ABSTRACT

Title of Dissertation: DOCUMENT INFORMATION EXTRACTION, STRUCTURE UNDERSTANDING AND MANIPULATION

Puneet Mathur
Doctor of Philosophy, 2023

Dissertation Directed by: Prof. Dinesh Manocha
Department of Computer Science

Documents play an increasingly central role in human communications and workplace productivity. Every day, billions of documents are created, consumed, collaborated on, and edited. However, most such interactions are manual or rule-based semi-automated. Learning from semi-structured and unstructured documents is a crucial step in designing intelligent systems that can understand, interpret, and extract information contained in digital PDFs, forms, receipts, contracts, infographics, etc. Our work tries to solve three major problems in the domain of information extraction from real-world multimodal (text+images+layout) documents: (1) multi-hop reasoning between concepts and entities spanning several paragraphs; (2) semi-structured layout extraction in documents consisting of thousands of text tokens and embedded images arranged in specific layouts; (3) hierarchical document representations and the need to transcend content lengths beyond a fixed window for effective semantic reasoning. Our research broadly binds together the semantic (document-level information extraction) and structural (document image analysis) aspects of document intelligence to advance user productivity.

The first part of the research addresses issues related to information extraction from characteristically long-range documents that consist of multiple paragraphs and require long-range contextualization. We propose augmenting the capabilities of the Transformer-based methods with graph neural networks to capture local-level context as well as long-range global information to solve document-level information extraction tasks. In this aspect, we first solve the task of document-level temporal relation extraction by leveraging rhetorical discourse features, temporal arguments, and syntactic features through a Gated Relational-GCN model to extend the capability of Transformer architecture for discourse-level modeling. Next, we propose DocTime, a novel temporal dependency graph parsing method that utilizes structural, syntactic, and semantic relations to learn dependency structures over time expressions and event entities in text documents to capture long-range interdependencies. We also show how the temporal dependency graphs can be incorporated into the self-attention layer of Transformer models to improve the downstream tasks of temporal questions answering and temporal NLI. Finally, we present DocInfer - a novel, end-to-end Document-level Natural Language Inference model that builds a hierarchical document graph, performs paragraph pruning, and optimally selects evidence sentences to identify the most important context sentences for a given hypothesis. Our evidence selection mechanism allows it to transcend the input length limitation of modern BERT-like Transformer models while presenting the entire evidence together for inferential reasoning that helps it to reason on large documents where the evidence may be fragmented and located arbitrarily far apart.

The second part of the research covers novel approaches for understanding, manipulation, and downstream applications of spatial structures extracted from digital documents. We first propose LayerDoc to extract the hierarchical layout structure in visually rich documents by

leveraging visual features, textual semantics, and spatial coordinates along with constraint inference in a bottom-up layer-wise fashion. Next, we propose DocEditor, a Transformer-based localization-aware multimodal (textual, spatial, and visual) model that performs the novel task of language-guided document editing based on user text prompts. Further, we investigated methods for building text-to-speech systems for semi-structured documents.

Finally, we will explore two applications of long-context document-level reasoning: (i) user-personalized speech recognition systems for improved next-word prediction in specific domains by utilizing retrieval augmentation techniques for ASR Language Models; (ii) Transformer-based methods to utilize multimodal information from long-form financial conference calls (document-level transcripts, audio-visual recordings, and tabular information) for improved financial time series prediction tasks.

DOCUMENT INFORMATION EXTRACTION, STRUCTURE UNDERSTANDING AND MANIPULATION

by

Puneet Mathur

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Dr. Dinesh Manocha, Chairperson
Dr. Jonathan Lazar, Dean's Representative
Dr. Ming Lin
Dr. Rachel Rudinger
Dr. Rajiv Jain (Adobe Research)

Dedicated To

To my father Pankaj Mathur, my mother Deepa Mathur, and my sister Archita Mathur whose lifelong sacrifices and unwavering belief in me have been my foundation; to all my teachers and mentors who have illuminated my path; my fiancée Kanika for her faith in my abilities, and to the eternal divine intelligence, which I seek to comprehend more deeply through my work.

Acknowledgments

The Ph.D. journey has been the most fulfilling part of my education. It started when my advisor, Prof. Dinesh Manocha saw a spark in me and motivated me to embark on this journey. He believed in me when I doubted myself. This journey started amidst the chaos of COVID-19. Prof. Dinesh provided me with intellectual and emotional support when the going was tough and encouraged me to be my best version in favorable times. His valuable guidance and inspiration helped me complete this long, intense Ph.D. journey. The tireless late nights we spent working on research, revising our papers, and responding to peer reviews, are as memorable now as they were stimulating at the time. His unwavering support in collaborations with the broader research community was a cornerstone of my Ph.D. journey. I am grateful to my Adobe collaborators, especially Rajiv Jain, Vlad Morariu, and Franck Dernoncourt for their constant mentorship and guidance. I sincerely thank my entire thesis committee, counting in Profs. Ming Lin, Jonathan Lazar, Rachel Rudinger, and Dr. Rajiv Jain for taking the time to read and evaluate my thesis and providing their valuable feedback throughout.

The two years I spent interning and collaborating with Adobe Research were the most memorable periods of this journey. I want to thank Rajiv and Vlad for taking this newcomer under their wings, helping me navigate the nuances of industrial research, and broadening my thought process to look for impactful problems. My internship experiences at Adobe formed the core of my thesis for which I will always be indebted to my manager Tong Sun. Franck is one

of the best mentors I have had as we successfully organized workshops at top-tier conferences and filed patents that helped me become a more effective researcher. His support for me as an international student remains invaluable and cherished. I thank Juixiang, Nedim, Ani, Varun, Chris, Ashutosh, and Verena for our chats about document AI and Ph.D. research.

I thank Prof. Sanghamitra Dutta for her support in Organizing workshops on AI and Finance gave me a creative outlet for my passion for finance. My journey in AI traces back to my undergraduate university where I met Ramit Sawhney and that single collaboration has been so instrumental in my life. I am grateful to him for being a wonderful friend. In Bharatiya traditions, a teacher is regarded as a guru who transforms one's life by setting them on a path of enlightenment. For me, this person was Prof. Rajiv Ratn Shah, my undergrad research advisor at IIIT-Delhi. His guidance and mentorship made me capable enough to dive into serious AI research and inclined me toward applied research.

No journey is truly complete without the companions who add value to it. I've been fortunate to have numerous supportive friends from GAMMA and elsewhere within and outside the University of Maryland. I want to thank Rohan who I came in contact with even before thinking about graduate studies. For a naive and confused undergrad afraid to take on the difficult journey of graduate studies in the US, he helped me give shape to my dreams by encouraging me to apply to UMD which eventually became my second home. His guidance on navigating grad school was critical at every step of this path. Saurya was my first friend in College Park who has been a constant source of camaraderie through my journey in the United States. Pulkit and Shantam were my first roommates in the US and we learned so much about managing life and were family-like friends away in this foreign land. I want to thank Shishira, Sai, Vasu, Naman, Noor, Shlok, and Pooja for the fond memories we created during our trips and through our

courses and studies. Sharing our highs and lows created deep bonds that I will cherish for a lifetime. There are very few people who are as selfless in helping juniors as Uttaran. I convey heartfelt gratitude to him for patiently answering all my questions and being my de-facto source of guidance in finding my way around as an international student in the US. Trisha helped me get started with my first projects in Gamma and set a high benchmark for us juniors to look up to. Vishnu, Sanjoy, Niall, Geonsun, Adarsh, Senthil, Utsav, and Kasun were always very helpful and made working at the Gamma Lab more fun.

My internships at Verisk, Meta, and Microsoft were instrumental parts of my journey. I would like to thank Manish Shrivastava, Gautam Kunapuli, Riyaz Bhat, Maneesh Singh, Zhe Liu, Ke Li, Yingyi Ma, Gil Keren, Zeeshan Ahmed, Xuedong Zhang, John Corring, Dinei Florencio for their valuable guidance and exhilarating discussions on post-Ph.D. careers that have helped me reach where I am today and will continue to help me in my future endeavors. I have had some great experiences collaborating with all my co-authors, discussing ideas, methods, and results feeding back into new ideas.

Wrapping all the parts of my Ph.D. journey together, I want to immensely thank my mom and dad who have sacrificed throughout their lives to enable me to take on this difficult yet rewarding journey. They believed in me when I was short on inspiration, held me back when I faced challenges and cared for me in innumerable ways. My sister has been my greatest source of joy and happiness. I am grateful to my fiancée Kanika who has been my biggest strength in my lows and my greatest cheerleader in my highs. I am indebted to her sacrifices of staying away from me for such an extended period of time to see me succeed and thank her for endlessly encouraging me to keep trying my best.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction and Overview

The goal of Document Artificial Intelligence is to develop systems and solutions that can understand, interpret, and extract information contained in semi-structured documents such as digital PDFs, forms, receipts, contracts, infographics, etc. In today's digitally connected world, documents play an increasingly central role in human communications and workplace productivity. Every day, billions of documents are created, consumed, collaborated on, and edited, however majority of such interactions are manual or rule-based semi-automated. The real challenge lies in enabling structural and semantic understanding of documents so that users can easily extract relevant information in an automated fashion without the loss of critical information. For example, the extraction of important dates mentioned in a contract and aligning the related events on a timeline may help improve the efficiency of lawyers in their day-to-day jobs. Similarly,

extracting itemized pairs from scanned receipts can help make accounting easier. However, such applications are challenged by the loss of structural and semantic understanding of documents through heuristic and semi-automated methods without the loss of critical information. Extracting information from real-world documents remains a challenging task as documents tend to be multimodal (text+images+layout), come in a variety of different layouts, encode multi-hop information flow between several constituents, and require input contextualization beyond a fixed aperture (context window) for effective applications. For instance, text documents may be composed of several paragraphs, often running into multiple pages; digital PDFs can have thousands of text tokens and embedded images arranged in specific layouts; financial documents may contain numerous tables, charts, and visualizations; legal documents may encode a dense hierarchical ordering of clauses and terminologies; narrative documents may cross-reference multiple concepts and entities across paragraphs, pages, and collections.

Effectively, our work tries to solve three major problems in the domain of information extraction from real-world multimodal (text+images+layout) documents: (1) multi-hop reasoning between concepts and entities spanning several paragraphs; (2) semi-structured layout extraction in documents consisting of thousands of text tokens and embedded images arranged in specific layouts; (3) hierarchical document representations and the need to transcend content lengths beyond a fixed window for effective semantic reasoning. Our research broadly binds together the semantic (document-level information extraction) and structural (document image analysis) aspects of document intelligence to advance user productivity tools. Semantic understanding involves designing and training methods for extracting information spread across the constituent passages and sentences of a long document. We overcome this limitation by leveraging the multi-hop qualities of graph neural networks which helps the Transformer encoder models to

transcend the input length limitations to reason over long document text. These advances help us improve upon the pertinent tasks of document-level temporal relation extraction, temporal dependency parsing, and natural language inference.

In this aspect, my first work explores the problem of temporal relation extraction (TempRel) which involves determining the temporal order between two events in a text (Pustejovsky et al. 2003a). Ordering events in time is useful for automatically inferring their relative occurrence and generating precise time anchoring for each event. Prior work focused on extracting temporal relations between event pairs present in the same sentence or adjacent sentences, mostly ignoring document-level pairs requiring long-range dependencies and multi-hop reasoning at the document level. Our main contribution to this work is the TIMERS model for document-level temporal relation extraction. TIMERS uses discourse features, temporal arguments, and structural and syntactic dependency parse connections to leverage long-range inter-sentential relationships in a text document to extend existing contextual BERT embeddings with structural and syntactic dependency parse connections. These rhetorical, syntactic, and temporal features are learned through relational Graph Convolutional Networks (R-GCN).

Further extending this work, we explore the task of temporal dependency parsing that aims to infer a graph of temporal relations rather than relying on densely annotated pairs of events in long documents. We introduce `DocTime` - a state-of-the-art temporal dependency parser that parses document-level text to produce temporal dependency graphs. Unlike previous approaches using contextual features such as BERT (Ross, Cai, and Min 2020), our model utilizes a graph network and a novel path prediction loss to reason over long-range multi-hop dependencies while maintaining global consistency of temporal ordering of inter-dependent events. We also propose `Time-Transformer`, a framework to incorporate temporal dependency graphs

into existing transformer-based architectures without retraining from scratch.

Next, we explored the task of document-level natural language inference where the premises are in the document granularity, whereas the hypotheses can vary in length from single sentences to passages with hundreds of words (Yin, Radev, and Xiong 2021). This textual reasoning task seeks to classify a presented hypothesis as entailed by, contradictory to or neutral to a premise (Dagan et al. 2010). Prior NLI datasets and studies have focused on sentence-level inference where both the premises and hypotheses are single sentences. Document-level NLI challenges modern approaches due to the limited input bottleneck of modern Transformer models (e.g. BERT model (Devlin et al. 2018) can only encode 512 input sub-tokens due to its quadratic self-attention complexity). Consequently, evidence in the document premise relevant to the hypothesis can potentially be distributed in several textual spans located arbitrarily far away from each other in long documents, and may not be simultaneously available to draw inference. We address the above challenge with a reasonable assumption that the portion of the premise (the ground truth evidence) necessary and sufficient for inference can fit entirely into the length limit of language model for effective representation learning, and hence propose DocInfer – a novel architecture that simultaneously performs successive optimal evidence selection and textual inference on large documents. It utilizes a novel graph representation of the document encoding structural, topical, concept and entity-based relationships. It performs subgraph pooling and asynchronous graph updates to provide a pruned, hypothesis-relevant and richer sub-document graph representation and uses a reinforcement-learning based subset selection module to provide the contextually-relevant evidences for inference.

Structural understanding of multimodal documents involves parsing the hierarchical spatial organization of document constituents to extract the text, layout boundaries, and visual attributes

of scanned or digital documents. Digital documents often contain images and scanned text. Parsing such visually rich documents is a core task for workflow automation, but it remains challenging since most documents do not encode explicit layout information, e.g., how characters and words are grouped into boxes and ordered into larger semantic entities. Current state-of-the-art layout extraction methods are challenged by such documents as they rely on word sequences to have correct reading order and do not exploit their hierarchical structure. To address these challenges, we propose the LayerDoc model for extracting hierarchical document layout in a layer-wise fashion, recursively grouping smaller spatial regions into larger, semantic elements. We are the first to formulate nested document hierarchy extraction using transformers. In addition, we propose a multimodal contextual encoder that maximizes the use of context by simultaneously modeling all possible parent-child pairs in a layer. For element type classification and semantic grouping, this leads to a relative improvement of 10-15% across several metrics. We also demonstrate how our extracted nested hierarchical document structure can improve the inferred token reading order and semantic word grouping by 8-12%.

We further look at how information about these extracted multimodal structure hierarchies can help us automatically edit documents based on user prompts. Digital documents are used extensively to help people improve business productivity (drafting contract agreements, presentation decks, letterheads, invoices, resumes, form filling) and communicate with customers through online advertisements, social media posts, flyers, posters, billboards, web and mobile app prototypes, etc. However, modern document editing tools require a skilled professional to work on a large screen. Challenges emerge when complex editing operations require multiple different functionalities wrapped within the editing tools for text and image region placement, grouping, spatial alignment, replacement, resizing, splitting, merging, and special effects. To

make editing tools accessible to increasingly novice users, we investigate intelligent document assistant systems that can make or suggest edits based on a user's natural language request. Such a system should be able to understand the user's ambiguous requests and contextualize them to the visual cues and textual content found in a document image to edit localized unstructured text and structured layouts. To this end, we propose a new task of language-guided localized document editing, where the user provides a document and an open vocabulary editing request, and the intelligent system produces a command that can be used to automate edits in real-world document editing software. We propose the DocEdit dataset which provides natural language edit requests on PDFs and design template documents. Each edit request is mapped to an executable command that can be used to automatically apply edits in real-world document editing software. We also propose the DocEditor model, a neural architecture to generate the executable computer command and ground the region of interest bounding box by solving inherent challenges in automated document editing, namely - (a) interpreting and localizing structured components and their relative positioning in the document; (b) matching document text tokens in a text-rich document formatted in varied spatial layouts (checkboxes, choice groups, text fields, columns, rows), (c) visually understanding the objects as per the user description.

We also explore several applications of document information mining and long-context multimodal content understanding such as (i) building text-to-speech systems for semi-structured documents, (ii) user-personalized speech recognition systems for improved next-word prediction in specific domains by utilizing retrieval augmentation techniques for ASR Language Models; (iii) Transformer-based methods to utilize multimodal information from long-form financial conference calls (document-level transcripts, audio-visual recordings, and tabular information) for improved financial time series prediction tasks.

The first application shows a use case of ordered document parsing through the task of synthesizing coherent speech from the text in documents which remains a challenging problem due to (1) the long sequence length of input text, and (2) the lack of correct reading order provided by off-the-shelf Optical Character Recognition (OCR) engines that tend to arrange all recognized tokens in a top-to-bottom and left-to-right manner, and disregard the layout of the long-form text (Clausner, Pletschacher, and Antonacopoulos 2013). Current TTS systems assume that the reading order sequence of input text tokens is correct. However, the reading order itself depends on the structure of the document and is not known apriori. In fact, current OCR systems cannot infer this correctly from complex spatial documents. Towards this end, we propose the task of document-level layout-informed text-to-speech synthesis that aims to generate human-level speech corresponding to the correct reading order of the text present in a semi-structured document. We present DocLayoutTTS, a neural baseline architecture that simultaneously learns text reordering, newline prediction, and mel-spectrogram prediction for synthesizing speech from documents in our proposed dataset in a multi-task fashion. Our proposed model uses curriculum learning to learn increasingly long document-level speech synthesis.

For the second application, combine publicly available earnings calls in MAEC (Li et al. 2020b)) and M&A calls (Sawhney et al. 2021a) datasets with tabular data extracted from SEC-EDGAR 10-Q and 10-K company-filing documents. We utilize tabular information from financial semi-structured documents with existing textual and audio modalities to show improvement in stock volatility and price movement prediction tasks across several baseline and state-of-the-art models. Induction of tabular data also reduces induced gender bias due to audio modality in the financial prediction models and demonstrates the usefulness of tabular data extracted from semi-structured financial documents as an alternative to audio modality for reducing gender

bias in audio-based neural networks, without significant performance degradation.

The third application deals with long-form content understanding for capturing rare word patterns associated with specific users/documents/domains, with a goal to improve domain-specific language modeling and personalized ASR using past data. To fulfill this goal, we introduce DomainRAG: Domain-distributed span-aggregated K-nearest N-gram retrieval augmentation to improve language modeling for Automatic Speech Recognition (ASR) personalization. DomainRAG leverages contextually similar n-gram word frequencies for recognizing rare word patterns associated with unseen domains. It aggregates the next-word probability distribution based on the relative importance of different domains to the input query. To achieve this, we propose the Span Aggregated Group-Contrastive Neural (SCAN) retriever that learns to rank external domains/users by utilizing a group-wise contrastive span loss that pulls together span representations belonging to the same group while pushing away spans from unrelated groups in the semantic space. We propose the ASAP benchmark for ASR LM personalization that consists of three user-specific speech-to-text tasks for meetings, TED talks, and financial earnings calls. Extensive experiments show that DomainRAG significantly outperforms strong baselines on popular Wikitext-103, UserLibri, and our ASAP dataset. We further demonstrate the usefulness of the SCAN retriever for improving user-personalized text generation and classification by retrieving relevant context for zero-shot prompting and few-shot fine-tuning of LLMs on the LAMP benchmark.

CHAPTER 2

TIMERS: Document-level Temporal Relation Extraction

Abstract

We present **TIMERS** - a **TIME**, **R**hetorical and **S**yntactic-aware model for document-level temporal relation classification. Our proposed method leverages rhetorical discourse features and temporal arguments from semantic role labels, in addition to traditional local syntactic features, trained through a Gated Relational-GCN. Extensive experiments show that the proposed model outperforms previous methods by 5-18% on the TDDiscourse, TimeBank-Dense, and MATRES datasets due to our discourse-level modeling.

2.1 Introduction

Temporal relation extraction (TempRel) is a challenging task that involves determining the temporal order between two events in a text (Pustejovsky et al. 2003a). Understanding the temporal ordering of events in a document plays a key role in downstream tasks such as timeline creation (Leeuwenberg and Moens 2018), time-aware summarization (Noh et al. 2020), temporal question-answering (Ning et al. 2020a), and temporal information extraction (Leeuwenberg and Moens 2019).

Prior work focuses on extracting temporal relations between event pairs (a.k.a., *TLINKS*) present in the same sentence (*Intra-sentence TLINKS*) or adjacent sentences (*Inter-sentence TLINKS*), mostly ignoring document-level pairs (*Cross-document TLINKS*) (Reimers, Dehghani, and Gurevych 2016). Past works have used RNN (Cheng and Miyao 2017; Meng, Rumshisky, and Romanov 2017; Goyal and Durrett 2019; Ning, Subramanian, and Roth 2019a; Han et al. 2019a; Han, Ning, and Peng 2019; Han et al. 2019b; Han, Zhou, and Peng 2020) and Transformer networks (Ballesteros et al. 2020; Zhao, Lin, and Durrett 2020b) for encoding a few sentences or a short paragraph but do not capture long-range dependencies and multi-hop reasoning at the document-level. This shortcoming is shown in the TDDiscourse dataset (Naik, Breitfeller, and Rosé 2019), which was designed to highlight global discourse-level challenges, e.g., multi-hop chain reasoning, future or hypothetical events, and reasoning requiring world knowledge.

We propose **TIMERS** - a **TIME**, **R**hetorical, and **S**yntactic-aware model for document-level temporal relation extraction. TIMERS uses discourse features in the form of connections from Rhetorical Structure Theory (RST) parsers (Bhatia, Ji, and Eisenstein 2015) to leverage long-range inter-sentential relationships. It also extends existing contextual embeddings with

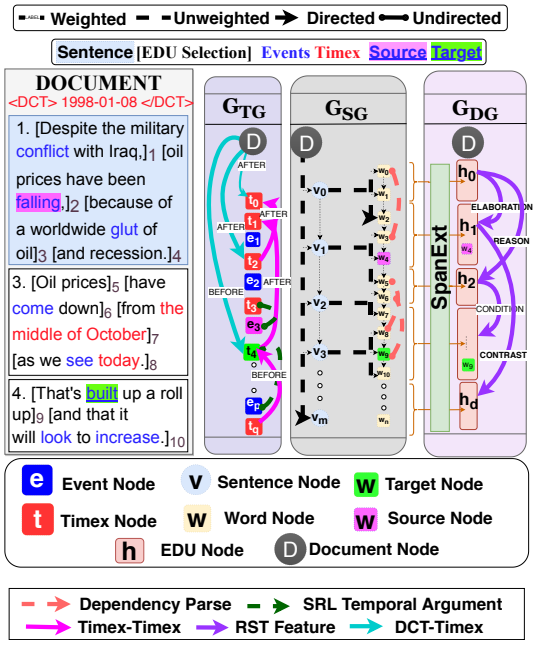


Figure 2.1: Three graphs are created from the input document. Time-aware Graph (G_{TG}): DCT-Timex associations, Timex-Timex associations, and Temporal Argument connections from semantic role labels; Syntactic-aware Graph (G_{SG}): structural and syntactic connections; and Rhetoric-aware Graph (G_{DG}): rhetorical relations between EDU's (h_i).

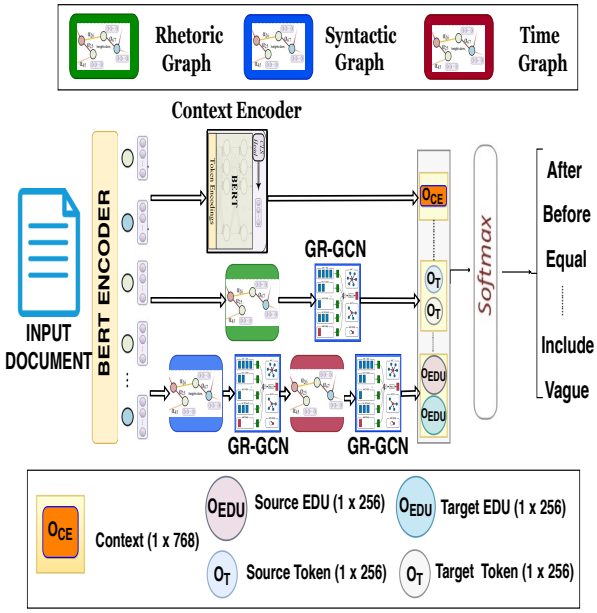


Figure 2.2: TIMERS learns rhetorical, syntactic, and temporal features through a Gated Relational-Graph Convolutional Networks (GR-GCN). The output of G_{SG} forms the input of G_{TG}. The output corresponding to the source and target nodes learned by G_{TG} (O_T) and G_{DG} (O_{EDU}) are concatenated with the output of the BERT-based context encoder (O_{CE}), which forms the final output h_G that passes through the Softmax layer to predict the temporal relation.

structural and syntactic dependency parse connections. Lastly, it uses timex-timex relations, *dct* (document creation date)-timex relations, and temporal arguments obtained via sentence-level semantic role labeling. These rhetorical, syntactic, and temporal features are learned through a modified version of Relational Graph Convolutional Networks (R-GCN) with a gating mechanism (GR-GCN) (Schlichtkrull et al. 2018), which learns highly relational data relationships in densely-connected graph networks.

Our **main contribution** is a document-level model that incorporates these three features to improve temporal relationship extraction. We obtain state-of-the-art performance across three datasets with **5-18% relative improvement**, showing improvement for events that require chain reasoning, causal prerequisite links, and future events.

2.2 Methodology

Let document D be defined as a sequence of n tokens $w_i \in W = \{w_1, \cdots, w_n\}$. The entire document is a list of m sentences $V = [v_1, \cdots, v_m]$. Each document has a set of p events $E = \{e_1, \cdots, e_p\}$ and q timexes $T = \{t_1, \cdots, t_q\}$, where $p, q \leq n$. The creation date of the document is represented by timestamp t_{DCT}. We denote the source and target events by e_s and e_t, respectively. The task is to identify the temporal relation $y \in R$ between the source and target event in a multi-class classification setup, where R is the set of all possible temporal links (*TLINKs*).

To solve this task, our model (Fig.2.1) builds the **TIMERS**-graph, which consists of a Syntactic Graph (Sec.2.2.1), a Time Graph (Sec. 2.2.2), and a Rhetorical Graph (Sec.2.2.3). Each graph is learned through GR-GCN to extract the embeddings used for temporal relation extraction

(Fig.2.2, Sec.2.2.4).

2.2.1 Syntactic-Aware Graph

The syntactic graph captures the document structure and word dependency. Our syntactic-aware graph ($\mathcal{G}_{SG}$) is made of separate nodes to represent the document D, each of its inherent sentences $v_i \in V$, and all the constituent words $w_i \in W$ of each sentence. The edges of the Syntactic Graph encode five relations: **(1) Document-Sentence Affiliation** and **(2) Sentence-Word Affiliation** model the hierarchical structure of the document through a directed edge from the document node to each sentence node and from a sentence node to each word in the sentence. **(3) Sentence-Sentence Adjacency** and **(4) Word-Word Adjacency** to preserve sequential ordering for consecutive sentence and word nodes. **(5) Word-Word Dependency** encodes the syntactical nature of the word-level relationships by adding an undirected edge between two word nodes if they share a parent-child relationship in the sentence-level dependency tree.

We use BERT to encode each w_i and obtain sentence embeddings $v_i^{'}$ by averaging the second-to-last hidden layer of BERT for each token. The document vector embedding $D_i^{'}$ was calculated as the average of all sentence embedding ($D_i^{'} = \sum_{i=0}^{m} v_i^{'}$).

2.2.2 Time-Aware Graph

When events are anchored to a specific time, it becomes easier to infer event relationships from their associated date and time. The time-aware graph ($\mathcal{G}_{TG}$) exploits this intuition and propagates relational information among events, timexes, and the Document Creation Time (DCT). The document node D is the node corresponding to the document creation date while the timexes t_i and events e_i are characterized by their corresponding word nodes in the Syntactic Graph.

We design three types of edge connections: **(1) *DCT*-Timex Association:** exploit the ordering of timexes with respect to the document creation time through directed weighted edges from *DCT* to timexes. **(2) Timex-Timex Association:** capture inherent non-local timeline ordering between timex pairs by a directed weighted edge. **(3) Predicate-Temporal Argument:** anchor local temporal relations at the sentence level by connecting each event verb predicate to its temporal argument with a directed edge. The connections formed between temporal entities help navigate information from the source event to the target event while exploring interactions with other events, timexes, *dct*, and temporal arguments.

We calculate timestamps for timexes and the *DCT* from the annotated TimeML format of input documents. The weight of the *DCT*-timex and timex-timex edges is determined based on the temporal order of the entities {*After*, *Before*, *Simultaneous*, *None*}. We added *None* as a relation when one of the timestamps cannot be anchored in time.

2.2.3 Rhetorical-Aware Graph

We use discourse features based on Rhetorical Structure Theory (RST) (Mann and Thompson 1988) to leverage long-range inter-dependencies through a discourse tree. The rhetorical discourse tree of a document contains nodes of phrases, where each phrase (a.k.a, Elementary Discourse Unit or EDU) is contiguous, adjacent and non-overlapping. The interdependencies among EDUs are represented by conventional rhetorical relations (Mann 1987), e.g. *Elaboration, Span, Condition, Attribution.* Prior work showed discourse features in the form of RST connections help leverage long-range document-level interactions between phrase units (Bhatia, Ji, and Eisenstein 2015) and identify background-foreground events (Aldawsari et al. 2020).

Elementary Discourse Unit (EDU), a sub-sentence phrase unit, is the minimal selection unit

for discourse segmentation of a document. We generate the document vector representations at EDU-level $h_i \in H = \{h_1, \cdots, h_d\}$ via the Self-Attentive Span Extractor (SpanExt) from (Lee et al. 2017) over the BERT token embeddings. We use the converted dependency version of the tree to build the Rhetorical-aware graph ($\mathcal{G}_{DG}$) by treating every discourse dependency from the i-th EDU to the j-th EDU as a directed edge weighted by the type of the rhetorical relation.

2.2.4 Temporal Relation Extraction

Each graph is instantiated as a gated variant of Relational Graph Convolutional Networks (R-GCN) (Schlichtkrull et al. 2018), which we term as Gated Relational Graph Convolution Network (GR-GCN). GR-GCN propagates messages among the nodes to obtain a learned node representation and is inspired by (Zhang et al. 2020b). Fig. 2.2 shows how the learned representations obtained from the syntactic-aware graph forms the input to the time-aware graph. For the time-aware graphs, the learned representations of nodes corresponding to the source event e_s and target event e_t are extracted (O_T). In the case of the rhetorical graphs, the span representations of the EDU span nodes corresponding to the source event (h_e) and target event (h_s) are extracted (O_{EDU}).

The output corresponding to the source and target nodes learnt by G_{TG} (O_T) and G_{DG} (O_{EDU}) are concatenated with output of BERT based context encoder (O_{CE}) (similar to BERT encoding in (Zhao, Lin, and Durrett 2020a)): $z_G = \text{ReLU}(W[O_T; O_{EDU}; O_{CE}] + b)$. This is followed by a Softmax layer to predict temporal relations.

Dataset	Train	Validation	Test	Labels
TDDMan (Naik, Breitfeller, and Rosé 2019)	4000	650	1500	a, b, s, i, ii
TDDAuto (Naik, Breitfeller, and Rosé 2019)	32609	1435	4258	a, b, s, i, ii
MATRES (Ning, Wu, and Roth 2018a) ##	231	25	20	e,a,b,v
TimeBank-Dense (Cassidy et al. 2014a)	4032	629	1427	a, b, s, i, ii, v

Table 2.1: Train/Val/Test data distribution for TDDMan, TDDAuto, MATRES, and TimeBank-Dense; a: After, b: Before, s: Simultaneous, i: Includes, ii: Is_included, v: Vague, e: Equal. (## (Ning, Subramanian, and Roth 2019a) use TimeBank and Aquaint for training, Platinum for test; 20% of train as validation)

Corpus	Model	F1
TB-Dense	(Vashishtha, Durme, and White 2019)	56.6
	EventPlus (Ma et al. 2021)	64.5
	CTRL-PG (Zhou et al. 2020)	65.2
	DEER (Han, Ren, and Peng 2020)	66.8
	TIMERS (ours)	**67.8**
MATRES	CogCompTime (Ning et al. 2018)	66.6
	(Goyal and Durrett 2019)	68.61
	BiLSTM+MAP (Han, Ning, and Peng 2019)	75.5
	EventPlus (Ma et al. 2021)	75.5
	(Wang et al. 2020a)	78.8
	DEER (Han, Ren, and Peng 2020)	79.3
	(Zhao, Lin, and Durrett 2020a)	79.6
	SMTL (Ballesteros et al. 2020)	81.6
	TIMERS (ours)	**82.3**

Table 2.2: Comparison of TIMERS with recent state-of-the-art models on TimeBank-Dense and MATRES dataset. TIMERS outperforms all recent top-performing systems.

