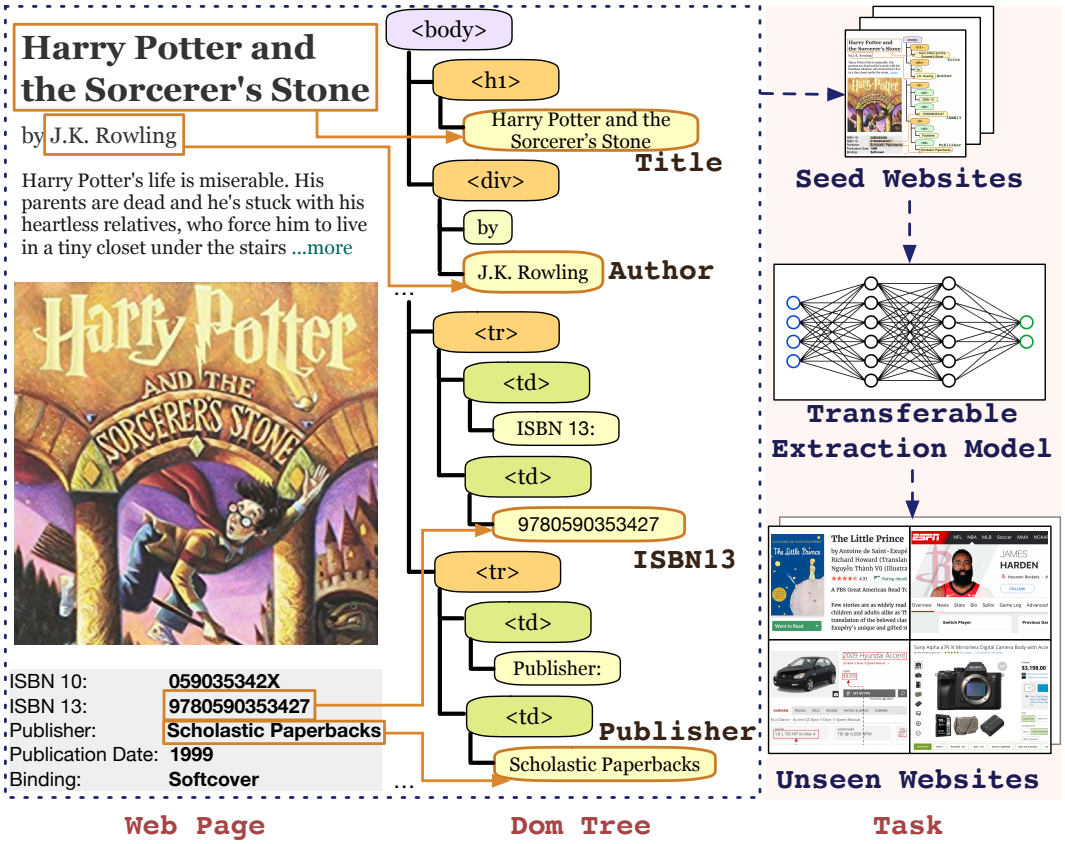


Figure 5.1: Learning a transferable model based on HTML DOM trees to extract attributes from unseen websites of various domains.

author, isbn13, publisher}) using a small amount of labeled data (e.g. a few websites). We consider two challenging scenarios in this work, (i) intra-domain few-shot extraction, where the training data consists of a few labeled seed websites from a given domain and the task is to extract the structured object from unseen websites in the same domain; (ii) cross-domain few-shot extraction, where the training data consists of a few labeled seed websites from a given domain (say A) *and* additional labeled websites from a *different* domain (say B) and the task is to extract structured objects from unseen websites in domain A. The key difference in the cross-domain setting is the availability of additional labeled websites from a *different* domain. At first glance, one may wonder why training data about one domain (say books) might help an extraction model on a completely unrelated domain (say cars). The experimental evidence in this work suggests that this is indeed helpful. We believe this

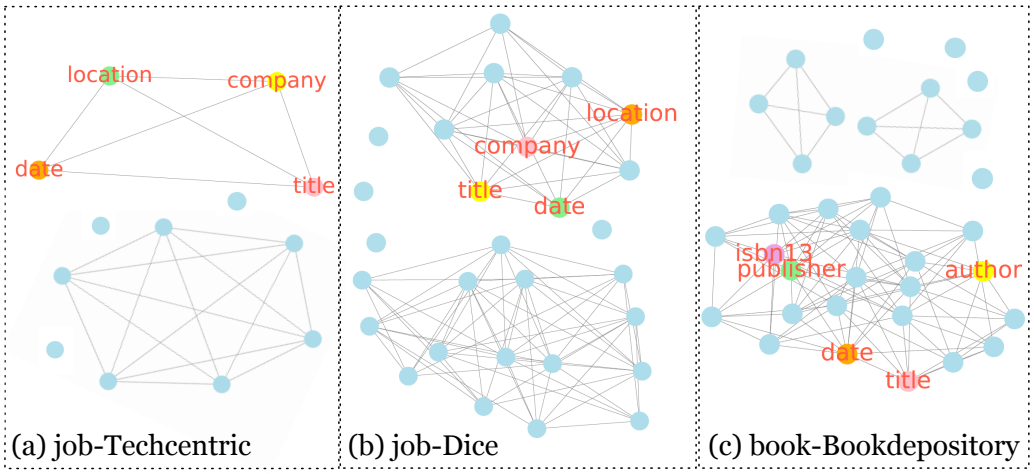


Figure 5.2: Graph visualization of the DOM node neighborhood.

is because the model is able to take advantage of this additional data to learn transferable contextual representations for the nodes. Practically, this is of significant value: the task of building each successive extractor is made easier by leveraging the websites labeled for previous tasks.

Our proposed `SimpDOM` model builds a rich representation for each node in the DOM tree by focusing on contextual features. This representation is then used to train a classifier to decide which attribute type the node belongs to. A key insight in this work is an algorithm to identify "friend" and "partner" nodes that are a particularly valuable context signal. For instance in Figure 5.1, we notice that the closest text node to "J. K. Rowling" contains information "by" which means "J. K. Rowling" is likely to be the *author* of this book. Knowing that the node containing "by" is a critical part of the contextual clue a human might rely on to determine that "J. K. Rowling" is the author of this book.

Summarizing node contexts using simplified neighborhood representations can allow us to leverage website-invariant features like semantically informative expressions and domain-invariant clues such as the co-occurrence of multiple attribute values. Visualizing the neighbor relationship of DOM nodes for three websites from two domains in figure 5.2 shows that in all three cases the nodes that contain attribute values are close to one another. One

explanation for this clustering is to draw readers' attention.

5.2 Contributions

- To the best of our knowledge, this is the first work to learn contextual representations for DOM tree nodes in a web page by leveraging the local tree structure.
- We are the first to present the cross-domain, few-shot attribute extraction task and demonstrate that it results in improved performance compared to the intra-domain setting.
- Extensive experiments show that `SimpDOM` significantly outperforms the SOTA method by 1.44% (F1 score), and the out-of-domain knowledge helps beat the SOTA by a further 1.37%.
- We open-source our implementations* to provide a testbed and facilitate future research in this direction.

There's a rich history of related work in this space which we address in the next section. In particular, we distinguish our work from the literature on wrapper induction [131, 185, 23] which relies on the fact that many websites are created from Document Object Model (DOM) tree [94] templates. These techniques have two drawbacks: 1) They typically require a labeled example for each site in the target domain to induce a wrapper for other pages on that site, 2) The wrappers yield work well for exact copies of the DOM structure but can be brittle in the face of minor structural variation or web page evolution over time [195]. Thus considerable human effort is required to periodically update templates. Our approach, `SimpDOM` eschews wrapper induction and learns attribute extraction models from a limited amount of annotated data that are capable of generalizing to web sites *not present* in the training data.

*The codes can be found at `https://github.com/google-research/google-research/tree/master/simpdom`.

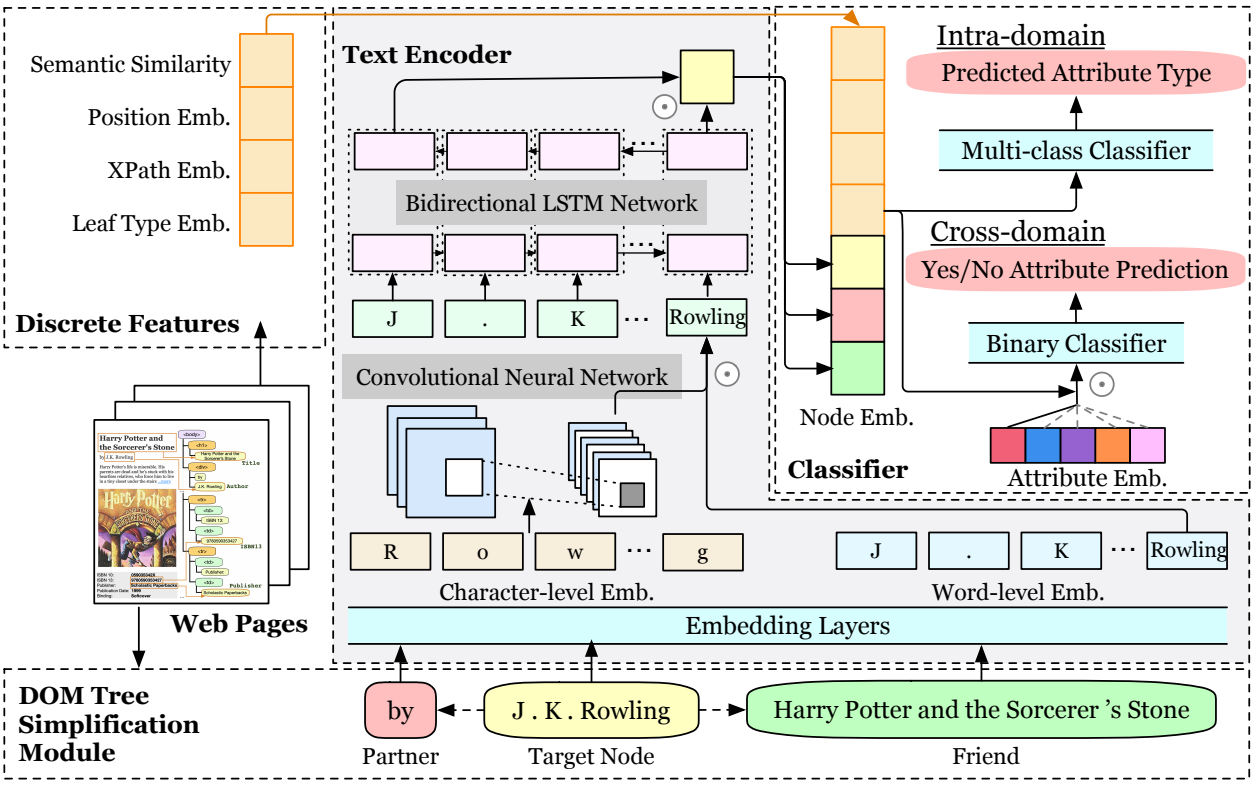


Figure 5.3: The overall architecture of SimpDOM.

5.3 Problem Formulation and Approach

In this section, we formally define the problem and introduce the outline of our proposed method, SimpDOM.

5.3.1 Few-shot attribute extraction from semi-structured websites

We tackle the problem of extracting structured objects from unseen websites. Each *domain* V has a set of websites. Each *website* W is composed of a collection of *detailed pages* which share a similar template. This is a fairly typical assumption since most such web pages are built by instantiating an HTML template with item details that are actually stored in an underlying database.

Attribute Extraction. Given a set of attributes of interest for the target domain, the task at hand is to extract a value (when present) for each attribute from each web page. We

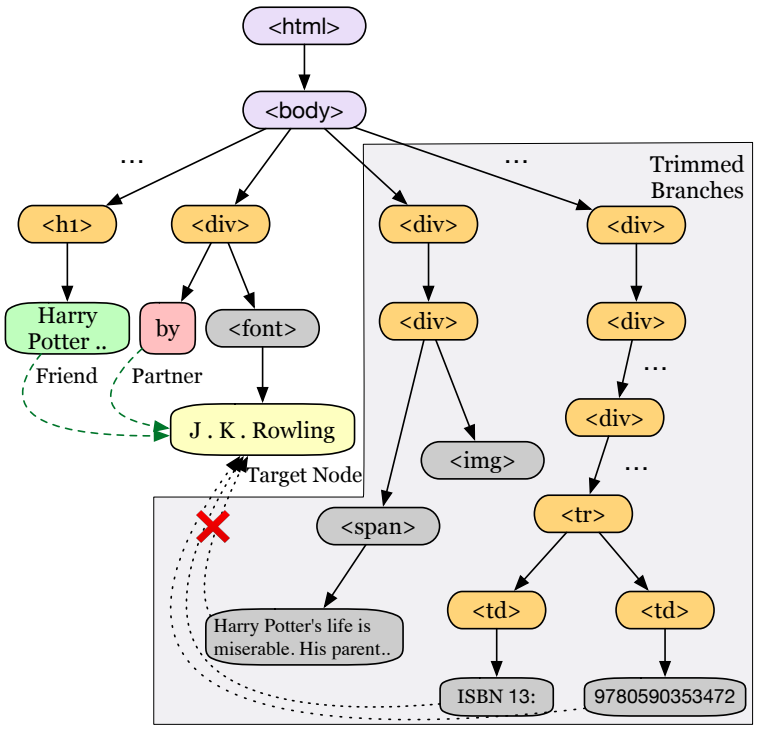


Figure 5.4: We extract the partner (by) and friends for each node by trimming unrelated branches.

make the simplifying assumption that one node can correspond to at most one pre-defined attribute type, consistent with prior work [102]. We formulate the attribute extraction as a node tagging task. Given a detailed page p with a set of variable nodes X, we aim to learn a model to classify each node $x \in X$ into one of the given attributes (e.g. *title, author, isbn13, publisher*) or *none* representing that this node does not contain any attribute values.

Few-shot Intra-domain Extraction.

Given a set of annotated seed websites $\{W_1^a, W_2^a, ..., W_i^a\}$ from domain V, we aim to learn a model $\mathcal{M}$ to extract attributes from a larger set of unseen websites $\{W_1^u, W_2^u, ..., W_j^u\}$ from the same domain.

Few-shot Cross-domain Extraction.

Given a set of annotated seed websites $\{W_1^a, W_2^a, ..., W_i^a\}$ from domain V_1, we aim to learn a model $\mathcal{M}$ to extract attributes from a larger set of unseen websites $\{W_1^u, W_2^u, ..., W_j^u\}$

from the same domain. However, in this setting, we also have access to annotated websites $\{W_1^{a'}, W_2^{a'}, ..., W_k^{a'}\}$ from domain V_2, where $V2 \neq V_1$.

5.3.2 Approach Overview

Figure 5.3 shows the overall framework of the proposed `SimpDOM` model for the few-shot attribute extraction task. First, we extract context features for each node called *friend* and *parent* nodes. Textual features corresponding to each node, its friends, and parents are then fed into a text encoder to generate a dense semantic embedding. We augment this with discrete features built from markup information such as XPath and leaf node types. We then add the relative position of each node as a global feature for the extraction task. The combined node embedding is used for predicting the type of a node. In the intra-domain scenario, we directly apply a multi-class classifier to the node embedding and output the attribute type probability distribution. In the cross-domain scenario, the attribute sets differ from domain to domain. Therefore, we have to alter the inference strategy to binary classification to achieve a matching probability for each attribute type. We select the attribute with the highest probability as the prediction.

Each page has a DOM tree T which contains a variable node set X and a fixed node set Y, where text contents are stored, and also a set of non-text nodes. Fixed nodes remain the same across different detailed pages on the same website (boilerplate like the site's name, navigation elements, etc.) while variable nodes may contain content specific to the object being described on the page. Without loss of generality, we enforce the constraint that the fixed nodes are always mapped to *none*.

5.4 Node Encoder and Classifier

The node encoder consists of three components: a module to extract friend and partner nodes from the DOM tree, the text encoder, and a discrete feature module.