	System	TDDMan			TDDAuto			MATRES			TB-Dense		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baselines	Majority	37.8	36.3	37.1	34.2	32.3	33.2	50.7	50.7	50.7	40.5	40.5	40.5
	CAEVO (Chambers et al. 2014a)	32.3	10.7	16.1	61.1	32.6	42.5	-	-	-	49.9	46.6	48.2
	SP (Ning, Feng, and Roth 2017)	22.7	22.7	22.7	43.2	43.2	43.2	66.0	72.3	69.0	37.7	37.8	37.7
	SP+ILP (Ning, Feng, and Roth 2017)	23.9	23.8	23.8	46.4	45.9	46.1	71.3	82.1	76.3	58.4	58.4	58.4
	BiLSTM (Cheng and Miyao 2017)	24.9	23.8	24.3	55.7	48.3	51.8	59.5	59.5	59.5	**63.9**	38.9	48.4
	BERT-base Transformer	36.5	37.1	37.5	62.0	61.7	62.3	65.6	78.1	77.2	59.7	60.7	62.2
	RoBERTa-base	35.7	36.5	37.1	60.6	62.7	61.6	77.3	79.0	78.9	58.1	57.6	61.9
	TIMERS (ours)	**43.7***	**46.7***	**45.5***	**64.3***	**72.7***	**71.1***	**81.1***	**84.6***	**82.3***	48.1	**65.2***	**67.8**
Ablation	TIMERS w\o Context Encoder	29.7	35.5	33.7	49.8	52.5	51.6	61.2	69.6	68.6	43.8	54.5	50.6
	TIMERS w\o $\mathcal{G}_{DG}$	39.6	39.6	41.8	61.7	66.8	65.4	71.8	79.1	79.7	51.4	63.0	63.3
	TIMERS w\o $\mathcal{G}_{SG}$	38.5	42.6	42.3	63.3	69.5	68.9	71.6	78.5	78.2	51.1	62.1	62.8
	TIMERS w\o $\mathcal{G}_{TG}$	37.5	39.8	39.5	58.7	68.3	67.1	72.8	78.5	77.7	50.5	62.9	61.8

Table 2.3: Results comparing the performance of TIMERS with baselines and ablative components on TDDMan, TDDAuto, MATRES and TimeBank-Dense datasets. We adopt the BERT and RoBERTa implementation from (Ballesteros et al. 2020). * indicates statistical significance over BERT Transformer ($p \leq 0.005$) under Wilcoxon's Signed Rank test. Darker green represents better F1 performance in ablation studies. Bold denotes the best-performing model. TIMERS improves substantially over all datasets. The ablation shows that context, discourse ($\mathcal{G}_{DG}$), and time-aware ($\mathcal{G}_{TG}$) graph encoders prove to be most beneficial.

2.3 Experiments

2.3.1 Data

We train and test our proposed model using the TDDMan and TDDAuto subsets of the TDDiscourse corpus (Naik, Breitfeller, and Rosé 2019), which was designed to explicitly focus on global discourse-level temporal ordering. We also train and evaluate our method on the MATRES and TimeBank-Dense datasets, both of which primarily consist of local TLINKs that occur in either the same or adjacent sentences. Table 2.1 reports the data statistics and label distributions. (Naik, Breitfeller, and Rosé 2019) shows the distribution of the distance between event-pairs for all TLINKs in the TDD test set and explains that nearly 53% TLINKs in the TDD dataset comprise of event pairs that are more than 5 sentences apart. Like (Cheng and Miyao 2017), we report results on non-vague labels of TimeBank-Dense. MATRES has no standard validation set. Hence, we follow the split used in (Ning, Subramanian, and Roth 2019a).

2.3.2 Experimental Settings

Token Encoding:The word-level token representations are obtained by summing the corresponding BERT embeddings from the last 4 layers of pre-trained BERT-base encoder. **Syntactic Dependency Parser**: The dependency parse tree of individual sentences is obtained via SpaCy[1] to form word-word dependency connections in the syntactic-aware graph. **Semantic Role Labeller**: We extract semantic role labels using AllenNLP's SRL parser[2] that internally uses SRL-BERT (Shi and Lin 2019) to obtain the temporal arguments corresponding to each verb event. **Timex Normalization**: Timex phrases are treated as a single unit for the purpose of graph construction by average pooling their BERT tokenized representations. Microsoft Recognizers-Text[3] is employed to normalize timexes and DCT date-time values. The normalized timex expressions are compared through Allen's interval algebra, where each timex has a start and an endpoint. The comparison is then made on the basis of the endpoints of the timexes, forming an edge going from earlier to later ending timex. **RST Discourse Parser**: We used the shift-reduce discourse parser proposed by (Ji and Eisenstein 2014) to build the discourse tree [4], which is post-processed using *discoursegraphs* library[5] (Neumann 2015) to build the rhetorical dependencies graph.

[1]https://spacy.io/
[2]https://demo.allennlp.org/semantic-role-labeling
[3]https://github.com/microsoft/Recognizers-Text
[4]Implementation used: https://github.com/jiyfeng/DPLP
[5]https://pypi.org/project/discoursegraphs/

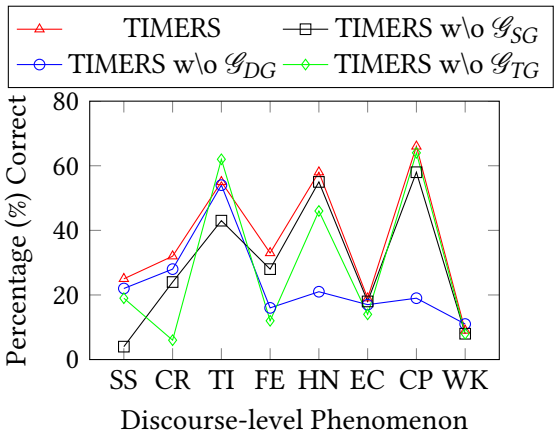


Figure 2.3: Error analysis on manually annotated discourse-level phenomena in the test set of TDDMan. SS: SingleSent, CR: Chain Reasoning, TI: Tense Indicator, FE: Future Events, HN: Hypothetical/Negated, EC: Event Coreference, CP: Causal/Prereq, WK: World Knowledge. TIMERS handles CR and CP phenomena but struggles on EC and WK.

2.3.3 Results

Table 6.2 compares our work to the baseline methods reported on the TDDMan, TDDAuto, MATRES, and TimeBank-Dense datasets. We also include results for BERT-based Transformer (Devlin et al. 2019a) and RoBERTa (Liu et al. 2019a) following (Ballesteros et al. 2020). To prevent truncation or memory errors otherwise caused by multi-sentence spans, we concatenate only sentences containing source and events as input to Transformer baselines. These methods outperform the existing reported results and provide strong benchmarks but still perform similarly to a majority class baseline for the TDDMan dataset. Our model shows a significant gain of 8.0 F1 and 8.8 F1 over the BERT baseline on the TDDMan and TDDAuto datasets. Table 3.6 compares TIMERS to additional rigorous state-of-the-art methods for TimeBank-Dense and MATRES. TIMERS achieves state-of-the-art performance on all four datasets, showing that it successfully handles intra-sentence, inter-sentence, and cross-sentence TLINK pairs through the same architecture.

2.3.4 Ablation Study

To assess the contribution of discourse, syntactic, and time-aware graphs, we performed an ablation experiment with different configurations (Table 6.2). Removing the context encoder significantly degrades performance, indicating that the graph components themselves cannot replace the contextual encoding. Removing any of the graph encoders hurts the model performance, motivating the need for all the constituent graph components. We also analyzed the relative importance of $\mathcal{G}_{DG}$, $\mathcal{G}_{SG}$, and $\mathcal{G}_{TG}$ represented by color shading in the table. The results show that the syntactic graph is least important for document level pairs in TDDMan and TDDAuto, which we believe is due to the longer range dependencies present in this dataset. However, removing the discourse graph for TimeBank-Dense and MATRES datasets leads to the least performance deterioration as inter and intra-sentence pairs do not fully utilize document-level rhetorical relations. TIMERS outperforms the BERT baseline even without $\mathcal{G}_{TG}$, demonstrating its useful in cases where document creation date or timexes cannot be obtained easily.

2.3.5 Error Analysis

The error analysis results of TIMERS and its ablations for TDDMan are shown in Fig. 8.8. The results provide evidence that the syntactic-aware graph ($\mathcal{G}_{SG}$) is most important for relations that can be extracted from a single sentence (SE). The time-aware graph ($\mathcal{G}_{TG}$) plays an important role in improving relationships requiring chain reasoning (multi-hop) and relationship determined by future events. We also note the role of the rhetorical-aware graph ($\mathcal{G}_{DG}$) for modeling future possibility (FE), hypothetical events (HN) and causal conditions for event occurrences (CP). This can be attributed to rhetorical relational features that extract plausible inter-dependencies such

as *cause, explanation, contrast* (Lioma, Larsen, and Lu 2012). None of the experimented models show improved performance on TLINK pairs which depend on world knowledge (WK) or event coreference (EC).

2.4 Conclusion

This work presents a neural architecture that utilizes local syntactic features, rhetorical discourse features, and temporal arguments in semantic role labels through a Gated Relational-GCN for document-level temporal relation extraction on TDDiscourse, MATRES, and TimeBank-Dense datasets. Experiments show that TIMERS shows substantial improvement for events that require chain reasoning and causal prerequisite links. Future work will focus on exploring real-world scenarios in which the temporal extraction task suffers from absent or erroneous event and timex annotations. We believe our proposed methods can also be adapted for other languages as well by overcoming possible limitations such as dependency parsing, semantic parsing, Timex normalization for the non-English corpora.

CHAPTER 3

DocTime: A Document-level Temporal Dependency Graph Parser

Abstract

We introduce `DocTime` - a novel temporal dependency graph (TDG) parser that takes as input a text document and produces a temporal dependency graph. It outperforms previous BERT-based solutions by a relative 4-8% on three datasets from modeling the problem as a graph network with path-prediction loss to incorporate longer-range dependencies. This work also demonstrates how the TDG graph can be used to improve the downstream tasks of temporal questions answering and NLI by a relative 4-10% with a new framework that incorporates the temporal dependency graph into the self-attention layer of Transformer models (`Time-transformer`). Finally, we develop and evaluate a new temporal dependency graph dataset for the domain of contractual documents, which has not been previously explored in this setting.

3.1 Introduction

Understanding the temporal relations between events mentioned in a document is an important natural language task with applications in downstream tasks such as timeline creation (Leeuwenberg and Moens 2018), time-aware summarization (Noh et al. 2020), temporal question-answering (Ning et al. 2020a), and temporal information extraction (Leeuwenberg and Moens 2019). This area of research remains important yet challenging due to several limitations such as confounded modalities (eg. events that are certain to happen vs the ones that might happen), event ambiguity (eg. agreeing to terms of a contract vs signing a contract) and need for complete annotation of all event pairs for precise temporal localization (Yao et al. 2020).

Early work densely annotated all pairs of events to address this problem (Cassidy et al. 2014b), but was limited to short passages or adjacent sentences due to the $\binom{n}{2}$ complexity of the task, especially for long documents. Recently this problem formulation was significantly simplified using temporal dependency trees (TDT) (Zhang and Xue 2019) and temporal dependency graphs (TDG) (Yao et al. 2020) by only capturing the reference TIMEX or event to build a dependency graph to capture this information. This enabled the development of temporal dependency parsers (**ross-etal-2020-exploring**; Zhang and Xue 2018a) to infer temporal relationships more robustly and efficiently.

We introduce `DocTime` - a state-of-the-art temporal dependency parser that parses document-level text to produce temporal dependency graphs. Unlike previous approaches using contextual features such as BERT (Ross, Cai, and Min 2020), our model utilizes a graph network and a novel path prediction loss to reason over long-range multi-hop dependencies while maintaining global consistency of temporal ordering of inter-dependent events.

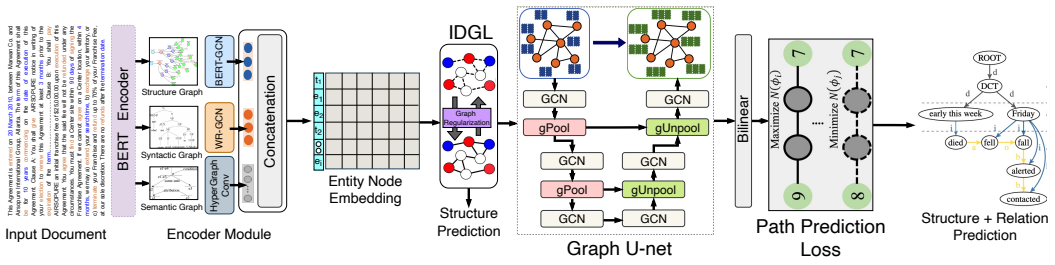


Figure 3.1: DocTime: encodes rich token level embeddings from an input document using structural, syntactic, and semantic graphs through BERT-GCN, WR-GCN and HyperGraph Conv layers, respectively. Token-level features are concatenated and passed through Iterative Deep Graph Learning (IDGL) to learn a noisy dependency structure over the TIMEX and Event entities. Graph U-net allows the model to incorporate longer-range dependencies for predicting the final temporal dependency graph structure and relationships. The model is trained with a novel auxiliary path prediction loss to learn multi-hop connections in TDG.

To validate the utility of DocTime and our generated temporal dependency graph, we go one step further than prior work and explore the question of whether temporal dependency graphs are useful for downstream tasks by introducing Time-Transformer. It is a framework to incorporate temporal dependency graphs into existing transformer-based architectures without retraining from scratch. We demonstrate the usefulness of our proposed Time-Transformer on temporal NLI (Vashishtha et al. 2020) and time-sensitive question answering (Chen, Wang, and Wang 2021) tasks.

Prior work on temporal relationship extraction and temporal dependency parsing have been mostly limited to news (Zhang and Xue 2019; Yao et al. 2020; Pustejovsky et al. 2003b), narrative stories (Zhang and Xue 2018b; Kolomiyets, Bethard, and Moens 2012) or clinical notes (Bethard et al. 2016). In addition to experimenting with existing temporal dependency parsing datasets, we introduce a dataset for temporal dependency graphs in a new domain - contractual documents, where temporal reasoning over events has real world legal and monetary implications for users.

Our main contributions include:

- A novel document-level temporal dependency parser (`DocTime`) that predicts the temporal dependency graph from text in an end-to-end manner with a novel path prediction loss, which outperforms the current SOTA by a relative 4-8% on three datasets.
- `Time-Transformer`, a novel framework to incorporate Temporal Dependency Graphs into transformer models for downstream tasks without needing to retrain from scratch. Results on natural language inference and question answering with a new self-attention module show a relative 4%-10% improvement.
- Development of new document-level (>1500 words) TDG dataset in the domain of contractual documents (ContractTDG).

3.2 Related Work

Temporal Dependency Parsing: Previous work has been devoted to pairwise classification of relations between events and time expressions, notably TimeBank (Pustejovsky et al. 2003c) and its extensions like (Cassidy et al. 2014b) annotated all relations. Pair-wise annotation have multiple problems including polynomial square complexity, global inconsistencies in predictions due to relation transitivity and forced annotation of vague relations (Ning, Wu, and Roth 2018b). Prior work focuses on extracting temporal relations between event pairs in the same sentence or adjacent sentences (Goyal and Durrett 2019; Ning, Subramanian, and Roth 2019a; Han et al. 2019a; Han, Ning, and Peng 2019; Han et al. 2019b; Han, Zhou, and Peng 2020; Ballesteros et al. 2020; Zhao, Lin, and Durrett 2020b). TIMERS (Mathur et al. 2021a) presented temporal relation extraction in long document.

Temporal Dependency Parsing (TDP): Temporal dependency trees were first proposed by

(Kolomiyets, Bethard, and Moens 2012). (Zhang and Xue 2018b) provided the the earliest TDT corpus on news data and narrative stories, (Zhang and Xue 2019) released the first English TDT corpus. (Yao et al. 2020) relaxed the assumption of single reference edge in dependency trees to form the improved TDG. (Zhang and Xue 2018a) built an end-to-end neural temporal dependency parser using BiLSTM and (Ross, Cai, and Min 2020) improved it further incorporating BERT. Our approach improves by modeling complex dependencies and introduces a new resource for TDG in contracts.

Linguistically-aware Transformers: Recent works have investigated using linguistic features as a prior for Transformer models. Syntax-bert (Bai et al. 2021) uses syntactic and constituency dependency on NLI and GLUE benchmarks. Coref-BERT Coreference-Informed Transformer (Liu, Shi, and Chen 2021) performs coreference-aware dialogue summarization. Temporal reasoning about event ordering can find applications in many tasks such as summarization (Noh et al. 2020), question answering (Chen, Wang, and Wang 2021; Ning et al. 2020b; Jin et al. 2020), commonsense reasoning (Qin et al. 2021), and natural language inference (Vashishtha et al. 2020). We propose to use TDG as priors to Transformer models to make them temporally-aware for use in downstream tasks.

3.3 DocTime: Document TDG Parsing

Task Formulation: Let document D be defined as a sequence of n tokens $[x_1, \cdots, x_n]$. The entire document can be seen as sequence of m sentences $[s_1, \cdots, s_m]$. Each document has a set of p events $E = [e_1, \cdots, e_p]$ and q timexes $T = [t_1, \cdots, t_q]$, where $p, q \leq n$. The creation date of the document is represented by timestamp t_{DCT}. (Yao et al. 2020) defines a temporal

dependency graph (TDG) where each timex node always has a reference timex, which is the most specific narrative time related to the event (Pustejovsky and Stubbs 2011). If such a narrative time is not available, the timex should be anchored to the DCT. An event node can either have a reference timex or be connected to a reference event, which is an event that provides the most specific temporal location. The task of temporal dependency graph parsing of a text document D results in a dependency graph $G = (C, V)$, where C represents the set of all events, timexes and the document creation date (DCT). V is the set of all edges in the graph, where each edge represents a temporal relationship $\Re$ between corresponding entity node pair $V = \{(t_i, t_j), (e_i, e_j), (e_i, t_j)\} \forall i, j \in C$.

Model Overview: Figure 3.1 shows an overview of our network architecture for temporal dependency parsing. We first extract token level BERT features from the input document, which are then enriched by three graph networks that encode structural, syntactic, and semantic relationships. This is followed by Iterative Deep Graph Learning over the TIMEX and Event entities to learn an initial dependency structure. This is passed through a Graph U-net to allow the model to incorporate longer range dependencies before predicting the final temporal dependency graph and relationships. The model is also trained with a novel auxiliary path prediction loss.

3.3.1 Feature Encoding

We leverage the pre-trained BERT language model to obtain the embeddings for each token as follows: $w_1, w_2, \cdots, w_n = \text{BERT}([x_1, x_2, \cdots, x_n])$, where w_i is the embedding of the token x_i. As document sequence lengths can be larger than 512, we use a *sliding window* encoding technique to encode whole documents. We average the embeddings of overlapping tokens of different windows to obtain the final representations. These token representations are enriched with

slightly enhanced variants of the structural (G_{str}), syntactic (G_{syn}) and semantic (G_{sem}) graphs utilized by (Mathur et al. 2021b) for document-level temporal relationship extraction. The key differences are the use of BERT-GCN (Lin et al. 2021) to combine contextual and structural graph features, the addition of co-reference relationships to the syntactic graph, and the use of a hypergraph convolution (Bai, Zhang, and Torr 2021) to allow for token level features in the semantic graph.

3.3.2 Temporal Dependency Prediction

We combine the learned representation for each entity node (timex, event, DCT) by concatenating the node embeddings learned from structural, syntactical and semantic graphs to obtain a D-dimensional feature vector for each of z entities in the document given by $\mathbf{F}(w_i) = g_i^{str} \oplus g_i^{syn} \oplus g_i^{sem}$, where $\oplus$ represents concatenation. We retain only the enriched node embeddings for each word. We then utilize Iterative Deep Graph Learning (IDGL)[1] (Chen, Wu, and Zaki 2020) to dynamically learn an initial dependency graph structure from the combined node embeddings. Given a noisy graph input feature matrix $\mathbf{F} \in R^{l*D}$, IDGL produces an implicitly learned graph structure $G^* = \{A^*, \mathbf{F}, f_l\}$ with a jointly refined corresponding graph node embeddings $\mathbf{F}'$ with adjacency matrix A^* by optimizing with respect to downstream link prediction task f_l between entity nodes.

3.3.2.1 Graph U-net For Higher Level Features

The Graph U-net (Gao and Ji 2019) is a U-shaped graph encoder-decoder architecture containing two down-sampling graph pooling (gPool) layers and two up-sampling graph unpooling

[1]Implementation: `https://github.com/graph4ai/graph4nlp`

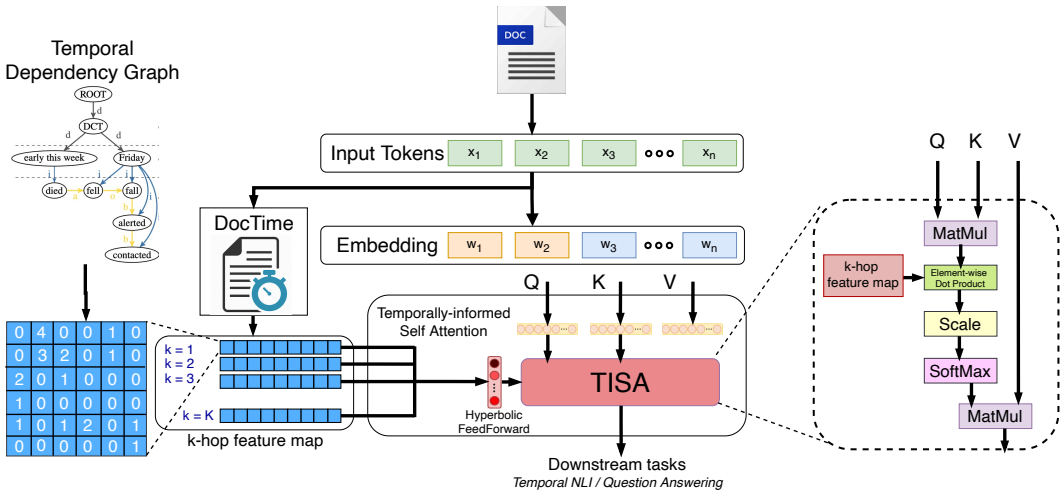


Figure 3.2: `Time-Transformer` is a variant of pre-trained Transformer models that augments temporal knowledge into the self-attention layer during fine-tuning of the Transformer model on different downstream tasks. The input text is converted into a temporal dependency graph using `DocTime` parser. The graph is then converted into a set of masks that encodes the temporal relationship between each token (i.e. After, Before) using the novel Temporally-informed Self-Attention (TISA). TISA creates K masks to represent the (k)-hop distance between two nodes in TDG for aggregating information across longer ranges in the input. TISA uses a hyperbolic feed-forward layer to learn the mask weights.

(gUnpool) layers with skip connections. gPool layers reduce the size of the graph to encode higher-order features, while the gUnpool layer restores the graph into its higher resolution structure, thereby promoting information exchange between entity pairs through an enlarged receptive field. Each graph pooling and unpooling layer is followed by a GCN layer to implicitly capture the topological information in the input graph. Taking the dynamically learned graph structure G^*, a graph embedding layer converts input node features **F'** into low-dimensional representations that are then passed through a graph U-net encoder-decoder $\mho$ to acquire entity-level relation matrix $\mathbf{Y} = \mho(\mathbf{F'})$, $\mathbf{Y} \in R^{l*l*D'}$.

3.3.2.2 *Temporal Dependency Link Prediction and Relation Classification*

Given entity adjacency matrix A^* and entity-level relation matrix $\mathbf{Y}$, we use a bilinear function to map them to link and relation probabilities $\mathbf{Z}_l$ and $\mathbf{Z}_r$, respectively. Formally, we have $\mathbf{Z}_l = \sigma(\mathbf{Y}W_l\mathbf{Y} + b_l)$ and $\mathbf{Z}_r = \sigma(A^*W_rA^* + b_r)$, where $W_l, W_r, b_l, b_r \in R^{D'*D'}$ represent learnable parameters. This is followed by a Softmax layer for link prediction and relations classification.

3.3.3 Training DocTime

Path Reconstruction Loss: In a document-level temporal parsing setup, the majority of node pairs may not have any ground truth link or temporal relation. Graph representation learning methods universally model relations between all entity pairs regardless of whether the entity pair has any relationship, leading to dispersion of attention in learning most non-existent edge connections. We propose path reconstruction loss L_{path}, which forces the model to pay more attention to learning entity pairs with relationships rather than ones without relationships. Equation 3.1 gives the cross entropy loss over all direct edge connection between all pairs of entities, where r_j^i indicates the relation between the entity pair and $P(r_j^i)$ is the probability of relation label r. Path reconstruction loss L_{path} modifies the cross entropy loss L_{ce} function as shown in Equation 3.2 by sampling all n^2 entity pairs and maximizing the probability of the shortest dependency path $\mathcal{N}(\phi)$ between the entity pair nodes. Finally, the path reconstruction loss and the existing classification loss are added as the training objective for DocTime, given by $L = L_{path} + L_{ce}$.

$$L_{ce} = -\frac{1}{\sum_{i=0}^{l} N_i} \sum_{i=1}^{l} \sum_{j=1}^{N_i} \{r_j^i \log P(r_j^i) + (1 - r_j^i) \log(1 - P(r_j^i))\} \tag{3.1}$$

$$L_{path} = -\frac{1}{\sum_{i=0}^{l} N_i} \sum_{i=1}^{l} \sum_{j=1}^{N_i} \{r_j^i \log \mathcal{N}(\phi_i) + (1 - r_j^i) \log(1 - \mathcal{N}(\phi_i))\} \tag{3.2}$$

Multi-task Training: Dependency link prediction and entity-level relation classification are correlated tasks and reinforce each other. We use multi-task training to optimize both tasks simultaneously using the path prediction cross-entropy loss. The final optimization uses a weighted sum of the dependency link prediction loss and entity-level relation classification loss $L = \lambda L_l + (1 - \lambda) L_r$, where the weighting factor λ is a hyperparameter.

3.4 Time-Transformer

We would also like to understand whether our temporal dependency parsing can be useful for downstream tasks requiring temporal reasoning. Here we introduce the Time-Transformer, which allows a TDG generated by `DocTime` to be combined with state-of-the-art transformer models for temporal tasks. The Time-Transformer augments the flow of information in a Transformer network via a temporally-informed self-attention mechanism. We first formulate the Time-Transformer architecture in §3.4 and then construct temporally-informed attention layers in §3.4.

Architecture: Time-Transformer was motivated by recent work incorporating syntax (Bai et al. 2021) or co-reference graphs (Liu, Shi, and Chen 2021) into the transformer architecture to improve downstream tasks. In each case, these approaches encode additional knowledge from the sparse graphs as a masked self attention layer into the transformer. Figure 3.2 shows the architecture of Time-Transformer incorporating temporal knowledge into the self-attention layer during fine-tuning of the Transformer model. The input text is converted into a temporal dependency

graph using DocTime parser. The graph is then converted into a set of masks that encodes the temporal relationship between each entity (i.e. After) explained in more detail in the next section: Temporally-informed Self-Attention. The input embedding (token+positional+attention masks) is passed through the Time-Transformer model which modifies the self-attention layer of the standard Transformer architecture with a *temporally-informed self-attention* layer to be fine-tuned on downstream tasks.

TISA: Temporally-informed Self-Attention : The TDG produced by DocTime is sparse and to effectively utilize the graph extracted by the temporal dependency parser for longer range temporal relationships, we utilize K self-attention layers that encode the temporal relationship if traversing K hops in the TDG as shown in 3.2. More formally starting from node A, the minimum number of hops (k) required to reach another node B can be regarded as k-hop distance between A and B, written as k-hop(A, B). We create K masks to represent the (k)-hop distance between two nodes to allow the model to aggregate information across longer ranges in the TDG. Specifically, a mask $M \in \{0, 1, 2, \cdots, r\}^{n \times n}$ denotes if there is a relation between entity i and j, and n is the number of tokens in the input text. The value of the mask is the relationship type for i and j. It is found by inferring the relationship using Allen's interval algebra (Allen 1983) and is set to 0 if there is no relationship or set to "Overlap" if there is a conflict. We adopt a soft-mask learning strategy to enable the self-attention layer to re-weight the importance of each mask and avoid the problem of vanishing gradient. A hyperbolic feed-forward layer is used to learn the mask weights as research has shown it can avoid distortion of the feature space in graph representations (Ganea, Bécigneul, and Hofmann 2018). The value of K is a hyperparameter that can be customized according to the nature of input dependency graph.

Training Time-Transformer: For each dataset, we optimize the hyper-parameters of

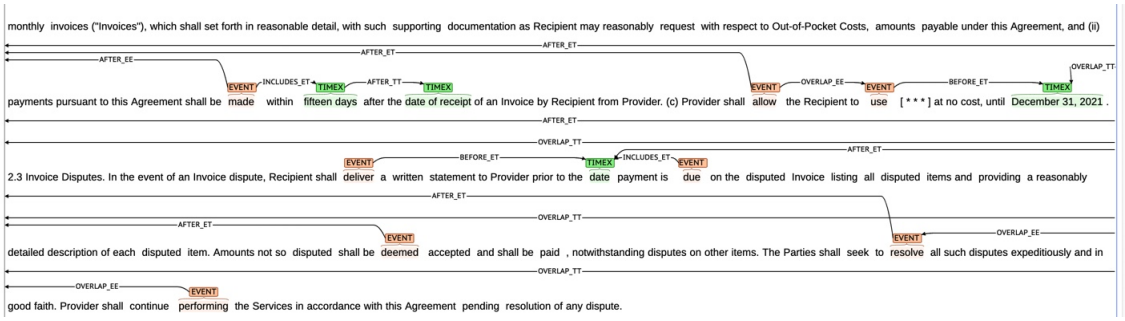


Figure 3.3: Example of a temporal dependency graph from ContractTDG dataset annotated using Brat Tool.

`Time-Transformer` through grid search on the validation data. In all our experiments, we limit the maximum value of k-hop to 15.

3.5 Experiment

3.5.1 Temporal Graph Parsing Datasets

We train and evaluate `DocTime` on three datasets. First is the **Temporal Dependency Graphs (TDG) dataset** (Yao et al. 2020) made up of 500 Wikinews articles annotated with document-level temporal dependency graphs. Second is the **Temporal Dependency Trees (TDT) dataset** (Zhang and Xue 2019) made from 183 documents derived from TimeBank (Pustejovsky et al. 2003b) annotated with a temporal dependency tree structure. The third dataset we created as part of this paper and is describe in more detail below.

Contract-TDG: Understanding the temporal relationship of events in contracts is an important business problem, where understanding event timelines can have legal and monetary consequences. Previous work on temporal relationships has largely focused on clinical, news, or narrative text, whereas to the best of our knowledge, the contractual domain has not been explored for this problem. To construct this dataset, we used 100 contracts from the Atticus con-

Dataset	Docs	Timex	Events	Rels
TimeBank (Pustejovsky et al. 2003c)	183	1,414	7,935	6,148
TB-Dense (Cassidy et al. 2014b)	36	289	1,729	12,715
MATRES (Ning, Subramanian, and Roth 2019b)	275	-	1,790	13,577
TDT-Crd (Zhang and Xue 2019)	183	1,414	2,691	4,105
TDG (Yao et al. 2020)	500	2,485	14,974	28,350
Contract-TDG) (Ours)	**100**	2354	11,752	12,909

Table 3.1: Comparison of ContractTDG data statistics to other temporal relation datasets. ContractTDG has fewer documents but a comparable number of TIMEX/Events/relations.

tracts dataset[2] (Hendrycks et al. 2021), which were sourced from public domain SEC contracts. Due to the multi-page length of these documents, we limited the annotations to the first 1500 words. We did not include definition sections, since they did not contain many events of interest for this task. The documents have a 70-10-20 split for training, validation, and testing.

To obtain the TDG annotations required for our task, we followed the 5 steps procedure outlined by the original TDG dataset in (**yao-etal-2020-annotating**): (i) TIMEX Identification (TE), (ii) Identifying reference times for TE, (iii) Event identification, (iv) Identifying reference times for events, (v) Identifying reference events for events. Document Creation Times (DCT) were provided as effective dates in the ATTICUS corpus.

Similar to (**yao-etal-2020-annotating**) for tasks 1 (TE) and 3 (Event ID), we used the Mechanical Turk platform to obtain two annotations to validate text spans of noisy TIMEXes extracted by HeidelTime software[3] (Strötgen and Gertz 2013) and verbs that were possible events. Disagreements were resolved by an expert annotator. However, for the reference tasks, we decided against using Mechanical Turk due to the difficulty and length of the contracts as well as the lower agreement faced by the original TDG system for the last two tasks. We instead

[2]https://www.atticusprojectai.org/cuad
[3]https://github.com/HeidelTime/heideltime

Task	TDG (F1)	Contract TDG (F1)
1: TIMEX ID	0.96	0.93
2: TIMEX RT	0.89	0.81
3: Event ID	0.79	0.76
4: RT ID (U)	0.67	0.83
4: RT ID (L)	0.61	0.75
5: RE ID (U)	0.59	0.85
5: RE ID (L)	0.52	0.79

Table 3.2: Inter-Annotator Agreement (IAA) for the Contract-TDG and TDG dataset. U = structure, L = structure + labels

	System	TD-Trees				TD-Graphs				ContractTDG			
		Structure-only		Structure+Relation		Structure-only		Structure+Relation		Structure-only		Structure+Relation	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Baselines	Majority Baseline	0.43	0.42	0.15	0.18	0.62	0.68	0.41	0.51	0.36	0.35	0.36	0.33
	Logistic Regression Baseline (Zhang and Xue 2018a)	0.64	0.70	0.26	0.29	0.62	0.69	0.49	0.58	0.42	0.39	0.45	0.38
	Neural Ranking Parser (BiLSTM) (Zhang and Xue 2018a)	0.75	0.79	0.53	0.60	0.69	0.79	0.55	0.66	0.49	0.46	0.52	0.48
	BERT Ranking Parser (Ross, Cai, and Min 2020)	0.77	0.83	0.59	0.68	0.71	0.80	0.62	0.71	0.67	0.65	0.62	0.61
	DocTime (ours)	**0.85***	**0.86***	**0.66***	**0.72***	**0.74***	**0.85***	**0.69***	**0.77***	**0.70***	**0.69***	**0.68***	**0.64***
Ablation	DocTime w\o Graph U-net	0.83	0.84	0.63	0.70	0.71	0.82	0.67	0.75	0.68	0.63	0.66	0.62
	DocTime w\o Structure Graph	0.81	0.80	0.62	0.65	0.67	0.72	0.65	0.73	0.67	0.63	0.64	0.60
	DocTime w\o Syntactic Graph	0.80	0.82	0.62	0.66	0.65	0.73	0.62	0.69	0.64	0.61	0.62	0.59
	DocTime w\o Semantic Graph	0.76	0.78	0.55	0.65	0.62	0.70	0.60	0.67	0.59	0.57	0.59	0.57
	DocTime w\ Graph Prediction	0.72	0.64	0.49	0.55	0.57	0.65	0.57	0.58	0.59	0.53	0.55	0.54
	DocTime w\ Pairwise Link Prediction	0.82	0.83	0.63	0.69	0.72	0.83	0.66	0.73	0.65	0.60	0.62	0.60
	DocTime w\ Path Prediction Loss	0.85	0.86	0.66	0.72	0.74	0.85	0.69	0.77	0.70	0.69	0.68	0.64

Table 3.3: Results comparing the performance of DocTime with baselines and ablative components on TDT, TDG, and ContractTDG datasets. We use majority and logistic regression baselines from (Zhang and Xue 2018a). * indicates statistical significance over BERT Ranking Parser (Ross, Cai, and Min 2020) ($p \leq 0.005$) under Wilcoxon's Signed Rank test. Darker green represents better F1 performance in ablation studies. Bold denotes the best-performing model. DocTime improves substantially over all datasets for both dependency structure and structure+relation prediction tasks. The ablation shows that semantic graph features prove to be most beneficial. Our proposed path prediction loss is critical for state-of-the-art performance of DocTime model.

used the BRAT annotation tool[4] (Stenetorp et al. 2012) with an expert annotator for tasks 2,4, and 5, following the (**yao-etal-2020-annotating**) guidelines . ContractTDG is annotated for four temporal relations - after, before, overlaps, and includes.

Table 3.1 compares the data statistics of the ContractTDG to previous temporal relationship and temporal dependency corpora. Even though this dataset has many fewer documents than the TDG dataset, it has a large number of TIMEX, Events, and Temporal relationships due to the document length. Table 3.2 reports the F1 IAA metrics for ContractTDG dataset to directly

[4]https://brat.nlplab.org/

compare to the original TDG dataset. For Tasks 1 and 3 we report IAA F1 for the two crowd sourced worker annotations and for the relationship tagging tasks (2,4,5), we report IAA metrics calculated on the test postion (20% of the data) that was reviewed by two experts. The agreement is slightly lower for the TIMEX/Event identification tasks but higher for the three relationship tasks. We evaluate DocTime for dependency structure as well as structure+relation prediction for both development and test splits.

3.5.2 Time-Transformer Experiments for Downstream Tasks

We adopt Time-Transformer on BERT (Devlin et al. 2019a), RoBERTa (Liu et al. 2019a), BigBird (Zaheer et al. 2020a) and FiD (Izacard and Grave 2021) for evaluation on two downstream tasks in §3.6.2. We utilized the official checkpoint for each pre-trained language model as provided by respective authors. First, we test Time-BERT and Time-RoBERTa on Temporal NLI dataset, which consists of 5 sub-datasets (Vashishtha et al. 2020) to study the effect of temporal reasoning for predicting event ordering and duration. Second, we run experiments on the TimeQA dataset (Chen, Wang, and Wang 2021) to evaluate the performance of Time-BigBird and Time-FiD for the long-document question-answering task. We report Exact Match (EM) and F1 scores as evaluation metrics on dev and test sets of easy and hard versions.

3.6 Results and Analysis

3.6.1 Temporal Graph Parsing

Performance of DocTime w.r.t. baselines: Table 6.2 compares the performance of DocTime against other baseline methods on TDT, TDG and ContractTDG. We also provide a majority

Model	UDS-duration	UDS-order	TempEval3	TimeBank-Dense	RED
Majority	50.00	54.52	54.57	50.54	52.51
NBOW (Iyyer et al. 2015)	82.54	54.52	54.57	50.54	52.51
Infersent (Conneau et al. 2017)	92.65	73.22	62.20	68.29	63.47
RoBERTa (Liu et al. 2019b)	94.51	80.17	54.57	94.60	80.59
Time-RoBERTa (E)	95.78	82.03	60.66	95.45	82.10
Time-BERT	96.01	82.97	61.32	96.08	82.15
Time-RoBERTa	**96.67**	**82.98**	**62.50**	**96.33**	**82.50**

Table 3.4: Accuracy comparison on the Temporal NLI dataset test set. Time-RoBERTa fine-tuned by utilizing temporal dependencies extract from DocTime model pre-trained on TDG dataset outperform all baselines provided by (Vashishtha et al. 2020)(see **bold**).

baseline ContractTDG to evaluate whether the methods work better than a random label assignment as implemented in (Yao et al. 2020). We also include the two current SOTA approaches for temporal dependency parsing: The BiLSTM attention-based Neural Ranking Parser proposed by (Zhang and Xue 2018a) [5] and the BERT Ranking Parser (Ross, Cai, and Min 2020) on each dataset . We also report results for a logistic regression baseline proposed by (Zhang and Xue 2018a). Results in Table 6.2 show that DocTime outperforms both Neural and BERT Ranking Parser by a significant margin on the TDT (2-4%) TDG (5-6%) and ContractTDG (3-4%) datasets. We believe its primarily because they formulate temporal dependency parsing as a ranking task designed to select the best reference event/timex for each node. However, TDG parsing requires the model to be able to reason over multiple dependencies originating from each node while maintaining global consistency of temporal ordering of inter-dependent events. We perform experiments for dependency structure prediction and structure+relation prediction and find that predicting labeled dependency edges is a much more challenging task across all datasets. DocTime achieves state-of-the-art performance on all three datasets (see **bold**), and shows that it can successfully handle document-level long-range dependencies in the challenging ContractTDG dataset from the 6-12% relative improvement over the BERT based ranking parser.

[5]Used: http://github.com/yuchenz/tdp_ranking

Ablation Study of DocTime: To assess the contribution of structure and syntactic and semantic graph features, we performed ablation experiments as reported in Table 6.2 highlighted in red. We also analyzed the effect of different types of training loss. We observe that removing the semantic graph consistently degrades performance, indicating the need for hypergraph learning over temporal arguments and RST features to capture document-level discourse relations. We see that removing structure graph reduced the performance to below the BERT Ranking Parser, as DocTime leverages BERT's contextual learning through a structural graph. Syntactic graph adds incremental value to DocTime due to its relational learning of syntactic dependencies within each sentence through relational GCN. We evaluated the model performance in case all edges of the TDG are used for one forward pass and call it "Graph Prediction". Training the model by evaluating a single edge in one pass (similar to temporal relation prediction in (Pustejovsky et al. 2003c) is referred to as "Pairwise Prediction". We explore the impact of different training losses for the proposed model (Table 6.2, highlighted in green). Learning DocTime by propagating losses over the entire document graph severely deteriorates model performance as the model has very limited training documents samples (182 for TDT, 400 for TDG, 80 for ContractTDG). Our proposed path prediction loss shows superior performance over pairwise link prediction as it jointly learns the relation label between a pair of nodes as well as the shortest dependency path linking them. As a result, the model can recover from structure prediction errors between nodes by learning an alternative path reconstructed through multi-hop connections.

Model	Easy-mode				Hard-mode			
	Dev		Test		Dev		Test	
	EM	F1	EM	F1	EM	F1	EM	F1
FT on TimeQA								
BigBird (Zaheer et al. 2020c)	16.4	27.5	16.3	27.1	11.4	20.6	11.9	20.3
`Time-BigBird (E)`	15.5	25.0	14.1	25.5	9.6	15.6	9.3	18.5
`Time-BigBird`	**18.9**	**29.5**	**18.9**	**29.5**	**13.0**	**22.5**	**13.0**	**22.8**
FiD (Izacard and Grave 2020)	15.9	27.1	15.7	28.0	10.7	19.1	10.3	19.7
`Time-FiD (E)`	13.8	25.2	12.1	25.6	8.9	17.3	8.8	17.6
`Time-FiD`	**17.5**	**29.3**	**18.1**	**30.3**	**12.5**	**22.2**	**12.5**	**21.5**
FT on TriviaQA								
BigBird (Zaheer et al. 2020c)	33.4	42.5	33.7	43.0	27.7	35.9	27.7	36.2
`Time-BigBird (E)`	31.3	40.4	32.3	41.8	25.9	33.6	25.8	35.5
`Time-BigBird`	**35.0**	**44.8**	**35.1**	**45.5**	**29.2**	**36.6**	**29.2**	**38.0**
FT on NQ + TimeQA								
FiD (Izacard and Grave 2020)	59.5	66.9	60.5	67.9	45.3	54.3	46.8	54.6
`Temp-FiD (E)`	57.9	65.6	58.5	65.2	41.1	52.6	44.5	52.8
`Time-FiD`	**61.3**	**68.2**	**62.4**	**69.6**	**46.7**	**56.2**	**48.2**	**56.4**

Table 3.5: Results comparing F1 score and exact match (EM) performance of `Time-BigBird` and `Time-FiD` for QA task on easy and hard sections of TimeQA dataset. We evaluate the Transformer models in 3 settings - fine-tune on TimeQA; fine-tune TriviaQA; and fine-tune on NQ then TimeQA. Green shows improvement due to our proposed `Time-Transformer` model, while we see degradation due to the Euclidean variant of `Time-Transformer (E)`

3.6.2 Application of Temporal Dependency Parsing for downstream tasks

We train the `DocTime` model on the TDG corpus, which can be used to infer a temporal dependency graph from raw text samples. We extract events and timexes using CAEVO (Chambers et al. 2014b) for all data samples in train,validate, and test. The temporal dependency graph acquired for each document is used as a prior for `Time-Transformer` to perform downstream tasks.

Performance of `Time-Transformer` on Temporal NLI: The temporal NLI task requires a model to identify the semantic relationship (entailed, not-entailed) between the context and corresponding hypothesis sentence based on temporal information from text. The temporal

dependency graphs extracted using the DocTime trained on the TDG corpus are used as prior for Time-BERT for entailment classification. Table 3.4 shows the test accuracies of Time-BERT-large, Time-RoBERTa-large and other competitive baselines [(Iyyer et al. 2015), (Conneau et al. 2017)] reported by (Vashishtha et al. 2020). The temporal information prior proposed in Time-Transformer helps the BERT and RoBERTa models perform much better on the NLI task. The accuracy improved by 1.5-2.3 F1 points by applying our framework on the RoBERTa model across the five subsets. We observe the performance gain in the case of the Euclidean version of Time-RoBERTa to be modest as compared to its hyperbolic counterpart.

Performance of Time-Transformer on TimeQA: The TimeQA task focuses on understanding the time scope of facts in the long text followed by answering questions conditioned on the query and the document using implicit temporal information. We then apply the DocTime model output trained on the TDG corpus to the Time-Transformer framework on BigBird and FiD language models for long document question answering task. Following (Chen, Wang, and Wang 2021), we experiment with three variants of pre-trained settings: (1) fine-tuned on the TimeQA training set; (2) fine-tuned on NQ/TriviaQA data (3) fine-tuned on NQ/TriviaQA data and TimeQA.

Table 3.5 shows the effectiveness of Time-BigBird and Time-FiD in consistently outperforming their corresponding baselines in all three settings. More specifically, we see a realtive gain of 10-14% in F1 and exact match scores (EM) for both easy and hard sections of the dataset. It is impressive to note that the improvements due to the Time-BigBird and Time-FiD models are steady with different pre-training setups with the addition of only a few extra parameters to the baseline model. An important observation here is that the Euclidean versions of Time-BigBird and Time-FiD show persistent performance deterioration across

all settings for TimeQA. We attribute this phenomenon to our initial hypothesis behind using hyperbolic operations in the proposed Temporally-informed self attention (TISA) layer. As the text length grows, the complexity of geometric operations increases, leading to vectorial distortions in Euclidean spaces (Ganea, Bécigneul, and Hofmann 2018). This is remedied by hyperbolic transformations of masked self-attention learning in the proposed `Time-Transformer`.

Our experiments provide evidence that temporal dependency graphs extracted using `DocTime` and then utilized as a prior by temporally-informed Transformer architectures such as `Time-Transformer` can improve the performance of several downstream tasks that require temporal reasoning at the sentence-level as well as at the document-level.

Impact of Long-term Dependency on `Time-Transformer` performance: We plot Fig. 3.4 to understand the capability of Transformer models to handle the long-term dependency in temporal reasoning on the TimeQA dataset. Plot shows the exact match (EM) accuracy vs length of the input document for hard samples. We use BigBird and FiD models fine-tuned on NQ + TimeQA as backbone models. BigBird's performance degrades rapidly as the length increases to over 5000 tokens, while the FiD's performance is quite uniformly distributed across different document lengths due to it's strong capability to deal with long-term dependency. `Time-BigBird` and `Time-FiD` follow a similar trend and maintain steady improvements over their corresponding baseline models with increasing in input lengths.

Space complexity analysis: We choose RoBERTa-base as the base model to analyze the space complexity. (Liu et al. 2019b) reported the number of trainable parameters in RoBERTa-Base to be about 123 million. `Time-RoBERTa` introduces an additional 2 million parameters in total due to k-hop mask learning in the TISA layer. Therefore, Time-BERT adds few parameters to the base model without affecting its original space complexity.

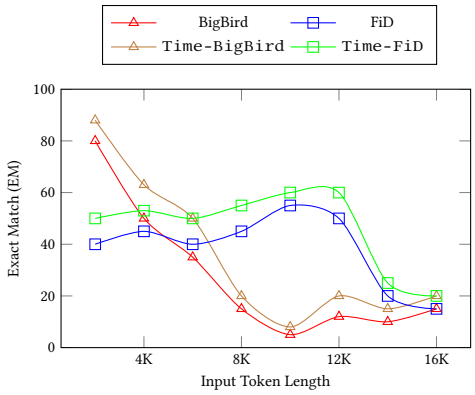


Figure 3.4: Impact analysis of long-distance dependencies on Transformer models for TimeQA task. The plot shows the exact match (EM) accuracy vs length of input document for hard samples. We use BigBird and FiD fine-tuned on NQ + TimeQA as backbone models. `Time-BigBird` and `Time-FiD` maintain steady improvement over baseline models even with the increase in input lengths.

Corpus	Model	Structure + Relation (F1)			
		te,te	e,te	e,e	full
TD-Graphs	Heuristic	0.82	0.58	0.34	0.51
	Neural Ranking Parser (Zhang and Xue 2018a)	0.93	0.66	0.58	0.66
	BERT Ranking Parser (Ross, Cai, and Min 2020)	0.93	0.74	0.58	0.71
	DocTime	**0.96**	**0.75**	**0.72**	**0.77**
Contract-TDG	Heuristic	0.45	0.36	0.18	0.33
	Neural Ranking Parser (Zhang and Xue 2018a)	0.57	0.45	0.29	0.48
	BERT Ranking Parser (Ross, Cai, and Min 2020)	0.70	0.54	0.33	0.61
	DocTime	**0.75**	**0.56**	**0.39**	**0.64**

Table 3.6: Performance (F1 score) of DocTime across timex-timex, event-timex and event-event pairs for dependency structure+relation prediction on TDG and ContractTDG datasets. DocTime outperforms all baselines on every setting.

Time Complexity analysis: We assume the number of tokens in each sentence to be n and extract k-hop mask matrices from a text document is $O(n^2)$ in the online inference phase. The time complexity of the Transformer embedding lookup layer is $O(n)$. The TISA layer calculates the attention score in $O(KD_q n^2)$ for both QK^T and learns the mask weights using a hyperbolic feedforward layer (MW^M), where D_q is dimension of Q and K is the number of sub-networks. The time complexity of the `Time-`BERT remains the same for small enough value of k ($k \leq 15$ in experiments).

3.7 Conclusion

We present `DocTime`, a new temporal dependency parsing approach that improves upon previous approaches by integrating longer term temporal information through a graph network with a novel path prediction loss. Additionally, we are able to show how a TDG can be incorporated into Transformer networks with Time-Transformer to improve on down stream tasks for NLI and question answering. Finally we introduce a TDG dataset in a new domain (Contractual documents) to expand research in this temporal reasoning to a new application domain. Future works will aim to explore more ways for integrating temporal dependency graphs into neural architectures across different application domains. In future, we would like to explore temporal event mining to aid various social media applications such as improving hate speech detection (Mathur et al. 2018b; Chopra et al. 2020), analyzing temporality in suicidal ideation detection (Mishra et al. 2019; Mathur et al. 2020) and abuse detection (Gautam et al. 2020; Sawhney et al. 2021b). The proposed Time-Transformer can find applications in augmenting financial tasks (Sawhney et al. 2020b), affective computing (Mittal et al. 2021), and AI for social good (Mathur

et al. 2018a) with temporal common sense reasoning.

CHAPTER 4

DocInfer: Document-level Natural Language Inference using Optimal Evidence Selection

Abstract

We present **DocInfer** - a novel, end-to-end Document-level Natural Language Inference model that builds a hierarchical document graph enriched through inter-sentence relations (topical, entity-based, concept-based), performs paragraph pruning using the novel *SubGraph Pooling* layer, followed by optimal evidence selection based on REINFORCE algorithm to identify the most important context sentences for a given hypothesis. Our evidence selection mechanism allows it to transcend the input length limitation of modern BERT-like Transformer models while presenting the entire evidence together for inferential reasoning. We show this is an important

property needed to reason on large documents where the evidence may be fragmented and located arbitrarily far from each other. Extensive experiments on popular corpora - DocNLI, ContractNLI, and ConTRoL datasets, and our new proposed dataset called CaseHoldNLI on the task of legal judicial reasoning, demonstrate significant performance gains of 8-12% over SOTA methods. Our ablation studies validate the impact of our model. Performance improvement of $\sim 3-6\%$ on annotation-scarce downstream tasks of fact verification, multiple-choice QA, and contract clause retrieval demonstrates the usefulness of `DocInfer` beyond primary NLI tasks.

4.1 Introduction

Natural Language Inference (NLI) is a fundamental textual reasoning task seeking to classify a presented hypothesis as *entailed by, contradictory to or neutral to* a premise (Dagan et al. 2010). Prior NLI datasets and studies have focused on sentence-level inference where both the premises and hypotheses are single sentences (SNLI (Bowman et al. 2015), MultiNLI (Williams, Nangia, and Bowman 2018), QNLI and WNLI (Wang et al. 2018)) Document-level NLI extends the reasoning of NLI beyond sentence granularity where the premises are in the document granularity, whereas the hypotheses can vary in length from single sentences to passages with hundreds of words (Yin, Radev, and Xiong 2021).

Document level NLI is an important problem for many tasks including verification of factual correctness of document summaries, fact-checking assertions against articles, QA on long texts, legal compliance of contracts, etc. Even so, it challenges modern approaches due to the limited input bottleneck of modern Transformer models. Consider that the universally used BERT model (Devlin et al. 2018) can only encode 512 input sub-tokens due to its quadratic

self-attention complexity. Consequently, evidence in the document premise relevant to the hypothesis can potentially be distributed in several textual spans located arbitrarily far away from each other in long documents, and may not be simultaneously available to draw inference.

Recent approaches, notably SpanNLI (Koreeda and Manning 2021), HESM (Hanselowski et al. 2018)) and others, have shown that chunking the premise into multiple document spans, scoring them, and aggregating the scores helps mitigate the limited input length problem. Such approaches do not allow the inference module to reason over the complete evidence. In contrast to encoding the document as a set of sentences fed into a transformer for inferential reasoning, a recent line of work, e.g. EvidenceNet (Chen et al. 2022), GEAR (Zhou et al. 2019) and HGRGA (Lin and Fu 2022)), encodes documents as graphs and uses graph reasoning to perform textual inference. Graphs allow encoding of various morphological and semantic relationships at various granularities. However, these approaches use graph-based processing subsequent to evidence selection.

We address the above challenge with a reasonable assumption that the portion of the premise (the ground truth evidence) necessary and sufficient for inference can fit entirely into the length limit of language model for effective representation learning. Our proposed system achieves this by selecting sentences in the document that are contextually relevant for a given hypothesis through pruning irrelevant paragraphs and reinforce learning based optimal sentence selection. Our **main** contributions:

- **`DocInfer` – a novel DocNLI model** that simultaneously performs successive optimal evidence selection and textual inference on large documents. It utilizes a novel graph representation of the document encoding structural, topical, concept and entity-based relationships. It performs subgraph pooling and asynchronous graph updates to provide a

pruned, hypothesis-relevant and richer sub-document graph representation and uses a reinforcement-learning based subset selection module to provide the contextually-relevant evidences for inference. Experimental results show that `DocInfer` outperforms the current SOTA on DocNLI, ContractNLI and ConTRoL datasets with a significant improvement of 8-12%.

- We propose **CaseHoldNLI - a new document-level NLI dataset** in the domain of legal judicial reasoning with over 270K document-hypotheses pair with maximum premise length of 3300 words. We observe similar performance gains on this dataset.
- **Application on downstream tasks:** We demonstrate the usefulness of the `DocInfer` evidence selection module on downstream tasks of fact verification, multiple choice QA and few shot clause retrieval from legal texts using no or small amounts of data for supervised fine-tuning. Results on FEVER-binary, MCTest, and Contract Discovery dataset show significant improvement of $\sim$ 3-6% F1.

4.2 Related Work

Document-level NLI Datasets: (Yin, Radev, and Xiong 2021) introduced Doc-level NLI on news and Wikipedia articles. (Liu et al. 2021a) proposed the multi-paragraph ConTRoL dataset focused on complex contextual reasoning (logical, coreferential, temporal, and analytical reasoning). Several datasets comprising legal documents like case laws, statutes, and contracts have been proposed. COLIE-2020 (Rabelo et al. 2020) and (Holzenberger, Blair-Stanek, and Van Durme 2020) support identification of relevant paragraphs from cases that entail the decision of a new case. However, the combined input length of their premise-hypothesis pairs remains

within 512 tokens with the premise lengths at paragraph-level, reasonably suited for input to BERT-like models. (Koreeda and Manning 2021) released ContractNLI dataset for document-level NLI task on multiple page NDA contract documents along with ground truth evidence labeling for interpretability. We benchmark `DocInfer` on DocNLI, ContractNLI and ConTRoL datasets. CaseHOLD dataset (Zheng et al. 2021) is a multiple choice QA dataset for selecting relevant governing laws required to reason about the legal decision text. **Document-scale and Corpus-Scale Reasoning**: In order to handle document-scale premises in the doc-NLI corpora, approaches like SpanNLI (Koreeda and Manning 2021), HESM (Hanselowski et al. 2018)) chunk the premise into multiple document spans for reasoning. A similar approach was followed by legal language models such as Legal-BERT (Chalkidis et al. 2020) and Custom Legal-BERT (Zheng et al. 2021) for legal reasoning tasks. More recently, language models (e.g., Longformer (Beltagy, Peters, and Cohan 2020) with 4096 token input) have been proposed to overcome the limited input field bottleneck. Fact Extraction and Verification (FEVER) (Thorne et al. 2018) tasks require extracting evidence and claim entailment given an input claim and the Wikipedia corpus. Prior works in this domain address the length limitation for claim verification by relevant evidence identification and its chunking which are individually scored and probabilistically aggregated (Subramanian and Lee 2020; Jiang et al. 2021a). Hierarchical graph modeling may be used to handle the large scale of the premise (Liu et al. 2019c; Zhou et al. 2019; Zhong et al. 2020; Zhao et al. 2020; Chen et al. 2022; Lin and Fu 2022; Si et al. 2021). **Context Selection for Document-level NLP**: Recent works have investigated selection of relevant context for document-level NLP tasks such as Neural Machine Translation (Kang et al. 2020), Event Detection (Ngo, Nguyen, and Nguyen 2020; Veyseh et al. 2021), Relation Extraction (Trong et al. 2022). Recently, some of the work on document-level NLP has looked at temporal relation extraction (Mathur et al. 2021b) ,

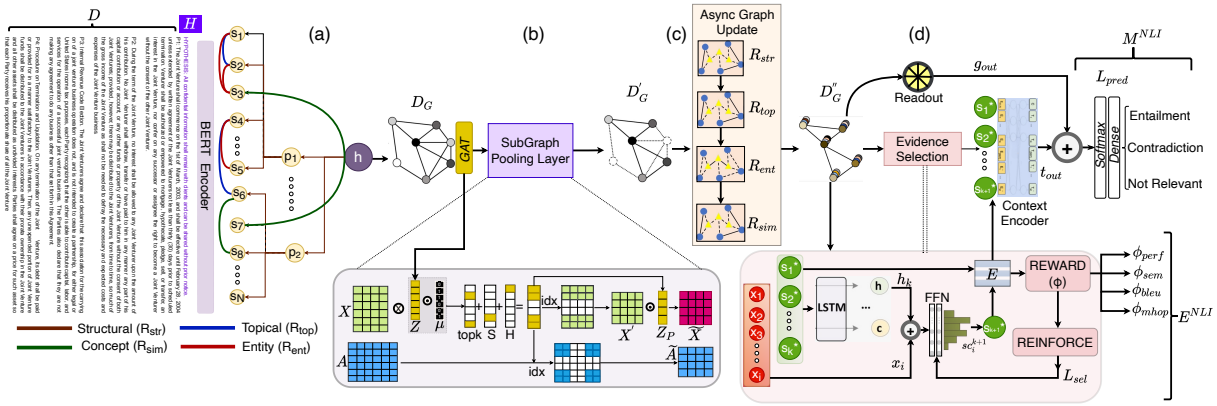


Figure 4.1: DocInfer Architecture: Document D and hypothesis H pass through BERT; Hierarchical graph using (R_{str}), (R_{top}), (R_{sim}) and (R_{ent}) relations. SubGraph Pooling extracts relevant paragraph nodes; Asynchronous graph updates learn relation-specific node embeddings. Evidence Selection optimized by REINFORCE rewards (ϕ_{perf}), (ϕ_{bleu}), (ϕ_{sel}), (ϕ_{mhop}).

temporal dependency parsing (Mathur et al. 2022d), and speech synthesis (Mathur et al. 2022a) using graphs and sequence learning. However, none of them have considered an end-to-end trainable approach for graph learning with to identify the relevant evidence extraction.

4.3 DocInfer

Given a textual hypothesis H, the task of document-level NLI is to classify whether the hypothesis is *entailed by*, *contradicting to* or *not mentioned by (neutral to)* the document D. We present DocInfer, a neural architecture (Figure 4.1) that can select a set of evidence sentences E from document D to form a shortened document D^e which is then used for NLI prediction. Here, for the document level NLI task, we need to constrain D^e to fall within the length limit of BERT-like context encoder to enable it to consume the evidence entirely for improved representation learning for NLI.

Our model can been seen as a sequence of four phases: (a) Representation of document D in the presence of the Hypothesis H to form a hierarchical document graph with sentences and

paragraphs as nodes and Structural, Topical, Entity-centric and Concept-similarity relations as edges. (b) Paragraph node pruning using the novel *Subgraph Pooling* layer to select highly relevant paragraphs. (c) Asynchronous graph update for improved node representations and finally. (d) Optimal evidence selection using REINFORCE from the graph for the task of document-level NLI.

Document Representation: Let premise document D be defined as a sequence of n sentences $s_1, s_2, \cdots, s_n$ such that $D = [s_1, s_2, \cdots, s_n]$. These sentences are naturally grouped into m consecutive paragraphs $P = [p_1, p_2, \cdots, p_m]$ such that each each sentence s_i belongs to only one paragraph p_j. We leverage pre-trained BERT language model to obtain the embedding of every sentence and paragraph nodes. The final representation for each sentence s_i and paragraph p_j is obtained by extracting the hidden vector of the CLS token as given by Emb(s_i) = BERT ([[CLS]; H; [SEP]; s_i; [SEP]]) and Emb(p_j) = BERT ([[CLS]; H; [SEP]; p_j; [SEP]]), respectively. Here H denotes the hypothesis text which is also encoded as h =BERT([[CLS]; H; [SEP]]). [CLS] and [SEP] are symbols that indicate the beginning and ending of a text input, respectively.

Document Graph Construction: The document is then modeled as a hierarchical graph $D_G = (V, E)$ to capture the premise document structure. Here, $V = \{V_p, V_s, V_h\}$, where V_p, V_s, V_h are nodes corresponding to all the paragraphs, all the sentences and the hypothesis, respectively. The set of edges (E) of the Document Graph encodes four types of relations between the nodes mentioned below:

(1) Structural Relations (R_{str}): Hypothesis-Paragraph edges and Paragraph-Sentence Affiliation edges model the hierarchical structure of the document through a directed edge from the hypothesis node to each paragraph node and from a paragraph node to each constituent sentence, respectively. Further, Paragraph-Paragraph Adjacency and Sentence-Sentence Adjacency links

preserve the sequential ordering for consecutive paragraph and sentence nodes through directed edges.

(2) Topical Relations (R_{top}): Sentence-Sentence Topical Consistency connections model the topical consistency between a pair of sentences by constructing sentence-level topical representations via latent Dirichlet allocation (Blei, Ng, and Jordan 2003). Given a pair of sentences s_i and s_j, we extract latent topic distribution $lda_i, lda_j \in R^l$ for each sentence which are joined if the Helinger $H(lda_i, lda_j)$ distance between them is greater than 0.5.

(3) Entity-centric Relations (R_{ent}): Sentence-Sentence Entity Overlap connections explicitly model the sentence-level interactions between entity spans by adding an undirected edge between two sentence nodes if they share one or more named entities. Further, Sentence-Sentence Entity Coreference connections join two sentences by an undirected edge if the sentences share mentions referring to the same real world entity.