5.4.1 Friend and Partner Nodes

Given a node x in the DOM tree, we define two kinds of nodes *partner* and *friends* that constitute the "friend circle" for the node. The whole DOM tree is a collection of nodes that originate from a unique starting node called the *root*. Recall that the set of nodes A on the path from *root* to node x (not including x) are ancestors of node x. The *friends* of x denotes a set of text nodes X^F such that for each $x^f \in X^F$, the distances from both x^f and x to their lowest common ancestor $a \in A$ is no more than a constant N. We compute the distance by counting the number of edges on the path. The *partner* x^p of x is a special *friend* node for which x and x^p are the only two text nodes in the tree that originate from their lowest common ancestor. Note that each node has at most one *partner* in the DOM tree while it could have zero or multiple *friends*. Usually, *partner* x^p is the closest *friend* to x in the DOM tree. The intuition behind defining partner and friend nodes is simple – while real-world DOM trees can be extremely complicated, most of the *context* for a node is present in DOM nodes that are either friends, and when there's a partner, it contains particularly important context. In Figure 5.5, we plot a common subtree structure (a) and its three possible variants (b,c,d). We simplify and normalize the three variants to (a) in order to extract the friend circle features.

For each variable node $x \in X$, we decode its XPath information to record the K closest ancestors of x. For instance, if the XPath of x is "/body/tr/td/", we consider both "/body/tr/" and "/body/" as the ancestor of x. Conversely, we can easily obtain all the descendants of each ancestor node to construct the candidate set for retrieving the partner and friends. By limiting the size of K, we can narrow down the search area in the tree such that the noisy textual features from distant branches can be efficiently trimmed, as shown in Figure 5.4.

In the extraction process, we keep all the basic HTML element tags like ¡tr¿ and ¡td¿ while remove the formatting and style tags such as ¡strong¿ and ¡font¿[†]. With partner and

[†]We refer the reader to the HTML tag categories described at `https://www.w3schools.com/TAGS/ref_byfunc.asp`.

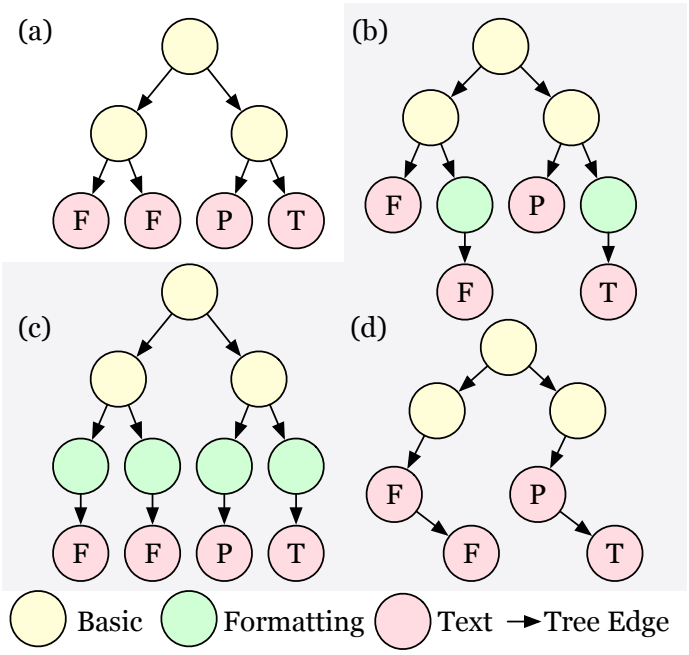


Figure 5.5: Subtree skeletons of web page DOMs including a common structure (a) and its three possible variants (b), (c) and (d).

friends extracted from the DOM tree for each node x, we feed the three sets of textual features separately into the text encoder as described in section 5.4.2 to generate three representations e_x, e_p, and e_f which are all d_w-dimensional vectors. We derive the joint semantic embedding e_s by simply concatenating the three representations as follows:

$$e_s = [e_x; e_p; e_f]\,.$$

Note that the joint embedding is a $3d_w$-dimensional vector.

5.4.2 Text Encoder

Node x contains a sequence of text $S_1 = [w_1, w_2, ..., w_{L1}]$, where $w_i \in \mathcal{W}$ and $L1$ denotes the word sequence length. We can easily split each word into a sequence of characters $S_2 = [c_1, c_2, ..., c_{L2}]$, where $c_i \in \mathcal{C}$ and $L2$ is the character sequence length. $\mathcal{W}$ and $\mathcal{C}$ are vocabularies of words and characters. We employ a hierarchical LSTM-CNN text encoder to encode the character-level and word-level features.

We notice that the attribute values usually contain useful morphological patterns in the character-level semantics [148]. For example, (*aa'bb* ft) and (*aa-bb* ft) are two common patterns of *height* attribute in the *nbaplayer* domain. Their character-level representation can be very important. Therefore, we leverage a Convolutional Neural Network to encode the character-level embeddings (dimension d_c) of each word w, resulting in h_w^c. We simply concatenate h_w^c with its word-level representation g_w retrieved from external pretrained word embeddings: $h_w = [g_w; h_w^c]$.

The LSTM [105] has been widely used as the unit of Recurrent Neural Network for learning the latent representation of sequence data [156]. Therefore, we feed the latent word representations $[h_{w_1}, h_{w_2}, ..., h_{w_{L1}}]$ into a bi-directional LSTM network, resulting in $e_x = \left[h_w^{forward}; h_w^{backward}\right]$.

Similarly, we can achieve the semantic representations for the node's partner and friends, e_p and e_f.

5.4.3 Discrete Feature Module

Xpath embeddings. Markup features such as XPath can be very useful for node tagging. An XPath of a DOM node "/html/body/tr/td/" can be seen as a sequence of HTML tags [¡html¿, ¡body¿, ¡tr¿, ¡td¿]. We learn a separate bi-directional LSTM to get the dense representation e_{xpath} of dimension d_{xpath} for each XPath sequence such that it can make use of all the meaningful tags in the sequence.

Leaf node type embeddings. The tag type of the DOM leaf node such as "¡h1¿" can also be meaningful. "¡h1¿" means the node is likely to be the title of the page, highly correlating with the *name* of a *nbaplayer* or the *title* of a *book*. We collect the vocabulary set of the HTML tags and randomly initialize an embedding e_{leaf} of dimension d_{leaf} for each of them.

Position embeddings. We also leverage the relative position of each node x as a discrete feature. This global information can benefit the task. For example in the *auto* domain, the *model* usually lies on the top of the page. We apply depth-first-search to traverse the tree

and get the occurrence position pos_x of each node. Then we compute its relative position via $\lceil \frac{pos_x}{\max_x\{pos_x\}} \rceil$. Similarly, a random embedding e_{pos} of dimension d_{pos} is initialized for each position.

Semantic similarity. We notice that for each node x the text in the partner node x^p can help determine x's attribute type and modeling the semantic relation between the text in x^p and the attribute types allows us to best leverage this data. Specifically, we compute the *cosine similarity*[‡] between the partner embedding e_p and each attribute embedding e_{a_i} to model their semantic relations, which results in a semantic similarity vector e_{cos} of dimension M, where M denotes the number of pre-defined attribute types.

Upon achieving these discrete features, we concatenate them into a vector $e_d = [e_{xpath}; e_{leaf}; e_{pos}; e_{cos}]$ of dimension $d_{xpath}+d_{leaf}+ d_{pos}+M$.

5.4.4 Inference and Optimization

We design different inference strategies for the two scenarios. Under the intra-domain scenario, the node embedding is connected to a multi-layer perceptron (MLP) for multi-class classification, as illustrated below:

$$e_n = [e_s; e_d]$$

$$\mathbf{h} = \text{MLP}(e_n), \mathbf{h} \in \mathbb{R}^{M+1}.$$

where $M + 1$ denotes the number of pre-defined attribute types plus a *none* type.

Under the cross-domain scenario, we notice that the MLP layer for multi-class classification can no longer be used for different domains which have different sizes of attribute sets. Therefore, we alter the inference strategy to binary classification. Specifically for each attribute type, we concatenate the node embedding e_n to a randomly initialized attribute embedding e_{a_i} of dimension d_a. We then feed it to a separate MLP and compute a score $\mathbf{h}_i$ to denote the probability of this attribute type:

$$e_{b_i} = [e_n; e_{a_i}], 1 \leq i \leq M + 1$$

[‡]We compute the scores via *cosine_similarity* $(e_p, e_{a_i}) = \frac{e_p \cdot e_{a_i}}{|e_p||e_{a_i}|}$.

$$\mathbf{h}_i = \text{MLP}(e_{b_i}), \mathbf{h}_i \in \mathbb{R}$$

Under both scenarios, we lastly apply the *softmax* function to normalize $\mathbf{h}$ and select the largest as the prediction $\hat{\mathbf{y}}$:

$$\mathbf{p}_i = \frac{\mathrm{e}^{\mathbf{h}_i}}{\sum_{j=1}^{M+1} \mathrm{e}^{\mathbf{h}_j}}; \hat{\mathbf{y}} = \underset{i}{\text{argmax}}\, \mathbf{p}_i.$$

The loss function optimizes the cross-entropy between the true labels $\mathbf{y}$ and the normalized probabilistic scores $\mathbf{p}$.

$$\text{loss} = -\sum_{n=1}^{|X|} \sum_{m=1}^{M+1} \mathbf{y}_{m,n} \log \mathbf{p}_{m,n}$$

5.5 Experiments

In this section, we first introduce the dataset and evaluation metrics. We also explain the implementation details to guarantee the reproducibility of our method. Then, a collection of baseline models are introduced to compare with our model under the intra-domain few-shot extraction scenario. We also conduct a series of ablation studies to answer the following questions: (i) *What are the contributions from each set of features?* (ii) *Will sequence modeling work well on DOM tree nodes?* (iii) *What is the performance of different word embedding strategies?* Lastly, we evaluate the effectiveness of the out-of-domain knowledge under the cross-domain few-shot extraction scenario.

5.5.1 Dataset

We rely on a public data set, SWDE [102] that consists of more than 124,000 web pages from 80 websites of 8 domains to train and evaluate the proposed model. Detailed statistics are shown in Table 5.1. Each domain consists of 10 websites and contains 3 to 5 attributes of interest. We notice that the *book* and *job* domains have the most variable nodes on average, roughly three times the number of variable nodes in the *auto* and *university* domains.

In the intra-domain few-shot experiments, we follow the methodology in FreeDOM [148]

Table 5.1: SDWE Dataset Statistics.

Domain	#Sites	#Pages	#Var. Nodes	Attributes
auto	10	17,923	130.1	model, price, engine, fuel
book	10	20,000	476.8	title, author, isbn13, pub, date
camera	10	5,258	351.8	model, price, manufacturer
job	10	20,000	374.7	title, company, location, date
movie	10	20,000	284.6	title, director, genre, mpaa
nbaplayer	10	4,405	321.5	name, team, height, weight
restaurant	10	20,000	267.4	name, address, phone, cuisine
university	10	16,705	186.2	name, phone, website, type

to select k seed websites as the training data and use the remaining $10 - k$ websites as the test set. For example, when $k = 2$, we build 10 training sets by picking 10 permutations from all the 2-seed-website combinations such as (*auto*, *book*), (*book*, *camera*), ..., and (*university*, *auto*). The corresponding test set for the first training set is the remaining 8 websites from *camera* to *university*. Note that in this few-shot extraction task, none of the pages in the $10 - k$ websites have been visited in the training phase. This setting is abstracted from the real application scenario where only a small set of labeled data is provided for specific websites and we aim to infer the attributes on a much larger unseen website set.

In the cross-domain few-shot experiments, we leverage one domain as the out-of-domain knowledge to train a model. Then we conduct the same intra-domain extraction experiments by loading the checkpoints from the pretrained model for parameter initialization. We create this experimental setting to enable a broader knowledge transfer across various domains, which can tackle the scenario where the domain of the existing annotation is inconsistent with the unseen websites.

5.5.2 Evaluation Metrics

We evaluate the extraction performance by page-level F1 scores, following the evaluation metrics from SWDE and FreeDOM [148, 102]. Page-level F1 score is the harmonic mean of extraction precision and recall in each page. Specifically, we evaluate the predicted attribute values with the true values for each detailed page. We compute an average F1 score over all the domains (Table 5.2) to compare with the baselines. We also compute the average F1 score for each domain (Figure 5.6) and each attribute (Figure 5.7) for detailed analysis.

5.5.3 Implementation details

For data pre-processing, we use the open-source LXML library[§] to extract DOM tree structures from each page. Then, we follow the simple heuristic used in [148] to filter nodes whose values are constant in all pages of a website. Thus most of the noisy page-invariant textual nodes such as the footer and navigation contents are removed and training speed is significantly accelerated. We use GloVe pretrained representations [199] to initialize our word embeddings. Other representations such as character embeddings and attribute embeddings are all randomly initialized. We also truncate every node's text to a maximum of 15 words. We set both maximum edge number N and maximum ancestor number K as 5 for extracting friend circle features and only keep the closest 10 friends for each DOM tree node by comparing their relative positions on the web page.

We conduct a grid search for all the hyper-parameters. We use 100 for both word embedding size d_w and character embedding size d_c. We select d_{path}, d_{leaf}, d_{pos} as $30, 30, 20$, respectively. For the CNN network, we use 50 filters and a kernel size of 3. For the LSTM network, we set the hidden layer size as 100. The model is implemented in Tensorflow. We train the model using 15 epochs and a batch size of 32. We apply a dropout mechanism following the MLP layer to avoid over-fitting issues. The dropout rate is 0.3. We use Adam as the optimizer with a learning rate of 0.001. It takes less than 30 minutes to finish the

[§]`https://lxml.de/`

complete training and evaluation cycle for each domain with one NVIDIA V100 GPU.

5.5.4 Baseline Models

We compare against several baselines:

Stacked Skews Model (`SSM`). `SSM` [46] utilizes expensive hand-crafted features and tree alignment algorithms to align the unseen web pages with seed web pages. Attribute values are extracted from each page of the unseen websites. Like our model, this method does not require visual rendering features.

Rendering-feature Model (`Render-full`). `Render-full` [102] employs visual features to express the distances between node blocks rendered with the web browser. Visual distances have proven to be a good method to encode the neighboring relationships among nodes [163] but this method requires the time-consuming rendering process and needs extra memory space to save the images, CSS, and JavaScript that can easily be out-of-date. `Render-full` employs a sophisticated heuristic algorithm to compute the visual distances, which gives the best performance [102], compared to other variants `Render-PL` and `Render-IP`.

Relational Neural Model (`FreeDOM-X`). `FreeDOM` leverages a relational neural network to encode features such as the relative distance and text semantics. This method is composed of two stages. The first stage model (`FreeDOM-NL`) learns a dense representation for each DOM tree node via node-level classification. The relational neural network in the second stage (`FreeDOM-Full`) claims to capture the distance and semantic relatedness between pairs of nodes in the DOM trees. This two-stage model does not rely on visual features but is hard to deploy in practice. Additionally, only modeling the relatedness between pairs of nodes neglects the rich structural information in the tree such as the friend circles. We compare with both `FreeDOM-NL` and `FreeDOM-Full` because the single-stage `FreeDOM-NL` is closer to our model and `FreeDOM-Full` achieves the state-of-the-art experimental results.

Table 5.2: Comparing the extraction performance (F1 score) of five baseline models to our method `SimpDOM` using different numbers of seed sites $k = \{1, 2, 3, 4, 5\}$.

Model \ #Seed Sites	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
`SSM`	63.00	64.50	69.20	71.90	74.10
`Render-Full`	**84.30**	86.00	86.80	88.40	88.60
`FreeDOM-NL`	72.52	81.33	86.44	88.55	90.28
`FreeDOM-Full`	82.32	86.36	90.49	91.29	92.56
`SimpDOM`	83.06	**88.96**	**91.63**	**92.84**	**93.75**

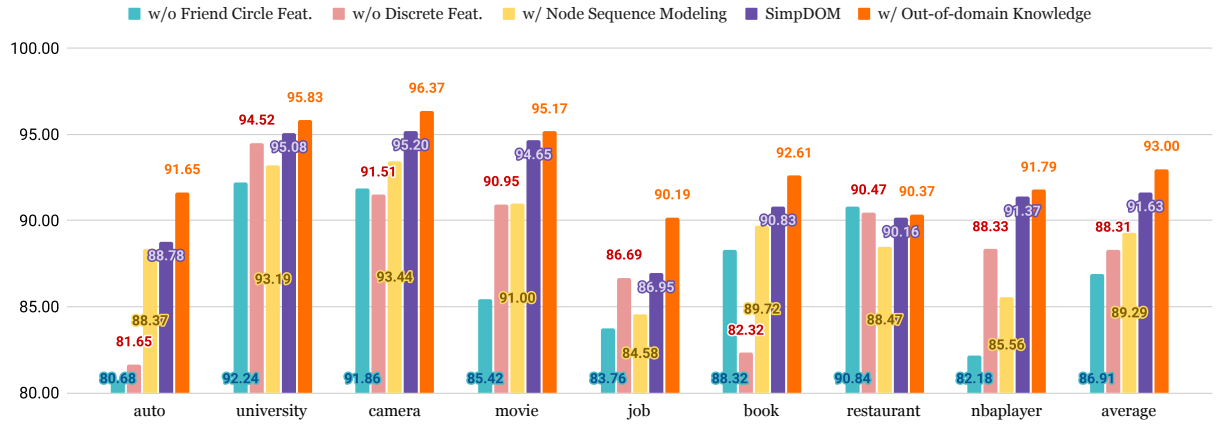


Figure 5.6: Ablation study results that demonstrate the contribution from different features and modules.

5.5.5 Intra-domain Few-shot Extraction Results

Table 5.2 shows the overall comparisons between our model `SimpDOM` and all four baselines using different numbers of seed websites. Our model achieves slightly worse performance when $k = 1$ while it largely outperforms `Render-Full` when $k = \{2, 3, 4, 5\}$. We can conclude that the delicately crafted visual features can capture more patterns in the scenario where extremely small training data exists. However, they are not as transferable as the rich semantic features extracted from our simplified DOM trees as k increases. Our method

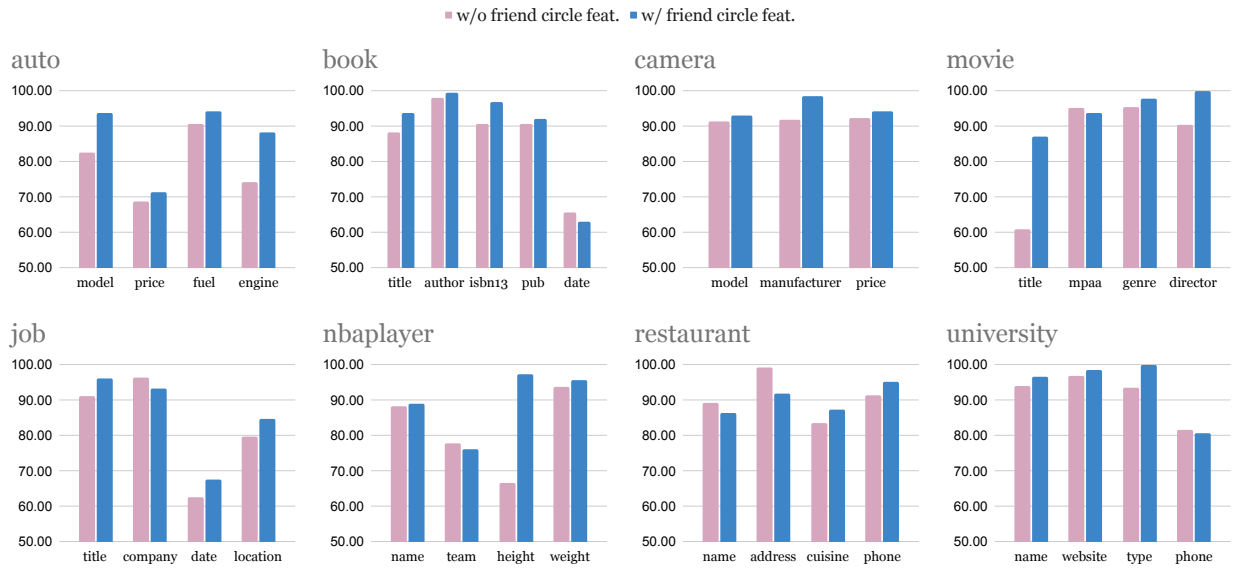


Figure 5.7: Per-attribute F1 performance comparisons between `SimpDOM` w/ and w/o friend circle features.

also consistently outperforms the state-of-the-art method `FreeDOM-Full` (an average improvement of 1.44% over all values of k) and achieves a 3.47%-10.54% improvement over the single-stage approach, `FreeDOM-NL`.

We plot the detailed performance of `SimpDOM` on different domains in figure 5.8. In general, the performance improves as k increases. This is not surprising because more training data yields better coverage of all possible instances. We also observe that the rate of performance growth slows down and sometimes the F1 scores of some domains (e.g. *nbaplayer* and *restaurant*) even fluctuate as more data is added to the training set (i.e. as k increases). We surmise that the reason for this behavior is that the model becomes more robust and less new knowledge can be transferred from annotated websites to unseen websites in these domains.

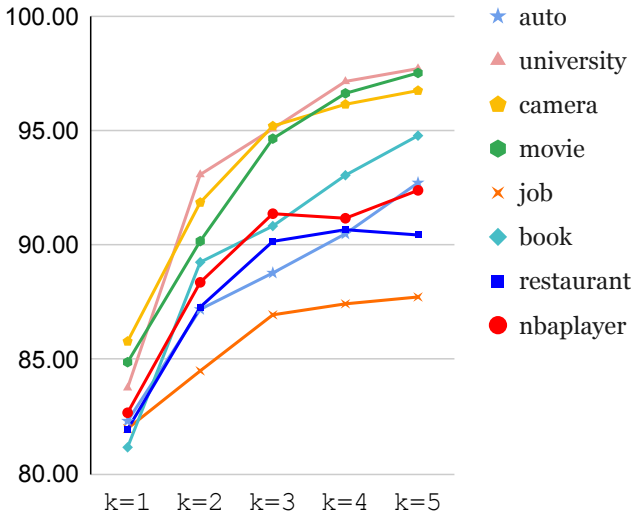


Figure 5.8: Comparing the extraction performance (F1 score) of different numbers of seed sites $k = \{1, 2, 3, 4, 5\}$ per domain.

5.5.6 Ablation Study

In Figure 5.6, we present an ablation study on different features of `SimpDOM`, including discrete features and friend circle features. We find that both sets of features improve the attribute extraction performance dramatically. For instance, the friend circle features increase the F1 score of the *nbaplayer* domain from 82.18% to 91.37% and the discrete features increase the performance on the *book* domain by 8.51%. However, *restaurant* is a special case where the score drops when we employ either of the two feature sets. We believe the node texts in some attribute values such as *name* and *address* are distinguishable enough and adding more features just adds more noise to the classification. This is also corroborated by Figure 5.7, which explains the detailed performance change when adding the friend circle features per attribute. We observe that the improvement on *height* of *nbaplayer* is significant. The nodes containing *height* value always share a similar pattern xx-yy[¶] with some other nodes on the same page. With the friend circle features, we find that *weight* is always a

[¶]For instance, NBA player Kobe Bryant's height (6-6) has the same value as his shooting record (6-6) in one game. It is impossible to distinguish two nodes by the text.

Table 5.3: Comparing different word embedding approaches when $k = 3$.

Embedding Approach	F1	Performance Change
GloVe Embedding Trainable	**91.63**	0
GloVe Embedding Fixed	91.25	-0.38
Randomized Word Embedding	89.66	-1.97
Contextualied Embedding	81.83	-9.80

friend node of *height*, which makes *height* distinguishable from other nodes with similar text patterns. The extraction of some attributes such as *company* in the *job* domain and *address* in the *restaurant* domain was not improved. We believe this is caused by the comparatively diverse positions of these attributes in different websites.

Another interesting ablation study is done with an additional sequence modeling layer[‖] which is commonly applied to sequence labeling tasks such as named entity recognition on plain text [134, 263]. We first obtain a sequence of node embeddings before the MLP classifier where all the nodes are from one web page. Then a new representation can be achieved from the sequence model for each node. The same classifier is used to predict the attribute type with the updated node representation. As shown in Figure 5.6 (marked as "w/ Node Sequence Modeling"), the additional sequence modeling layer fails to optimize the node representations for all the domains especially those with more variable nodes such as *nbaplayer* and *job*. We suppose that the information from all other DOM tree nodes can be selectively attended to the current node with this mechanism, however this introduces more noise than useful knowledge. This further demonstrates the importance of utilizing the structure of the DOM tree to eliminate noise from distant and irrelevant nodes.

We also compare different embedding approaches for encoding textual features. As shown in Table 5.3, we conduct experiments to test the randomized word embedding, fixed GloVe word embedding, and trainable GloVe word embedding. In the trainable setting we can con-

[‖]We utilize the Transformer [248] as the sequence modeling layer. LSTM is another option.

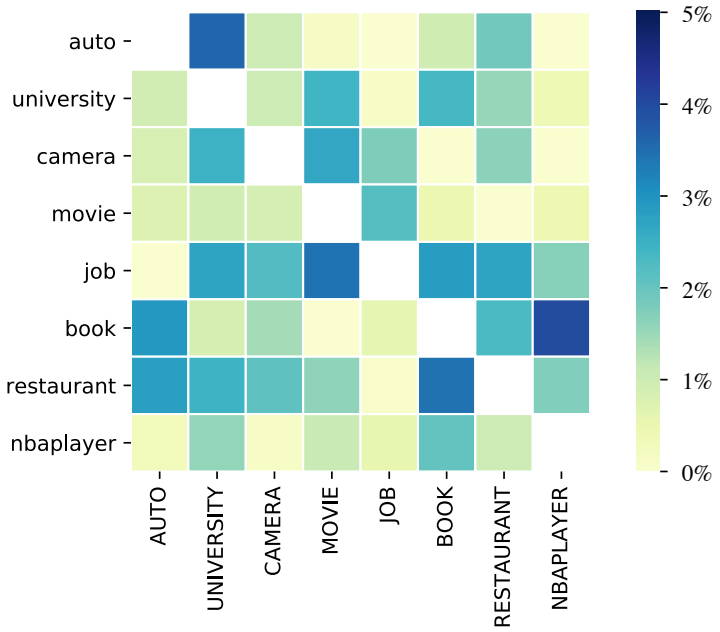


Figure 5.9: Heatmap denoting the performance improvements per F1 score from the out-of-domain knowledge ($k = 3$).

tinue to optimize the parameters in the embedding layer, initialized from GloVe, achieving the best performance. We think that a specific "web-language" model can serve the web information extraction tasks better. Drawing on recent developments in contextualized language models, we also tried using BERT [72][**] to generate the contextualized embeddings but it decreases the performance by 9.8%. This is not surprising given that the context in each node is very limited[††] and the huge size of parameters (110M in BERT-BASE) for fine-tuning can easily cause an over-fitting problem.

5.5.7 Cross-domain Few-shot Extraction Results

We plot a heatmap in Figure 5.9 to denote the performance improvements from the out-of-domain knowledge. Each entry in the heatmap relates to a pair of domains, where the domain

[**]We choose BERT without loss of generality. It can be replaced by its alternatives like ELMo [202] or XLNet [265].

[††]On average, each variable node contains only 2-5 words in different domains.

in the upper case is used as the out-of-domain knowledge while the domain in the lower case is used to train and test the model. We do not plot the scores in the diagonal because domains cannot serve as their own out-of-domain resource. One interesting observation is that this heatmap is roughly symmetric with respect to the diagonal, which demonstrates a mutual relationship between pairs of domains. For instance, *job* and *movie*, *book* and *nbaplayer*, *restaurant* and *book* can all significantly improve the extraction performance for each other, while *auto* and *job*, *camera* and *nbaplayer* seem to be irrelevant to each other. We show the performance of each domain achieved by using the most helpful out-of-domain knowledge in Figure 5.6. We achieve the highest average F1 score of 93% over all the domains ($k = 3$), which improves the performance of our intra-domain experiment by a further 1.37% (absolute value). This evidence proves our assumption that a better contextual node representation can be learned from additional knowledge, which is extremely helpful in the scenario where only a few labeled data are provided for specific domains.

5.6 Conclusion

In this work, we present a simple but effective method, `SimpDOM`, for the attribute extraction task. `SimpDOM` uses the tree structure of the neighborhood around a node to learn a contextual representation for each node in the DOM tree. This method does not require the expensive generation of visual features and is more robust than wrapper induction.

5.7 Acknowledgment

This chapter introduces a work during my summer internship at Google AI [278]. I want to thank my colleagues at Google AI for their valuable discussions and work on revising the paper.

CHAPTER 6

Social Media Information Extraction for Pandemic Surveillance

COVID-19 has caused lasting damage to almost every domain in public health, society, and economy. In this chapter, we present a work that (i) takes advantage of the social media data to construct heterogeneous knowledge graphs based on the events and relationships; (ii) conducts time series prediction to provide both short-term and long-term forecasts of the confirmed cases and fatality at the state-level in the United States and simultaneously discovers risk factors for COVID-19 interventions.

6.1 Motivation

Over 200 countries and territories have been deeply impacted by the outbreak of the coronavirus disease 2019 (COVID-19). As of 2021 May, a total of 164 million cases and 3.4 million deaths were reported all over the world*. It is critical to forecast the short-term and long-term trends of the epidemic, to help governments and health organizations determine the prevention strategies and help researchers understand the transmission characteristics of the virus.

Modeling the COVID-19 pandemic is challenging. Previous studies present three types of disease transmission approaches to explain and model the pandemic, which are exponential growth models [160], self-exiting branching process [128], and compartment models (e.g., Susceptible-infected-resistant (SIR) [121], Susceptible-Exposed-Infected-Removed

*https://covid19.who.int/

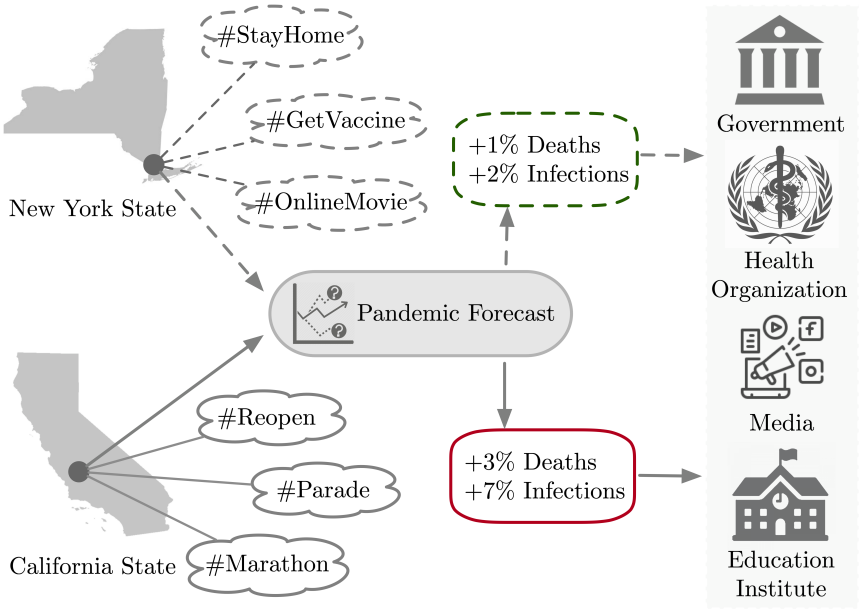


Figure 6.1: Social media users can serve as a "social sensor" for monitoring the pandemic trend.

(SEIR) [19] and Herd Immunity [81]). However, exponential growth models can only address the initial outbreak while self-exiting-branching process and compartment models favor the development and peak stages [34]. Besides, the pandemic trend varies dramatically across different locations and times in response to real-time breaking events. To tackle these challenges, some data-driven approaches [68, 15] that ensembles statistical and machine learning models emerge for monitoring the confirmed cases, fatality, and hospitalizations. [193, 89] leverage graph neural networks to incorporate the population mobility data, i.e., how many people traveled from one place to another, to encode the underlying diffusion patterns into the learning process. However, these models take into consideration only a small number of homogeneous features. They are incapable of capturing potential risk factors and identifying various intervention mechanisms of this new pandemic as well.

As the quarantine life takes over the world and people turn to online platforms for communication and information, social media become more influential than ever [101, 186]. The vast collections of social media streams can capture local activities (e.g., public gatherings and vaccination progress) that may affect the transmission of the virus in real-time. Over

170 million tweets are posted every day in the United States related to observations, behaviors, and thoughts of individual users [65]. The social media users can be naturally treated as robust "social sensors" [119] to unveil the surveillance evidence over time and space. For example, in Figure 6.1, the severe discussions related to the coming social events such as "Marathon" and "Parade" may indicate a potential risk of virus spread while some hot hashtags like "#StayHome" or "#GetVaccine" may represent the safety awareness of individuals in the prevailing areas. Over the past decades, researchers have successfully applied social media data to monitor the earthquakes [213] or air quality [119]. Inspired by these works, we aim to incorporate social media content to forecast the pandemic.