(4) Concept-Similarity Relations (R_{sim}): Sentences conceptually similar to other sentences and the hypothesis are connected to each other to account for presence of related events and topics in two sentences. We propose Sentence-Sentence ConceptNet Similarity using ConceptNet Numberbatch (CN). Let $A_i^{cn} = [a_1, a_2, \cdots, a_l]$ be the ConceptNet Numberbatch embeddings for the words in sentence $s_i = [w_1, w_2, \cdots, w_M]$ respectively. Here, if a word w_q does not have its corresponding embedding in CN, we simply set its vector a_q to zero. Further, we introduce Hypothesis-Sentence Knowledge Similarity (using KnowBert embedding) connections that add weighted undirected edges between sentence-sentence and hypothesis-sentence node pairs, respectively. KnowBert representations are obtained by encoding text using the pre-trained KnowBert language model as $A_i^{kbrt} = \text{KnowBERT}([s_i])$. The edge weights $\varepsilon_(i, j)$ between the input vector pairs (a_i, a_j) is cosine similarity between the knowledge-based semantic embeddings

of the input texts.

$$\varepsilon_(i,j) = \begin{cases} cosine(A_i^{cn}, A_j^{cn}) & \text{if } a_i, a_j \in S \\ cosine(A_i^{kbrt}, A_j^{kbrt}) & \text{if } a_i == H, a_j \in S \end{cases}$$

Paragraph Pruning using *Subgraph Pooling*: Long documents are structured as a sequence of paragraphs such that each paragraph may be topically coherent to itself and neighboring paragraphs. As such, paragraphs unrelated to a given hypothesis may be ignored to reduce distractor cues. Graph pooling (Grattarola et al. 2021) is a popular method for graph coarsening. Unlike previous methods such as gPool (Gao and Ji 2019) and SAGPool (Lee, Lee, and Kang 2019) that pool entire graph, we propose attention-based *Subgraph Pooling* layer which can select top rank nodes from a predefined subset of nodes in the graph. *Subgraph Pooling* layer can selectively drop irrelevant paragraph nodes while retaining the remaining paragraph nodes, their corresponding sentence nodes and the hypothesis node in the graph.

Suppose there are N nodes in document graph D_G with node embedding of size C with adjacency matrix $A \in \Re^{NxN}$ and feature matrix $X \in \Re^{NxC}$ .We apply GAT (Veličković et al. 2017) over D_G to obtain self-attention scores Z for all nodes. The pooling ratio η is a hyperparameter that determines the number of paragraph nodes to keep based on the value of Z. We want to select the top-rank nodes only from the set of paragraph nodes. Hence, we use a hard mask $\mu = \{1|x_i \in P \forall X; 0\}$ that is 1 for all paragraph nodes P, otherwise zero. We perform an element-wise multiplication ($\odot$) between attention scores and mask values to get a soft mask $Z_P = Z \odot \mu$. Top-rank operation ranks returns the indices of top η paragraphs based on Z_P. Node indices corresponding to the set of selected top-η paragraphs added to the set of sentence nodes minus those belonging to the pruned paragraphs ($\text{idx}_{S-S_{P-P_\eta}}$) and hypothesis ($\text{idx}_H$) are

selected as follows: idx = top-rank(Z_P, η) + $\text{idx}_{S-S_{P-P_\eta}}$ + idx_H. The combined index tensor (idx) contains the indices of all the nodes selected in the final graph $D_G^{'}$. $X^{'}$(idx, :) and $\widetilde{A} = A(\text{idx}, \text{idx})$ perform the row and/or column extraction to form the adjacency matrix and the feature matrix of $D_G^{'}$. The attention scores for selected nodes Z_{idx} act as gating weights for node features after filtering which controls the information flow and makes the whole procedure trainable by back-propagation as given by: $\widetilde{X} = X^{'} \odot (Z_{idx})$.

Asynchronous Graph Update: Graph Neural Networks (GNN) are useful for multi-hop reasoning on hierarchical graphs comprising of different levels of granularity (questions, paragraphs, sentences, entities) (Fang et al. 2019; Zhang et al. 2020a; Chen et al. 2021). However, GNN's perform message passing synchronously at each step of the graph update, ignoring the fact that different relationship (edge) types may have different priorities. In order to overcome this challenge, we propose to use Asynchronous Graph Update (Li et al. 2021a) to perform sequential graph updates corresponding to all relationship types in $R \in \{R_{str}, R_{top}, R_{ent}, R_{sim}\}$ to enhance the effectiveness of multi-hop reasoning. **Optimal Evidence Selection (E^{NLI})**: To select the set of most relevant evidence sentences E, we hypothesize that a sentence s_i from document D is important for NLI prediction if including the corresponding sentence as part of evidence set can improve the performance of NLI label prediction model (M^{NLI}). We design an iterative process for sentence selection such that at step $k+1$ in the process ($k \geq 0$), a sentence s_i^{k+1} is chosen which has not been selected previously in evidence set $E_k = \{s_1*, \cdots s_k*\}$ at step k. We employ a Long Short Term Memory Network (LSTM) over previously selected k sentences to select a relevant sentence at time step $k+1$. At step 0, the initial hidden state h_0 for LSTM is set to zero. At step $k+1$, we use the hidden state h_k of LSTM from prior step to assign a score sc_i^{k+1} for each sentence node $s_i \in S - E_k$. The sentence with highest selection score is considered for selection

at this step as given by $sc_i^{k+1} = sigmoid(FFN([x_i : h_k]))$ and $s_{k+1}* = argmax_{s_i \in S-E_k}(sc_i^{k+1})$, where FFN is a two-layer feed-forward network. In particular, if selecting $s_{k+1}*$ causes the number of words in the selected sentences so far to exceed the context encoder length limit (eg., 512 tokens for BERT), the selection process stops and $s_{k+1}*$ is not included in the evidence set E (i.e., $E = \{s_1*, \cdots, s_k*\}$ in this case). Otherwise, the selection process continues to the next step and $s_{k+1}*$ will be chosen and included in E (i.e.,$E = \{s_1*, \cdots, s_{k+1}*\}$). The hidden state of LSTM is also updated for the current step, i.e., $h_{k+1} = LSTM(h_k, x_{k+1}*)$, to prepare for the continuation of sentence selection.

Evidence Selection Reward Function: In order to train the evidence selection module, we employ the REINFORCE algorithm (Williams 1992) and incorporate the following information signals in the reward function of REINFORCE to better supervise the training process. In order to train the evidence selection module, we employ the REINFORCE algorithm (Williams 1992). We incorporate the following information signals in the reward function of REINFORCE to better supervise the training process:

(1) Task Reward ϕ_{perf}: We compute this reward based on the NLI task prediction performance. In order to measure the impact of the selected context, we use a T-5 model (Raffel et al. 2019a) pre-trained on MNLI corpus (Williams, Nangia, and Bowman 2017) to predict the NLI label for the given hypothesis + context pair. $\phi_{perf}(E)$ is set to 1 if the final prediction is correct; and 0 otherwise.

(2) Semantic Reward ϕ_{sem}: We propose that the evidence sentences should be semantically similar to the hypothesis. Our motivation is that similar context sentences (e.g., discussing the same events or entities) provide more relevant information for the NLI prediction. We include the semantic similarity between the selected evidence sentences in E and the hypothesis as

measured by the cosine similarity (i.e.,$\odot$) between their sentence embeddings computed using SimCSE[1] (Gao, Yao, and Chen 2021).

(3) Evidence Reward ϕ_{bleu}: We seek to promote evidence sentences having a high overlap with the target ground truth evidence. In many cases, the target evidence length may be way less than 512 token limit. Hence, our motivation is to reward the lexical overlap while penalizing verbosity arising at evidence selection stage. We calculate the BLEU score between the selected evidence E and ground truth evidence E_{gt}: $\phi_{bleu} = BLEU(E, E_{gt})$ This reward can only be applied for cases where ground truth evidence annotation is present.

(4) Multihop Reward ϕ_{mhop}: The motivation for this reward is that a sentence should be preferred to be included in E by the selection process if there are common entities mentions with the hypothesis. Moreover, connected sentences by the virtue of common entity mentions are more likely to refer to the same events. Hence, we leverage the subgraph similarity of the learned node embeddings of the selected evidence and their first degree node connections through entity-centric relations with the hypothesis node in G''_D. We perform max-pooling operation over the concatenated node embeddings of the corresponding evidence sentences and their first degree node connections joined by R_{ent}: $\hat{E} = maxpool(v_1 \bigoplus v_2, \cdots, v_k | s_i \in E, i \in \{1, \cdots, k\})$, where $\bigoplus$ means embedding concatenation. Finally, we compute the dot-product between $\hat{E}$ and node embedding of the hypothesis node h as $\phi_{mhop} = \hat{E}.h$.

4.3.1 Training DocInfer

NLI Prediction Loss: We combine the final representations corresponding to the learnt graph structure (g_{out}) and selected evidence text (t_{out}). We aggregate the embeddings corresponding to

[1] https://github.com/princeton-nlp/SimCSE

the selected sentence nodes in $D_G^{''}$ and the hypothesis node using a summation-based graph-level readout function (Xu et al. 2018b) as $g_{out} = \rho(\sum_{v \in D_G^{''}} W_g V_i^T)$. The words in the evidence sentences are joined in order of their appearance in document D and input to the context encoder $t_{out} = Encoder([CLS]s_1; s_2, \cdots, s_k)$. g_{out} and t_{out} are concatenated and passed through two dense fully-connected layers: $z = ReLU(Dense(t_{out} \bigoplus g_{out})$. This is followed by a Softmax layer to predict entailment/contradiction/neutral by utilizing the negative log-likelihood loss: $L_{pred} = -P(y|z)$.

Evidence Selection Loss: The overall reward function to train our evidence selection module is $\phi(E) = \phi_{perf} + \phi_{sem} + \phi_{bleu} + \phi_{mhop}$. Using REINFORCE, we seek to minimize the negative expected reward $\phi(E)$ over the possible choices of E as $L_{sent} = -\mathbb{E}_{E \sim P(E|D,H)}[\phi(E)]$, and $L_{sent} = -\mathbb{E}_{E \sim P(E|D,H)}[\phi(E)] \nabla \log(P(E|H, D))$. Finally, the probability of the selected sequence E is computed via $P(E|H, D) = \prod_{k=0,\cdots,K-1} P(s_{k+1} * |H, D, s_{i \leq k} *)$, which is obtained via softmax over selection scores for sentences in S at selection step $k + 1$.

Joint NLI Prediction and Evidence Selection: During training, the NLI prediction model M^{NLI} and the evidence selection module E^{NLI} are trained alternatively. At each update step, E^{NLI} first selects optimal evidence sentences E that form a shortened document D^e. M^{NLI} uses E to predict the NLI label. The parameters of M^{NLI} are updated using the gradient of NLI prediction loss L_{pred}, keeping the parameters of the evidence extraction module constant. Next, the parameters of the evidence selection module are updated using the gradient of L_{sent}, keeping parameters of M^{NLI} constant. This process repeats until convergence. At test time, evidence sentences are first selected and then consumed by the prediction model to perform NLI prediction.

4.4 Experiments

4.4.1 Datasets for Document-level NLI

We use the following three datasets to benchmark document-level NLI approaches. **(1) DocNLI** (Yin, Radev, and Xiong 2021): A large-scale document-level NLI dataset obtained by reformatting mainstream NLP tasks such as question answering and document summarization. **(2) ContractNLI** (Koreeda and Manning 2021): NLI dataset of 607 contract documents annotated with ground truth evidence sentences. **(3) ConTRoL (Liu et al. 2021a)**: A passage-level NLI dataset of exam questions that requires logical, analytical, temporal, coreferential reasoning, and information integration over multiple premise sentences. **(4) CaseHoldNLI**, the fourth and novel NLI dataset introduced in this paper, in the legal judicial reasoning domain for identifying the governing legal rule (also called "Holding") applied to a particular set of facts. It is sourced from the CaseHOLD dataset (Zheng et al. 2021) comprising over 53,000+ multiple choice questions. Each multiple choice question comprises of a snippet from a judicial decision along with 5 semantically similar potential holdings, of which only one is correct. We obtain the NLI-version by combining the question and the positive (negative) answer candidate as a positive (negative) hypothesis. To evaluate the dataset quality, we asked an expert to select the NLI using only the hypothesis for 10% of the test data sampled at random. The poor performance of this human baseline ($\sim 0.24F1$) validates that the dataset doesn't suffer from hypothesis bias. CaseHoldNLI dataset is comparable to challenging document-level NLI datasets with average premise length at document-scale and exceeds the maximum input length limit of BERT models. We report train/dev/test splits of each dataset.

	System	DocNLI	
		Dev F1	Test F1
Baselines	Majority	19.7	19.9
	BERT (Hypothesis-only)	21.9	22.0
	BERT$_{base}$ (Devlin et al. 2018)	63.1	60.1
	BERT$_{large}$ (Devlin et al. 2018)	63.5	61.1
	RoBERTa$_{base}$ (Liu et al. 2019a)	61.0	59.5
	RoBERTa$_{large}$ (Liu et al. 2019a)	63.1	61.3
	T5 (Raffel et al. 2019b)	62.9	61.1
	Longformer (Beltagy, Peters, and Cohan 2020)	46.1	44.4
	GEAR (Zhou et al. 2019)	67.8	63.3
	KGAT (Liu et al. 2019c)	68.5	64.8
	HESM (Subramanian and Lee 2020)	68.9	65.0
	DREAM (Zhong et al. 2020)	69.7	65.9
	TARSA (Si et al. 2021)	70.4	66.4
	EvidenceNet (Chen et al. 2022)	72.6	68.5
Ours	**DocInfer (w/ RoBerta)**	**75.5**	**72.3**

Table 4.1: Results comparing performance of DocInfer with baselines on DocNLI dataset. **Bold** denotes the best-performing model. LightCyan and Yellow show best performing baseline and Transformer model.

	System	ContractNLI				
		Acc (%)	F1 (C)	F1 (E)	mAP	PR@80
Baselines	Majority	67.4	8.3	42.8	-	-
	BERT$_{large}$ (Devlin et al. 2018)	77.5	25.7	76.4	0.822	0.763
	T5 (Raffel et al. 2019b)	73.2	21.2	69.1	0.786	0.575
	Longformer (Beltagy, Peters, and Cohan 2020)	71.2	19.2	70.4	0.755	0.648
	BigBird (Zaheer et al. 2020b)	71.5	18.8	70.9	0.776	0.630
	GEAR (Zhou et al. 2019)	78.4	26.9	78.3	0.909	0.774
	KGAT (Liu et al. 2019c)	78.9	27.8	79.2	0.914	0.773
	HESM (Subramanian and Lee 2020)	28.2	79.5	79.9	0.916	0.789
	DREAM (Zhong et al. 2020)	79.8	29.3	80.4	0.919	0.786
	TARSA (Si et al. 2021)	80.4	29.4	80.5	0.916	0.783
	SpanNLI-Bert$_{large}$ (Koreeda and Manning 2021)	87.5	35.7	83.4	0.922	0.793
Ours	**DocInfer (w/ Bert)**	**91.8**	**38.2**	**89.1**	**0.956**	**0.832**

Table 4.2: Results comparing performance of DocInfer with baselines on ContractNLI dataset. **Bold** denotes the best performing model. LightCyan and Yellow show best performing baseline and Transformer model.

	System	ConTRoL				
		Acc (%)	F1 (E)	F1 (N)	F1 (C)	F1 (O)
Baselines	Majority	40.6	57.7	0.0	0.0	19.2
	BERT_{base} (Devlin et al. 2018)	47.4	42.4	50.2	46.0	46.2
	BERT_{large} (Devlin et al. 2018)	50.6	45.9	53.1	49.3	49.4
	RoBERTa_{base} (Liu et al. 2019a)	45.9	45.3	45.9	45.6	45.6
	BART (Lewis et al. 2020a)	56.3	49.1	59.5	53.8	54.0
	Longformer (Beltagy, Peters, and Cohan 2020)	49.8	45.6	46.8	46.2	46.2
	BigBird (Zaheer et al. 2020b)	49.3	46.0	45.1	46.0	46.1
	BART-NLI (Liu et al. 2021a)	57.2	49.0	60.4	54.2	54.5
	BART-NLI-FT (Liu et al. 2021a)	57.5	49.3	60.6	54.6	55.0
	KGAT (Liu et al. 2019c)	59.1	50.6	61.8	55.7	56.6
	HESM (Subramanian and Lee 2020)	59.3	50.9	62.3	56.1	56.6
	DREAM (Zhong et al. 2020)	59.8	51.1	62.0	56.1	56.3
	HGRGA (Lin and Fu 2022)	60.6	52.9	62.4	58.7	58.0
	EvidenceNet (Chen et al. 2022)	61.8	56.4	64.2	64.3	61.6
Ours	**DocInfer (w/ BART)**	**66.7**	**60.6**	**67.1**	**69.6**	**67.4**

Table 4.3: Results comparing performance of DocInfer with baselines ConTRoL dataset. **Bold** denotes the best-performing model. LightCyan and Yellow show best performing baseline and Transformer model.

	System	CaseHoldNLI		
		P	R	F1
Baselines	Majority	0.0	1.0	0.0
	BERT_{base} (Devlin et al. 2018)	42.2	46.3	44.2
	RoBERTa_{base} (Liu et al. 2019a)	42.2	46.3	44.2
	T5 (Raffel et al. 2019b)	41.5	43.5	42.5
	Legal-BERT (Zheng et al. 2021)	46.5	47.9	47.1
	Longformer (Beltagy, Peters, and Cohan 2020)	40.1	43.3	41.6
	GEAR (Zhou et al. 2019)	42.9	46.8	44.8
	HESM (Subramanian and Lee 2020)	44.0	48.0	45.9
	HGRGA (Lin and Fu 2022)	45.4	49.4	47.3
	EvidenceNet (Chen et al. 2022)	47.3	50.5	48.8
Ours	**DocInfer (w/ Legal-Bert)**	**51.3**	**53.1**	**52.2**

Table 4.4: Results comparing performance of DocInfer with baselines on CaseHoldNLI dataset. **Bold** denotes the best-performing model. LightCyan and Yellow show best performing baseline and Transformer model.

4.4.2 Experiments on Downstream Tasks

(1) Fact Verification: The NLI-version of FEVER (Thorne et al. 2018) task, released by (Nie, Chen, and Bansal 2019), considers each claim as a hypothesis while the premises consist of ground truth textual evidence and other randomly sampled related text.

(2) Multi-choice Question Answering: The NLI-version MCTest (Richardson, Burges, and Renshaw 2013) combines the question and the positive (negative) answer candidate as a positive (negative) hypothesis. Presence of limited labeled data makes them both good benchmarks to investigate the performance of document-level NLI models on annotation-scarce tasks. We evaluate `DocInfer` trained on DocNLI dataset and report F1 scores for both tasks. We follow the "FEVER-binary" and "MCTest-NLI" settings proposed in (Yin, Radev, and Xiong 2021).

(3) Contract Clause Retrieval (Borchmann et al. 2020): is a task to identify spans in a target document representing clauses analogous (i.e. semantically and functionally equivalent) to the provided seed clauses from source documents. We reformulate this as an NLI task where the seed clauses are concatenated to form the hypothesis, and the target document is the premise. We test the evidence selection capabilities of `DocInfer` trained on ContractNLI dataset for identifying relevant sentence-level spans in the premise for the clause retrieval task. The dataset has 1300 examples each for validation and test to tune and test the paragraph selection hyperparameter η. We followed the evaluation framework specified in (Borchmann et al. 2020) of few (1-5) shot setting and report Soft F1 score.

4.5 Results

Table 4.1-4.4 compares the performance of `DocInfer` against other baselines on DocNLI, ContractNLI, ConTRoL, and CaseHoldNLI datasets. Similar to (Yin, Radev, and Xiong 2021), we truncate the hypothesis-premise pair sequence to the appropriate maximum input length for input to Transformer models. BERT (Devlin et al. 2018), RoBERTa (Liu et al. 2019a), DeBERTa (He et al. 2020), BART (Lewis et al. 2020a) show superior performance for DocNLI, ContractNLI, and ConTRoL datasets, respectively. Legal-BERT (Chalkidis et al. 2020) outperforms other Transformer language models on CaseHoldNLI dataset due to its high domain-specificity of legal language. However, they are challenged by their input length restriction of 512 tokens for contextually reasoning over long premise lengths. Consistent with observations of (Yin, Radev, and Xiong 2021), large input Transformer models such as Longformer (Beltagy, Peters, and Cohan 2020) and BigBird (Zaheer et al. 2020b) that can handle up to 4096 tokens underperform traditional BERT-like models on all four datasets. We attribute this to the presence of distractors in long documents and the inability of these models to reason in a multihop fashion. BART-NLI which is pretrained on sentence-level NLI (Liu et al. 2021a) improves over naive Transformers but still struggles due to limited captured context.

We also re-purpose several strong baseline methods from the Fact Extraction and Verification (FEVER 1.0) task. by reformulating the document retrieval and claim verification steps to paragraph retrieval and textual entailment, respectively. GEAR, KGAT, and HGRGA model the document as a dense fully-connected graph, leading to distractor interactions confounding the reasoning process. They are also devoid of linguistic information about entities, topics or commonsense knowledge. HESM uses document chunking which hinders contextual reasoning

	System	DocNLI		ContractNLI					ConTRoL					CaseHoldNLI		
		Dev F1	Test F1	Acc (%)	F1 (C)	F1 (E)	mAP	PR@80	Acc (%)	F1 (E)	F1 (N)	F1 (C)	F1 (O)	P	R	F1
	Context Encoder	RoBERTa		BERT					BART					Legal-BERT		
Ours	DocInfer	75.5	72.3	91.8	38.2	89.1	0.956	0.832	66.7	63.6	67.1	69.6	67.4	51.3	53.1	52.2
Ablation	DocInfer w\Concept Relations	72.6	70.7	90.7	35.4	87.2	0.928	0.810	62.3	62.6	66.6	66.1	64.2	50.6	52.9	51.7
	DocInfer w\Topical Relations	72.2	70.4	90.6	33.6	86.6	0.925	0.819	60.5	59.4	66.7	63.2	63.1	49.5	51.5	50.5
	DocInfer w\Entity Relations	72.5	70.2	90.2	31.1	85.7	0.921	0.812	59.8	58.8	66.0	62.8	62.5	49.0	51.4	50.2
	DocInfer w\o Asynchronous Graph Update	73.2	71.5	89.5	36.3	84.3	0.923	0.813	57.6	61.0	59.7	65.1	64.8	48.8	50.5	49.6
	Greedy Evidence Selection	67.5	64.9	88.3	36.0	84.1	0.876	0.780	56.5	56.3	56.1	55.2	55.8	46.8	46.2	46.7
	DocInfer w\o Paragraph Pruning	65.6	64.5	85.4	35.9	83.9	0.855	0.742	51.8	47.0	45.5	48.0	46.8	44.8	46.2	45.5
	DocInfer w\o Evidence Selection	65.0	63.6	83.7	35.5	83.5	0.825	0.715	51.6	46.7	45.0	47.8	46.5	44.4	45.9	45.1
	DocInfer w\Task Reward	70.5	68.5	90.1	33.9	86.7	0.907	0.769	61.9	61.0	64.0	65.2	63.4	47.4	47.4	47.4
	DocInfer w\Evidence Reward	-	-	90.8	36.4	87.5	0.916	0.805	-	-	-	-	-	-	-	-
	DocInfer w\Semantic Reward	71.4	69.0	90.6	34.2	84.4	0.912	0.778	63.3	62.5	63.9	64.7	63.7	46.6	46.8	46.7
	DocInfer w\Multihop Reward	71.6	69.5	90.5	35.5	85.6	0.910	0.785	61.6	62.0	63.2	64.0	63.0	46.1	46.9	46.5

Table 4.5: Results comparing ablative components of DocInfer model and analysis of using a single reward/relation at a time. Darker green represents better F1 performance, darker red shows negative impact. Evidence reward is applicable only for ContractNLI which has ground truth evidence annotations.

for far-away chunks. DREAM and TARSA use semantic role labeling and topic modeling, respectively, to identify phrase interaction but lack entity-level information required to resolve coreferences across document. EvidenceNet and SpanNLI emerge as strong baseline models for our work. DocInfer outperforms SpanNLI and EvidenceNet due to its ability to iteratively select important evidence sentences in the premise and simultaneously utilize multihop interactions between related evidences. **Impact of Input Length**: DocInfer achieves SOTA performance on all four datasets and maintains steady improvements over corresponding baseline models with increasing in input lengths. **Choice of context encoder in NLI prediction**: One of the merits of the our approach is that it is extensible and can utilize any domain-specific transformer language models for context encoding to further augment performance. We evaluate the choice of context encoder for different datasets. DocInfer gives SOTA performance using RoBERTa for DocNLI, BERT for ContractNLI, BART for ConTRol, and Legal-BERT for CaseHoldNLI, in the prediction model.

Ablation Study of DocInfer: Table 7.3 shows ablations for the document graph relations, module components, and reward functions. We observe that concept relation is critical in all data settings due to the need for external knowledge-based semantic representation for

connecting related concepts across sentences. Removing any of the relations does not degrade the performance below EvidenceNet (Chen et al. 2022) or SpanNLI baselines. This is important for adapting our method to new domains where existing linguistic parsers maybe noisy or non-existent. Cells in Table 7.3 highlighted in red shows the ablation of individual components such that removing paragraph pruning mechanism severely deteriorates model performance as the model has to evaluate an exponentially larger number of candidate evidences during evidence selection stage. In absence of optimal evidence selection, we treat evidence extraction as a binary classification task over each sentence node along with NLI label given by the "readout" function similar to KGAT (Liu et al. 2019c). The severe performance drop of `DocInfer` model in absence of evidence selection component highlights its importance for document NLI task. Asynchronous graph update adds incremental value to `DocInfer` owing to its relation-specific message passing. Evidence Selection and Paragraph Pruning components are most critical for SOTA performance of `DocInfer`. Greedy selection instead of REINFORCE significantly decreases performance. Concept relations are most beneficial for `DocInfer`, followed by topical and entity relations. Evidence, semantic, multihop and task rewards most help ContractNLI, ConTRoL, DocNLI, and CaseHoldNLI.

Impact of reward function: Table 7.3 shows that removing any reward component (i.e., task, semantic, evidence, multihop) significantly hurts the overall performance, thus clearly demonstrating their individual importance. To assess the necessity of the multi-step selection using REINFORCE, we eliminate the multistep selection strategy and perform a one-shot sentence selection where the top k sentences with the highest selection scores from the first step are selected. We call this setting greedy evidence selection and show that the elimination of multistep selection drops performance, suggesting that selecting sentences incrementally conditioning on

System	Fine-tune	FEVER binary	MCTest v160	MCTest v500
RoBERTa	✗	88.4	90.0	85.8
EvidenceNet	✗	88.7	90.6	86.0
DocInfer†	✗	**89.2***	**91.0***	**86.4***
DocInfer† w/o *R*	✗	86.3	87.5	82.5
RoBERTa	✓	89.4	91.0	91.0
EvidenceNet	✓	89.9	90.8	90.6
DocInfer†	✓	**90.5***	**91.5***	**91.2***
DocInfer† w/o *R*	✓	86.6	88.5	87.5

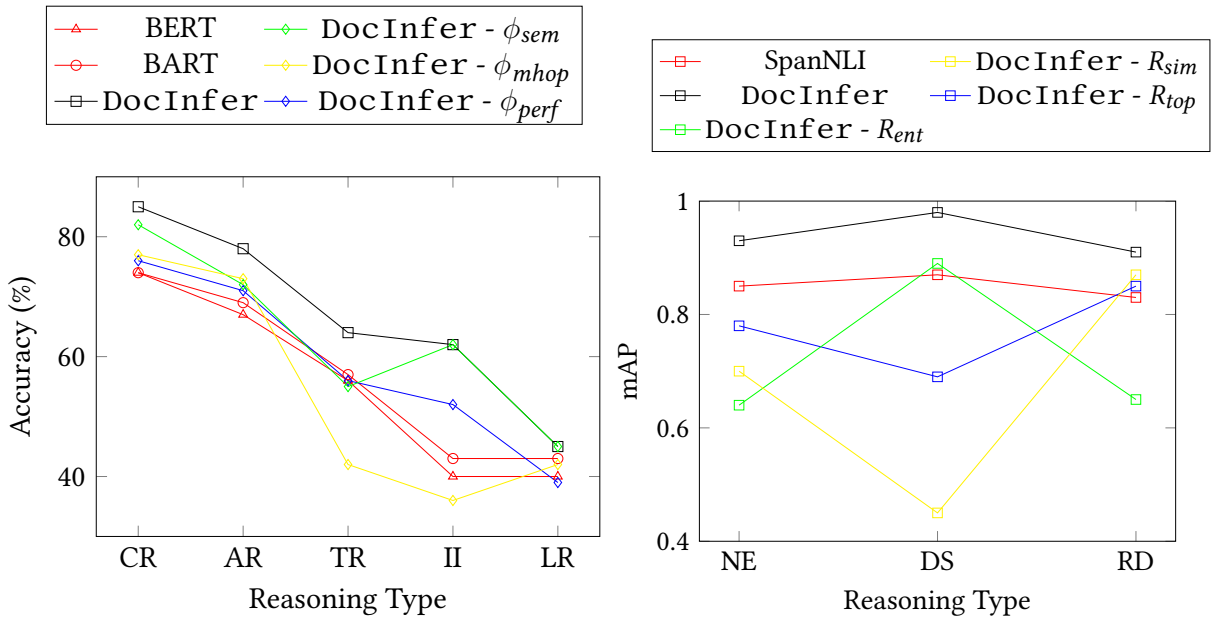
Table 4.6: Performance comparison of DocInfer† with RoBERTa (large) and EvidenceNet on FEVER-binary and MCTest-NLI. † means using RoBERTa as a context encoder.

previously selected sentences is advantageous.

Performance of DocInfer on downstream tasks: Table 4.6 shows the evaluation of DocInfer along with RoBERTa-large and EvidenceNet (Chen et al. 2022) baselines and RoBERTa model from (Yin, Radev, and Xiong 2021) on FEVER-binary and MCTest tasks.

We train all models on the DocNLI dataset to benefit from cross-task transfer and to minimize domain shift. We then infer all models in two settings: (i) without task-specific fine-tuning, and (ii) with fine-tuning on the end task. DocInfer model consistently outperforms baselines across both tasks in case of without fine-tuning (FEVER-binary: +0.8 F1, MCTest v160: +1 F1, MCTest v500: +0.6 F1) and with fine-tuning (FEVER-binary: +0.9 F1, MCTest v160: +0.5 F1, MCTest v500: +0.2 F1). We observe that both tasks require the models to capture topic coherence, knowledge-based semantics, and entity interactions as removing graph relations severely degrades the performance.

Evidence selection for clause retrieval focuses on selecting evidence spans in the target



(a) ConTRoL Dataset

(b) ContractNLI Dataset

Figure 4.2: Error analysis across reasoning types (accuracy%) and challenging phenomenon (mAP) on the test set of ConTRoL and ContractNLI datasets.

document (premise) given the entailment relation with seed clauses (hypothesis). The task is unsupervised in nature (has no training set). We test the evidence selection module (E^{NLI}) of the DocInfer model and its ablated variants (without paragraph pruning and reward functions), all pre-trained on the ContractNLI dataset. Table 4.7 shows that DocInfer model with BERT as the context encoder outperforms strong baselines by approximately 5%. Removing paragraph pruning significantly degrades the performance, highlighting the need to prune distractor paragraphs for retrieving relevant information. The presence of each reward function to maintain the performance of DocInfer indicates the linguistic importance of each reward. Formulating the task as NLI helps contextualize the seed clauses with the premise as opposed to earlier techniques of isolated vectorization and naive aggregation by (Borchmann et al. 2020).

Qualitative Analysis: Figure 4.2 shows qualitative analysis across different reasoning types on the test set of the ConTRoL dataset. The results provide evidence that the multihop and semantic

System	Soft F1
Tf-IDF	0.39
GloVe (300D, EDGAR)	0.41
Sentence-BERT	0.32
USE	0.38
BERT	0.35
RoBERTa	0.31
GPT-1	0.49
GPT-2 (large)	0.51
DocInfer‡ (pretrain ContractNLI)	**0.53***
DocInfer‡ w\o Paragraph Pruning	0.42
DocInfer‡ w\o Task Reward (ϕ_{perf})	0.48
DocInfer‡ w\o Semantic Reward (ϕ_{sem})	0.45
DocInfer‡ w\o Evidence Reward (ϕ_{bleu})	0.45
DocInfer‡ w\o Multihop Reward (ϕ_{mhop})	0.44
Human	0.84

Table 4.7: Performance comparison of DocInfer and its configurations pretrained on ContractNLI and tested for clause retrieval without fine-tuning on Contract Discovery dataset (Borchmann et al. 2020). ‡: BERT as context encoder.

similarity rewards are important for coreference reasoning (CR) due to reasoning over multiple mentions and noun phrases. Multihop reward also helps improve Information aggregation (II) which requires combining information from multiple paragraphs. Task reward benefits logical reasoning as it focuses on logical inference of human language. DocInfer is unable to handle temporal and analytical reasoning cases. We further analyze the evidence extraction mAP on the ContractNLI dataset across diverse challenging phenomena. Entity relations are critical for resolving reference to definitions (RD) as they are anchored together through common mentions. Concept similarity links play an important role in resolving information spread out between discontinuous spans based on commonsense reasoning. DocInfer handles evidence identification for all studied phenomena better than SpanNLI.

4.6 Conclusion and Future Work

We introduce DocInfer, a document-level NLI model that uses enriched hierarchical document graph through inter-sentence relations, performs paragraph pruning using *SubGraph Pooling* layer, and optimally selects evidence sentences using REINFORCE algorithm to outperform SOTA methods on four doc-NLI datasets, including our propose CaseHoldNLI on legal judicial reasoning.DocInfer is useful for downstream fact verification, multi-choice QA and legal clause retrieval tasks. For future work, we intend to integrate temporal knowledge and analytical reasoning into our model to improve the performance.

graphicx enumitem

CHAPTER 5

LayerDoc: Layer-wise Extraction of Spatial Hierarchical Structure in Visually-Rich Documents

Abstract

Digital documents often contain images and scanned text. Parsing such visually-rich documents is a core task for workflow automation, but it remains challenging since most documents do not encode explicit layout information, e.g., how characters and words are grouped into boxes and ordered into larger semantic entities. Current state-of-the-art layout extraction methods are challenged by such documents as they rely on word sequences to have correct reading order and do not exploit their hierarchical structure. We propose `LayerDoc`, an approach that uses visual features, textual semantics, and spatial coordinates along with constraint inference to extract

the hierarchical layout structure of documents in a bottom-up layer-wise fashion. LayerDoc recursively groups smaller regions into larger semantic elements in 2D to infer complex nested hierarchies. Experiments show that our approach outperforms competitive baselines by 10-15% on three diverse datasets of forms and mobile app screen layouts for the tasks of spatial region classification, higher-order group identification, layout hierarchy extraction, reading order detection, and word grouping.

5.1 Introduction

Structured documents such as forms, invoices, receipts, resumes, contracts and web/app screen interfaces are ubiquitously used in industry (Harley, Ufkes, and Derpanis 2015) and contain a rich variety of components such as tables, check boxes, widgets, buttons, input fields. Structured documents make use of spatial layout to convey information through potentially nested spatial grouping. However, digital documents (eg. PDF) generally discard most structure and encode only low-level binary information, while document images produced by a scanner or mobile phone scan app are stored in rasterized format (as pixels). Neither of these document formats encode spatial structure explicitly to identify which pieces of text belong together. This leads to challenges for state-of-the-art information extraction techniques, which generally assume that the reading order of text is known (Wang et al. 2021a).

A number of techniques–e.g., LayoutLM (Xu et al. 2020), LayoutLMv2 (Xu et al. 2021), DocStruct (Wang et al. 2020b), Form2Seq (Aggarwal et al. 2020a)–model the textual semantics, visual appearance, and spatial location of text to solve sequence labeling and classification tasks. These techniques are able to model spatial information implicitly to assign semantic labels to

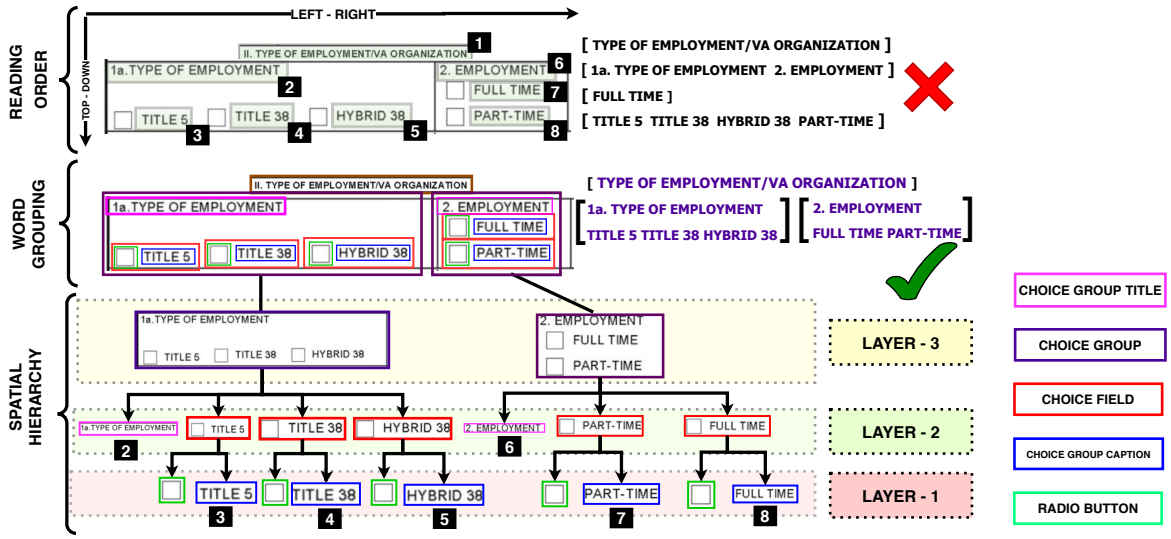


Figure 5.1: Example of a scanned form document showing the true reading order using numbered black boxes; word grouping based on spatial arrangement; spatial document hierarchy of elements. Reading order extracted naively in a linear fashion (top → down, left → right) is incorrect (top-right). However, the document can be decomposed into a hierarchy, where text fragments group into choice group caption, radio buttons, and choice labels grouped into choice fields, etc. `LayerDoc` extracts this spatial hierarchy to group elements and assign them correct semantic labels. The correct reading order is obtained by leaf node traversal of the hierarchy.

words, classify a sequence of words (or sub-word tokens), or predict relationships between given regions. However, these methods do not infer the 2D grouping of individual words into semantic elements (e.g., DocStruct assumes candidate regions are provided as preprocessed inputs), nor do they produce the nested structure of a document as output. While LayoutLM is capable of grouping multiple word or sub-word tokens into semantic elements via BIO encoding, the encoding assumes that the reading order of input tokens is correct–but reading order itself is dependent on the structure of the document and is not known, and most OCR systems cannot infer it correctly for complex spatial structure (Clausner, Pletschacher, and Antonacopoulos 2013).

To illustrate the importance of modeling the structure of a document, consider the example shown in Figure 5.1. For the use case of digital form authoring, where the goal is to convert a

scanned form into a digital format, an algorithm would need to extract characters/words, group them into semantic elements (e.g., a choice label), and further group them into larger elements (a label and the checkbox to its left form a *choice field* element, multiple choice fields form a *choice group*, etc). All of these nested group relationships are important since the labels need to be displayed next to the corresponding checkboxes, and the choice group must consist of mutually exclusive choices that affect the UI, as checking one box should cause the other boxes in the group to be unchecked. Besides form authoring, other uses of this type of structure include reflowability across devices (Gupta et al. 2007; Khemakhem, Herold, and Romary 2018), adaptive editing of user-interfaces (Murray 1999), and improving accessibility for user-interactions (Zhang et al. 2021a)

Even for other extraction tasks, where the structure itself is not of interest, an understanding of the hierarchical arrangement of text regions is useful for the purpose of producing sequences with accurate reading order. This is important for modern Transformer based language models such as BERT (Devlin et al. 2019a) and LayoutLM (Xu et al. 2020) which depend on the correct order of the input text for downstream tasks and are sensitive to incorrect order (Hong et al. 2021). Once the hierarchical structure is extracted as in Figure 5.1, a traversal of the structure can produce reading order that respects group structure and avoids the errors that OCR algorithms would produce.

We propose **LayerDoc**, a model that uses multimodal deep learning on visual features, textual semantics and spatial geometry as well as constraint inference to generate a complete bottom-up ordered hierarchical arrangement of document layout structure. Within this hierarchy, each node is a rectangular region which is assigned a semantic label, with the leaf nodes consisting of OCR tokens or embedded images. This structure is generated in a layer-wise fashion: given

an input set of regions, LayerDoc hypothesizes candidate 2D groupings of these regions without the need for IOB tagging, evaluates candidate parent-child links between a child region and parent region (the group it belongs to), then commits to a global parent-child assignment through constraint optimization. The multi-modal nature of LayerDoc benefits not only those cases where spatial signals are effective (e.g., where layout based models excel) but also where visual and textual signals are needed, as evident from experiments on diverse datasets of semi-structured forms and scanned user-interfaces. Our novel **contributions**:

1. We propose LayerDoc for extracting hierarchical document layout in a layer-wise fashion, recursively grouping smaller spatial regions into larger, semantic elements. We are the first to formulate **nested document hierarchy extraction using transformers**.
2. We propose a multimodal contextual encoder that **maximizes use of context by simultaneously modeling all possible parent-child pairs in a layer**. For element type classification and semantic grouping, this leads to a relative improvement of 10-15% across several metrics.
3. We demonstrate how our **extracted nested hierarchical document structure can improve the inferred token reading order and semantic word grouping** by 8-12%.

5.2 Related Work

Document layout hierarchy extraction involves two main tasks: spatial element detection and spatial region relationship extraction. Early works (Lebourgeois, Bublinski, and Emptoz 1992; Simon, Pret, and Johnson 1997; Ha, Haralick, and Phillips 1995) used heuristics for both tasks independently, which were later replaced by computer vision models (object detectors) (Yang et al. 2017; He et al. 2017; Deng et al. 2018; Liao et al. 2017) to detect lower-level elements

and group them based on spatial overlap. (Li et al. 2020c) utilized Faster-RCNN (Ren et al. 2015) for document object detection. Recent 2D transformer-based object detectors such as DETR (Carion et al. 2020) do not explicitly model the visual hierarchy or leverage multimodal (semantic, spatial and visual) information or contextual modeling. Transformer-based models such as LayoutLM (Xu et al. 2020), LayoutLMv2 (Xu et al. 2021), LamBERT (Garncarek et al. 2021), DocFormer (Appalaraju et al. 2021), BROS (Hong et al. 2020), and TILT (Powalski et al. 2021), have been used for sequence labeling and classification of spatial regions in documents. However, they do not reason about hierarchy or grouping in an end-to-end fashion. Form2Seq (Aggarwal et al. 2020a) and MMPAN (Aggarwal et al. 2020b) extracted limited types of higher-order structures (Choice Groups, Text Fields and Choice Fields) in form documents. Although Form2Seq utilized a seq2seq network to leverage context, it could not be applied in general settings for end-to-end document spatial hierarchy construction. Recently, DocStruct (Wang et al. 2020b) proposed a multimodal model for extracting parent-child relationships between regions. However, it does not utilize the context of neighboring spatial regions for link prediction, nor does it predict the parent element type, as it is designed for naive key-value pair extraction. Our method uses Transformers to analyze multimodal contextual input from lower-level elements to detect and classify higher-level elements, and reconstruct all layers of the layout hierarchy.

5.3 Methodology

The document hierarchy is constructed by iteratively grouping elements (“child-boxes") in the current layer into larger regions (“parent-boxes") in the next layer. The child-boxes in the first layer consist of elementary tokens extracted directly from a document page image: textual

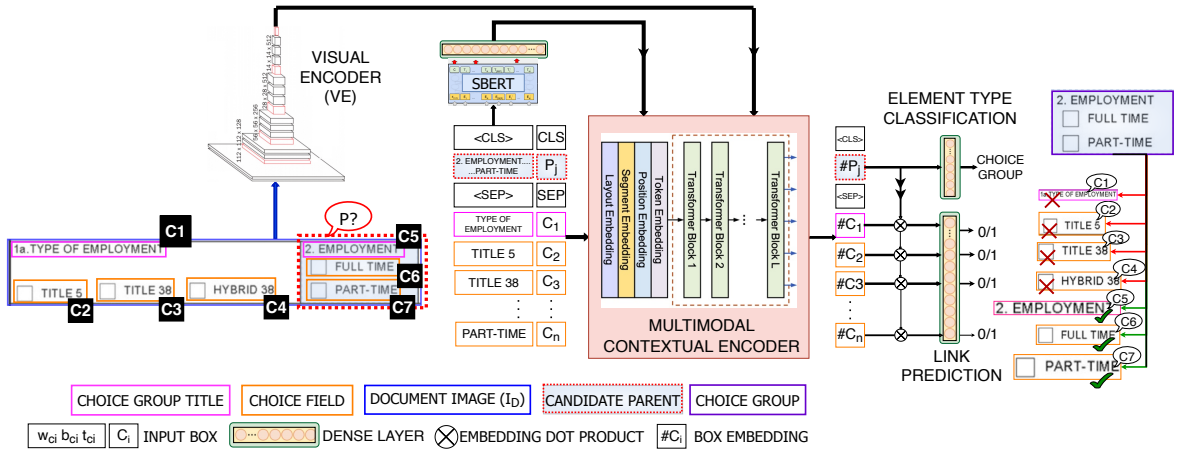


Figure 5.2: LayerDoc takes raw documents and OCR'ed text as inputs and outputs the spatial hierarchy by grouping lower-level elements into parent boxes, predicting parent-child links and the element type of the parent boxes. The model operates on one layer at a time, considering all child boxes C_i in one layer and candidate parent boxes P_j in the next layer. The model encodes the visual, textual, and spatial features of each element into a sequence to compute an encoding, which is then used for link prediction and element type classification. Candidate parents are generated using the child boxes C_i, the final parent set is selected via constraint optimization, and the model is applied recursively to build the hierarchy bottom-up.

tokens are extracted by an off-the-shelf OCR system and visual regions (e.g., widgets, radio-buttons, and embedded images in the form use case) predicted by a high-precision object detector. For intermediate layers, our approach hypothesizes a high-recall set of geometrically feasible "potential parent-boxes" directly from the child-boxes, such that each box can group one or more child-boxes and form the next layer in the hierarchy. At the core of our approach is a multimodal model (illustrated in Fig 5.2 and described in Sec 5.3.1) that predicts links between a potential parent-box and all of its child-boxes in consecutive layers and jointly predicts the semantic label of the parent box. Not all potential parent-boxes are actual elements, so we use constraint inference to keep the parent-boxes that maximize the child-box link probabilities and satisfy hierarchical constraints. This process is repeated one layer at a time, starting from the lowest layer of elementary tokens and recursively grouping the lower-level elements into higher-level

constructs to form a hierarchical arrangement of spatial boxes (see Sec 5.3.3). We next formalize the problem and provide model details.

Problem Statement: Let I_D represent the input document page of which elementary tokens (OCR, embedded widgets, and icons) and their rectangular bounding boxes are extracted by OCR and a high precision object detector, respectively. The ground truth document hierarchy for a scanned document comprises of spatial boxes b_i, each represented by its coordinates (x_1, y_1, x_2, y_2), where (x_1, y_1) and (x_2, y_2) are the top-left and bottom-right coordinates, respectively. Each box has a predefined type label t_i. The textual content (w_i) present in a box is acquired by linearly serializing OCR text tokens lying within the box boundaries. The constituent bounding boxes are arranged in a tree-like format where a box in a higher layer may be a parent of one or more boxes in the layer immediately below it. Thus, each box of the document hierarchy tree contains the list of nested child boxes contained within such that: (i) each child-box is grouped into one and only one parent box i.e. the parent-boxes do not mutually overlap, and (ii) each parent-box groups together all geometrically possible child-boxes within its bounds. Unlike previous works (Wang et al. 2020b; Wang et al. 2021b), this task does not assume the ground truth parent bounding boxes in each layer to be previously known as part of the input at test time.

5.3.1 LayerDoc Model

We denote the set of n child boxes serialized in a left-to-right and top-to-bottom fashion in the k^{th} layer as $c_i \in \{c_1, c_2 \cdots c_n\}$ and the j^{th} potential parent box candidate under consideration as p_j. We represent each box with three input modalities: (i) Semantic Cues, (ii) Spatial Cues, and (iii) Structural Cues. We also utilize the visual encoding of the entire scanned document image

to augment the spatial and semantic signals with visual cues.

Semantic Cues: Using an off-the-shelf pre-trained language model (*SBert*), we encode the textual content of each box (w_i) into a sentence embedding $s = SBert([[CLS], w_i])$ of dimension $1 \times d_S$, where d_S is the hidden states of pre-trained language model. We concatenate the sentence embedding of the potential parent box s_{p_j} with the sequence of sentence embeddings of child boxes $(s_{c_1}, s_{c_2}, \cdots s_{c_n})$ and pass them through a fully connected layer to form the semantic input sequence $S_j^n = \sigma(W_1([s_{p_j} \oplus s_{c_1} s_{c_2} \cdots s_{c_n}]) + \delta_1)$, where W_1, δ_1, $\sigma(\cdot)$, and $\oplus$ denote the weight matrices, bias, Sigmoid activation function, and concatenation, respectively.

Spatial Cues: We extract the bounding box coordinates to derive the relative layout information of each box. Each bounding box b is represented through its upper-left $([x_1, y_1])$ and bottom-right $([x_2, y_2])$ co-ordinates that are normalized, $b = [\frac{x_1}{W}, \frac{y_1}{H}, \frac{x_2}{W}, \frac{y_2}{H}]$, where H and W are the height and width of the scanned document page. The normalized parent bounding box b_{p_j} is concatenated with the sequence of normalized child bounding boxes $(b_{c_1} b_{c_2} \cdots b_{c_n})$ to form the spatial input sequence $B_n^j = [b_{p_j} \oplus b_{c_1} b_{c_2} \cdots b_{c_n}]$.

Structural Cues: Each child box has a box type t. The parent box type is not known at input. Hence, it is represented by a dummy value of < PBOX > in the input sequence. We concatenate the category types of the parent box followed by the linearly serialized child boxes to obtain the structural input sequence $T_j^n = [<\text{PBOX}>: t_{c_1} t_{c_2} \cdots t_{c_n}]$.

Visual Cues: Given the document image I_D, we resize it to a fixed size (h,w,3). It is passed through a visual encoder (VE) to obtain the visual feature map $\eta = VE(I_D)$. We utilize the same input visual feature map across all layers and parent box configurations in a given document.

Multimodal Contextual Encoder (Λ) combines the structural, spatial and visual cues extracted from the input potential parent box and the sequence of child-boxes through a Transformer-based

language model. The semantic cues are concatenated with the embedding of each input box using late fusion as denoted by $\bigoplus$ due to the limitation dictated by the LayoutLM backbone. The final box embedding sequence is $X_j^n = \Lambda([B_j^n; T_j^n; \eta] \bigoplus S_j^n)$. The box embedding sequence X_j^n is matrix multiplied with the parent box embedding $X_j^n[: p_j]$ as $X_j^n[: p_j] \otimes X_j^n$ vector, where $\otimes$ means matrix multiplication. This results in a dot product of each child box embedding with parent box embedding to obtain $[\hat{p}_j; \hat{c}_1, \hat{c}_2, \cdots \hat{c}_n]$.

Link Prediction and Element Type Classification: The child box representations ($[\hat{c}_1, \hat{c}_2, \cdots \hat{c}_n]$) are passed through a dense fully-connected layer followed by a *Sigmoid* layer to generate the link probabilities between each child box c_i and a potential parent box p_j: $\alpha_{j1}, \cdots \alpha_{jn} = \sigma(W_2([\hat{c}_1, \cdots \hat{c}_n]) + \delta_2)$, where W_2, δ_2 and $\sigma(\cdot)$ are the weight matrices, bias and Sigmoid activation function, respectively. The parent box representation ($\hat{p}_j$) is passed through a dense fully-connected layer followed by a *softmax* to predict element type $\varphi_j = \sigma(W_3(\hat{p}_j) + \delta_3)$.

5.3.2 Training LayerDoc

Negative Parent Sampling: Most potential parent-boxes will be false positives, so to deal with the sparsity of positive samples at test time, we introduce negative sampling (Mikolov et al. 2013) in the training regime inspired by (Wang et al. 2021b; Wang et al. 2020b). For each training sample having at least one positive link between the potential parent-box and any of the input child-boxes, we add an unrelated parent-box example to the training set for the same setting to make the training robust to negative samples.

Multi-task Training: Element type classification uses a weighted cross-entropy loss to adjust for class imbalance, while link prediction uses a negative sample cross-entropy loss (Wang et al. 2021b) to account for negative data augmentation. Both tasks are correlated and reinforce

each other, so we use multi-task training to optimize both tasks simultaneously. The final optimization uses a weighted sum of the link prediction loss and element classification loss $L = \lambda L_{Link} + (1 - \lambda)L_{Class}$, where the weighting factor λ is a hyperparameter.

5.3.3 Inferring Document Layout Hierarchy

We recursively group child-boxes into parent-boxes such that the parent-boxes of the k^{th} layer become child-boxes of the $k + 1^{th}$ layer, iterating until only one parent-box remains. Each iteration involves three steps: (i) parent-box candidate generation, (ii) candidate link prediction and type classification, and (iii) constraint inference. The first iteration uses elementary token boxes t_i (OCR text, widgets, icons, etc) as child-boxes. Step (i) hypothesizes geometrically feasible potential parent-boxes (m candidates with an upper limit of $O(n^4)$ due to all relevant combinations of box co-ordinates) ensuring a high-recall collection of potential parent-boxes. Step (ii) predicts parent-child links and element types for each candidate parent-box with all the child-boxes as input, returning link probabilities $\alpha_{ji} \forall i \leq n; j \leq m$. Step (iii) selects the subset of parent boxes that are mutually non-overlapping, cover all child-boxes, and maximize the constraint optimization function described next.

Parent Box Proposal are created by utilizing the geometric constraints of the child boxes. We obtain sets of horizontal (x_{min}, x_{max}) and vertical (y_{min}, y_{max}) coordinates from the child box coordinates and merge them if lying within a threshold distance of each other to cluster closely placed coordinates and reduce the search space of coordinates. We choose two coordinate points from both x and y sets to form one rectangular parent box.

Constraint Inference: For the k^{th} layer, LayerDoc predicts link probabilities α_{ij} between each pair of potential parent box p_i and child box c_j. The best set of parent boxes is selected by

solving a constraint optimization problem, maximizing the cost function $\hat{Y} = \max_{y_i \in \Upsilon} \sum_i^m \omega_i y_i$, where $\omega_i = \kappa + \frac{\sum_j \hat{\alpha}_{ij}}{Ar(p_i)^k / Ar(\cup_{j=0}^n c_j)}$, $\hat{\alpha}_{ij}$ is the adjusted link probability between p_i and c_j such that $\hat{\alpha}_{ij} = \alpha_{ij} - 1$. κ is a large constant added to make all parent scores positive to avoid trivial solution of all weights as zero. $Ar(.)$ defines area of a box and $(\cup_{j=0}^n c_j)$ is the union of all n child boxes. $y_i \rightarrow 1$ represents the case where the potential parent box p_i is accepted as a valid parent box. The optimization is subject to constraint space $\Upsilon = \Upsilon_1 \cap \Upsilon_2$ defined over the set of all pairs of potential parent boxes $\Re^m$, where $\Upsilon_1 : y_i \in \{0, 1\}$ and $\Upsilon_2 : y_a + y_b \leq 1 \mid \forall\ a, b \in \Re \times \Re,\ Ar(p_a \cap p_b) > 0$. This is a typical Maximum Independent Set Problem (Tarjan and Trojanowski 1977) when reduced to a simple linear-programming relaxation by constraining y_i to be binary. It can be solved using Integer Linear Programming (ILP). However, the number of parent boxes grow exponentially, forcing us to further relax the ILP solution by greedily selecting one parent with highest ω_i at a time, leaving improved solutions to future work.

5.4 Experiments

Datasets: We train and test the `LayerDoc` model on three datasets, Hierarchical Forms, RICO and FUNSD, which provide scanned document images as input.

(1)Hierarchical Forms (Aggarwal et al. 2020a) is a rich corpus of scanned form documents from diverse domains like insurance, finance, and government agencies. The documents are human-annotated with labelled bounding boxes, element type, and element relations for a set of 14 constituent elements such as Text Fields, Checkboxes, Choice Groups, Widgets, Tables, Image, Header, Footer, etc. **(2) RICO** (Deka et al. 2017) is a dataset of more than 66k layout hierarchies of mobile app screens augmented with semantic annotations of UI components. The bounding

boxes, element labels and nested hierarchies are from app source code. **(3) FUNSD** (Jaume, Ekenel, and Thiran 2019) is a dataset of noisy scanned forms with shallow hierarchies and filled form fields.

Training: We experiment with four ablation settings using LayoutLM (Xu et al. 2020) (LayerDoc$_{LLM}$) or LayoutLMv2 (Xu et al. 2021) (LayerDoc$_{LLMv2}$) in multimodal context encoder. LayoutLMv2 extracts visual cues via Detectron2 (Wu et al. 2019). We experiment with and without SentenceBERT (Reimers and Gurevych 2019) for extracting semantic cues (LayerDoc$_{LLM+SBERT}$ and LayerDoc$_{LLMv2+SBERT}$). We use an equally balanced train-validation split. **Object detector**: We utilize Faster-RCNN trained on the training set of Forms/RICO/FUNSD dataset to infer lower-level elementary tokens such as widgets, images, etc. **Box Types** of elementary boxes are obtained object detector predictions.

Evaluation Tasks: We evaluate LayerDoc on five tasks: Element Type Classification and Group Identification for specific components of the architecture; Element Detection, Reading Order and Grouping for full hierarchy. **Element Type Classification:** Evaluates the parent-box type classification using weighted F1-score for each type, using ground truth child-boxes as input at test time. **Group Identification:** Evaluates link prediction between the candidate parent-box and child-boxes using macro F1 score using ground-truth child-boxes as input at test time. **Hierarchy Reconstruction:** Elementary tokens (words+bounding boxes) are given as input and all other layers use the predictions from the previous layer. We evaluate document layout hierarchy predicted in Sec 5.3.3 using Mean Average Precision (mAP) (0.5 IoU threshold) between ground truth and predicted bounding boxes using the standard teacher forcing technique (Williams and Zipser 1989a). We also utilize the Adjusted Rand-index (Rand 1971) to measure the similarity between two hierarchies in each layer as well as for the whole layout hierarchy

	Modality	Model	TableRow	ChoiceGroup	Footer	Section	ListItem	Table	TextRun	TableCell	TextBlock	List	Field	Header	Overall
Baseline	Visual	MFCN (Yang et al. 2017)	–	0.0	–	0.71	0.54	–	0.11	–	0.46	0.90	–	0.69	–
	Visual	DLV3+ (Peng, Yin, and Yang 2020)	–	0.57	–	0.55	0.75	–	0.69	–	0.86	0.48	–	0.83	–
	Spatial + Text	Form2Seq (Aggarwal et al. 2020a)	–	0.78	–	0.67	0.90	–	0.85	–	0.91	0.93	–	**0.85**	–
Ablation	Spatial	LayerDoc_{LLM}	0.36	0.74	0.65	0.57	0.74	0.20	0.61	0.57	0.41	0.35	0.42	0.15	0.43
	Spatial + Text	$\text{LayerDoc}_{LLM+SBERT}$	0.48	0.68	0.75	0.67	0.79	0.29	0.74	0.89	0.84	0.41	0.80	0.72	0.76
	Spatial + Visual	LayerDoc_{LLMv2}	0.69	0.89	0.90	0.76	0.94	**0.96**	0.82	0.97	0.97	0.93	0.94	0.35	0.89
	Spatial + Visual + Text	$\text{LayerDoc}_{LLMv2+SBERT}$	**0.92**	**0.90**	**0.92**	**0.86**	**0.96**	0.94	**0.88**	**0.98**	**0.98**	**0.95**	**0.94**	0.67	**0.90**

(a) Hierarchical Forms Dataset

	Modality	Model	List Item	Text	Checkbox	TextButton	Modal	Toolbar	Card	Drawer	Multi-Tab	WebView	Input	Button Bar	Tile	Overall
Baseline	Visual	Faster-RCNN (Ren et al. 2015)	0.55	0.54	0.29	0.36	0.48	0.63	0.18	0.61	0.45	0.45	0.11	0.03	0.48	0.48
	Visual	UIED (Chen et al. 2020a)	0.62	0.61	0.35	0.41	0.62	0.83	0.27	0.74	0.51	0.45	0.19	0.10	0.60	0.71
	Visual	DETR (Carion et al. 2020)	0.67	0.65	0.39	0.46	0.67	0.86	0.30	0.75	0.52	0.48	0.20	0.12	0.63	0.72
	Spatial + Visual + Textual	LayoutLMv2 (Xu et al. 2021)	0.82	0.72	0.39	0.50	0.73	0.88	0.42	0.78	0.55	0.61	0.16	0.18	0.67	0.75
Ablation	Spatial	LayerDoc_{LLM}	0.80	0.74	0.42	0.51	0.69	0.94	0.35	0.81	0.60	0.53	0.22	0.18	0.68	0.76
	Spatial + Text	$\text{LayerDoc}_{LLM+SBERT}$	0.82	0.74	0.46	0.53	0.59	0.94	0.45	0.83	0.61	0.56	0.20	0.70	0.68	0.78
	Spatial + Visual	LayerDoc_{LLMv2}	0.87	**0.77**	0.46	0.53	**0.78**	0.93	0.46	0.86	0.59	0.65	**0.28**	0.2	0.68	0.79
	Spatial + Visual + Text	$\text{LayerDoc}_{LLMv2+SBERT}$	**0.88**	0.76	**0.47**	**0.55**	0.65	**0.96**	**0.49**	**0.87**	**0.71**	**0.68**	0.20	**0.78**	**0.73**	**0.80**

(b) RICO Dataset

Table 5.1: Results comparing F1 scores of LayerDoc with baselines and ablative components for **element classification task** for label-wise and overall spatial elements in (a) **Hierarchical Forms** and (b) **RICO dataset**. Our proposed approach outperforms the baselines, and ablation analysis shows that each individual component contributes to the overall performance.

in aggregate. We consider the child-boxes in a given layer linked to the same parent-box as one cluster and consider the predicted parent-boxes to match if the ground truth if IoU > 0.5.

Reading Order Comparison: Following (Wang et al. 2021b), we sort the predicted layout hierarchy and traverse the bounding boxes in-order to recover the sequence of OCR tokens. We then compare the predicted reading order sequence against the ground truth reading order using Average Page-level BLEU (p-BLEU) and Average Relative Distance (ARD) (Wang et al. 2021a). **Grouping Comparison:** We evaluate the word and element grouping. Similar to (Lee et al. 2021; Wang et al. 2021a), we utilize the word grouping metric to calculate the F1, precision and recall of intervals in the predicted word sequence belonging to an element compared to the ground truth sequence.