To this end, we want to answer the following interesting research questions:

- *Can social media contents further enhance the short-term and long-term COVID-19 forecasts?*
- *How to identify potential risk factors from the social media data as these factors may vary over time and space?*

Motivated by them, we collaborate with Twitter and use their COVID-19 stream API service to crawl large-scale tweets related to COVID-19 based on Twitter's internal COVID-19 annotations. We propose a novel framework, Social Media enhAnced pandemic suRveillance Technique (`SMART`), which is composed of two modules, information extraction module and time series prediction module [273]. Specifically, in the information extraction process, we recognize named entities and identify relationships among them from the large-scale tweet corpora. Based on the entities and relationships, we build a spatial-temporal heterogeneous knowledge graph. We then propose a Dynamic Graph Neural Network (DGNN) with a Bidirectional Recurrent Neural Network (Bi-RNN) to forecast pandemic trends and suggest risk factors for each location.

6.2 Contributions

- To the best of our knowledge, we are the first to simultaneously detect social events for pandemic surveillance and suggest the risk factors.
- We propose a novel framework, `SMART`, for domain-specific information extraction from social media data and time series prediction on dynamic spatial-temporal graphs. Extensive experiments show the effectiveness of our approach. We achieve 7.3% and 7.4% improvements from the state-of-the-art methods for confirmed case/fatality predictions.

6.3 Pandemic Forecast

Epidemic Prediction Models. There are three types of epidemic prediction models in literature, including exponential growth models [160], self-exiting branching process [128], and compartment models [211, 19, 103, 30, 129, 219, 121, 26, 183, 39, 104, 81]. The dynamics of infectious diseases are expressed by the compartment models for predicting the epidemic trends using ordinary differential equations [211]. SIR [121], as the most prevailing compartment model, segments the population into three parts: Susceptible, Infectious, and Recovered and express the population flow among them with evolving equations. Later, many cumulative studies based on SIR emerge, including SEIR [19], SEIS [252], MSEIR [104], SuEIR [280], and MSIR [183]. In specific, SEIR includes the Exposed compartment and SEIS, MSIR, MSEIR, SuEIR extend SEIR by taking into account either Immunity or untested/unreported compartments. However, as concluded in [34], the exponential growth models can only address the initial outbreak while self-exiting-branching process and compartment models favor the development and peak stages. None of these models are expected to be precise and robust in the long-term pandemic prediction.

Statistical and Machine Learning Models. Researchers also apply statistical time series prediction models such as ARIMA and PROPHET for COVID-19 pandemic predic-

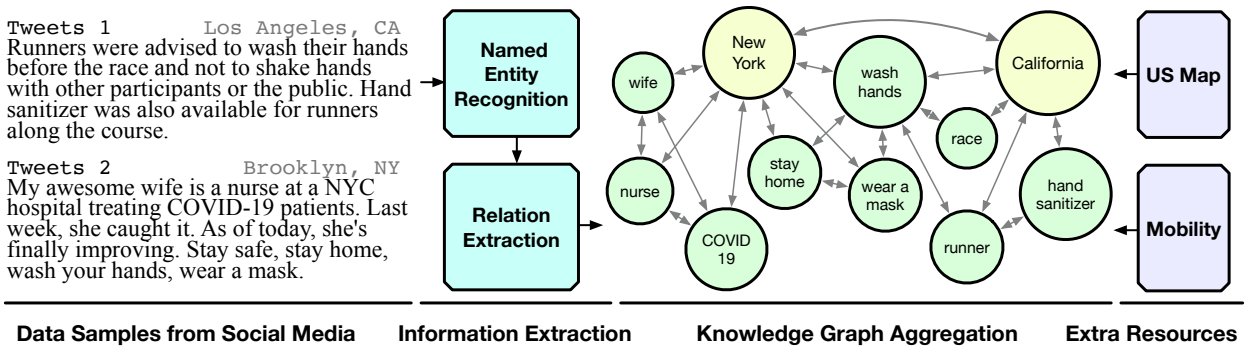


Figure 6.2: Overview of the information extraction pipeline on social media data.

tion [130, 167]. ARIMA [43] is an Autoregressive Integrated Moving Average model, relying on a basic assumption that the future time series are linear aggregations of the past ones. PROPHET [240] is an additive model that emphasizes seasonal effects so that the model works better on time series with periodical patterns. [212, 210, 62] aggregate neural networks to an Autoregressive model, to enhance inter-region connections or temporal dependencies. However, these models conduct pandemic forecasts highly depending on the trend and seasonality instincts behind the historical COVID-19 statistics, incapable of incorporating heterogeneous features. [193, 89, 120] apply graph neural networks to take advantage of the mobility data across different regions but still cannot detect hidden risk factors for the pandemic modeling. Therefore, in this study, we propose a social media enhanced pandemic forecast framework to incorporate the extracted entities and relationships for confirmed case/fatality prediction with strong interpretability.

6.4 Social Media Enhanced Pandemic Surveillance

Given a large-scale collection of social media data together with the historical confirmed cases/fatalities and the population mobility statistics, we aim to forecast the pandemic trend and recognize potential risk factors. The framework of our `SMART` model consists of two components: (i) information extraction module including a named entity recognizer and a relation identifier (as shown in Figure 6.2); (ii) spatial-temporal dynamic graph encoder for

pandemic trend forecast (as shown in Figure 6.3).

6.4.1 Constructing Dynamic Knowledge Graphs from Social Media Data

We propose a bottom-up solution to extract entities and relations to construct the heterogeneous dynamic knowledge graphs.

Named Entity Recognition (NER). NER is a natural language processing (NLP) task which labels the tokens in a sequence with tags from a desired tag pool. In this work, we adopt the NER setting to extract entities of interest from the social media data by labeling the words or phrases in the tweet sentences. As examples in Figure 6.2, we want to recognize *nurse* as OCCUPATION, *stay home* as INDIVIDUAL_BEHAVIOR, *race* as EVENT, and so on.

Traditional NER approaches [47, 84, 196] heavily rely on expensive and time-consuming feature engineering including parsing the Part-of-Speech tags of each word and the syntactic dependency structures of the sentences. Some recent studies [67, 111, 155] incorporate neural networks with statistical models, such as conditional random fields [133], to improve the model performance. With deep language models like BERT [72] and RoBERTa [159], the NER performance can be further improved. Without the loss of generality, we leverage BERT model to provide contextualized embeddings and learn a supervised named entity recognizer. To overcome the problem with the nonexistence of annotated tweets as training data, we collect the benchmark corpora and their annotations for multiple NER tasks, including I2B2-2010 [69], CORD-NER [255] and MACCROBAT-2018 [50]. Based on those external datasets, we jointly learn a recognition model to extract entities on the COVID-19 related tweets data. On average, we extracted 10,040 unique entities of 45 entity types 270k tweets corpus every day.

Relation Extraction. Given the extracted entities, the next step is to identify the relationships among the entities. Note that we only extract intra-tweet relations. In other words, we do not predict the relation between entities in different tweets. Existing solu-

tions [250, 143, 194, 270] formulate the problem as a sequence classification task, given a textual sequence and the positions of two named entities. Specifically, a multi-class classification is conducted to assign a label from a desired set for the relationship. However, this formulation highly depends on the quality and quantity of the annotated datasets to achieve satisfactory performance. It is obviously incapable of identifying emerging new relation types.

To overcome the above challenge, we convert the multi-class prediction task to a binary classification problem of only identifying the existence of a potential relationship between any entity pair in each tweet instance. We aggregate datasets from multiple tasks including Wiki80 [100], I2B2-2012 [235], and MAACROBAT-2018 [50] to create the positive training data (labeled as 'True'). In order to achieve balanced training, validation and test datasets, we apply negative sampling to create the same number of instances with the label 'False'. Note that we assume no relation between any two entities exists if the entities were not annotated. Similarly, we acquire the sequence representations from the fine-tuned BERT language model and feed them into a binary classification layer for label prediction. During the inference stage, we enumerate all possible pairs of entities in each tweet and assign binary labels for them.

Domain-specific Pre-trained Language Model. To tackle domain-specific tasks, such as Clinical information extraction [271] and Bioinformatics knowledge acquisition [135], recent studies pre-train new language models with large-scale corpora collected from those domains [140, 13] to learn customized token and sequence representations. Motivated by these approaches, we leverage all COVID-19 relevant text corpora together with the social media data to pre-train a `CoronaBERT` language model with 12 layers of Transformers and over 110 million parameters, in order to equip our models with powerful input embeddings. We ceaselessly fine-tune the parameters in `CoronaBERT` as more COVID-19 stream corpora become available and release the models on a quarterly basis.

Heterogeneous Knowledge Graph Aggregation. After named entity recognition and relation extraction, we apply the DBSCAN clustering model [79] to merge semantically

similar entities for reducing the noises in the entity sets. This step is essential for cleaning the entities extracted from tweets. For example, "Marathon" and "Marathon:)" are supposed to be merged and "COVID-19" is indeed the same as "COVID2019". In specific, we cluster the entities based on the similarity among their entity embeddings acquired by `CoronaBERT`. We assign the node in each cluster with the highest occurrence in tweets as the cluster head. Other nodes in the same cluster will be replaced by the cluster head.

Based on the clustering results, we aggregate the denoised knowledge pieces into a heterogeneous knowledge graph. Two types of nodes exist in the graph, including location nodes and entity nodes. Here we set the location nodes as the 50 states in the United States while our methods can be easily extended to the county-level locations or applied to other countries and regions. Next, we build three types of edges as follows:

- **Entity-Entity** edges: we add an edge between any two entities if there is a 'True' relationship identified.
- **Location-Entity** edges: we look up the geo-location attribute of the tweet where each entity is extracted and add an edge between the entity node and the geo-location.
- **Location-Location** edges: we add an edge between a location pair under two circumstances, (i) two locations are adjacent to each other on the US map; (ii) we detected population transition from one location to another according to the mobility data. More details of the mobility data are provided in Section 6.5.1.

We build one knowledge graph for each day. Later, knowledge graphs within a certain time period will be further aggregated for time series prediction, as described in Section 6.4.2.

6.4.2 Time Series Prediction with Dynamic Graph Attention Network

Dynamic graph aggregation. We represent the heterogeneous knowledge graph of the t-th day as $G^{(t)} = (V^{(t)}, E^{(t)})$ where $n = |V^{(t)}|$ denotes the number of nodes, $V^{(t)} = V_L^{(t)} \cup V_E^{(t)}$, where $V_L^{(t)}$ is the location node set and $V_E^{(t)}$ is the entity node set. Given a sequence of

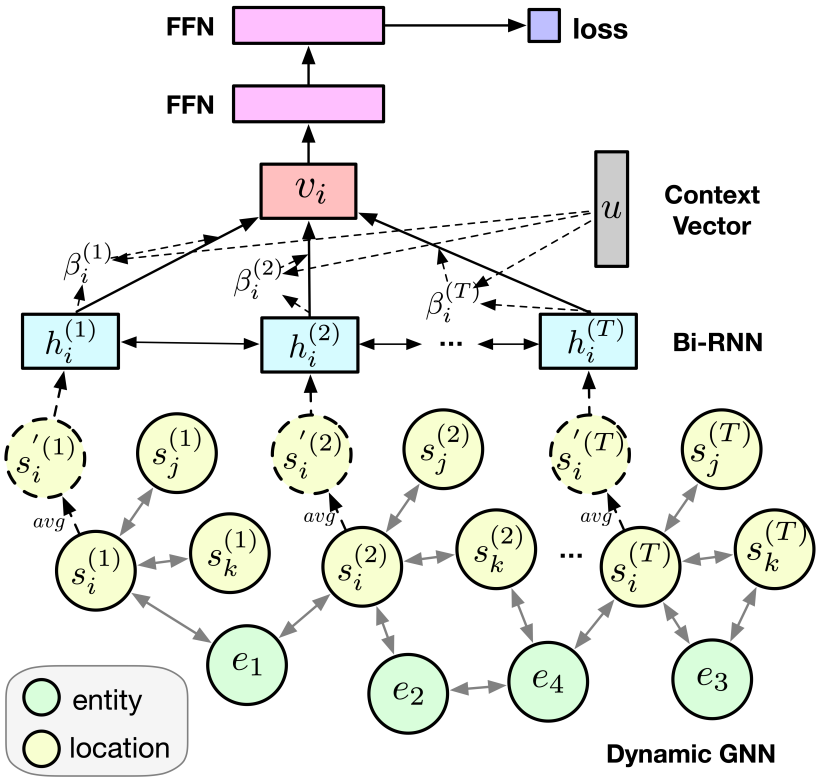


Figure 6.3: Overview of the time series prediction module.

knowledge graphs $\{G^{(1)}, G^{(2)}, ..., G^{(T)}\}$ of length T, we aim to predict the COVID-19 courses including confirmed cases and fatality cases on the day $T + l$. We regard it as a short-term prediction when $l < 14$ or a long-term prediction when $l \geq 14$. We formulate the time series prediction problem as a regression task.

We continue to aggregate the length-T graph sequence into one spatial-temporal graph $G^S = (V^S, E^S)$ as shown in Figure 6.3. First, we keep all the location nodes from different times in the period, i.e. $V_L^S = V_L^{(1)} \cup V_L^{(2)} \cup ... \cup V_L^{(T)}$. On the other hand, we merge entity nodes of different times, i.e. $V_E^S = V_E^{(1)} \cup_{\backslash t} V_E^{(2)} \cup_{\backslash t} ... \cup_{\backslash t} V_E^{(T)}$, where $\cup_{\backslash t}$ denotes a time-unaware set union. For example, the entity node e_1 is recognized in the location s_i on both time 1 and time 2, but we only keep one e_1 in V_E^S by connecting e_1 to $s_i^{(1)}$ and $s_i^{(2)}$. In this way, we introduce the inter-time propagation edges to expand the node neighbors along the temporal dimension so that we can easily model the structural temporal dependencies among the nodes.

Node Features. Our pre-trained `CoronaBERT` is applied to generate the initial semantic features x_i^{se} of dimension d_e for node i of any type. We also incorporate the historical COVID-19 statistics x_{st} of d_t days ahead of the current time as an extra feature set for location nodes, resulting in a node feature embedding $x_i = x_i^{se}||x_i^{st}$ of dimension $d_e + d_t$, where $||$ denotes a vector concatenation. Note that we keep the embedding dimensions of location nodes and entity nodes the same, in order to smooth the graph propagation computation. Hence, we append a zero vector of length d_t at the end of each entity vector.

Dynamic Graph Neural Network. We propose a multi-head DGNN architecture to perform the graph propagation. We first conduct a linear transformation on the input node embeddings:

$$z_{i,p} = W_p x_i,$$

where W_p is a learnable weight matrix; $p = \{1, ..., H\}$; H is the number of heads. Then, we compute a pair-wise un-normalized attention score of an edge between any two neighbors (two nodes i and j) in the graph:

$$e_{ij,p} = \text{LeakyReLU}(w_p^T(z_{i,p}||z_{j,p})),$$

where w_p is a learnable weight vector and LeakyReLU [261] is applied as a non-linear transformation. We use the attention score to indicate the importance of a neighbor node in the message passing process, especially when we interpret the risk entities to each location. A Softmax is applied to normalize the attention weights to a probability distribution so that we can easily interpret and compare the importance of all incoming edges,

$$\alpha_{ij,p} = \frac{\exp(e_{ij,p})}{\sum_{k \in \mathcal{N}_S(i) \cup \mathcal{N}_E(i)} \exp(e_{ik,p})},$$

where $\mathcal{N}_S(\cdot)$ and $\mathcal{N}_E(\cdot)$ denote the sets of neighboring location nodes and entity nodes. We finally aggregate the embeddings of neighboring nodes. The aggregation is scaled by the

normalized attention scores. We compute the averaged embeddings over different heads,

$$x_i' = \sigma\left(\frac{1}{H}\sum_{p=1}^{H}\sum_{j\in\mathcal{N}_S(i)\cup\mathcal{N}_E(i)}\alpha_{ij,p}z_{j,p}\right).$$

Attentive Bi-Recurrent Neural Network. We intend to further encode the temporal dependencies between the **location nodes** over times and learn a hidden state of the overall graph using an Attentive Bi-RNN module. We collect embeddings from the same location of different times $[x_i^{'(1)}, x_i^{'(2)}, x_i^{'(T)}]$ and recursively feed them into a Bi-RNN with Gated Recurrent Units (GRU) [63]. We choose GRU instead of Long Short Term Memory (LSTM) [105] unit due to its computational efficiency and capability of tackling shorter sequences like tweets [64]. The hidden representation of each location in time t is learned from two directions,

$$\overleftarrow{h}_i^{(t)} = \text{GRU}(\overleftarrow{h}_i^{(t+1)}, x_i^{'(t)}),\ \overrightarrow{h}_i^{(t)} = \text{GRU}(\overrightarrow{h}_i^{(t-1)}, x_i^{'(t)}),$$

$$h_i^{(t)} = \overleftarrow{h}_i^{(t)} \oplus \overrightarrow{h}_i^{(t)},$$

We then aggregate the hidden states with another attention mechanism,

$$v_i = \sum_{t=1}^{T}\beta_i^{(t)}h_i^{(t)}, \beta_i^{(t)} = \frac{\exp(u^T h_i^{(t)})}{\sum_k \exp(u^T h_i^{(k)})},$$

where u denotes a context vector and $\beta_i^{(t)}$ are attention scores reflecting the contribution of the hidden representation in time t.