	Modality	Model	TableRow	ChoiceGroup	Footer	Section	ListItem	Table	TextRun	TableCell	TextBlock	List	Field	Header	Overall
Baseline	Visual	MFCN (Yang et al. 2017)	–	0.28	–	–	–	–	–	–	–	–	0.19	–	–
	Visual	DLV3+ (Peng, Yin, and Yang 2020)	–	0.47	–	–	–	–	–	–	–	–	0.51	–	–
	Spatial + Text	Form2Seq (Aggarwal et al. 2020a)	–	0.61	–	–	–	–	–	–	–	–	0.86	–	–
	Spatial + Text + Visual	MMPAN (Aggarwal et al. 2020b)	–	0.63	–	–	–	–	–	–	0.88	–	0.90	–	–
	Spatial + Text + Visual	DocStruct (Wang et al. 2020b)	0.39	0.20	0.18	0.21	0.16	0.09	0.28	0.40	0.27	0.14	0.30	0.07	0.36
Ablation	Spatial	LayerDoc$_{LLM}$	0.36	0.64	0.45	0.28	0.20	0.14	0.44	0.41	0.51	0.22	0.38	0.38	0.41
	Spatial + Text	LayerDoc$_{LLM+SBERT}$	0.38	0.68	0.51	0.48	0.33	0.68	0.52	0.78	0.65	0.45	0.56	0.45	0.55
	Spatial + Visual	LayerDoc$_{LLMv2}$	**0.90**	0.75	0.75	**0.78**	0.82	**0.83**	0.49	0.76	0.90	0.75	0.85	0.15	0.79
	Spatial + Visual + Text	LayerDoc$_{LLMv2+SBERT}$	0.85	**0.78**	**0.80**	0.67	**0.85**	0.70	**0.79**	**0.82**	**0.92**	**0.77**	**0.92**	**0.49**	**0.81**

(a) Hierarchical Forms Dataset

	Modality	Model	List Item	Text	Checkbox	TextButton	Modal	Toolbar	Card	Drawer	Multi-Tab	WebView	Input	Button Bar	Tile	Overall
Baseline	Visual	Faster-RCNN (Ren et al. 2015)	0.20	0.24	0.29	0.36	0.28	0.31	0.15	0.21	0.25	0.18	0.21	0.15	0.35	0.27
	Visual	UIED (Chen et al. 2020a)	0.24	0.35	0.45	0.40	0.32	0.48	0.27	0.56	0.49	0.55	0.69	0.50	0.56	0.52
	Visual	DETR (Carion et al. 2020)	0.32	0.39	0.49	0.45	0.38	0.54	0.33	0.61	0.55	0.62	0.72	0.54	0.59	0.55
	Spatial + Visual + Textual	LayoutLMv2 (Xu et al. 2021)	0.77	0.80	0.69	0.75	0.42	0.83	0.72	0.69	0.81	0.82	0.81	0.74	0.69	0.72
Ablation	Spatial	LayerDoc$_{LLM}$	0.25	0.40	0.52	0.40	0.44	0.52	0.25	0.33	0.40	0.63	0.36	0.53	0.40	0.50
	Spatial + Text	LayerDoc$_{LLM+SBERT}$	0.26	0.41	0.54	0.42	0.44	0.68	0.28	0.35	0.45	0.81	0.39	0.55	0.58	0.55
	Spatial + Visual	LayerDoc$_{LLMv2}$	0.81	0.86	0.75	0.80	0.45	0.88	0.77	0.76	0.86	0.86	0.87	0.81	0.74	0.86
	Spatial + Visual + Text	LayerDoc$_{LLMv2+SBERT}$	**0.83**	**0.88**	**0.77**	**0.82**	**0.47**	**0.90**	**0.79**	**0.77**	**0.88**	**0.88**	**0.96**	**0.84**	**0.76**	**0.89**

(b) RICO Dataset

Table 5.2: Results comparing F1 scores of LayerDoc with baselines and ablative components for **group identification task** for label-wise and overall spatial elements in (a) **Hierarchical Forms** and (b) **RICO dataset**. Our proposed approach outperforms the baselines, and ablation analysis shows that each individual component contributes to the overall performance.

5.5 Results and Analysis

We present our experimental results, where **bold** in tables denotes the best performing model. Colored text represents the proposed LayerDoc with LayoutLMv2 backbone and Sentence-BERT for semantic cues. Values not reported by the baseline models are indicated by (–) dashes.

5.5.1 Element Type Classification

Hierarchical Forms: Table 5.1a shows element classification where we compare LayerDoc with MFCN (Yang et al. 2017), DLV3+ (Peng, Yin, and Yang 2020), Form2Seq (Aggarwal et al. 2020a) as they report strong baseline performance for this task. Form2Seq is a competitive baseline that uses seq2seq modeling of spatial regions for element classification and extraction. However, it struggles to handle long-range dependencies in dense forms with large sequences of tokens. MFCN and DLV3+ are strong convolution based baselines utilized specifically in the

Model	Element Classification (F1)	Group Identification (F1)
BERT (Jaume, Ekenel, and Thiran 2019)	0.64	0.29
GNN + MLP (Carbonell et al. 2021)	0.64	0.39
UniLMv2-large (Appalaraju et al. 2021)	0.70	–
SPADE (Hwang et al. 2020)	0.71	0.41
StrucTexT (Li et al. 2021b)	0.83	0.44
LayoutLMv1-large (Xu et al. 2020)	0.78	0.42
FUDGE (Davis et al. 2021)	0.66	0.56
SERA (Zhang et al. 2021b)	–	0.65
BROS (Hong et al. 2020)	0.81	0.66
MSAU-PAF (Dang et al. 2021)	0.83	0.75
LayoutLMv2-large (Xu et al. 2021)	0.84	–
DocFormer-large (Appalaraju et al. 2021)	0.84	–
LayerDoc$_{LLMv2+SBERT}$ **(Ours)**	**0.86**	**0.78**

Table 5.3: Comparison of LayerDoc (w/ LayoutLMv2 and SentenceBert) with baseline models for **element type classification (entity labeling)** and **group identification (linking)** on the **FUNSD** dataset. LayerDoc outperforms all recent top-performing systems in terms of F1 score.

document understanding domain. All three baselines were designed to work for a limited set of elements found in the lowest layers of the hierarchy, preventing comparison between all element types.

RICO: Table 5.1b reports results for RICO dataset. We establish a strong baseline UEID (Chen et al. 2020a) that uses a mix of text detector and traditional computer vision techniques to classify and extract spatial elements. Inspired by (Li et al. 2020d), we compare Faster-RCNN (Ren et al. 2015) which is a traditional object detector. We also fine-tune and evaluate recent transformer based object detection models such as DETR (Carion et al. 2020) and Swin Transformer (Liu et al. 2021b) on UI interfaces from RICO dataset. Visual object detectors are not able to leverage semantic context necessary for document understanding. LayoutLMv2 (Xu et al. 2021) model utilizes visual, spatial as well as semantic context. However, it is pre-trained for language modeling tasks as opposed to layout hierarchy extraction objective. **Performance of LayerDoc** with LayoutLMv2 backbone and SentenceBERT shows significant gains across all element types as

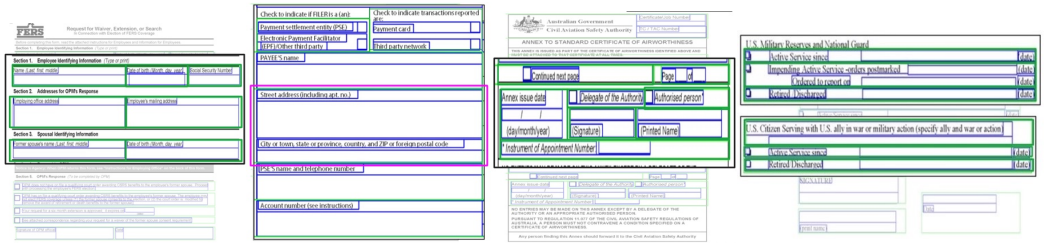
Dataset	Model	Reading Order		Word Grouping		
		p-BLEU (↑)	ARD (↓)	P	R	F1
FUNSD	Heuristics	0.69	8.46	–	–	–
	LayoutLMv1 (Xu et al. 2020)	0.89	2.54	0.82	0.88	0.85
	LayoutLMv2 (Xu et al. 2021)	0.92	2.21	0.84	0.87	0.86
	LayoutReader (Wang et al. 2021a)	0.98	1.75	–	–	–
	ROPE (Lee et al. 2021)	–	–	0.88	0.90	0.89
	LayerDoc$_{LLM}$	0.98	1.68	0.82	0.79	0.80
	LayerDoc$_{LLM+SBERT}$	0.99	1.65	0.85	0.90	0.87
	LayerDoc$_{LLMv2}$	0.98	1.63	0.86	0.92	0.89
	LayerDoc$_{LLMv2+SBERT}$	**0.99**	**1.60**	**0.92**	**0.93**	**0.92**
RICO	Heuristics	0.49	1.77	–	–	–
	Faster-RCNN (Ren et al. 2015)	0.55	1.76	0.45	0.76	0.57
	UEID (Chen et al. 2020a)	0.61	1.75	0.45	0.76	0.57
	DETR (Carion et al. 2020)	0.63	1.74	0.48	0.79	0.60
	LayoutLMv2 (Xu et al. 2021)	0.65	1.72	0.63	0.83	0.72
	LayerDoc$_{LLM}$	0.65	1.70	0.68	0.95	0.79
	LayerDoc$_{LLM+SBERT}$	0.67	1.68	0.68	0.94	0.79
	LayerDoc$_{LLMv2}$	0.69	1.62	0.73	0.95	0.83
	LayerDoc$_{LLMv2+SBERT}$	**0.70**	**1.60**	**0.77**	**0.97**	**0.87**

Table 5.4: Results for **reading order** (p-BLEU and ARD) and **word grouping** (F1, P, R) for **FUNSD and RICO** dataset.

it benefits from contextual modeling of spatial regions, multimodal input to the contextual encoder, and multi-tasking objective aimed at optimizing the element type classification and group identification simultaneously. *Header* type elements in Hierarchical Forms dataset are an exception where our model underperforms the Form2Seq baseline. Lower performance of header can be attributed to model overfitting as the header class is a minority in the dataset. LayerDoc is trained to predict several different components simultaneously as opposed to Form2Seq and DLV3+ baselines which are specifically trained on selective components. Moreover, visual modality does not help element type prediction as headers are usually localized in a small part of the document and do not benefit from contextual modeling.

Dataset	Model	Hierarchy Reconstruction mAP (↑)	Rand-index Test P	R	F1
FUNSD	LayoutLMv1 (Xu et al. 2020)	0.27	0.51	0.54	0.52
	LayoutLMv2 (Xu et al. 2021)	0.35	0.61	0.62	0.61
	LayerDoc$_{LLM}$	0.45	0.77	0.72	0.74
	LayerDoc$_{LLM+SBERT}$	0.48	0.72	0.81	0.76
	LayerDoc$_{LLMv2+SBERT}$	**0.50**	**0.78**	**0.83**	**0.80**
RICO	Faster-RCNN (Ren et al. 2015)	0.15	0.32	0.33	0.33
	UEID (Chen et al. 2020a)	0.21	0.35	0.48	0.39
	DETR (Carion et al. 2020)	0.21	0.43	0.48	0.45
	LayoutLMv2 (Xu et al. 2021)	0.23	0.55	0.54	0.55
	LayerDoc$_{LLM}$	0.19	0.70	0.74	0.72
	LayerDoc$_{LLM+SBERT}$	0.22	0.74	0.73	0.74
	LayerDoc$_{LLMv2+SBERT}$	**0.27**	**0.86**	**0.84**	**0.85**
Hierarchical Forms	DocStruct (Wang et al. 2020b)	0.10	0.35	0.36	0.36
	LayerDoc$_{LLM}$	0.10	0.33	0.51	0.40
	LayerDoc$_{LLM+SBERT}$	0.11	**0.36**	0.51	**0.42**
	LayerDoc$_{LLMv2+SBERT}$	**0.12**	0.31	**0.55**	0.40

Table 5.5: Results for **hierarchy reconstruction** (mAP and Rand-index test) for **FUNSD, Hierarchical Forms and RICO** dataset.



(a) (b) (c) (d)

Figure 5.3: Example illustrations of predictions by LayerDoc$_{LLMv2+SBERT}$ on the test set of the Hierarchical Forms dataset. Blue boxes denote input child boxes while green boxes indicate detected parent boxes in the hierarchy. The pink box in (b) highlights the semantically unique spatial groups inferred from the document layout hierarchy.
(a) *Field - widget* pairs are detected with high precision with spatially consistent boxes being grouped together.
(b) Non-trivial form fields aggregated based on their semantic meaning. Eg. addresses in *text fields* are grouped into *text block*.
(c) Extracts difficult non-symmetric *TextBlocks* despite multiple levels of nesting.
(d) Errors in *Choice fields* grouping due to initial mistakes in grouping of *widgets* propagates to later *choice groups* grouping.

5.5.2 Group Identification

Hierarchical Forms: Table 5.2a shows group identification results where we compare against image segmentation baselines - DeepLabV3+ and MFCN for element extraction. These models often make mistakes in case of closely spaced text blocks and text fields, struggling to predict complete choice fields and choice groups due to their inability to capture complete horizontal context. Form2Seq (Aggarwal et al. 2020a) and MMPAN (He et al. 2017) baselines use LSTM-based seq2seq models to extract multimodal hierarchical associations. We consider the settings where ground truth is given as input to the next step of the pipeline. Results for DeepLabV3+, MFCN, Form2Seq, and MMPAN are derived from (Aggarwal et al. 2020a) which evaluated them to work with specific inputs (text blocks) and to give certain outputs (text blocks, choice groups, choice fields), hence the sparsity in their results. We additionally evaluate DocStruct (Wang et al. 2020b) on Hierarchical Forms, a recent state-of-the-art method for layout structure extraction by re-implementing it for generic semi-structured documents. **RICO:** We evaluate the task of group identification on RICO using hybrid deep networks (UIED), traditional (Faster-RCNN) as well as Transformer-based DETR for 2D object detection baselines. The input to the model is the raw document image while the outputs are predicted bounding boxes with class labels. LayoutLMv2 model is fine-tuned and evaluated similarly to (Li et al. 2020g). **Performance of LayerDoc**: LayerDoc is significantly better compared to all baselines by a large margin for Hierarchical Forms, except for *Choice Fields*. Form2Seq and MMPAN outperform in grouping *text blocks* and *widgets* into *choice field* elements as they were designed to selectively handle such elements. DocStruct severely underperforms against LayerDoc on complex hierarchical forms due to the lack of document-level context and inability to generalize beyond simple key-value

pair elements. For RICO, both Faster-RCNN and DETR are weaker than LayerDoc as they do not leverage multimodal input. LayerDoc outperforms LayoutLMv2 due to its superior recursive parent-child link prediction approach. **Performance on FUNSD:** Table 5.3 compares multiple state-of-the-art methods for element classification and group identification on FUNSD. LayerDoc outperforms all other models on extracting and classifying key-value pairs in noisy forms. **Ablation Study:** We denote a darker green shade to indicate better F1 performance, and ablation is indicated by the "Modality" column across the tables. We observe a consistent benefit of using both visual as well as textual modalities in LayerDoc across all tasks. Visual cues extracted by Detectron2 in LayoutLMv2 backbone improves performance as most semi-structured documents have visually rich elements such as tables, check boxes, widgets, buttons, input fields. Semantic cues help improve identification of most elements, except *table* and *sections* elements as they rely more heavily on spatial boundaries and neighbouring white spaces for accurate extraction.

Hierarchy Reconstruction: We evaluate the predicted document layout hierarchy in Table 5.5. Elementary tokens (words+bounding boxes) are input and each layer uses the predictions from the previous layer in a recursive manner. Unlike past hierarchy extraction techniques applied to FUNSD (Wang et al. 2020b), we do not assume ground truth parent boxes to be a part of the input during hierarchy inference. We evaluate using Mean Average Precision (mAP) of predicted boxes with a 0.5 IoU threshold between ground truth and predicted bounding boxes. To generate hierarchies from baseline models, we use the elements detected at inference to arrange them in a bottom-up hierarchy based on geometric constraints. We show that LayerDoc with LayoutLMv2 and SentenceBERT outperforms other configurations on all three datasets where ablations show the usefulness of visual, spatial and textual cues.

Reading Order: Table 5.4 compares reading order of OCR tokens based on the extracted layout hierarchy. We implement a heuristics baseline that linearly sorts the words from left to right and top to bottom based on OCR box coordinates. We report results on FUNSD and RICO datasets and conclude that LayerDoc achieves the SOTA results. Comparing LayerDoc's performance on FUNSD with LayoutLmv1, LayoutLMv2, LayoutReader (Wang et al. 2021a) and ROPE (Lee et al. 2021) shows competitive p-BLEU performance and reduction in ARD by approximately 10%. For RICO, we compare the reading order derived from the layout hierarchy as extracted by UIED, Faster-RCNN, DETR, and LayoutLMv2. LayerDoc generates better reading order compared to competitive object detection methods. Our contribution becomes significant for RICO dataset where reading order is complicated by deep nested hierarchies. Ablation experiments show that both layout and textual information play equally important roles.

Word Grouping: We observe an improvement of 4% and 10% in F1-score of word grouping performance on FUNSD and RICO datasets, respectively. LayerDoc is able to capture the complete text layout that helps it recover missing words that flow over to the start of the next line at line end. This is especially important for grouping check boxes and text fields into choice groups, and table components present in deeply nested scanned forms.

Impact of SBERT and layer-wise structure on LayerDoc: We observe a 8-14% performance drop by removing SBERT in element classification, group identification, reading order and word grouping tasks, demonstrating its importance to LayerDoc with a LayoutLMv2 backbone. However, even without SBERT, LayerDoc outperforms the LayoutLMv2 baselines on RICO by 10-14%, demonstrating that the layer-wise structure of LayerDoc is also important. **Computational cost**: On an average, LayerDoc requires ≈10 times less forward passes to generate complete hierarchy compared to the DocStruct/LayoutLMv2 baseline as it can perform link

prediction between a proposed parent box and all child boxes in a layer through its contextual modeling instead of comparing all possible pairs of parent-child pairs across different layers one at a time. This results in reduced search space. LayerDoc has comparable parameters to LayoutLMv2, with the additional parameters from linear layers. Hence their time complexity is comparable, yet LayerDoc outperforms due to algorithmic modifications rather than model size. Figure 5.3 presents some **illustrative examples** with inferred layout hierarchies by `LayerDoc`.

Error Analysis: (i) Recursive Error Propagation: Grouping performance reduces higher up the predicted hierarchy as elements detected in the initial layers are used for predicting elements in the subsequent levels of the hierarchy, causing error propagation. (ii) Lack of parent-box context: Our approach infers one parent box at a time in a given layer. Despite optimal layer-wise parent-box selection, errors produced at this step cannot be backpropagated during training. Restricted backtracking in future work may alleviate error accumulation at higher levels.

5.6 Conclusion and Future Work

We present `LayerDoc` that uses visual, textual and spatial signals along with constraint inference to extract the documents hierarchy in a bottom-up layer-wise fashion. Extensive experiments demonstrate the advantages of our method for extracting specific components of the hierarchy (element type classification and group identification) as well as its downstream applications in reading order detection and word grouping on three diverse semi-structured document datasets. `LayerDoc` enables full-scale hierarchy extraction from diverse documents to enable form authoring, document re-flow, and adaptive editing of user-interfaces. Our current work is limited by its iterative nature and restricted to greedy optimizations. Future work can

focus on integrating restricted backtracking in parent selection, layer embedding for different levels, cross-dataset generalization, semi-greedy approaches. Hierarchy construction can aid **long-context document understanding** for tabular parsing (Mathur et al. 2022b), layout-enriched speech synthesis (Mathur et al. 2022a), and NLP tasks like temporal information extraction (Mathur et al. 2021b), temporal dependency parsing (Mathur et al. 2022d), and NLI (Mathur et al. 2022c).

CHAPTER 6

DocEdit: Language-Guided Document Editing

Abstract

Professional document editing tools require a certain level of expertise to perform complex edit operations. To make editing tools accessible to increasingly novice users, we investigate intelligent document assistant systems that can make or suggest edits based on a user's natural language request. Such a system should be able to understand the user's ambiguous requests and contextualize them to the visual cues and textual content found in a document image to edit localized unstructured text and structured layouts. To this end, we propose a new task of language-guided localized document editing, where the user provides a document and an open vocabulary editing request, and the intelligent system produces a command that can be used to automate edits in real-world document editing software. In support of this task, we curate the

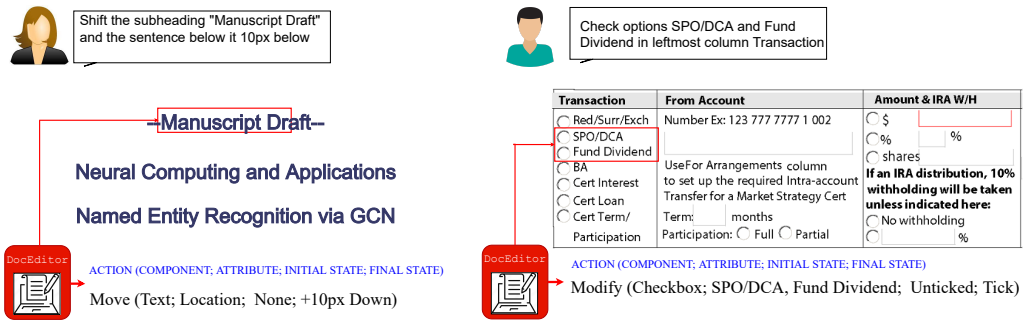


Figure 6.1: The DocEdit dataset provides natural language edit requests on PDFs and design template documents. Each edit request is mapped to an executable command that can be used to automatically apply edits in real-world document editing software. We propose DocEditor, a neural architecture to generate the executable computer command and ground the region of interest bounding box. Examples from the Hierarchical Forms dataset and the public Enron corpus in this figure illustrate several challenges where an intelligent system needs to (a) interpret and localize structured components and their relative positioning in the document; (b) match document text tokens in a text-rich document formatted in varied spatial layouts (checkboxes, choice groups, text fields, columns, rows), (c) visually understand the objects as per the description.

DocEdit dataset, a collection of approximately 28K instances of user edit requests over PDF and design templates along with their corresponding ground truth software executable commands. To our knowledge, this is the first dataset that provides a diverse mix of edit operations with direct and indirect references to the embedded text and visual objects such as paragraphs, lists, tables, etc. We also propose DocEditor, a Transformer-based localization-aware multimodal (textual, spatial, and visual) model that performs the new task. The model attends to both document objects and related text contents which may be referred to in a user edit request, generating a multimodal embedding that is used to predict an edit command and associated bounding box localizing it. Our proposed model empirically outperforms other baseline deep learning approaches by 15-18%, providing a strong starting point for future work.

6.1 Introduction

Digital documents are used extensively to help people improve business productivity (drafting contract agreements, presentation decks, letterheads, invoices, resumes, form filling) and communicate with customers through online advertisements, social media posts, flyers, posters, billboards, web and mobile app prototypes, etc. However, modern document editing tools require a skilled professional to work on a large screen. Challenges emerge when complex editing operations require multiple different functionalities wrapped within the editing tools for text and image region placement, grouping, spatial alignment, replacement, resizing, splitting, merging, and special effects. As the creation and editing of documents become more ubiquitous and increasingly used by novice users on mobile devices, there is an increasing need to improve the accessibility of these tools through an intelligent assistant system that can understand user's intent and translate it into executable code that can be processed by the editing tool to fulfill a user's editing needs.

We formulate a new task for language-guided document editing and create a new dataset, `DocEdit`, as illustrated in Figure 10.1, wherein an intelligent system is expected to generate the executable commands (e.g., move components, modify attribute values and special effects, add/delete text, etc.) and visually ground the region of interest given the natural language edit request expressed by human users over a document image. To do so, document editing systems should not only understand the user intent, but also extract and interpret the textual content of the document images along with the visual cues including layout (paragraphs, lists, tables, headers, footers), non-textual elements (marks, tick, shapes, diagrams, graphs), and style (font, colors, size, highlighting, special effects). Departing from generic language-guided image editing

tasks (Shi et al. 2020; Lin et al. 2020c), our task warrants a different approach to exploit the above visual cues and high-density of textual tokens by making use of the relative positioning of objects and text tokens.

The new dataset for this task, DocEdit, provides natural language edit requests on PDFs and design template documents, along with the result of a human carrying out the edit request. Each edit request is mapped to an executable command that can be simulated in real-world document editing software. To collect such a dataset, we utilized User Interface (UI) experts to, edit a set of input documents, provide a description of the edit, and generate the ground truth executable command corresponding to a set of diverse and creative edit requests posed by freelance designers. DocEdit contains more than 17k PDF and 10k design templates with a diverse mix of edit operations (add, delete, modify, split, merge, replace, move, copy) and reference types (direct, object referring, text referring) from the users.

Our work also takes the first step towards automating *Language-guided Localized Document Editing (LLDE)* using DocEditor, a Transformer-based localization-aware multimodal (textual, spatial, and visual) model. The model represents the visual appearance of document elements (e.g., paragraphs, images) through their bounding boxes and document semantics (the meaning of the text in the document) through document text tokens obtained via OCR. It uses multi-head attention to obtain a text-enriched visual box embedding which is fused with a text embedding and a regression token. The fused representation is provided to a Transformer decoder, which generates the command text in an autoregressive fashion. Additionally, we employ a layout graph to encode the relative position of boxes and document text tokens to regress the RoI bounding box coordinates. We perform node classification as an auxiliary task for anchor box prediction to ground the edit location in terms of relevant object and document text token boxes.

Dataset	Size	OCR	LE	IR	Doc
CAISE	6.1K	✗	✗	✗	✗
DialEdit	8.8K	✗	✗	✗	✗
Edit me	9.1K	✗	✓	✗	✗
ILLC-IER	2.5K	✗	✓	✓	✗
DocEdit (Ours)	28.3K	✓	✓	✓	✓

Table 6.1: . Comparison of DocEdit with related language-guided image editing datasets. Our dataset is the largest document-centric corpus with localized edits (LE), OCR'ed text, and indirect references (IR) to local objects.

DocEditor proves as a strong benchmark for this task and outperforms other unimodal and multimodal baselines. Our **contributions** are:

- We introduce a new task and dataset for document edit command generation for language-guided localized document editing. The DocEdit dataset consists of document-edit pairs on PDFs and design templates along with corresponding ground truth executable commands.
- We propose DocEditor, a novel multimodal transformer that takes a language-based edit request and produces a spatially localized set of edit commands. To the best of our knowledge, no such multimodal language-guided document editing model exists, with existing models lacking in terms of their understanding of document text and their ability to perform localized edits. Our proposed DocEditor model empirically outperforms other baseline deep learning approaches by 15-18%, providing a strong starting point for future work.

6.2 Related Work

Table 6.1 shows prior tasks and datasets for language-guided image editing systems such as (Shi et al. 2020; Lin et al. 2020c). However, most of these tasks are designed to work with natural

images instead of documents that are usually text-rich and may contain a wide range of structured components in varied layouts. Recent GAN-based methods (Jiang et al. 2021b; El-Nouby et al. 2018; Wang et al. 2022; Li et al. 2020a; Jiang et al. 2021c) are popular for natural image editing tasks as they perform end-to-end pixel-wise image generation, but are unsuitable for digital-born PDF documents with rich text. Such methods still cannot handle spatial and semantic understanding of embedded text present in the documents. Previous research works (Kim et al. 2022; Shi et al. 2021a; Shi et al. 2022; Chen et al. 2018) have explored language-driven image editing to map image edits into actionable computer commands, they are largely limited to global requests where the entire image is uniformly modified. Complex and unstructured documents, for example, images of receipts, invoices, and forms have a large number of relatively small text objects scattered throughout an unstructured document and surrounded by "distraction" objects which are not of interest. Hence, there is a need to spatially localize the objects of interest by modeling text and image content and relating it to the user's text description. Efforts to investigate such localized editing of spatial regions remain limited. Previous attempts at intent/action/goal identification from user edit requests (Manuvinakurike et al. 2018a; Manuvinakurike et al. 2018b; Manuvinakurike et al. 2018c; Lin et al. 2020a) have only explored a limited set of edit functions constrained to changes in brightness, contrast, background color, and their numeric values. Hence, there is a significant gap in the space of operations possible through automated document editing, thus necessitating the development of methods that can generalize to ambiguous open vocabulary user requests and convert them into executable commands grounded in specific action, component, and attribute taxonomy.

6.3 Task Description

We introduce the task of generating an executable command from a linguistic user request for editing a document according to the user's intent. Formally, given a document D to be edited and the user request defined as a sequence of n tokens $W = [t_1, t_2, \cdots t_l]$, we predict the executable command C of the format: $ACTION(< Component >, < Attribute >, < Initial\_State >, < Final\_State >, [x, y, h, w])$. Here, *Action* describes the executable function belonging to the following taxonomy - Add, Delete, Copy, Move, Replace, Split, Merge, Modify. It is followed by arguments corresponding to the document *components* to be edited, *attributes* to be modified, *initial state* of the attributes, and the *final state* of the attributes expected in the edited version. The Region of Interest (RoI) is represented by the bounding box $[x, y, h, w]$ enclosing the components to be edited in the input document image, such that (x, y) refers to the top-left coordinate while h and w refer to the height and width of the bounding box, respectively. We perform end-to-end command generation task along with the RoI bounding box regression grounded in the document image.

6.4 `DocEdit` Dataset

Language-guided image editing has been studied in the past. However, there is no existing dataset that captures language-guided editing of structured documents such as PDFs, PowerPoint presentations, and design templates, in which the spatial arrangement of content (text, images, etc) may be as important as the content itself, and edit operations are localized to specific regions of a document. Such documents are rich in layout due to the presence of a high variety of structured components such as tables, graphs, text fields, checkboxes, widgets, lists, and

backgrounds along with the unstructured text. Therefore, we present the DocEdit dataset which provides pairs of the document image and user edit requests along with the ground truth edit command and the final edited version of the document. We present two variants of the DocEdit dataset: (1) DocEdit-PDF comprising of edits performed on publicly available PDF documents and (2) DocEdit-Design comprising of edits on design templates.

Data Acquisition: We extracted 20K anonymized PDF documents from the publicly available Enron corpus and Hierarchical forms (Aggarwal et al. 2020a) datasets with all personally identifiable information (PII) removed for DocEdit-PDF. We downloaded 12K publicly available and freely distributed design templates from the Adobe Express platform for DocEdit-Design.

Document Edit Creation: We employed 15 freelance annotators from Upwork with verified past experience in graphic design and Word/PDF document editing. The annotators were provided with examples and online tutorials for editing PDF and design templates and were encouraged to provide creative edit requests unique to each document. The edit requests are shuffled and each annotator is asked to utilize Adobe Acrobat and Adobe Express tools for physically editing PDF documents and design templates, respectively. We trained both sets of annotators to make them familiar with the edit creation process so as to guarantee the quality of the dataset. In the training session, we provided feedback for 100 practice edit requests and the corresponding edited version of the document per annotator for consistency. We performed this training session multiple times until the quality of the data has no obvious/critical issues. **Ground Truth Collection**: We developed a taxonomy of possible actions, components, and attributes. The annotators were asked to select the most relevant edit action along with one or more relevant options for components and attributes. Additionally, we asked the annotators to provide ground truth labels for the initial state of the component prior to editing, and the final state of the component post-

editing as text inputs filled in by the annotator based on the user request description and visual context from the document image. In order to uniquely identify the location of the component to be edited, we asked the annotators to mark a tightly enclosed bounding box region surrounding the corresponding component in the document image. We concatenated the labels and bounding box coordinates to form the output command. Ground truth labels were not sourced from the same annotator providing the edit request description. **Data Quality Estimation**: We report a high degree of agreement (Krippendorff's alpha) between annotators for the test portion (20% of the dataset). **Edit Reference Types**: We further categorize the editing requests as direct, object-referencing, or text-referencing requests. Direct requests are self-contained with specific cues about the component to be modified. We see that a majority of samples in our dataset are indirect requests that refer to a component or text in a document through their relative position, making the task challenging due to the necessity to resolve indirect object references. **Data Splits**: We split both DocEdit-PDF and DocEdit-Design into train, validation, and test in the ratio of 70:10:20.

6.5 Methodology

We present DocEditor (see Figure 6.2), a neural architecture that takes in the user request text and document image as input and predicts an executable edit command by generating the textual functional arguments along with regressing the bounding box coordinates of the edit RoI. Our model can be seen as a sequence of five phases: (a) multimodal feature extraction to obtain the user request embedding, visual object embeddings, and document text token embeddings, (b) obtaining text-enriched visual object representations, (c) generating an executable command by

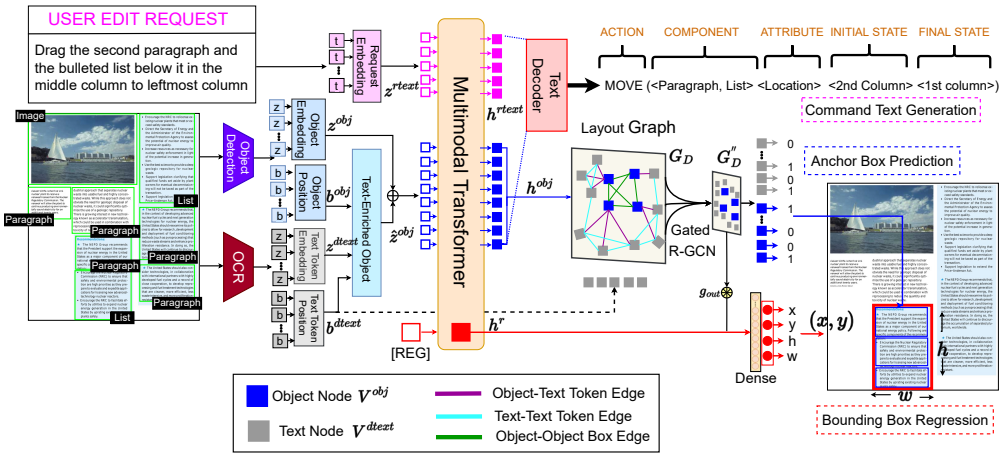


Figure 6.2: Overview of our proposed system, DocEditor: Object boxes and document text tokens obtained by the object detector and OCR system from the document image are combined using multi-headed attention to form text-enriched visual object embeddings. These are concatenated with the encoded text request and [REG] token to form the multimodal input to the Transformer model. A text decoder generates command text in an auto-regressive way. The output hidden states of the object boxes and document text token embeddings are used to create a Layout Graph with nodes joining Object boxes and document text boxes learned through a Gated R-GCN. We perform node classification for anchor box prediction. The graph embedding obtained through the readout function is combined with the [REG] token embedding to regress the RoI bounding box coordinates.

combining the linguistic and visual input representations and passing the combination through a Transformer encoder-decoder, and (d) a layout graph for encoding spatial relationships (e) RoI bounding box regression of the command's target region.

6.5.1 DocEditor Model

Multimodal Feature Extraction: Our model receives input from three modalities: textual request description, the document's visual objects, and document text tokens. We extract embeddings corresponding to each modality and project them into a common d-dimensional latent space. **(1) Textual Request Embedding**: Given the user request, we encode the request words $w_1, w_2, \cdots, w_i$ into a sequence of T WordPiece tokens using SentencePiece (Kudo and Richardson 2018). We use a vocabulary of 32,000 workpieces obtained from a pre-trained

Transformer model to convert the tokens into the request text embedding, and then project them into a d dimensional embedding, yielding $z^{rtext} \in R^{d \times T}$. **(2) Visual Object Embedding**: Given a document image, we use pre-trained object detectors to obtain a set of N visual objects in the document. Inspired by (Singh et al. 2019), we extract the visual object features from the object detector's output. These features are linearly transformed into d-dimensional vector space to get the object embedding as $z^{obj} \in R^{d \times N}$. Further, we extract the normalized 2D-bounding box coordinate b_n^{obj} of each object box. **(3) Document Text Embedding**: We obtain a set of M document text tokens from the document image using the OCR system. We extract the 300-dimensional FastText vector (Bojanowski et al. 2017), 604-dimensional Pyramidal Histogram of Characters (PHOC) (Almazán et al. 2014) vector, and normalized 2D bounding box coordinates b_m^{dtext}. We concatenate all the features and linearly project them into a d-dimensional space to get the final document text embedding as $z^{dtext} \in R^{d \times M}$.

Text-enriched Visual Object Representation: Building a common embedding space for user request text, image features, and document text is challenging because there may be hundreds of document text tokens in a text-rich document. Fitting the entire set of document text tokens in the input space may become infeasible due to the increasing computational complexity of multi-headed attention that grows quadratically to the input dimension space. Moreover, not all document text contributes equally to grounding the edit text in the image. There is a need to better exploit the associations between bounding boxes corresponding to document objects (e.g., paragraphs) and the nearby document text at a document level to handle such edit requests that indirectly reference local document objects through their associated document text tokens. Thus, we propose the Text-enriched Document Object representation module (as shown in Fig 6.2) which contextually integrates the visual objects with their overlapping document text by

computing the position-guided attention score vector a_n between the n^{th} visual object and m document text tokens for all $n = 1, \cdots, N$ as follows,

$$a_n = softmax((W^Q b_n^{obj})^T * [W^K b_1^{dtext}, \cdots, W^K b_M^{dtext}] \tag{6.1}$$

where W^Q and W^K are the query projection matrix and key projection matrix, respectively. The document text attended embedding representation for the n^{th} visual object is calculated as the weighted sum of the M document text embeddings given by following equation $z_n^{obj|dtext} = [z_1^{dtext}, \cdots, z_M^{dtext}] * a_n^T$. Each n^{th} object is then represented by aggregating the object feature embedding z_n^{obj}, document text attended object representation $z_n^{obj|dtext}$ and the linear projection of the object bounding box coordinate $W^{obj} b_n^{obj}$ given as: $\hat{x}_n^{obj} = z_n^{obj} + x_n^{obj|dtext} + W^{obj} b_n^{obj}$. The input sequence of object embeddings is represented by $\hat{z}^{obj} = [\hat{z}_1^{obj}, \cdots, \hat{z}_N^{obj}]$.

Multimodal encoder-decoder for command generation: We first fuse the multimodal input context comprising of user request embedding z^{rtext} and Text-enriched visual object representation $\hat{z}^{obj}$. We further pre-append a learnable embedding (called [REG] token (Deng et al. 2021), and denoted by r) to the multimodal input for mapping the spatial location of the edit intent. The combined multimodal embedding input for our encoder-decoder model is formulated as $z^{input} = z^r \oplus \hat{z}^{obj} \oplus r$, where $\oplus$ represents concatenation. The [REG] token is randomly initialized at the beginning of the training stage and optimized with the whole model.

We then utilize the Text-to-Text (T5) Transformer (Raffel et al. 2020) as our base encoder-decoder architecture to take our input and generate a command sequence. We retain the originally proposed model while modifying the input and output layers to accommodate the additional [REG] token. The multi-head attention mechanism in the Transformer model allows each pair of tokens from the joint embedding to attend to each other across modali-

ties. As a result, the decoder's hidden states as well as the output state of the [REG] token can leverage a consolidated multi-modal representation for localization-aware and layout-oriented command generation and box coordinates regression tasks. The output hidden states from the Transformer model can be represented as $h^{out} = \text{Transformer}(z^{input})$ such that $h^{output} = [h_1^{rtext}, \cdots, h_T^{rtext}; h_1^{obj}, \cdots, h_N^{obj}; h^r]$, where h^{rtext}, h^{obj}, and h^r refer to the output hidden states corresponding to the request text, object, and [REG] embeddings, respectively. We perform greedy decoding, i.e. choose the highest-probability logit at every time step, to generate the output command text.

A Layout Graph to Encode Spatial Relationships: User requests often indirectly reference the components relative to other neighboring objects or text in the document. We hypothesize that the model should reason about the local layout within the region of interest for improving its predictive performance. Hence, we build a Document Layout Graph $G_D = (V, E)$ to encode the relative spatial relations between visual object boxes and text positions. Here, $V = \{V^{obj}, V^{dtext}\}$, where V^{obj}, V^{dtext} are the set of nodes corresponding to N object nodes $V_1^{obj}, \cdots, V_N^{obj}$, and M document text token nodes $V_1^{dtext}, \cdots, V_M^{dtext}$, respectively. The node embeddings of object nodes are extracted from the output hidden states corresponding to the object boxes $h_n^{obj} \forall n \in \{1, \cdots, N\}$. In the case of document text token nodes, we directly use the document text token embedding $z_m^{dtext} \forall m \in \{1, \cdots, M\}$ as the node embedding. The Layout Graph contains three types of edges E: **(1) Object-Text Token Edges**: directed edges for node affiliation if the document text token box lies entirely within the object box. **(2) Text-Text Token Edges**: Connecting all neighboring document text token boxes may lead to dense and isolated components, while joining adjacent tokens in the same line may produce disconnected components. We instead build a β-skeleton of all document text token boxes in the document image with $\beta = 1$ (Kirkpatrick and Radke 1985)

since such edges provide a balance between connectivity within a local cluster of document text tokens and ensure that the whole graph is one connected component (Berg et al. 1997). The graph is constructed on peripheral points of the document text token boxes with at most one edge between each pair of boxes. All connections in the β-skeleton graph are added as undirected edges to the layout graph. **(3) Object-Object Box Edges**: directed edges weighted by the type of spatial position between two object boxes in the document. Inspired by (Yao et al. 2018), we define ten types of spatial relations – inside, overlap, and 8-way orientations including up, down, left, right, upper-left, upper-right, bottom-left, bottom-right.

We use the Gated Relational Graph Convolution Network (GR-GCN), a gated variant of R-GCN to model our layout graph. GR-GCN is able to learn highly relational data relationships in densely-connected graph networks. The layout graph is passed through two layers of GR-GCN to obtain enriched graph node embeddings $G_D^{''}$.

Bounding Box Prediction: Our proposed model directly infers the bounding box coordinates of the region of interest over the document image. We aggregate the node embeddings corresponding to all object and document text token nodes in $G_D^{''}$ using a summation-based graph readout function (Xu et al. 2018b) which is mathematically denoted as $g_{out} = \rho(\sum_{v_i \in G_D^{''}} W_g v_i)$, where W_g is a learnable matrix. We concatenate the output state of [REG] token from the Transformer decoder h^r and the readout output g_{out}, and pass it through a regression block which is implemented as an MLP with a ReLU activated fully-connected layer and a prediction head with four outputs for each bounding box coordinate $b^{'}$ as $b^{'} = ReLU(Dense(h^r \oplus g_{out})$.

	System	EM (%)	Word Overlap F1	ROUGE-L	Action (%)	Component (%)
Baselines	Generator-Extractor	6.6	0.25	0.22	36.7	8.5
	GPT2	11.6	0.76	0.76	79.7	27.2
	BART	19.7	0.78	0.76	81.2	29.5
	T5	20.4	0.79	0.76	81.4	29.8
	BERT2GPT2	7.3	0.37	0.39	45.2	9.2
	LayoutLMv3-GPT2	8.7	0.39	0.40	47.6	10.3
	CLIPCap	8.5	0.25	0.27	44.5	9.34
	DiTCap	23.6	0.81	0.80	82.5	25.5
	Multimodal Transformer	31.6	0.82	0.83	83.1	32.4
Ours	**DocEditor**	**37.6**	**0.87**	**0.86**	**87.6**	**40.7**
Ablation	w/o Text Embedding	6.7	0.15	0.12	6.75	6.5
	w/o Visual Embedding	33.6	0.74	0.75	77.5	36.9
	w/o Layout Graph	32.7	0.75	0.76	82.2	37.5
	w/o Bounding Box Regression Loss	33.6	0.80	0.79	85.2	38.2
	w/o Anchor Box Prediction Loss	35.8	0.84	0.83	84.4	39.5

(a) DocEdit-PDF

System	EM (%)	Word Overlap	ROUGE-L	Action (%)	Component (%)	Attribute (%)
Generator-Extractor	10.1	0.33	0.31	33.4	15.9	14.5
GPT2	16.6	0.78	0.76	76.4	24.5	18.2
BART	19.5	0.79	0.77	77.1	25.1	25.3
T5	20.0	0.80	0.78	77.5	25.8	25.7
BERT2GPT2	6.5	0.31	0.30	36.0	18.6	9.5
LayoutLMv3-GPT2	9.6	0.36	0.34	38.3	20.1	12.6
CLIPCap	9.3	0.24	0.25	19.78	13.6	14.2
DiTCap	18.9	0.79	0.77	77.8	25.4	25.6
Multimodal Transformer	32.8	0.83	0.81	79.5	48.6	35.2
DocEditor	**38.2**	**0.86**	**0.86**	**84.5**	**52.2**	**43.5**
w/o Text Embedding	6.1	0.13	0.11	6.4	6.9	6.5
w/o Visual Embedding	34.0	0.77	0.77	79.5	44.2	37.7
w/o Layout Graph	33.5	0.79	0.77	79.1	46.1	38.3
w/o Bounding Box Regression Loss	34.2	0.83	0.82	82.5	47.1	39.2
w/o Anchor Box Prediction Loss	35.0	0.82	0.78	83.3	49.8	41.7

(b) DocEdit-Design

Table 6.2: Results comparing the performance of DocEditor for command generation with baselines and ablations on DocEdit-PDF and DocEdit-Design datasets. Bold represents the best-performing model. DocEditor outperforms all baseline methods.

6.5.2 Training DocEditor

Command Generation Loss: For generating the textual part of the desired output command, we utilize the pre-trained weights of T5 which were obtained by performing a denoising pre-training task on 750 GB cleaned English text data from the publicly-available Common Crawl web archive. We fine-tune the backbone Transformer architecture using standard maximum likelihood, i.e. using teacher forcing (Williams and Zipser 1989b) and a cross-entropy loss between predicted token $t_i^{'}$ and ground truth token t_i as $L_{gen} = -\sum_i t_i \log(t_i^{'})$, where $t_i = 1$ for

token predicted correctly.

Bounding Box Regression Loss: To address the problem of scaling effects due to varying sizes of the predicted boxes, we predict normalized bounding box coordinates between 0 and 1000, which are then scaled by the document image dimensions to retrieve original dimensions. We utilize a weighted sum of the scale-invariant generalized IoU loss (GIoU) (Rezatofighi et al. 2019) and the smooth L1 loss for the standard regression problem. Let $b = (x, y, w, h)$ denote the prediction the normalized ground-truth box as $b^{'} = (x^{'}, y^{'}, w^{'}, h^{'})$. The training objective of our bounding box regression is: $L_{bbox} = L_{smooth-l1}(b, b^{'}) + \lambda L_{giou}(b, b^{'})$, where $L_{smooth-l1}$ and L_{giou} are the smooth L1 loss and GIoU loss, respectively. λ is a hyperparameter. **Anchor Box Prediction Loss**: Not all object or document text token boxes are relevant to the edit intent. Hence, the model should have the ability to select the ones that highly overlap with the ground truth RoI. We treat each node in the layout graph as an anchor and perform binary node classification to predict if the object or document text token box lies entirely within the ground truth region of interest (RoI). We optimize the anchor prediction as an auxiliary task through the binary cross-entropy loss as $L_{anchor} = -\sum_{V_i \in G_D^{''}} y_i \log V_i$ where $y_i = 1$ if the object box overlaps with RoI, else 0.

Multitask Training: Command generation, bounding box regression and anchor box prediction tasks are all correlated as they share a common linguistic, spatial and visual latent space, and can reinforce each other. Hence, we use multi-task training to optimize both tasks simultaneously. The final optimization uses a weighted sum of L_{gen}, L_{reg}, L_{anchor} such that total loss $L = \lambda_1 L_{gen} + \lambda_2 L_{reg} + (1 - \lambda_1 - \lambda_2) L_{anchor}$, where the weighting factors λ_1, λ_2 are hyperparameters.

6.6 Experiments

Baselines: We compare DocEditor against several unimodal and multimodal baselines for the command generation task:**Seq2seq Text-only**: We use GPT2 (Radford et al. 2019), BART (Lewis et al. 2020b), and T5 (Raffel et al. 2020) that input only the user text description. **Generator-Extractor** uses BERT+DETR with an autoregressive decoding head for command generation. **Tranformer Encoder-Decoder** (Rothe, Narayan, and Severyn 2020): Combines GPT2 decoder with LayoutLMv3 encoder (LayoutLMv3-GPT2) or BERT encoder (BERT2GPT2). **Prefix Encoding** (Mokady et al. 2021): We utilize intermediate learned representations from a pre-trained encoder (CLIP (Radford et al. 2021) and DiT (Li et al. 2022)) as a prefix to the GPT2 decoder network and fine-tune on downstream tasks. **Multimodal Transformer (M4C)** (Hu et al. 2020): Combines multimodal input from user description, visual objects, and document text with a text generation decoder instead of the copy pointer mechanism. For the RoI bounding box prediction task, we compare DocEditor against visual grounding methods such as ReSC-Large (Yang et al. 2020b) and TransVG (Deng et al. 2022) for direct coordinates regression. **Evaluation Metrics**: We report exact match accuracy (EM %), Word overlap F1and ROUGE-L (Lin 2004). In order to evaluate at a more granular level, we compute the exact match accuracy for actions, components, and attributes. We evaluate bounding box prediction in terms of top-1 accuracy (%) (Jaccard overlap ≥ 0.5).

6.7 Results

Performance Comparison of command generation: Table 6.2 compares the performance of DocEditor model against other contemporary baselines on the DocEdit-PDF and

System	DocEdit-PDF Top-1 Acc (%)	DocEdit-Design Top-1 Acc (%)
ReSC-Large	17.04	15.89
TransVG	25.34	24.89
DocEditor	**36.50**	**34.34**
w/o Text Embedding	3.33	3.25
w/o Visual Embedding	22.45	20.47
w/o Layout Graph	14.48	15.56

Table 6.3: Results comparing the performance of DocEditor for RoI bounding box regression with baselines and ablations on DocEdit-PDF and DocEdit-Design datasets.

Dataset	GPT2	BART	T5
CAISE (Kim et al. 2022)	60.1	59.5	42.8
ILLC-IER (Lin et al. 2020b)	57.7	55.8	46.9
DocEdit-PDF (Ours)	11.6	19.7	20.4
DocEdit-Design (Ours)	16.6	19.5	20.0

Table 6.4: Results comparing the difficulty of contemporary language-driven image-editing datasets.

DocEdit-Design datasets. Our proposed model achieves significantly better performance across both PDF and design template documents when compared to the text-only and multi-modal baselines used in prior command generation work. We attribute this to DocEditor's ability to localize structured components through Text-enriched object box embeddings and contextualize relevant visual objects and document text tokens through multi-head attention in contrast to text-only approaches that lose these visual cues and prior multi-modal approaches that do not leverage the document structure. Moreover, DocEditor exploits the anchor box prediction loss to determine the mutual importance of each object and document text token box which helps it improve over the multimodal transformer baseline. However, it can also be observed that there is ample room for improvement in both types of document settings. We attribute this to the inherent difficulty of the task and motivate further research by discussing current shortcomings through error analysis.

Performance Comparison of RoI Prediction: We compare the RoI bounding box prediction performance of baselines with the proposed model in Table 6.3. We re-purpose scene-text visual grounding baselines for our task due to similarity in the input space. Transformer based TransVG (Deng et al. 2021) model outperforms other competitive baselines as it contextual learns the visual and linguistic information through a common embedding space. Our method further improves this architecture by enhancing the output of [REG] token embedding by output from the layout graph. **Ablation Analysis**: Table 6.2 and 6.3 analyze the ablations for each component of the `DocEditor`. The textual modality of the user request is most critical–removing it yields the random baseline. Removing any other model component does not degrade the performance below this benchmark, which aligns with the fact that the edit command generation task cannot be solved without the edit request descriptions. Removing the Layout Graph severely degrades bounding box regression performance as well as text match accuracy because the model loses the ability to spatially localize the relevant objects and document text tokens. Removing the text-enriched object box embedding significantly affects the consistency of text being generated and the regression box overlap as the model can no longer utilize the document text to match the referred component in the descriptions. **Comparison with contemporary tasks**: We compare the difficulty of our proposed language-guided document editing task with existing image editing tasks through their performance on naive text generation models. We hypothesize that if the text-only modality can provide enough information for solving the task, it will make the image modality redundant and trivialize the overall task to seq2seq generation. Table 6.4 summarizes the performance of GPT2, BART, and T5 across language-guided image editing datasets - CAISE (Kim et al. 2022) and ILLC-IER (Lin et al. 2020b). We observe that text-only models achieve a high exact match accuracy ($\tilde{6}0\%$). We conclude that samples in these datasets contain many

generic edit commands that are neither user-specific nor require a visual or spatial understanding of localized components. Our dataset struggles to achieve one-third of performance ($\leq$ 20%) compared to other datasets, necessitating research in non-trivial multimodal methods for closing the performance gap with expert humans. We observe that the proposed model is unable to handle commonsense reasoning on world knowledge and makes errors when it is required to parse several attribute modifications simultaneously for the same component.

6.8 Conclusion and Future Work

We present a dataset for language-guided document editing with instances of user edit requests on PDFs and design templates and their ground truth executable command for real-world document editing automation. We also present `DocEditor`, a Transformer-based localization-aware multimodal model that outperforms the competitive baseline for command generation tasks and edit RoI prediction tasks. We provide qualitative analysis with examples to gain insights on the limitations of the proposed model to motivate future work along several interesting directions of conversational document editing and intelligent document assistance.

CHAPTER 7

DocLayoutTTS: Dataset and Baselines for Layout-informed Document-level Neural Speech Synthesis

Abstract

We propose a new task of synthesizing speech directly from semi-structured documents where the extracted text tokens from OCR systems may not be in the correct reading order due to the complex document layout. We refer to this task as layout-informed document-level TTS and present the DocSpeech dataset which consists of 10K audio clips of a single-speaker reading layout-enriched Word document. For each document, we provide the natural reading order of text tokens, its corresponding bounding boxes, and the audio clips synthesized in the correct reading order. We also introduce DocLayoutTTS, a Transformer encoder-decoder architecture that

generates speech in an end-to-end manner given a document image with OCR extracted text. Our architecture simultaneously learns text reordering and mel-spectrogram prediction in a multi-task setup. Moreover, we take advantage of curriculum learning to progressively learn longer, more challenging document-level text utilizing both DocSpeech and LJSpeech datasets. Our empirical results show that the underlying task is challenging. Our proposed architecture performs slightly better than competitive baseline TTS models with a pre-trained model providing reading order priors.

7.1 Introduction

Text-to-speech (TTS) is an important task in speech language processing to enable human-machine interaction that is intelligible and indistinguishable from human speech. Prior works in neural TTS have achieved near human-level speech synthesis ability using recent attention-based autoregressive methods such as Tacotron 2 (Shen et al. 2018) and Transformer-based end-to-end speech synthesis models (eg. Transformer-TTS) (Li et al. 2019). However, synthesizing coherent speech from text in documents remains a challeng problem due to two reasons: (1) long sequence length of input text; (2) lack of correct reading order provided by off-the-shelf Optical Character Recognition (OCR) engines that tend to arrange all recognized tokens in a top-to-bottom and left-to-right manner, and disregard the layout of the long-form text (Clausner, Pletschacher, and Antonacopoulos 2013).

The capability to perform long-context speech synthesis is needed for several tasks such as singing voice synthesis (SVS) systems (Hono et al. 2021), document readers and screen reading systems (Pradhan et al. 2022), reading out audio-based online content (e.g. news

articles, audiobooks, and podcasts), and conversational speech generation (Cong et al. 2021). Typically, the raw input text comprising of large coherent speech units such as paragraphs is pre-processed into utterances (e.g., sentences) and are treated independently from each other, thus discarding the original ordering. Moreover, directly concatenating parts of independently synthesized audio units can make it unnatural due to various reasons. These include lack of spontaneous prosodic phenomena like filled pauses, prolongations, voice modulation, variations in fundamental frequency through time, or the speech rate peaking around the middle of a larger unit (Cambre et al. 2020). Prosodic variation is governed by context at different levels, and contextual information is expressed through prosody (Cole 2015).

Current TTS systems assume that the reading order sequence of input text tokens is correct. However, the reading order itself depends on the structure of the document and is not known apriori. In fact, current OCR systems cannot infer this correctly from complex spatial documents. Linearizing text tokens based on bounding box coordinates is not optimal for certain document types, such as multi-column templates, tables, forms, and invoices, where text may be structured spatially in a layout. Synthesizing speech from a scrambled sequence of text tokens may result in unacceptable results, thereby deteriorating the quality of human computer interactions and making documents inaccessible for people with visual disabilities (Pradhan et al. 2022). Recently, some deep learning-based methods have been proposed to perform reading order detection (Wang et al. 2021a). However, pre-processing raw text sequences discards most orthographic knowledge of the structure of the original text data, making it harder to accurately model prosodic variations in speech. Additionally, mere reordering of text tokens followed by synthesis of isolated short sentences distorts the natural *phrasing* (Klimkov et al. 2017) of text, and may discard the larger context and structure of the long-form text.

Main Contributions: We propose the task of document-level layout-informed text-to-speech synthesis that aims to generate human-level speech corresponding to the correct reading order of the text present in a semi-structured document. Furthermore, we release DocSpeech, a benchmark dataset of 10K speech samples corresponding to Word documents with a wide variety of semi-structured layouts.

The process of synthesizing layout-informed speech for semi-structured documents can benefit from solving the reading order unscrambling and long-form audio generation tasks simultaneously. Therefore, we present a strong neural model, DocLayoutTTS, to jointly model text reading order detection as well as speech mel-spectrogram synthesis using a Transformer encoder-decoder. Using curriculum learning (Kong et al. 2021), our DocLayoutTTS model demonstrates competitive performance on the task of layout-informed document-level TTS. Some novel aspects of our work include:

1. We propose a new task for layout-informed document-level TTS to generate speech from text present in semi-structured documents. We curate a public dataset, DocSpeech, which consists of 10K audio clips of a single speaker reading documents with complex layouts. We provide OCR reading order as well as unscrambled text transcription for each clip. The resulting audio clips have a total length of approximately 830 hours, with an average clip duration of 5 minutes, compared to 10 seconds in the LJSpeech dataset.
2. We present DocLayoutTTS, a neural baseline architecture that simultaneously learns text reordering, newline prediction, and mel-spectrogram prediction for synthesizing speech from documents in our proposed dataset in a multi-task fashion.
3. Our proposed model uses curriculum learning to learn increasingly long document-level

speech synthesis, starting with short speech utterances from the LJSpeech dataset. We compare the performance of DocLayoutTTS with other strong baselines and find that DocSpeech is a challenging dataset for the proposed task.

7.2 Related Work

Long-form document-level TTS: Recurrent neural network models such as Tacotron (Wang et al. 2017) use attention-mechanism to align the target text and output a spectrogram. On the other hand, Tacotron 2 (Shen et al. 2018) system uses location-sensitive attention to extend the alignment between the encoder and decoder to the information of the previous time step. However, they are still limited to synthesizing few sentences of text into speech due to constraints on long-range input sequences. Transformer-based end-to-end text-to-speech synthesis models such as Transformer-TTS (Li et al. 2019) use multi-headed attention to solve the long range dependency problem. However, as the sequence length of the input increases, the computational complexity of training the Transformer model rises quadratically. (Hwang and Chang 2021) used attention-masking along with curriculum learning to extend the maximum synthesis length to 5 minutes. However, most of these prior works are limited by the fact that they rely on well-formed phoneme sequences as input to generate the mel-spectrograms. We hypothesize that a multi-task objective that simultaneously learns reading order detection, newline prediction, and mel-spectrogram generation can help exploit latent layout signals for recovering the correct reading order, thereby preserving the natural prosody required for human-like document-level speech.

Reading Order Detection in Text: Several previous studies have explored reading order

Statistics	LJSpeech	DocSpeech
Total Clips	13,100	10,000
Total Words	225K	1800K
Total Characters	1308K	10500K
Total Duration	24 hr	830 hr
Average Clip Duration	6.57 sec	5 min
Min Clip Duration	1.11 sec	1.05 min
Max Clip Duration	10.10 sec	10.2 min
Mean Words per Clip	17.23	156

Table 7.1: Dataset statistics comparing the proposed DocSpeech with LJSpeech dataset. DocSpeech has fewer total speech samples but significantly larger total and mean clip duration.

prediction in text. (Aiello, Smeulders, et al. 2003) was one of the earliest works to propose a rule-based learning method for identifying reading order sequences in text. (Malerba, Ceci, and Berardi 2008) applied domain knowledge to determine the reading order relationship between logical document components. More recently, deep learning-based methods have been widely used for this task. (Li et al. 2020f) used visual layout features encoded through a graph neural network to reorganize OCR text into a proper sequence. Most recently, (Wang et al. 2021a) provided an seq2seq model for text reordering based on semantic, visual and spatial signals. In contrast, our work is the first attempt in terms of studying the necessity of reading order detection for text-to-speech tasks.

7.3 DocSpeech Dataset

We create DocSpeech, a synthetic dataset by re-purposing the open-source ReadingBank dataset (Wang et al. 2021a) which provides semi-structured Word documents with the reading sequence of words as extracted from DocX files, correct reading order sequence based on structured layout, as well as corresponding bounding boxes for each text token extracted from the PDF versions of the DocX files. We sample a subset of 10K documents from ReadingBank such that each file has more than 50 words.

We used the Gentle Forced Aligner[1], a Kaldi[2]-based audio alignment tool to perform forced alignment of words and audio snippets on LJSpeech[3] dataset. We obtain the word-level audio alignment from the generated time-marked conversation file which is used to construct an audio mapping of each unique word with its corresponding mel-spectrogram. We combine mel-spectrograms corresponding to each token in the correct reading order of the document text file. However, if simply joined, an unnatural voice may be generated due to the audio concatenation step. To prevent this, we insert an empty m-token (mel spectrogram token) between each consecutive word-level mel-spectrogram. The m-token, allows the speech to pause naturally between consecutive mel-spectrograms, giving the effect of naturally linked words. DocSpeech contains 10,000 document-speech pairs with an average clip duration of 5 minutes, out of which 100 files are used for testing and the remaining for training. Table 7.1 summarizes the dataset statistics about the DocSpeech dataset.

[1]https://lowerquality.com/gentle/
[2]https://kaldi-asr.org/
[3]https://keithito.com/LJ-Speech-Dataset/

7.4 Our Approach

In this section, we describe the problem statement, individual components of DocLayoutTTS model as illustrated in Figure 7.1, and training paradigm for optimizing the model.

7.4.1 Problem Formulation

We formally define the document-level layout-informed text-to-speech task. Given a semi-structured document $\mathscr{D}$ with words w_i acquired through an OCR along with their corresponding bounding box coordinates (x_1, y_1, x_2, y_2) (where (x_1, y_1) and (x_2, y_2) are the top-left and bottom-right coordinates, respectively), we aim to synthesize speech mel-spectrogram $\mathscr{S}$ such that the constituent words are sorted into their correct reading order in the speech output. We derive the ground truth reading order from the embedded XML metadata of Word documents. Further, the WORD documents are converted into the PDF format to extract the 2D bounding box of each word using Google Tesseract[4].

7.4.2 Textual Layout Encoder

Inspired by Transformer-TTS architecture (Li et al. 2019), we include a text-to-phoneme converter to learn the mapping between different regularities between the text syllabi and phonemes. Each incoming phoneme is passed through an encoder prenet to embed the phoneme input into a trainable embedding of 512 dimensions, followed by a batch normalization, ReLU activation, and a dropout layer. We add positional encoding (*PE*) (Vaswani et al. 2017) scaled by a factor of α to the processed phoneme input to take advantage of the relative token sequence of input.

[4]https://github.com/tesseract-ocr/tesseract

Additionally, we add four 2D-positional encoding ($PE^{2D}_{x_0}$, $PE^{2D}_{x_1}$, $PE^{2D}_{y_0}$, $PE^{2D}_{y_1}$) to the phoneme input for learning the relative spatial position in a document. The four 2D-positional embedding layers correspond to the upper (y_0), lower (y_1), left (x_0), and right (x_1) coordinate directions, respectively. Each input phoneme h_i being fed to the encoder is represented by the following equation:

$$h_i = prenet(phoneme_i) + \alpha * PE(i) + \beta * (PE^{2D}_{x_0}(i) + PE^{2D}_{x_1}(i) + PE^{2D}_{y_0}(i) + PE^{2D}_{y_1}(i))$$

7.4.3 Decoder

Reading Order Sequence Decoder: In the sequence decoding stage, the source and target are reordered sequences. We constrain the target sequence prediction to be the correctly ordered indices in the source sequence. Additionally, we also predict if a particular input position indicates the start of a new line in the text document. Specifically, we perform a binary classification at each decoding step to check if the token denotes the end of reading order line due to layout constraints or end of page width.

Melspectrogram Decoder: Similar to TransformerTTS (Li et al. 2019), we use a Transformer decoder using multi-head attention to integrate the encoder hidden states in multiple perspectives. We experiment with a larger embedding space of $d = \{1024, 2048, 4096\}$ compared to 512 embedding space of Transformer TTS to better model long-range context vectors. We employ a WaveNet vocoder to synthesize audios from the generated mel-spectrograms.