Learning Objective. We feed the context-aware node representation v_i into two layers of Feed Forward Networks (FFN) and lastly generate a scalar $\hat{y}_i^{(\bar{t}+l)}$ representing the predicted COVID-19 confirmed case or fatality number in l days ahead of time $\bar{t}$. We compute the loss with the following Mean-Squared-Error (MSE) objective [215],

$$\mathcal{L} = \frac{1}{mn}\sum_{\bar{t}=1}^{n}\sum_{i=1}^{m}(y_i^{(\bar{t}+l)} - \hat{y}_i^{(\bar{t}+l)})^2,$$

where m is the number of location nodes and n is the number of days that requires a prediction.

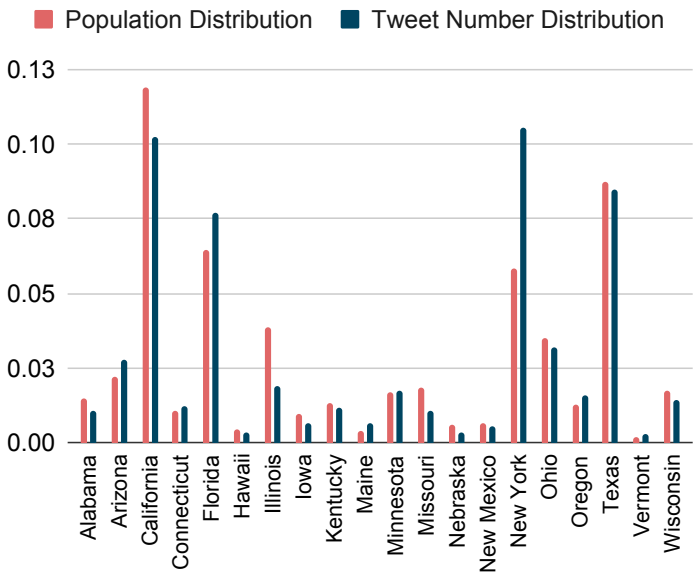


Figure 6.4: Comparison between the spatial distributions of US population and the number of tweets over 20 states.

6.5 Experiments

6.5.1 Datasets

Twitter Stream Data. We collaborate with Twitter and build a real-time tweet crawler to steadily acquire relevant social media tweets using their COVID-19 streaming API[†] [169]. In detail, the streaming API returns real-time tweets related to COVID-19 based on Twitter's internal COVID-19 tweet annotation system. The data collected for this study start on May 15, 2020 and end on April 8, 2021. Figure 6.4 compares the distributions of the US population and the number of tweets over 20 states. We notice except that New York people are more passionate about posting COVID-19 related tweets while California people do the opposite, other states have relatively similar spatial distributions over the population and number of tweets.

Mobility Data. As [193] conclude a strong relationship between the population transition

[†]`https://developer.twitter.com/en/docs/labs/covid19-stream/api-reference/`.

and regional COVID-19 trends, we also collect the mobility data that describe the population transition in the United States from SafeGraph[‡] for pandemic forecast.

COVID-19 Statistics. We leverage the US state-level COVID-19 statistics gathered by the New York Times[§] based on reports from state and local health agencies for building the ground truths of pandemic forecasts. We use the statistics of confirmed new cases and fatalities from May 5, 2020 to April 8, 2021. Note that the start date is the earliest date when we have Twitter Stream data available. The average new confirmed cases and fatalities over 50 states are 1788.3 and 28.7 per day while the standard deviations are 3374.8 and 63.5. California has the highest average number of new confirmed cases (10988.5) and fatalities (173.4). Vermont has the lowest numbers (60.0 new confirmed cases and 0.5 fatalities).

6.5.2 Experimental Setup and Evaluation Metrics

Following the experimental setup in [193], we train a model with the data from time 1 to time $\bar{t}$ and use it to predict the numbers on time $\bar{t}+l$[¶]. We evaluate the model on short-term ($l = \{1, 7\}$) and long-term ($l = \{14, 28\}$) predictions. Note that we learn a different model to predict the cases for time $\bar{t} + l_i$ and $\bar{t} + l_j$, where $i \neq j$. In the training process, we select 5 data points from the training set as the validation set to identify the best model.

We evaluate the performance of our method by computing the Mean-Absolute-Error (MAE) [214],

$$\text{error}_{\text{MAE}} = \frac{1}{mn} \sum_{\bar{t}=1}^{n} \sum_{i=1}^{m} |y_i^{(\bar{t}+l)} - \hat{y}_i^{(\bar{t}+l)}|,$$

where m and n denote the numbers of test instances and location nodes. We also follow [125, 8] to compute the symmetric Mean-Absolute-Percentage-Error (sMAPE) to show the

[‡]`https://www.safegraph.com/`.

[§]`https://github.com/nytimes/covid-19-data`.

[¶]For example, if we predict the next-day (i.e., $l = 1$) case number for date 12-31-2020, we make use of all the data between 5-15-2020 and 12-31-2020 to build the training set.

average error rate over times and locations,

$$\text{error}_{\text{sMAPE}} = \frac{1}{mn} \sum_{\bar{t}=1}^{n} \sum_{i=1}^{m} \frac{|y_i^{(\bar{t}+l)} - \hat{y}_i^{(\bar{t}+l)}|}{|y_i^{(\bar{t}+l)} + \hat{y}_i^{(\bar{t}+l)}|}.$$

6.5.3 Baselines

We select three types of baselines and benchmark models to compare to our approach.

Compartment models. As there are a large number of compartment models proposed in recent days for COVID-19 forecast, we select three of them with the top performance and complete results in the desired time period from the COVID-19 Forecast Hub[‖]: `JHU_IDD-CovidSP` [142], `UCLA-SuEIR` [280], and `RobertWalraven-ESG` [251]. In detail, `JHU_IDD-CovidSP` proposes a modified SEIR compartment model where the time in the Infected compartment follows an Erlang distribution to produce more realistic infectious periods. `RobertWalraven-ESG` is a mathematical model that approximates the SEIR method with a particular skewed Gaussian distribution. `UCLA-SuEIR` extends SEIR by explicitly modeling the untested and unreported compartment. Note that the 1-day-ahead pandemic forecast results are not provided in the COVID-19 Forecast Hub.

Statistical time series prediction models. Two commonly used statistical models are compared to our approach: `ARIMA` and `PROPHET`. `ARIMA` [130] is an autoregressive moving average model, explaining a given time series based on its past values. `PROPHET` [167] is a time series prediction model[**] where non-linear trends can be fit with seasonality, plus holiday effects.

Neural network-based models. A simple two-layer LSTM-based neural network (`LSTM`) is used for COVID-19 pandemic prediction [62], taking the sequence of case numbers from the previous week as the input. `MPNN` [193] is a message passing neural network, building graphs to aggregate the historical case numbers from the neighboring locations based on

[‖]The model descriptions and up-to-date predicted results can be found at `https://github.com/reichlab/covid19-forecast-hub`.

[**]`https://github.com/facebook/prophet`.

the mobility magnitude. `MPNN+LSTM` [193] takes advantage of both `MPNN` and `LSTM` by jointly learning the graph propagation and temporal dependencies over case numbers of different times.

6.5.4 Implementation Details

Information Extraction. We train the named entity recognition and relation extraction models both for a maximum of 10 epochs. The models are implemented in PyTorch and we use Adam optimizer [123] to optimize the model parameters. We randomly select 10% instances from the training set as the validation set to select the optimal models. To avoid the GPU out-of-memory problem, we filter out tweets with more than 40 words (around 0.17%). In this work, we focus on the information extraction from English tweets so we also remove the tweets if 90% of the contents are non-English.

Time Series Prediction. We train the model for a maximum of 300 epochs. Early stopping occurs after 100 epochs. Similarly, we utilize PyTorch to implement the model and leverage Adam [123] for parameter optimization. Batch normalization [116] and dropout [230] are applied to the outputs of DGNN and FFN layers to avoid over-fitting. It takes around 8 hours to finish the complete training and evaluation cycle with one NVIDIA V100 GPU. We employ grid search to find the optimal hyperparameters of our model. Detailed hyperparameter values are listed in Table 6.1.

6.5.5 Results

Confirmed Case Forecast. Results of the confirmed case short-term and long-term forecasts are shown in Table 6.2. Compared to the best baseline method `MPNN+LSTM`, our model improves the average MAE and sMAPE by 7.3% and 2.3%, respectively. The results show `SMART` significantly outperforms the compartment models, such as `JHU_IDD-CovidSP` and `UCLA-SuEIR`. We think the big gap between our method and the compartment models results from the serious over-fitting issue in the SEIR model and its extensions. The SEIR

Table 6.1: Grid-search is used to find the optimal hyperparameters of our model.

Hyperparameter	Value
Learning Rate	0.001
Batch Size	4
Dropout Ratio	0.5
Bi-RNN Hidden State Size	64
DGNN Hidden Unit Size	64
Graph Sequence Length T	7
Semantic Feature Dim. d_e	768
Historical COVID-19 Statistics Feature Dim. d_t	7

model tends to assume that the peak would come right after the current data and is especially weak at predicting the progression at the early pandemic stage [89]. We also notice that the two statistical time series prediction models perform differently, and `ARIMA` gets much lower errors than `PROPHET` especially in the long-term prediction. This could be because `PROPHET` is supposed to work best with time series that have strong seasonal effects which is obviously not the situation in the COVID historical statistics. It turns out that a simple linear aggregation over the past case numbers can achieve relatively good performance. Besides, `MPNN` gets higher errors compared to its temporal variant, `MPNN+LSTM`, denoting the effectiveness of learning the temporal dependencies together with the graph aggregation. However, solely using `LSTM` to conduct the pandemic forecast achieves quite inaccurate predictions. We think it is because sequence modeling approaches like `LSTM` are unstable to handle the sequential inputs with sharply changing patterns [193]. For instance, it may be hard for `LSTM` to recognize turning points, such as lockdowns and reopens. `SMART` initially outperforms other models by a small margin (1-day-ahead forecast) while the improvement increases as the model predicts on later days. Compared to `MPNN+LSTM`, `SMART` achieves the largest error reduction of 9.5% and 9.4% while forecasting the case numbers in the next 7th and 14th day. This could be because the ongoing events discussed on social media would

Table 6.2: Performance of the short-term (1 day & 7 days ahead) and long-term (14 days & 28 days ahead) new confirmed case number forecast.

Confirmed Case	1 day ahead		7 days ahead		14 days ahead		28 days ahead		Average	
	MAE	sMAPE	MAE	sMAPE	MAE	sMAPE	MAE	sMAPE	MAE	sMAPE
JHU_IDD-CovidSP	-	-	1123.721	0.387	1253.138	0.409	1534.643	0.452	1303.834	0.416
RobertWalraven-ESG	-	-	768.433	0.310	978.533	0.369	2472.093	0.466	1406.353	0.382
UCLA-SuEIR	-	-	755.365	0.258	1099.761	0.335	1591.006	0.439	1148.711	0.344
ARIMA	604.181	0.200	802.977	0.250	961.297	0.286	1300.487	0.364	917.235	0.275
PROPHET	791.066	0.296	991.049	0.697	1341.798	0.810	2019.242	0.518	1285.789	0.581
LSTM	1262.333	0.393	1248.080	0.381	1235.201	0.357	1204.188	0.347	1237.450	0.369
MPNN	485.520	0.193	567.745	0.213	825.410	0.266	1304.112	0.352	795.697	0.256
MPNN+LSTM	455.677	0.172	523.770	0.209	672.049	**0.211**	967.123	0.286	654.655	0.220
SMART	**430.007**	**0.163**	**474.164**	**0.203**	**608.984**	0.216	**913.202**	**0.279**	**606.589**	**0.215**

not immediately affect the COVID-19 confirmed case numbers. More precisely, we need 1-2 weeks on average for the newly infected cases to be self-identified, tested and confirmed, based on our observations.

To observe the detailed forecast performance on every test instance, we plot the smoothed MAE curve for SMART and three neural network-based baselines (LSTM, MPNN, MPNN+LSTM). Note that every data point on the curves represents the MAE over all the test instances before the corresponding date. We observe that an error explosion becomes more and more clearly visible at the early stage of MPNN. We think MPNN is quite unstable especially when the training data are limited. In contrast, our SMART model remains stable of all time. In addition, we observe the average MAE comes to a peak in the middle of January for all the models. This is consistent with the fact that the new confirmed case numbers in the US come to a peak at around the same time. We also plot the smoothed sMAPE curve in Figure 6.6 which shows the sMAPE over the test instances before that date. All the curves quickly converge as the models obtain enough training instances, denoting the stability of our method.

Fatality Forecast. We show the results of fatality forecasts in Table6.3. SMART achieves 7.4% and 5.5% lower MAE and sMAPE, compared to the best baseline model MPNN+LSTM.

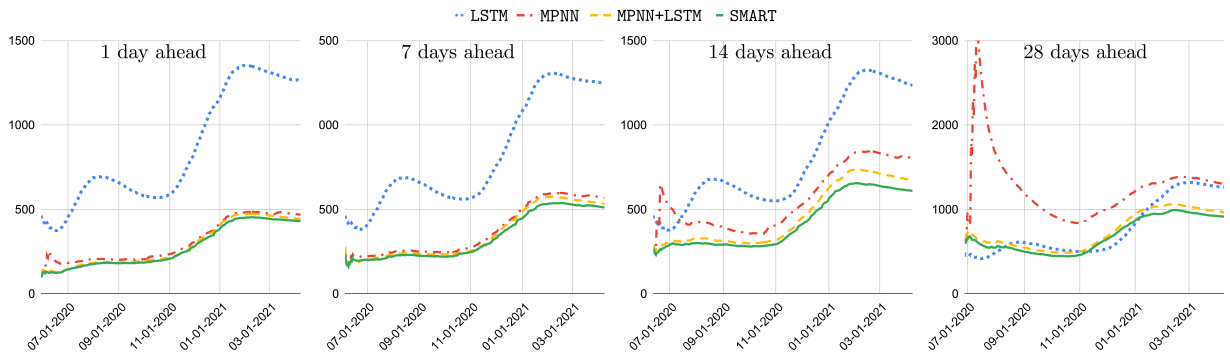


Figure 6.5: The comparison between SMART and three neural network-based baselines (LSTM, MPNN, MPNN+LSTM) on the smoothed MAE curve.

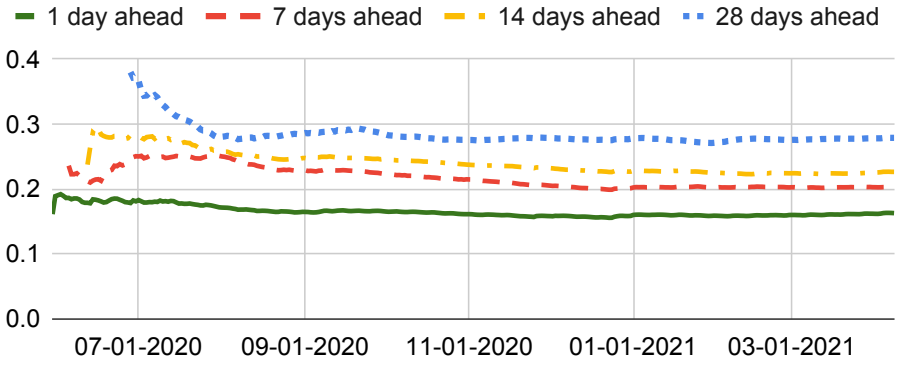


Figure 6.6: The comparison of smoothed sMAPE curve of SMART on four forecast tasks.

Among the three compartment models, UCLA-SuEIR performs the best. We surmise that taking unreported/untested cases leads to better modeling on fatalities. We notice the MAE of LSTM model is lower than SMART by 2.9% while its sMAPE is higher than SMART by 26.1%. We believe the LSTM model has been over-fitted to some extremely large or small values so that a large MAE can be avoided but the sMAPE will explode. Again, we find that the improvements of SMART on the 7,14-28-day-ahead forecast tasks (7.3%, 9.1%, and 8.7%) are much more significant than the 1-day-ahead forecast task (3.2%), demonstrating the long-term advantages of our method.

Table 6.3: Performance of the short-term (1 day & 7 days ahead) and long-term (14s day & 28 days ahead) new fatality number forecast.

Fatality	1 day ahead		7 days ahead		14 days ahead		28 days ahead		Average	
	MAE	sMAPE	MAE	sMAPE	MAE	sMAPE	MAE	sMAPE	MAE	sMAPE
JHU_IDD-CovidSP	-	-	18.911	0.465	19.851	0.480	24.362	0.516	21.041	0.487
RobertWalraven-ESG	-	-	15.490	0.452	18.590	0.484	26.179	0.541	20.086	0.492
UCLA-SuEIR	-	-	14.235	0.429	15.603	0.451	19.064	0.495	16.301	0.458
ARIMA	16.589	0.372	18.649	0.492	22.223	0.437	31.766	0.591	22.307	0.473
PROPHET	19.323	0.423	21.914	0.445	24.469	0.464	29.204	0.500	23.728	0.458
LSTM	18.039	0.423	17.937	0.432	17.770	0.542	**17.744**	0.531	17.872	0.482
MPNN	12.129	0.356	12.897	0.372	14.871	0.380	19.733	0.434	14.908	0.386
MPNN+LSTM	12.175	0.354	12.785	0.351	14.572	0.379	20.005	0.446	14.884	0.383
SMART	**11.783**	**0.346**	**11.847**	**0.331**	**13.236**	**0.349**	18.263	**0.421**	**13.782**	**0.362**

6.5.6 Ablation Study

We present the ablation study on the 7-day-ahead new confirmed case forecast task to demonstrate the effectiveness of each module in our framework. We observe similar results on other forecast tasks. Here we explain the different settings of our model variants as follows.

w/o RE module. Under this setting, we exclude the Entity-Entity edges in the heterogeneous knowledge graphs so that we can observe the improvement from our relation extraction module.

w/o NER module. We continue to exclude the Location-Entity edges to check the contribution of our named entity recognition module. Under this setting, all the edge propagation between location nodes and entity nodes are eliminated.

w/o Attentive Bi-RNN module. We remove the Attentive Bi-RNN module from our framework. We alternatively compute an element-wise averaged representation for each location node and feed it into the FNN layer for the pandemic forecast.

w/o DGNN module. To verify the contribution of our DGNN module, we remove the DGNN module but instead recursively feed the sequence of historical COVID-19 statistics features into the Attentive Bi-RNN units for each location node.

Table 6.4: Ablation study on the 7-day-ahead forecast task. Similar results can be achieved from other forecast tasks.

Model	MAE	Error Lift(%)
SMART	474.164	-
w/o RE module	495.688	+4.3
w/o NER module	518.389	+8.9
w/o Attentive Bi-RNN module	528.025	+10.4
w/o DGNN module	1112.334	+120.8
w/o CoronaBERT Language Model	500.878	+5.6

w/o CoronaBERT Language Model. We also observe the contribution from our pre-trained CoronaBERT language model by replacing it with a BERT language model (BERT-BASE) to initialize the semantic representations for each node.

In summary, every component in our framework is proved effective. Removing Entity-Entity and Location-Entity edges leads to 4.3% and 8.9% error lifts, respectively. When we jump over the DGNN module, the error dramatically increases, proving the capability of the heterogeneous graph to encode a rich spatial-temporal representation for each location node. The Attentive Bi-RNN module also makes a significant improvement of 10.4% on the forecast performance.

6.5.7 Risk Factor Discovery

To identify the potential location-wise *risk factors* of the COVID-19 pandemic, we make use of the normalized attention score $\alpha_{i,j}$ (introduced in Section 6.4.2) which indicates the contribution of each entity node i when node i's message is passed to the location node j. For each location, we first rank all the dates based on the number of confirmed cases in decreasing order. We then pick the top 20% dates with the biggest numbers from all the dates to build a *high set*. Ultimately, we aim at discovering a group of significant entities from the tweets

Table 6.5: Top-5 *risk factors* in six different states related to COVID-19 pandemic.

	California	**New York**	**Florida**
#1	pharmacists	traveler	workers
#2	#endthelockdown	doctors	#stopcovidcorruption
#3	mexico city	test results	crimes
#4	covid-positive	bill gates	voting
#5	msm	public health	#stayconnected
	Ohio	**Hawaii**	**Vermont**
#1	golf	mental health	#endthelockdown
#2	#hydroxychloroquine	immigrants	rape
#3	#wwg1wgaworldwide	surf	#wakeupamerica
#4	crush	2ndwave	burger
#5	traveler	patients	sickness

that are used to predict the confirmed cases on the dates from the *high set*. Specifically, during each inference process, we retrieve the attention scores of all the Location-Entity edges for each location node. We then compute a *risk score* for each (Location, Entity) pair by averaging the attention scores over all dates in the *high set*. Finally, the entities with top-k *risk scores* for each location can be considered as the *risk factors*.

Table 6.5 shows the top-5 *risk factors* of six states: California, New York, Florida, and Ohio, Hawaii, and Vermont with distinct spatial distributions as shown in Figure 6.4. Some of the entities can be easily connected with the increasing trend of the COVID-19 pandemic. For example, when people are seeking for *ending the lock down* in California and Vermont, or *staying connected* to each other in Florida, they are likely to go out, inevitably facilitating the spread of the virus. When people pay more attention to the local doctor resource or public health condition in New York, the peak of the pandemic should not be far away. However, it may be hard to interpret some entities like *msm* without the contexts since *msm* can be the abbreviation of either *mainstream media* or *master of science in management*.

Table 6.6: Top-5 *risk factors* under four different entity categories related to COVID-19 pandemic.

	HASHTAG	**SIGN_OR_SYMPTOM**
#1	#wakeupamerica	cough
#2	#covidiot	sneezes
#3	#breakingnews	headaches
#4	#staysafe	chill
#5	#ppeshortage	sickness
	SOCIAL_INDIVIDUAL_BEHAVIOR	**ORGANIZATION**
#1	genocide	@youtube
#2	loyalty	@nytimes
#3	discord	nih
#4	voting	amazon
#5	racism	msm

We also incorporate the named entity recognition results to show in Table 6.6 the top5 *risk factors* under 4 different categories: HASHTAG, SIGN_OR_SYMPTOM, SOCIAL_INDIVIDUAL_BEHAVIOR and ORGANIZATION. We notice *msm* is categorized as an organization, so it is more likely to be interpreted as the *mainstream media*. It is obvious that the pandemic is getting more serious if we are facing the *personal protective equipment shortage*. The government and health institutes are better to be prepared if more and more people become sick and have the symptoms such as *cough* and *sneezes*. There are limitations if we only rely on the entities with high attention scores to interpret the *risk factors*. For example, we cannot simply conclude that the prevailing entity *amazon* results in an increasing trend of the pandemic. The relationship between *amazon* and increasing trend might not be causal but just co-occurrence.

6.6 Conclusion

In this study, we conduct the first trial to incorporate the entities and relationships extracted from social media data to simultaneously enhance the pandemic surveillance and detect the potential risk factors. We propose a dynamic graph neural network to learn the temporal dependency among nodes of different times and propagate the messages among the heterogeneous nodes. Extensive experiments show the effectiveness and robustness of our forecast model. We will open-source our framework and release the pre-trained `CoronaBERT` language model to facilitate future research in this direction.

6.7 Acknowledgment

This work is a version of [273] and was partially supported by the National Science Foundation [NSF-DGE-1829071, NSFIIS- 2031187] and the National Institutes of Health [NIH-R35-HL135772, NIH/NIBIB-R01-EB027650].

CHAPTER 7

Discussion and Future Directions

In this thesis, we present different strategies for information extraction under a low resource scenario including acquiring supervision from auxiliary knowledge and transferable representations. In all cases, we show the efficient information extraction can be achieved by carefully learning the representations of words, sentences and documents. We analyze the proposed algorithms and provide extensive experimental results. In this chapter, we conclude the thesis by discussing some directions for future research.

For the clinical information extraction, we present a comprehensive extraction pipeline. We first propose `ACROBAT`, a clinical typing system to facilitate producing resources for training clinical information extraction models and better understanding the concepts with clinical documents and publications. The typing system is proved to be able to reflect diverse vocabulary and phenomena described with the clinical documents without requiring direct connections to curated concepts or terminology. This typing system serves as a precious test bed for clinical researchers considering the limited public resources in this domain. We then propose a contextualized language model enhanced named entity recognition model to extract significant entities and events from the clinical narratives. Two pre-trained clinical language models, `Clinical-ELMo` and `Clinical-Flair` are released to facilitate further research in the clinical community. Next, we propose `CTRL-PG` which focuses on the temporal relation extraction task by incorporating the probabilistic soft logic rules to model the transitivity constraints and symmetric dependencies among relevant relationships. In this study, we prove the effectiveness of taking advantage of the auxiliary supervision for information extraction. Last but not least, we introduce our `CREATe` system, which incorporates all

the aforementioned advanced information extraction algorithms for building an end-to-end query system. The system is capable of indexing relevant case reports from the heterogeneous entity graphs according to the matched keywords and chronologies (prioritized). In this thesis, we focus on the research topics of named entity recognition and temporal relation extraction, while there remains many significant directions to explore in the clinical domain, such as co-reference resolution in the clinical narratives and knowledge graph construction. Aligning the knowledge graphs based on the information extraction results can benefit many down-streaming applications including medical question answering and medicine discovery.

Chapter 4 discusses the effectiveness of learning multi-modal representations for information extraction. We build a cross-modality encoder to digest both linguistic and visual features in Section 4.1 to facilitate the theme (keyphrase) recommendation process for ad images. This cross-attentive method can be applied broadly. For example, it can be applied for solving the visual question answering problems and tackling the visual semantic role labeling tasks. The `PCPR` framework proposed in Section 4.2 jointly learns the multi-modal representations for pun recognition in the figurative language processing task based on both word and pronunciation embeddings. We notice that the pronunciation embeddings can facilitate also facilitate the homographic pun detection, implying the potential of pronunciation for enhancing general language modeling. This observation opens up several directions for future research. Though some advanced language models such as BERT and RoBERTa have pushed the performance on many NLP tasks dramatically, there remains room to improve the language modeling, for example, by incorporating the pronunciation modeling. Moreover, the proposed framework can also be applied to other figurative language processing tasks such as irony detection and poem generation.

We propose `SimpDOM` to build rich DOM node representations for the few-shot attribute extraction tasks in chapter 5. There are several directions worth exploring to improve `SimpDOM` algorithms including a better implementation that could further cut down the clock time for extracting the contextual information for each node, experimental study of additional cross-domain prediction problems, and the study on extracting attributes from

further more unannotated websites and pages. Automatic evaluation should also be explored for the open information extraction tasks.

We introduce the `SMART` framework in Chapter 6 which extracts named entities and relationships from the social media corpus and builds heterogeneous knowledge graphs to enhance the pandemic time series prediction model. Overall, we provide a generic solution for taking advantage of the informative entities and relationships in the social media data. It is straightforward to apply our approach to any future epidemiological surveillance. Our approach also has the potential to tackle other real-world problems, such as environment monitoring and crime detection. In the future, it will be beneficial to detect the risk factors in a more strict manner by identifying the relationship between the risk factors and the pandemic trends or predicted targets.

Bibliography

[1] Automatic understanding of image and video advertisements. http://people.cs.pitt.edu/~kovashka/ads, 2019.

[2] Banner blindness. https://en.wikipedia.org/wiki/Banner_blindness, 2019.

[3] Facebook business: Optimize your ad results by refreshing your creative. https://www.facebook.com/business/m/test-ads-on-facebook, 2019.

[4] Marketing land: Social media ad fatigue. https://marketingland.com/ad-fatigue-social-media-combat-224234, 2019.

[5] Match zoo. https://github.com/NTMC-Community/MatchZoo, 2019.

[6] Shutterstock: Search millions of royalty free stock images, photos, videos, and music. https://www.shutterstock.com/, 2019.

[7] Taboola-trends. https://trends.taboola.com/, 2019.

[8] Hossein Abbasimehr and Reza Paki. Prediction of covid-19 confirmed cases combining deep learning methods and bayesian optimization. *Chaos, Solitons & Fractals*, 142:110511, 2021.

[9] Kiran Adnan and Rehan Akbar. An analytical study of information extraction from unstructured and multidimensional big data. *Journal of Big Data*, 6(1):1–38, 2019.

[10] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649. Association for Computational Linguistics, 2018.

[11] Ghada Alfattni, Niels Peek, and Goran Nenadic. Extraction of temporal relations from clinical free text: A systematic review of current approaches. *Journal of Biomedical Informatics*, page 103488, 2020.

[12] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*, 2019.

[13] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

[14] Han Altae-Tran, Bharath Ramsundar, Aneesh S Pappu, and Vijay Pande. Low data drug discovery with one-shot learning. *ACS central science*, 3(4):283–293, 2017.

[15] Nick Altieri, Rebecca L Barter, James Duncan, Raaz Dwivedi, Karl Kumbier, Xiao Li, Robert Netzorg, Briton Park, Chandan Singh, Yan Shuo Tan, et al. Curating a covid-19 data repository and forecasting county-level death counts in the united states. *arXiv preprint arXiv:2005.07882*, 2020.

[16] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

[17] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[18] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[19] Joan L Aron and Ira B Schwartz. Seasonality and period-doubling bifurcations in an epidemic model. *Journal of theoretical biology*, 110(4):665–679, 1984.

[20] A. R. Aronson. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. *Proc AMIA Symp*, pages 17–21, 2001.

[21] Alan R Aronson and François-Michel Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.

[22] Agnese Augello, Gaetano Saccone, Salvatore Gaglio, and Giovanni Pilato. Humorist bot: Bringing computational humour in a chat-bot system. In *CISIS 2008*, pages 703–708, 2008.

[23] Mohd Amir Bin Mohd Azir and Kamsuriah Binti Ahmad. Wrapper approaches for web data extraction: A review. In *2017 6th International Conference on Electrical Engineering and Informatics (ICEEI)*, pages 1–6. IEEE, 2017.

[24] Stephen H Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. Hinge-loss markov random fields and probabilistic soft logic. *The Journal of Machine Learning Research*, 18(1):3846–3912, 2017.

[25] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[26] Norman TJ Bailey et al. *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd, 5a Crendon Street, High Wycombe, Bucks HP13 6LE., 1975.

[27] L. Balbuena, D. Hayes, S. G. Ramirez, and R. Johnson. Eagle's syndrome (elongated styloid process). *Southern Medical Journal*, 90(3):331–4, Mar 1997.

[28] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

[29] David W Bates, Suchi Saria, Lucila Ohno-Machado, Anand Shah, and Gabriel Escobar. Big data in health care: using analytics to identify and manage high-risk and high-cost patients. *Health Affairs*, 33(7):1123–1131, 2014.

[30] Ross Beckley, Cametria Weatherspoon, Michael Alexander, Marissa Chandler, Anthony Johnson, and Ghan S Bhatt. Modeling epidemics with differential equation. *Tennessee State University Internal Report*, 2013.

[31] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[32] Sagie Benaim and Lior Wolf. One-shot unsupervised cross domain translation. *arXiv preprint arXiv:1806.06029*, 2018.

[33] Dario Bertero and Pascale Fung. Predicting humor response in dialogues from tv sitcoms. In *ICASSP 2016*, pages 5780–5784, 2016.

[34] Andrea L Bertozzi, Elisa Franco, George Mohler, Martin B Short, and Daniel Sledge. The challenges of modeling and forecasting the spread of covid-19. *Proceedings of the National Academy of Sciences*, 117(29):16732–16738, 2020.

[35] Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. SemEval-2015 task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814, Denver, Colorado, June 2015. Association for Computational Linguistics.

[36] Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. SemEval-2016 task 12: Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062, San Diego, California, June 2016. Association for Computational Linguistics.

[37] Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. SemEval-2017 task 12: Clinical TempEval. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[38] Lidong Bing, Tak-Lam Wong, and Wai Lam. Unsupervised extraction of popular product attributes from e-commerce web sites by considering customer reviews. *ACM Transactions on Internet Technology (TOIT)*, 16(2):1–17, 2016.

[39] Md Haider Ali Biswas, Luís Tiago Paiva, and MDR De Pinho. A seir model for control of infectious diseases with constraints. *Mathematical Biosciences & Engineering*, 11(4):761, 2014.

[40] Vladislav Blinov, Valeria Bolotova-Baranova, and Pavel Braslavski. Large dataset and language model fun-tuning for humor recognition. In *ACL 2019*, pages 4027–4032, 2019.

[41] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Research*, 32(90001):D267–70, 2004.

[42] Florian Boudin. pke: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, December 2016.

[43] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[44] Alberto J Cabán-Martinez and Wilfredo F García-Beltrán. Advancing medicine one research note at a time: the educational value in clinical case reports. *BMC Research Notes*, 5(1):293, 2012.

[45] Yitao Cai, Yin Li, and Xiaojun Wan. Sense-aware neural models for pun location in texts. In *ACL 2018*, pages 546–551, 2018.

[46] Andrew Carlson and Charles Schafer. Bootstrapping information extraction from semi-structured web pages. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 195–210. Springer, 2008.

[47] Xavier Carreras, Lluís Màrquez, and Lluís Padró. Named entity extraction using adaboost. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

[48] Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, June 2014.

[49] J. H. Caufield, Y. Zhou, A. O. Garlid, S. P. Setty, D. A. Liem, Q. Cao, J. M. Lee, S. Murali, S. Spendlove, W. Wang, L. Zhang, Y. Sun, A. Bui, H. Hermjakob, K. E. Watson, and P. Ping. A reference set of curated biomedical data and metadata from clinical case reports. *Scientific Data*, 5:180258, Nov 2018.

[50] J Harry Caufield, Yichao Zhou, Yunsheng Bai, David A Liem, Anders O Garlid, Kai-Wei Chang, Yizhou Sun, Peipei Ping, and Wei Wang. A comprehensive typing system for information extraction from clinical narratives. *medRxiv*, page 19009118, 2019.

[51] J Harry Caufield, Yijiang Zhou, Anders O Garlid, Shaun P Setty, David A Liem, Quan Cao, Jessica M Lee, Sanjana Murali, Sarah Spendlove, Wei Wang, et al. A reference set of curated biomedical data and metadata from clinical case reports. *Scientific data*, 5:180258, 2018.

[52] Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284, 2014.

[53] Chia-Hui Chang, Mohammed Kayed, Moheb R Girgis, and Khaled F Shaalan. A survey of web information extraction systems. *IEEE transactions on knowledge and data engineering*, 18(10):1411–1428, 2006.

[54] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

[55] Jonathan H Chen, Tanya Podchiyska, and Russ B Altman. Orderrex: clinical order decision support and outcome predictions by data-mining electronic medical records. *Journal of the American Medical Informatics Association*, 23(2):339–348, 2016.

[56] Lei Chen and Chong MIn Lee. Predicting audience's laughter using convolutional neural network. *arXiv preprint arXiv:1702.02584*, 2017.

[57] Peng-Yu Chen and Von-Wun Soo. Humor recognition using deep learning. In *NAACL 2018*, pages 113–117, 2018.

[58] Wei Chen, Lang Zong, Weijing Huang, Gaoyan Ou, Yue Wang, and Dongqing Yang. An empirical study of massively parallel bayesian networks learning for sentiment extraction from unstructured text. In *Asia-Pacific Web Conference*, pages 424–435. Springer, 2011.

[59] Xuelu Chen, Muhao Chen, Weijia Shi, Yizhou Sun, and Carlo Zaniolo. Embedding uncertain knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3363–3370, 2019.

[60] Fei Cheng and Yusuke Miyao. Classifying temporal relations by bidirectional LSTM over dependency paths. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, July 2017.

[61] Veera Raghavendra Chikka. CDE-IIITH at SemEval-2016 task 12: Extraction of temporal information from clinical documents using machine learning techniques. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1237–1240, San Diego, California, June 2016. Association for Computational Linguistics.

[62] Vinay Kumar Reddy Chimmula and Lei Zhang. Time series forecasting of covid-19 transmission in canada using lstm networks. *Chaos, Solitons & Fractals*, 135:109864, 2020.

[63] Kyunghyun Cho and Van Merrienboer. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[64] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[65] J. Clement. Countries with most twitter users 2020, Feb 2020.

[66] K. B. Cohen, A. Lanfranchi, M. J.-Y. Choi, M. Bada, W. A. Baumgartner, N. Panteleyeva, K. Verspoor, M. Palmer, and L. E. Hunter. Coreference annotation and resolution in the Colorado Richly Annotated Full Text (CRAFT) corpus of biomedical journal articles. *BMC Bioinformatics*, 18(1):372, Dec 2017.

[67] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.

[68] IHME COVID, Christopher JL Murray, et al. Forecasting covid-19 impact on hospital bed-days, icu-days, ventilator-days and deaths by us state in the next 4 months. *MedRxiv*, 2020.

[69] Berry De Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *Journal of the American Medical Informatics Association*, 18(5):557–562, 2011.

[70] Lingjia Deng and Janyce Wiebe. Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 179–189, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[71] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[72] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[73] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[74] Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. Neural temporal relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 746–751, 2017.

[75] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610, 2014.

[76] Samuel Doogan, Aniruddha Ghosh, Hanyang Chen, and Tony Veale. Idiom savant at semeval-2017 task 7: Detection and interpretation of english puns. In *SemEval-2017*, pages 103–108, 2017.

[77] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. Ncbi disease corpus. *J. of Biomedical Informatics*, 47(C), February 2014.

[78] Matthijs Douze, Arthur Szlam, Bharath Hariharan, and Hervé Jégou. Low-shot learning with large-scale diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3349–3358, 2018.

[79] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[80] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[81] Paul Fine, Ken Eames, and David L Heymann. "herd immunity": a rough guide. *Clinical infectious diseases*, 52(7):911–916, 2011.

[82] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[83] Corina Florescu and Cornelia Caragea. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, July 2017.

[84] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 168–171. Association for Computational Linguistics, 2003.

[85] G.D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[86] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2):115–130, 2000.

[87] Diana Galvan, Naoaki Okazaki, Koji Matsuda, and Kentaro Inui. Investigating the challenges of temporal relation extraction from clinical text. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 55–64, 2018.

[88] Diana Galvan-Sosa, Koji Matsuda, Naoaki Okazaki, and Kentaro Inui. Empirical exploration of the challenges in temporal relation extraction from clinical text. *Journal of Natural Language Processing*, 27(2):383–409, 2020.

[89] Junyi Gao, Rakshith Sharma, Cheng Qian, Lucas M Glass, Jeffrey Spaeder, Justin Romberg, Jimeng Sun, and Cao Xiao. Stan: spatio-temporal attention network for pandemic prediction using real-world evidence. *Journal of the American Medical Informatics Association*, 28(4):733–743, 2021.

[90] Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. Semeval-2015 task 11: Sentiment analysis of figurative language in twitter. In *SemEval 2015*, pages 470–478, 2015.

[91] David Graff. The aquaint corpus of english news text ldc2002t31. Linguistic Data Consortium, Philadelphia, 2002.

[92] Hong Guan, Jianfu Li, Hua Xu, and Murthy Devarakonda. Robustly pre-trained neural model for direct temporal relation extraction. *arXiv preprint arXiv:2004.06216*, 2020.

[93] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016.

[94] Suhit Gupta, Gail Kaiser, David Neistadt, and Peter Grimm. Dom-based content extraction of html documents. In *Proceedings of the 12th international conference on World Wide Web*, pages 207–214, 2003.

[95] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 2017.

[96] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[97] Rujun Han, I-Hung Hsu, Mu Yang, Aram Galstyan, Ralph Weischedel, and Nanyun Peng. Deep structured neural network for event temporal relation extraction. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, November 2019.

[98] Rujun Han, Qiang Ning, and Nanyun Peng. Joint event and temporal relation extraction with shared representations and structured prediction. 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)*, November 2019.

[99] Rujun Han, Yichao Zhou, and Nanyun Peng. Domain knowledge empowered structured neural net for end-to-end event temporal relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5717–5729, Online, November 2020. Association for Computational Linguistics.

[100] Xu Han, Tianyu Gao, Yuan Yao, Demin Ye, Zhiyuan Liu, and Maosong Sun. Opennre: An open and extensible toolkit for neural relation extraction. *arXiv preprint arXiv:1909.13078*, 2019.

[101] Xuehua Han, Juanle Wang, Min Zhang, and Xiaojie Wang. Using social media to mine and analyze public opinion related to covid-19 in china. *International Journal of Environmental Research and Public Health*, 17(8):2788, 2020.

[102] Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. From one tree to a forest: a unified solution for structured web data extraction. In *Proceedings of the 34th international*

ACM SIGIR conference on Research and development in Information Retrieval, pages 775–784, 2011.

[103] Tiberiu Harko, Francisco SN Lobo, and MK Mak. Exact analytical solutions of the susceptible-infected-recovered (sir) epidemic model and of the sir model with equal death and birth rates. *Applied Mathematics and Computation*, 236:184–194, 2014.

[104] Herbert W Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

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

[106] Judy Hoffman, Eric Tzeng, Jeff Donahue, Yangqing Jia, Kate Saenko, and Trevor Darrell. One-shot adaptation of supervised deep convolutional models. *arXiv preprint arXiv:1312.6204*, 2013.

[107] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[108] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany, August 2016. Association for Computational Linguistics.

[109] Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. Attention-based multimodal neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 639–645, 2016.

[110] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

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

[112] Lluís-F Hurtado, Encarna Segarra, Ferran Pla, Pascual Carrasco, and José-Angel González. Elirf-upv at semeval-2017 task 7: Pun detection and interpretation. In *SemEval-2017*, pages 440–443, 2017.

[113] Zaeem Hussain, Mingda Zhang, Xiaozhong Zhang, Keren Ye, Christopher Thomas, Zuha Agha, Nathan Ong, and Adriana Kovashka. Automatic understanding of image and video advertisements. In *CVPR*, 2017.

[114] Ilija Ilievski and Jiashi Feng. Multimodal learning and reasoning for visual question answering. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 551–562, 2017.

[115] Vijayasaradhi Indurthi and Subba Reddy Oota. Fermi at semeval-2017 task 7: Detection and interpretation of homographic puns in english language. In *SemEval-2017*, pages 457–460, 2017.

[116] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[117] Aaron Jaech, Rik Koncel-Kedziorski, and Mari Ostendorf. Phonological punderstanding. In *ACL 2016*, pages 654–663, 2016.

[118] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[119] Jyun-Yu Jiang, Xue Sun, Wei Wang, and Sean Young. Enhancing air quality prediction with social media and natural language processing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2627–2632, 2019.

[120] Xiaoyong Jin, Yu-Xiang Wang, and Xifeng Yan. Inter-series attention model for covid-19 forecasting. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 495–503. SIAM, 2021.

[121] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

[122] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. GENIA corpus–a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl 1):i180–i182, Jul 2003.

[123] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[124] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[125] İsmail Kırbaş, Adnan Sözen, Azim Doğuş Tuncer, and Fikret Şinasi Kazancıoğlu. Comparative analysis and forecasting of covid-19 cases in various european countries with arima, narnn and lstm approaches. *Chaos, Solitons & Fractals*, 138:110015, 2020.

[126] George Klir and Bo Yuan. *Fuzzy sets and fuzzy logic*, volume 4. Prentice hall New Jersey, 1995.

[127] Furkan Kocayusufoglu, Ying Sheng, Nguyen Vo, James Wendt, Qi Zhao, Sandeep Tata, and Marc Najork. Riser: Learning better representations for richly structured emails. In *The World Wide Web Conference*, pages 886–895, 2019.

[128] Quyu Kong, Rohit Ram, and Marian-Andrei Rizoiu. Evently: Modeling and analyzing reshare cascades with hawkes processes. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1097–1100, 2021.

[129] Martin Kröger and Reinhard Schlickeiser. Analytical solution of the sir-model for the temporal evolution of epidemics. part a: time-independent reproduction factor. *Journal of Physics A: Mathematical and Theoretical*, 53(50):505601, 2020.

[130] Tadeusz Kufel et al. Arima-based forecasting of the dynamics of confirmed covid-19 cases for selected european countries. *Equilibrium. Quarterly Journal of Economics and Economic Policy*, 15(2):181–204, 2020.

[131] Nicholas Kushmerick, Daniel S Weld, and Robert Doorenbos. *Wrapper induction for information extraction*. University of Washington Washington, 1997.

[132] Roland Kwitt, Sebastian Hegenbart, and Marc Niethammer. One-shot learning of scene locations via feature trajectory transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 78–86, 2016.

[133] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[134] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[135] Kun Lan, Dan-tong Wang, Simon Fong, Lian-sheng Liu, Kelvin KL Wong, and Nilanjan Dey. A survey of data mining and deep learning in bioinformatics. *Journal of medical systems*, 42(8):1–20, 2018.

[136] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[137] Hee-Jin Lee, Hua Xu, Jingqi Wang, Yaoyun Zhang, Sungrim Moon, Jun Xu, and Yonghui Wu. UTHealth at SemEval-2016 task 12: an end-to-end system for temporal information extraction from clinical notes. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, June 2016.

[138] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR*, abs/1901.08746, 2019.

[139] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, 2019.

[140] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[141] Artuur Leeuwenberg and Marie-Francine Moens. Structured learning for temporal relation extraction from clinical records. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, April 2017.

[142] Joseph Chadi Lemaitre, Kyra H Grantz, Joshua Kaminsky, Hannah R Meredith, Shaun A Truelove, Stephen A Lauer, Lindsay T Keegan, Sam Shah, Josh Wills, Kathryn Kaminsky, et al. A scenario modeling pipeline for covid-19 emergency planning. *medRxiv*, 2020.

[143] Jake Lever and Steven Jones. Painless Relation Extraction with Kindred. *BioNLP 2017*, pages 176–183, 2017.

[144] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, 2019.

[145] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[146] Stan Z. Li and Anil Jain, editors. *Forward-Backward Algorithm*, pages 580–580. Springer US, Boston, MA, 2009.

[147] Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*, 2019.

[148] Bill Yuchen Lin, Ying Sheng, Nguyen Vo, and Sandeep Tata. Freedom: A transferable neural architecture for structured information extraction on web documents. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1092–1102, 2020.

[149] Chen Lin, Timothy Miller, Dmitriy Dligach, Hadi Amiri, Steven Bethard, and Guergana Savova. Self-training improves recurrent neural networks performance for temporal relation extraction. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 165–176, 2018.

[150] Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. Representations of time expressions for temporal relation extraction with convolutional neural networks. In *BioNLP 2017*, pages 322–327, 2017.

[151] Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. A BERT-based universal model for both within- and cross-sentence clinical temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 65–71, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[152] Chen Lin, Timothy Miller, Dmitriy Dligach, Farig Sadeque, Steven Bethard, and Guergana Savova. A BERT-based one-pass multi-task model for clinical temporal relation extraction. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 70–75, Online, July 2020. Association for Computational Linguistics.

[153] M. Tamer Ozsu Ling Liu. *Encyclopedia of Database Systems*. Springer New York, 2nd ed. edition, 2018.

[154] H. Liu, Z.-Z. Hu, M. Torii, C. Wu, and C. Friedman. Quantitative assessment of

dictionary-based protein named entity tagging. *Journal of the American Medical Informatics Association*, 13(5):497–507, 2006.

[155] Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. Empower sequence labeling with task-aware neural language model. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[156] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.

[157] Qi Liu, Matt J Kusner, and Phil Blunsom. A survey on contextual embeddings. *arXiv preprint arXiv:2003.07278*, 2020.

[158] Sijia Liu, Liwei Wang, Vipin Chaudhary, and Hongfang Liu. Attention neural model for temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 134–139, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

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

[160] George Livadiotis. Statistical analysis of the impact of environmental temperature on the exponential growth rate of cases infected by covid-19. *PLoS one*, 15(5):e0233875, 2020.

[161] Hector Llorens, Estela Saquete, and Borja Navarro. Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291, 2010.

[162] Colin Lockard, Xin Luna Dong, Arash Einolghozati, and Prashant Shiralkar. Ceres: Distantly supervised relation extraction from the semi-structured web. *arXiv preprint arXiv:1804.04635*, 2018.

[163] Colin Lockard, Prashant Shiralkar, Xin Luna Dong, and Hannaneh Hajishirzi. Zeroshotceres: Zero-shot relation extraction from semi-structured webpages. *arXiv preprint arXiv:2005.07105*, 2020.

[164] Patrice Lopez. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *International conference on theory and practice of digital libraries*, pages 473–474. Springer, 2009.

[165] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[166] Xuezhe Ma and Eduard H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *CoRR*, abs/1603.01354, 2016.

[167] Sakib Mahmud. Bangladesh covid-19 daily cases time series analysis using facebook prophet model. *Available at SSRN 3660368*, 2020.

[168] Bodhisattwa Prasad Majumder, Navneet Potti, Sandeep Tata, James Bradley Wendt, Qi Zhao, and Marc Najork. Representation learning for information extraction from form-like documents. In *proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pages 6495–6504, 2020.

[169] Kevin Makice. *Twitter API: Up and running: Learn how to build applications with the Twitter API.* " O'Reilly Media, Inc.", 2009.

[170] Johanna McEntyre and David Lipman. Pubmed: bridging the information gap. *Cmaj*, 164(9):1317–1319, 2001.

[171] Alan K Melby and Terry Warner. *The possibility of language: A discussion of the nature of language, with implications for human and machine translation*, volume 14. John Benjamins Publishing, 1995.

[172] Yuanliang Meng and Anna Rumshisky. Context-aware neural model for temporal information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, July 2018.

[173] Rada Mihalcea and Carlo Strapparava. Making computers laugh: Investigations in automatic humor recognition. In *EMNLP 2005*, pages 531–538, 2005.

[174] Elena Mikhalkova and Yuri Karyakin. Punfields at semeval-2017 task 7: Employing roget's thesaurus in automatic pun recognition and interpretation. *arXiv preprint arXiv:1707.05479*, 2017.

[175] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *LREC 2018*, 2018.

[176] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[177] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. 2013.

[178] George Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[179] Stephanie A Miller and Lenhart K Schubert. Time revisited 1. *Computational Intelligence*, 6(2):108–118, 1990.

[180] Tristan Miller, Christian Hempelmann, and Iryna Gurevych. Semeval-2017 task 7: Detection and interpretation of english puns. In *SemEval-2017*, pages 58–68, 2017.

[181] Shaunak Mishra, Manisha Verma, and Jelena Gligorijevic. Guiding creative design in online advertising. RecSys, 2019.

[182] Shaunak Mishra, Manisha Verma, Yichao Zhou, Kapil Thadani, and Wei Wang. Learning to create better ads: Generation and ranking approaches for ad creative refinement. CIKM '20, page 2653–2660, New York, NY, USA, 2020. Association for Computing Machinery.

[183] Islam Abdalla Mohamed, Anis Ben Aissa, Loay F Hussein, Ahmed I Taloba, et al. A new model for epidemic prediction: Covid-19 in kingdom saudi arabia case study. *Materials Today: Proceedings*, 2021.

[184] Louis-Philippe Morency and Tadas Baltrušaitis. Multimodal machine learning: integrating language, vision and speech. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 3–5, 2017.

[185] Ion Muslea, Steve Minton, and Craig Knoblock. A hierarchical approach to wrapper induction. In *Proceedings of the third annual conference on Autonomous Agents*, pages 190–197, 1999.

[186] Teagen Nabity-Grover, Christy MK Cheung, and Jason Bennett Thatcher. Inside out and outside in: How the covid-19 pandemic affects self-disclosure on social media. *International Journal of Information Management*, 55:102188, 2020.

[187] Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):10, 2009.

[188] Anton Nijholt, Andreea I Niculescu, Valitutti Alessandro, and Rafael E Banchs. Humor in human-computer interaction: a short survey. 2017.

[189] Azadeh Nikfarjam, Ehsan Emadzadeh, and Graciela Gonzalez. Towards generating a patient's timeline: extracting temporal relationships from clinical notes. *Journal of biomedical informatics*, 46:S40–S47, 2013.

[190] Qiang Ning, Zhili Feng, and Dan Roth. A structured learning approach to temporal

relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, September 2017.

[191] Dieke Oele and Kilian Evang. Buzzsaw at semeval-2017 task 7: Global vs. local context for interpreting and locating homographic english puns with sense embeddings. In *SemEval-2017*, pages 444–448, 2017.

[192] Mary E Okurowski. Information extraction overview. Technical report, NATIONAL COMPUTER SECURITY CENTER FORT GEORGE G MEADE MD, 1993.

[193] George Panagopoulos, Giannis Nikolentzos, and Michalis Vazirgiannis. Transfer graph neural networks for pandemic forecasting. 2020.

[194] Nagesh C Panyam, Karin Verspoor, Trevor Cohn, and Kotagiri Ramamohanarao. Exploiting graph kernels for high performance biomedical relation extraction. *Journal of biomedical semantics*, 9(1):7, 2018.

[195] Aditya Parameswaran, Nilesh Dalvi, Hector Garcia-Molina, and Rajeev Rastogi. Optimal schemes for robust web extraction. *Proceedings of the VLDB Conference*, 4(11), September 2011.

[196] Alexandre Passos, Vineet Kumar, and Andrew McCallum. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*, 2014.

[197] Panupong Pasupat and Percy Liang. Zero-shot entity extraction from web pages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 391–401, 2014.

[198] Ted Pedersen. Duluth at semeval-2017 task 7: Puns upon a midnight dreary, lexical semantics for the weak and weary. *arXiv preprint arXiv:1704.08388*, 2017.

[199] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP 2014*, pages 1532–1543, 2014.

[200] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

[201] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[202] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[203] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

[204] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[205] Aniket Pramanick and Dipankar Das. Ju cse nlp @ semeval 2017 task 7: Employing rules to detect and interpret english puns. In *SemEval-2017*, pages 432–435, 2017.

[206] James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK., 2003.

[207] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[208] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):18, 2018.

[209] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[210] Alexander Rodriguez, Anika Tabassum, Jiaming Cui, Jiajia Xie, Javen Ho, Pulak Agarwal, Bijaya Adhikari, and B Aditya Prakash. Deepcovid: An operational deep learning-driven framework for explainable real-time covid-19 forecasting. *medRxiv*, 2020.

[211] Ronald Ross. An application of the theory of probabilities to the study of a priori pathometry.—part i. *Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character*, 92(638):204–230, 1916.

[212] Amal I Saba and Ammar H Elsheikh. Forecasting the prevalence of covid-19 outbreak in egypt using nonlinear autoregressive artificial neural networks. *Process safety and environmental protection*, 141:1–8, 2020.

[213] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860, 2010.

[214] Claude Sammut and Geoffrey I. Webb, editors. *Mean Absolute Error*, pages 652–652. Springer US, Boston, MA, 2010.

[215] Claude Sammut and Geoffrey I. Webb, editors. *Mean Squared Error*, pages 653–653. Springer US, Boston, MA, 2010.

[216] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: a large-scale dataset for multimodal language understanding. *arXiv preprint arXiv:1811.00347*, 2018.

[217] G. K. Savova, W. W. Chapman, J. Zheng, and R. S. Crowley. Anaphoric relations in the clinical narrative: corpus creation. *Journal of the American Medical Informatics Association : JAMIA*, 18(4):459–465, Jul 2011.

[218] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical text analysis

and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[219] Reinhard Schlickeiser and Martin Kröger. Analytical solution of the sir-model for the temporal evolution of epidemics. part b. semi-time case. *Journal of Physics A: Mathematical and Theoretical*, 2021.

[220] Susanne Schmidt and Martin Eisend. Advertising repetition: A meta-analysis on effective frequency in advertising. *Journal of Advertising*, 44(4):415–428, 2015.

[221] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[222] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *An introduction to information retrieval*. Cambridge University Press,, 2007.

[223] Shishir K Shandilya and Suresh Jain. Automatic opinion extraction from web documents. In *2009 International Conference on Computer and Automation Engineering*, pages 351–355. IEEE, 2009.

[224] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

[225] Golnar Sheikhshabbafghi, Inanc Birol, and Anoop Sarkar. In-domain context-aware token embeddings improve biomedical named entity recognition. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, October 2018.

[226] David Smiley, Eric Pugh, Kranti Parisa, and Matt Mitchell. *Apache Solr enterprise search server*. Packt Publishing Ltd, 2015.

[227] Xinying Song, Jing Liu, Yunbo Cao, Chin-Yew Lin, and Hsiao-Wuen Hon. Automatic extraction of web data records containing user-generated content. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 39–48, 2010.

[228] Ergin Soysal, Jingqi Wang, Min Jiang, Yonghui Wu, Serguei Pakhomov, Hongfang Liu, and Hua Xu. Clamp–a toolkit for efficiently building customized clinical natural language processing pipelines. *Journal of the American Medical Informatics Association*, 25(3):331–336, 2018.

[229] Lucia Specia, Stella Frank, Khalil Sima'An, and Desmond Elliott. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 543–553, 2016.

[230] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[231] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii. brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations Session at EACL 2012*, 2012.

[232] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[233] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

[234] Jingyuan Sun, Shaonan Wang, and Chengqing Zong. Memory, show the way: Memory based few shot word representation learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1435–1444, 2018.

[235] Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813, 2013.

[236] Javier Tamames. Text detective: a rule-based system for gene annotation in biomedical texts. *BMC Bioinformatics*, 6(Suppl 1):S10, 2005.

[237] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP-IJCNLP*, 2019.

[238] Patrick Tan, Yichao Zhou, Xinxin Huang, Giuseppe M Mazzeo, Chelsea Ju, Vincent Kyi, Brian Bleakley, Justin Wood, Peipei Ping, and Wei Wang. Aztec: A cloud-based computational platform to integrate biomedical resources. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 1365–1366. IEEE, 2017.

[239] Buzhou Tang, Yonghui Wu, Min Jiang, Yukun Chen, Joshua C Denny, and Hua Xu. A hybrid system for temporal information extraction from clinical text. *Journal of the American Medical Informatics Association*, 20(5):828–835, 2013.

[240] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

[241] The UniProt Consortium. Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, 45(D1):D158–D169, 2017.

[242] Julien Tourille, Olivier Ferret, Aurelie Neveol, and Xavier Tannier. Neural architecture for temporal relation extraction: a bi-lstm approach for detecting narrative containers. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 224–230, 2017.

[243] Yao-Hung Hubert Tsai and Ruslan Salakhutdinov. Improving one-shot learning through fusing side information. *arXiv preprint arXiv:1710.08347*, 2017.

[244] O. Uzuner, A. Bodnari, S. Shen, T. Forbush, J. Pestian, and B. R. South. Evaluating the state of the art in coreference resolution for electronic medical records. *Journal of the American Medical Informatics Association : JAMIA*, 19(5):786–791, Sep 2012.

[245] 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*, 18(5):552–556, 2011.

[246] Ankit Vadehra. Uwav at semeval-2017 task 7: Automated feature-based system for locating puns. In *SemEval-2017*, pages 449–452, 2017.

[247] Ali Moradi Vartouni, Saeed Sedighian Kashi, and Mohammad Teshnehlab. An anomaly detection method to detect web attacks using stacked auto-encoder. In *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, pages 131–134. IEEE, 2018.

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

[249] Olga Vechtomova. Uwaterloo at semeval-2017 task 7: Locating the pun using syntactic characteristics and corpus-based metrics. In *SemEval-2017*, pages 421–425, 2017.

[250] Patrick Verga, Emma Strubell, and Andrew McCallum. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 872–884, 2018.

[251] Robert Walraven. Emperical skewed gaussian, 2021.

[252] Hui Wan et al. An seis epidemic model with transport-related infection. *Journal of theoretical biology*, 247(3):507–524, 2007.

[253] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Multi-task feature learning for knowledge graph enhanced recommendation. In *The World Wide Web Conference*, pages 2000–2010, 2019.

[254] Wei Wang, Brian Bleakley, Chelsea Ju, Vincent Kyi, Patrick Tan, Howard Choi, Xinxin Huang, Yichao Zhou, Justin Wood, Ding Wang, et al. Aztec: A platform to render biomedical software findable, accessible, interoperable, and reusable. *arXiv preprint arXiv:1706.06087*, 2017.

[255] Xuan Wang, Xiangchen Song, Yingjun Guan, Bangzheng Li, and Jiawei Han. Comprehensive named entity recognition on cord-19 with distant or weak supervision. *arXiv preprint arXiv:2003.12218*, 2020.

[256] Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. Cross-type biomedical named entity recognition with deep multi-task learning. *CoRR*, abs/1801.09851, 2018.

[257] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

[258] Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, 41(W1):W518–W522, 2013.

[259] Sen Wu, Luke Hsiao, Xiao Cheng, Braden Hancock, Theodoros Rekatsinas, Philip Levis, and Christopher Ré. Fonduer: Knowledge base construction from richly formatted data. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1301–1316, 2018.

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

[261] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[262] Yan Xu, Yining Wang, Tianren Liu, Junichi Tsujii, and Eric I-Chao Chang. An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):849–858, 2013.

[263] Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. Tener: Adapting transformer encoder for name entity recognition. *arXiv preprint arXiv:1911.04474*, 2019.

[264] Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. Humor recognition and humor anchor extraction. In *EMNLP 2015*, pages 2367–2376, 2015.

[265] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[266] Keren Ye and Adriana Kovashka. ADVISE: symbolism and external knowledge for decoding advertisements. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, pages 868–886, 2018.

[267] Wenpeng Yin. Meta-learning for few-shot natural language processing: A survey. *arXiv preprint arXiv:2007.09604*, 2020.

[268] Seounmi Youn and Seunghyun Kim. Newsfeed native advertising on facebook: young millennials' knowledge, pet peeves, reactance and ad avoidance. *International Journal of Advertising*, 38(5):651–683, 2019.

[269] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages

2335–2344, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[270] Yijia Zhang and Zhiyong Lu. Exploring semi-supervised variational autoencoders for biomedical relation extraction. *Methods*, 166:112–119, 2019.

[271] Yichao Zhou, Wei-Ting Chen, Bowen Zhang, David Lee, J Harry Caufield, Kai-Wei Chang, Yizhou Sun, Peipei Ping, and Wei Wang. Create: Clinical report extraction and annotation technology. *arXiv preprint arXiv:2103.00562*, 2021.

[272] Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4904–4913, Hong Kong, China, November 2019. Association for Computational Linguistics.

[273] Yichao Zhou, Jyun-yu Jiang, Xiusi Chen, and Wei Wang. stayhome or marathon? social media enhanced pandemic surveillance on spatial-temporal dynamic graphs. CIKM '21. Association for Computing Machinery, 2021.

[274] Yichao Zhou, Jyun-Yu Jiang, Jieyu Zhao, Kai-Wei Chang, and Wei Wang. "the boating store had its best sail ever": Pronunciation-attentive contextualized pun recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 813–822, Online, July 2020. Association for Computational Linguistics.

[275] Yichao Zhou, Chelsea Ju, J Harry Caufield, Kevin Shih, Calvin Chen, Yizhou Sun, Kai-Wei Chang, Peipei Ping, and Wei Wang. Clinical named entity recognition using contextualized token representations. *arXiv preprint arXiv:2106.12608*, 2021.

[276] Yichao Zhou, Shaunak Mishra, Jelena Gligorijevic, Tarun Bhatia, and Narayan Bhamidipati. Understanding consumer journey using attention based recurrent neu-

ral networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3102–3111, 2019.

[277] Yichao Zhou, Shaunak Mishra, Manisha Verma, Narayan Bhamidipati, and Wei Wang. Recommending themes for ad creative design via visual-linguistic representations. In *Proceedings of The Web Conference 2020*, WWW '20, page 2521–2527, New York, NY, USA, 2020. Association for Computing Machinery.

[278] Yichao Zhou, Ying Sheng, Nguyen Vo, Nick Edmonds, and Sandeep Tata. Simplified dom trees for transferable attribute extraction from the web. *arXiv preprint arXiv:2101.02415*, 2021.

[279] Yichao Zhou, Yu Yan, Rujun Han, J. Harry Caufield, Kai-Wei Chang, Yizhou Sun, Peipei Ping, and Wei Wang. Clinical temporal relation extraction with probabilistic soft logic regularization and global inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14647–14655, May 2021.

[280] Difan Zou, Lingxiao Wang, Pan Xu, Jinghui Chen, Weitong Zhang, and Quanquan Gu. Epidemic model guided machine learning for covid-19 forecasts in the united states. *medRxiv*, 2020.

[281] Yanyan Zou and Wei Lu. Joint detection and location of english puns. In *NAACL 2019*, pages 2117–2123, 2019.