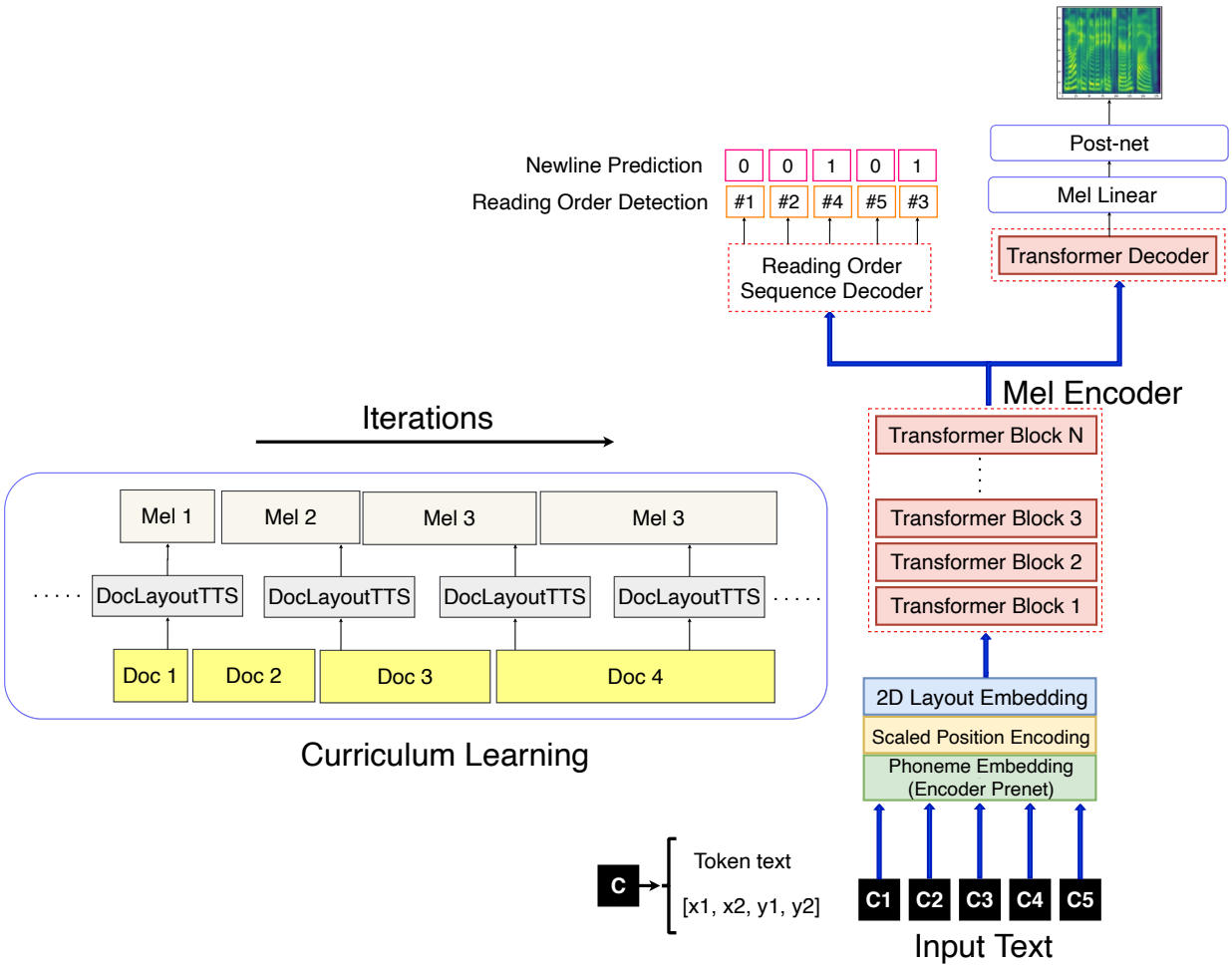


Figure 7.1: DocLayoutTTS model takes a sequence of text tokens as input along with their bounding box coordinates. Encoder Prenet converts the input into a sequence of phonemes which are passed through a phoneme embedding. Scaled Position encoding and Layout encoding are added to the input and passed through the Transformer encoder. Mel decoder predicts the mel spectrograms while the reading order sequence decoder predicts the indices corresponding to each word. We also predict the newline which denotes a break in the left-to-right traversal of reading order sequence in the document. Curriculum learning feeds increasingly long text sequences as input to gradually train the model with more difficult input samples.

7.4.4 Multi-task Training

We use mean absolute error (MAE) to predict the mel-spectrogram. Reordered sequence index classification uses categorical cross-entropy loss, while newline prediction uses a weighted binary cross-entropy loss to adjust for class imbalance. All three tasks are correlated and reinforce each other, so we use multi-task training to optimize them simultaneously. The final optimization uses a weighted sum of the link prediction loss and element classification loss where the weighting factors λ and γ are hyperparameters as shown in the following Equation:

$$\mathscr{L}_{total} = \lambda\mathscr{L}_{mel} + \gamma\mathscr{L}_{reorder} + (1 - \lambda - \gamma)\mathscr{L}_{newline}$$

7.4.5 Curriculum Learning

Inspired by (Hwang and Chang 2021), we utilize curriculum learning (Bengio et al. 2009) to improve the training process for document-level TTS. Curriculum learning is a deep learning training process where the difficulty of learning becomes gradually more complex. We apply curriculum learning to train the encoder-decoder network to help generate longer sequence inputs without losing long-range context. We utilize LJSpeech and our proposed DocSpeech to achieve an increasingly difficult curriculum in terms of the input sequence length. We initially start with sentence-level input of LJSpeech. In the subsequent iterations, we input document text with increasing lengths, until all text input sequences have been exhausted. In order to fit the data in the limited GPU capacity, the model was set to automatically reduce the batch size to 1/2 whenever the GPU capacity limit was reached. This process continued until the batch size was reduced to 1 and could not go down further.

7.5 Experiments

In this section, we detail our experiments to test the proposed DocLayoutTTS model with the DocSpeech dataset. We compare our method with strong baselines and evaluate the synthesized audio quality in terms of MOS score.

7.5.1 Baselines

We compare DocLayoutTTS with two state-of-the-art pre-trained TTS models - Tacotron 2 (Shen et al. 2018) and TransformerTTS (Li et al. 2019). The input to these models is the original sequence of text extracted by the OCR. LayoutReader (Wang et al. 2021a) is a state-of-the-art reading order detection model. We also test using a separate model for text reordering and using the reordered output as input to TTS models. In this direction, we use a pre-trained LayoutReader for reading order detection, followed by Tacotron 2 and TransformerTTS for TTS. This provides an opportunity to analyze the effects of decoupling the reading order detection process from speech synthesis. We also compare our method to DLTTS, an RNN-based document-level TTS model.

7.5.2 Evaluation

We select 100 testing examples from the DocSpeech dataset. Each test sample consists of the document image, extracted text tokens along with their bounding boxes, and ground truth speech output. We evaluate the mean option score (MOS) on these 100 documents generated by different models. We randomize the speech samples from baselines, our model, and the ground truth, followed by equal-sized sampling to ensure that the expert testers don't know the sources

of speech files. MOS is performed by 5 fluent English speakers who are experts in the domain of audio processing. MOS evaluation involves testers rating the quality of audio on a scale from 1 to 5 with 0.5-point increments. We also compare the mel-spectrograms generated by our model with baseline models to visualize the effect of our contributions to the quality of generated audio.

7.5.3 Training Details

We used the Pytorch framework for deep learning models. We used two Nvidia Tesla P100 GPUs to train the models. We enabled multi-GPU training to enlarge the batch size. In order to accommodate large input sequences, we used dynamic batch sizes to maximize GPU utilization. DocLayoutTTS inputs phoneme sequence. Hence, the input text was pre-processed to get the phoneme sequences by following sentence separation, text normalization, word segmentation and pronunciation. Similar to TransformerTTS (Li et al. 2019), we use a WaveNet vocoder conditioned on mel-spectrograms and trained it simultaneously using teacher forcing with a sample ground truth rate of 16000 and frame rate of ground truth mel-spectrogram equal to 80. Hyperparameters $\alpha, \beta, \gamma, \lambda$ were sampled between 0 and 1, with equal intervals of 0.1. We use Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-6}$ and a learning rate of 10^{-3}.

7.6 Results and Analysis

Comparison with Baselines: Table 7.2 shows the comparison of our proposed DocLayoutTTS with baseline models. Tacotron 2 and TransformerTTS perform poorly on the DocSpeech dataset due to their inability to handle text sequences longer than a few sentences as well as lack of layout information for token reordering. Hence, they suffer from missing words, and repetitions and the output speech is unintelligible. This is also evident in the mel-spectogram

visualizations where parts of the generated spectrogram is destroyed at multiple locations indicating failure in the decoding step. Using pre-trained LayoutReader (Wang et al. 2021a) to unscramble the reading order and feed that as input to TTS models performs significantly better. However, the long-form text still hurts the performance despite being fed the reordered input. Additionally, we experiment with the DLTTS model, which claims to be able to synthesize speech sequences for up to 5 minutes. LayoutReader + DLTTS forms the strongest baseline with a remarkable improvement in MOS score compared to others. However, it still suffers from error propagation from decoupled reading order prediction and mel-spectrogram prediction. We observe that DocLayoutTTS outperforms other strong baseline models. This improvement can be attributed to its ability to encode layout information in an end-to-end fashion as well as document-level training using curriculum learning. However, it can also be observed that the outperformance of our architecture is not quite large in magnitude, and significantly falls short compared to the ground truth recordings. We attribute this to the inherent difficulty of the task. Further research in more sophisticated models may help improve the performance on DocSpeech.

Ablation Results: We perform a detailed study the contributions of each component in our proposed architecture to attribute the source of improvements. We experiment by training the model by removing the newline prediction loss and reorder detection loss, one at a time. We observe that text reorder detection is crucial for our model to successfully produce coherent speech clips with correct word order. Newline prediction loss helps the model to take advantage of the geometric information provided by visually rich documents. Without the newline prediction loss, the model struggles to generate speech samples with naturalness in pauses and voice modulations. Finally, the addition of curriculum learning helps the model to smoothly extend

System	MOS
Tacotron2 (Shen et al. 2018)	1.75 ±0.04
TransformerTTS (Li et al. 2019)	1.82 ±0.02
LayoutReader (Wang et al. 2021a) + Tacotron2 (Shen et al. 2018)	2.05 ±0.08
LayoutReader (Wang et al. 2021a) + TransformerTTS (Li et al. 2019)	2.08 ±0.03
LayoutReader (Wang et al. 2021a) + DLTTS (Hwang and Chang 2021)	2.25 ±0.02
DocLayoutTTS (Ours)	**2.32 ±0.02**
GroundTruth	4.65 ±0.08

Table 7.2: **Quantitative Results:** Comparison of MOS score with baseline models on DocSpeech dataset. Our proposed model improves the MOS score by 0.07 over the layoutReader + DLTTS.

System	MOS
DocLayoutTTS (Ours)	**2.32 ±0.07**
w/o newline prediction loss	2.15 ±0.05
w/o curriculum learning	2.04 ±0.06
w/o reorder detection loss	1.79 ±0.04

Table 7.3: **Ablation Results:** Comparison of MOS score with ablation models on DocSpeech dataset. We perform ablation experiments to show the usefulness of different components of DocLayoutTTS model highlighted in red.

the hidden space to learn long-range text sequences, preventing catastrophic forgetting at the decoding step.

7.7 Conclusion and Future Work

We present a new dataset titled as DocSpeech for the task of synthesizing speech directly from semi-structured documents where the text tokens may not be present in correct reading order. We also present DocLayoutTTS, a strong Transformer-based TTS model that can simultaneously

learn to predict reading order of the document-level text and synthesize speech corresponding to the same. Further, our approach uses curriculum learning to extend the self-attention based audio alignment to long-form document-level input sequences. Although experiments on the proposed dataset display the effectiveness of our contributions, the task is challenging due to the large gap in speech quality between the ground truth recording and model-generated audio. In the future, we aim to explore non-autoregressive solutions that are not impacted by the sequential inference nature of the current approaches so as to scale well to document-level input text lengths.

CHAPTER 8

MONOPOLY: Financial Prediction from MONetary POLicY Conference Videos Using Multimodal Cues

Abstract

Risk prediction and price movement classification are essential tasks in financial markets. Monetary policy calls (MPC) provide important insights into the actions taken by a country's central bank on economic goals related to inflation, employment, prices, and interest rates. Analyzing visual, vocal, and textual cues from MPC calls can help analysts and policymakers evaluate the economic risks and make sound investment decisions. To aid the analysis of MPC calls, we curate the Monopoly dataset, a collection of public conference call videos along with their corresponding audio recordings and text transcripts released by six international banks between

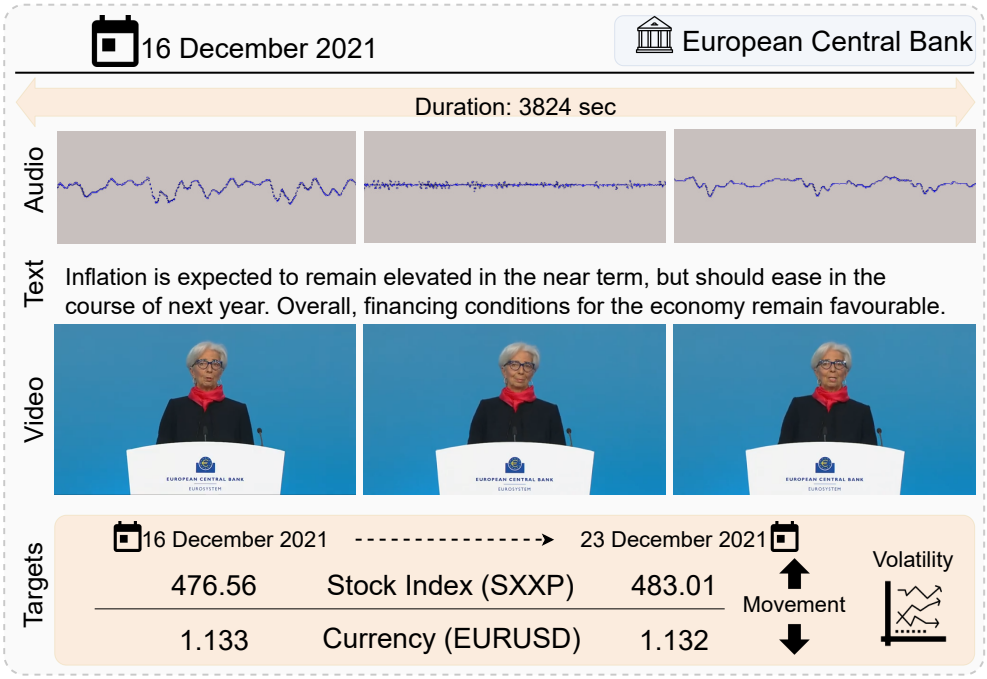


Figure 8.1: A sample from a Monetary Policy Call held by the Europen Central Bank. The Governor first presents a prepared press speech, followed by a spontaneous question and answer (Q&A) session with journalists. The meeting ended with an adverse market reaction that led to declining currency value and a high volatility in stock prices.

2009 and 2022. Our dataset is the first attempt to explore the benefits of visual cues in addition to audio and textual signals for financial prediction tasks. We introduce `MPCNet`, a competitive baseline architecture that takes advantage of the cross-modal transformer blocks and modality-specific attention fusion to forecast the financial risk and price movement associated with the MPC calls. Empirical results prove that the task is challenging, with the proposed architecture performing 5-18% better than strong Transformer-based baselines. We release the MPC dataset and benchmark models to motivate future research in this new challenging domain.

8.1 Introduction

Predicting how the prices of a financial asset will vary over a certain period is an important financial analysis task for investors and policymakers (Lewellen 2006). Understanding the

sentiment of the economy and it's associated risk perceptions can help analysts make better decisions about investment returns, while policymakers can implement cautionary monetary measures in order to maintain a healthy economy (Cai, Camara, and Capel 2021; Shapiro and Wilson 2021). With unparalleled advances in multimodal learning, a massive amount of unstructured data is accessible to investors for financial forecasting (Jiang 2021). One such rich source of information is the Monetary Policy Conference (MPC's) call. These hour-long, public video conferences are held periodically where the governors of a country's central bank[1] (eg., the Federal Reserve Bank in the United States) meet to discuss the actions undertaken to improve the financial conditions of the country, explain their stance on the monetary policy, and assess the risks to economic growth. The MPC calls are a combination of a prepared press speech by the governor followed by a spontaneous question-answering session with the journalists (Marchal 2021). The public presentation sheds light on the announcements regarding policy decisions and gives indications about the future path of the economy. The question-answer session involves the call participants like media reporters and market analysts engaging in a dialogue with the governors to analyze a range of economic factors such as inflation, employment, the value of currency, stock market growth, and interest rates on loans.

Prior works (Boukus and Rosenberg 2006; Rosa 2013) have highlighted the impact of MPC calls on financial stock markets as evident from a "higher than normal" trading across different financial assets. For instance, (Gómez-Cram and Grotteria 2022) gives an example of how the volatility of S&P 500 index can be observed to be roughly three times larger on days when the Federal Reserve Bank conducts its MPC calls in the US compared to other times. Hence, shareholders critically analyze the multimodal MPC calls to forecast stock market indices,

[1] https://www.investopedia.com/terms/c/centralbank.asp

treasury bonds, prices of gold, and currency exchange rates post the conference call (Tadle 2022). Prior findings (Gorodnichenko, Pham, and Talavera 2021) suggest that the minutes of the MPC calls can provide important market-relevant information for several financial assets as mentioned in Table 8.1 and need to be assessed systematically.

There is anecdotal evidence that non-verbal cues such as complexity of language, vocal tone and facial expressions of the speakers can be indicative and correlated with trading activities in the financial markets (Cao 2022; Li, Wu, and Bu 2016). Although existing research has used text and audio for financial predictions (Qin and Yang 2019a; Sawhney et al. 2020b; Sawhney et al. 2021a; Ramachandran and DeRose Jr 2018), use of visual cues as part of multimodal input has been largely limited. Existing NLP literature has focused on what is being said during the press conferences while there is a need to focus on how it is being said. This can accomplished by exploiting the visual aspects of the conferences for scrutinizing the human behavior such as eye-movements, facial expressions, postures, and gaits (Marchal 2021). According to (Weiss 2011), behavioral clues may reflect emotions that subjects might want to hide. Variability across different speakers makes it extremely difficult to detect these expressions in real time. For instance, Figure 10.1 depicts an MPC call held by European Central bank where the tone of the conference takes a more negative turn when the governor tries to evade questions on future inflation. The textual content indicates an optimistic outlook on long-term inflation despite an overall pessimistic vocal tone. The followup discussion depicts the speaker hesitancy in indulging more details to the reporters, accompanied with facial expressions that could indicate stress. Consequently, the meeting ended with an adverse market reaction that led to declining currency value and a high volatility in stock prices. Motivated by prior works, we explore multimodal deep learning approaches that can extract complementary information from multiple modalities

Financial assets	Impact of MPC announcements
Stock Prices	Indications of healthy, steadily growing economy increases stock prices
(Large/ Small)	Size of stock market - large *vs.* small, indicates set of all stocks *vs.* top performing stocks
Gold Price	Rise in inflation expectations raises prices of precious metals
Treasury bond yields	Higher perceived risk of recession and rising interest rates leads to price increase
(Short/Long-term)	Duration of bond term (short *vs.* long) indicates time expectation of interest rates hike
Currency Exchange Rate	Increase in employment and regulated inflation leads to appreciation in value

Table 8.1: Importance of MPC call analysis for financial forecasting.

to improve financial modeling. Our work takes the first step in multimodal financial modeling on MPC calls by utilizing the visual, vocal, and verbal modalities simultaneously.

Our **contributions** in this work can be summarized as:

- We curate a public dataset, Monopoly: Monetary Policy Call Dataset, consisting of 340 video conference calls spanning over 350 hours between 2009 to 2022 extracted from 6 major English-speaking economies - USA, Canada, European Union, United Kingdom, New Zealand, and South Africa.
- We accompany the dataset with several strong neural baselines. Our proposed methodology, MPCNet utilizes video frames, audio recordings, and utterance-aligned transcripts, learnt through a cross-modal transformer architecture and modality-specific attention fusion for volatility and price movement prediction of stock market indices, gold price, currency exchange rates, and bond prices. We provide a cumulative of 24K data points for experimentation.
- MPCNet empirically outperforms other competitive deep learning approaches by 5-18% in this new task domain.

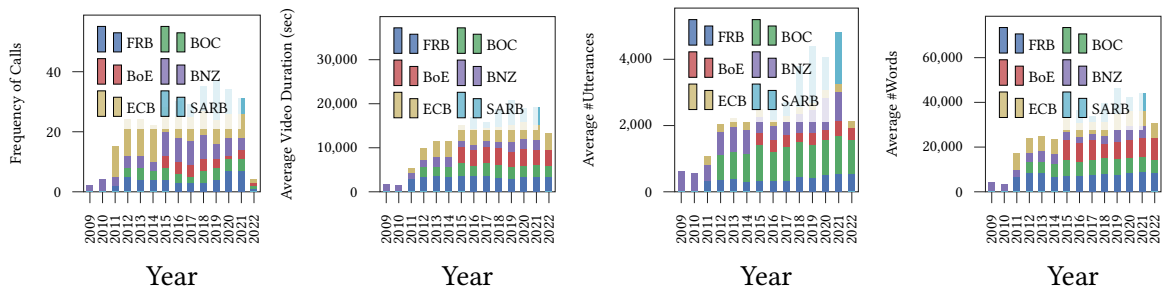
8.2 Related Work

AI in Finance Traditional financial forecasting techniques have been applied in areas such as stock markets (Rundo et al. 2019; Ariyo, Adewumi, and Ayo 2014), currency exchange markets

(Walczak 2001; Kamruzzaman and Sarker 2003), and energy economics (Ghoddusi, Creamer, and Rafizadeh 2019; Bento et al. 2018). Conventional financial models previously relied only on numerical features (Nikou, Mansourfar, and Bagherzadeh 2019), which include discrete (ARIMA (Ariyo, Adewumi, and Ayo 2014), GARCH (Bollerslev 1986), rolling regression (Peng et al. 2018)), continuous (Markov chain (Jacquier, Polson, and Rossi 2002) and stochastic volatility (Andersen 2008)), and neural approaches (Kim et al. 2019; Luo et al. 2018). Efforts have since shifted towards utilizing textual data such as social media posts, news reports, web searches, etc. (Xu and Cohen 2018a; Sawhney et al. 2021c). These approaches limit their analyses to stock markets. (Sawhney et al. 2020b) explored a multi-task setting for financial risk forecasting in stock markets using earnings calls. However, the multi-task setting is limited to simultaneous prediction of movement and volatility of a single target variable, and simultaneous prediction of multiple economic variables presents a new avenue for research in financial forecasting.

Monetary Policy Calls Previous research has shown that MPC calls provide key economic indicators that determine how the policy impacts the financial markets, and can improve financial predictions (Boukus and Rosenberg 2006; Rosa 2013). Studies have also been carried out exclusively for MPC calls (Ehrmann and Fratzscher 2007; Tadle 2022), which show that monetary policy meeting minutes affect policy expectations, often exerting an even larger effect on financial markets than the release of the policy decisions. Furthermore, the Q&A portion of the press conference serves as a clarifier of the economic outlook, particularly during times of high macroeconomic uncertainty (Ehrmann and Fratzscher 2007). There is, however, a gap in leveraging neural predictive modeling using visual, verbal and vocal cues pertaining to MPC calls for financial forecasting.

Multimodality in Financial Forecasting Existing work in the financial realm utilize vocal



(a) MPC call frequency (b) Video duration (c) Mean # utterances. (d) Mean # words.

Figure 8.2: Year-wise statistics for each bank (FRB: Federal Reserve Bank of USA, BOC: Bank Of Canada, BoE: Bank of England, BNZ: Reserve Bank of New Zealand, ECB: European Central Bank, SARB: South African Reserve Bank).

and textual cues from earnings conference calls (Qin and Yang 2019a; Sawhney et al. 2020b), and mergers and acquisitions calls (Sawhney et al. 2021a) for stock volatility prediction. Multimodal architectures that use these cues for financial predictions have seen significant improvements in their performances (Yang et al. 2020a; Sawhney et al. 2020b). However, the vision modality, which may offer important cues that correlate with the performance of financial markets (Cao 2022) remains underexplored, which we seek to address with this work.

8.3 Problem Formulation

We consider a monetary policy meeting χ which consists of three components: $\chi = [v; a; t]$. The sequence of textual utterances[2] $t = [t_1, t_2, \cdots, t_N]$ is extracted from the meeting transcript where t_i is the i^{th} utterance of the call and N is the maximum number of utterances in any call. Similarly, a is the sequence of corresponding audios for the textual utterances and is represented as $[a_1, a_2, \cdots, a_N]$. Finally, v corresponds to the sequence of the video frames corresponding to each audio segment, given by $[v_1, v_2, \cdots, v_N]$. Each utterance in a given call belongs to speaker $s \in \{governor, reporter\}$. Our goal is to forecast predictions for the set of six principle financial

[2]Due to higher complexity and noise of processing long length of videos, we segment at sentence level as opposed to the word level.

targets: $\mathbb{U}$ = {*Stock Index (Small), Stock Index (Large), Gold Price, Currency Exchange Rate, Long-term bond yield (10-years), Short-term bond yield (3-months)*}. We experiment simultaneously predicting all target variables using shared model parameters. For volatility prediction, we stack all computed volatility values $v^u_{[d,d+\tau]}, \forall u \in \mathbb{U}$ into an $|\mathbb{U}|$-dimensional target vector $\mathbf{v}_{[d,d+\tau]}$. For the movement prediction task, we similarly stack all computed movement labels $y^u_{[d,d+\tau]}, \forall u \in \mathbb{U}$ into an $|\mathbb{U}|$-dimensional target vector $\mathbf{y}_{[d,d+\tau]}$. We will now describe the two kinds of prediction tasks that we explore in this work i.e volatility and movement prediction.

Volatility: Following (Kogan et al. 2009a), we define volatility prediction as a regression problem. For a given target variable $u \in \mathbb{U}$ with price p_i on day i, the volatility is the natural log of the standard deviation of return prices r in a window of τ days, given as,

$$v^u_{[d,d+\tau]} = \ln\left(\sqrt{\frac{\sum_{i=d}^{d+\tau}(r_i - \bar{r})^2}{\tau}}\right), \quad v \in \mathbb{R} \tag{8.1}$$

where $r_i = \frac{p_i - p_{i-1}}{p_{i-1}}$ is the return price on day i of the target m, and $\bar{r}$ is the average of these returns over a period of τ days.

Price movement Following (Xu and Cohen 2018b), we define price movement $y_{[d,d+\tau]}$ over a period of τ days as a binary classification task. For a given target, whose price p can either rise or fall on a day $d + \tau$ compared to a previous day d , we formulate the classification task,

$$y^u_{[d,d+\tau]} = \begin{cases} 1, & p_{d+\tau} \geq p_d \\ 0, & p_d \geq p_{d+\tau} \end{cases} \tag{8.2}$$

Bank	Year Range	# of Data Samples
Federal Reserve	2011-22	3804
European Central Bank	2011-22	7416
Bank of England	2015-22	1728
Bank of Canada	2012-22	2808
Reserve Bank of New Zealand	2009-22	5040
South African Reserve Bank	2016-22	3384

Table 8.2: Data distribution of conference video files for each bank. The number of data samples corresponds to total data points in the Monopoly dataset corresponding to each bank.

8.4 Monopoly Dataset

Conference call transcripts and audios have been extensively studied in the past (Qin and Yang 2019a; Sawhney et al. 2021a). However, there is no existing financial conference dataset that captures the visual modality. Therefore, we present the Monopoly dataset with videos, audio recordings and text transcripts corresponding to the monetary policy committee meetings conducted by the central banks of six major economies - United States, United Kingdom, European Union, Canada, New Zealand, and South Africa. To limit the scope, we ensured all audios and transcripts were in English, and had "Monetary Policy" mentioned in their titles.

8.4.1 Dataset Acquisition

We extract the conference call videos from the official websites of the respective central banks. We used BeautifulSoup[3] Python package to web scrape the dates, video links, and transcripts of the monetary policy calls, and download the MP4 videos and PDF transcripts using Urllib[4]. Textual components of the PDF were extracted using PDFPlumber[5] python library. We use the Bloomberg Terminal[6] to extract the time series of daily prices between Jan 2000 to Mar 2022

[3]https://www.crummy.com/software/BeautifulSoup/
[4]https://pypi.org/project/urllib3/
[5]https://pypi.org/project/pdfplumber/
[6]https://bba.bloomberg.net/

corresponding to the six financial target for each conference call.

8.4.2 Dataset Statistics

Since conference calls started being reliably released post 2009, we filter and list all MPC calls between January 2009 and March 2022. These meetings are held 8 times in a year. A total of 464 MPC conference calls were downloaded. However, we discarded conference calls where text-audio-video alignment was not possible due to missing media or transcription files. The final dataset comprises of 340 conference calls of a combined duration of 15, 729 minutes with the average duration of the calls around 53 minutes. The scripted opening statement during the press conference is on average just shy of 10 minutes long, while the Q&A session usually lasts for about 44 minutes, with the governor answering an average of 22 questions and follow-ups. Table 8.2 shows the data distribution for conference calls originating from different banks. The mean number of audio utterances across the calls is 587.54 ± 38.32, with a maximum of 2462 utterances. Similarly, we observe varying lengths of conference calls with mean and maximum number of words as 6280 and 17,258 words, respectively. Looking at year-wise trends in Figure 8.2, we see that the availability of calls gets more consistent every year as more and more countries mandate public release of conference recordings. We also see a positive trend of progressively increase in all three modalities of the conference calls - total duration (visual), number of utterances (vocal), as well as the number of words (textual) each year. The dataset is split chronologically into a train, validation, and test set in the ratio of 70 : 10 : 20, respectively, to ensures that future data is not used for forecasting past data.

8.5 Methodology

8.5.1 Multi-Modal Segmentation and Alignment

Given the three modalities v, a and t, it is essential to segment them into sequences such that they align and correspond with each other. To perform segmentation, we follow existing work (Sawhney et al. 2021a) and use utterance-level embeddings, where we consider each sentence or phrase as an utterance. We perform forced alignment using the library Aeneas[7] to align the audio segments with textual utterance. Aeneas uses the Sakoe-Chiba Band Dynamic Time Warping (DTW) (Sakoe and Chiba 1978a) forced alignment algorithm, which shows high discrimination between words. The Forced Alignment algorithm takes as input a text file divided into segments $t = [t_1, t_2, \cdots, t_N]$, an unfragmented audio file a, and returns a mapping which associates each text fragment $t_j \in t$ with a corresponding time-interval in the audio file, given as $\hat{a} = [a(\tau_s^1, \tau_e^1), a(\tau_s^2, \tau_e^2), \cdots, a(\tau_s^N, \tau_e^N)]$, where $\hat{a}_j = a(\tau_s^j, \tau_e^j)$ is the j-th audio segment between timestamps τ_s^j and τ_e^j. Video frames are already aligned to the audio, i.e for a given audio segment a_j with start and end times of τ_s^j and τ_e^j respectively, we obtain the corresponding video segment $v_j = [v_j^1, v_j^2, \cdots, v_j^N]$ as a sequence of frames, given as $v_j = [v(\eta\tau_s^j), v(\eta\tau_s^j + 1), \cdots, v(\eta\tau_e^j)]$, where $v(k)$ denotes the k-th frame of the full video, η is the frame rate (in fps), and $s < e$. We use audio sampling rate of 44kHz and video frame rate of 12 fps for audio and video time series.

8.5.2 Multi-Modal Feature Extraction

Textual Features: We compute the feature representation of each utterance using BERT (Devlin et al. 2019b), which has shown to be an effective pre-trained language-based model for extracting

[7] https://github.com/readbeyond/aeneas

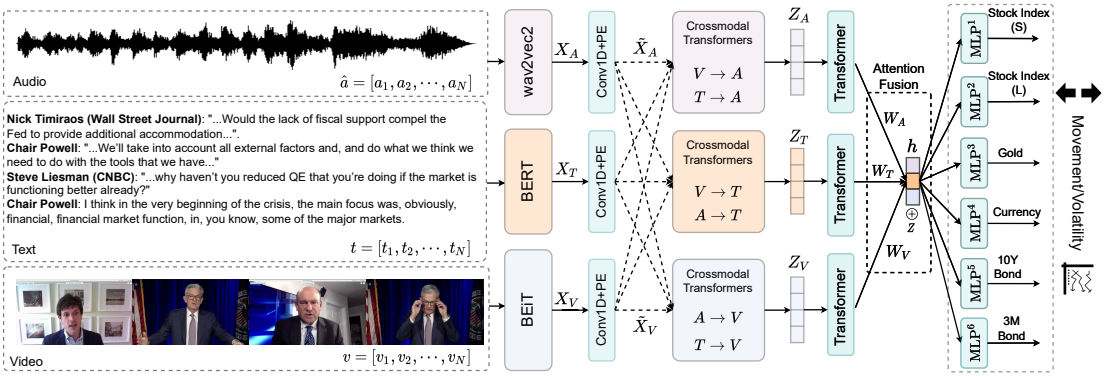


Figure 8.3: We illustrate each building block in the architectural pipeline of MPCNet, starting with (i) feature extraction (ii) locally-aware position encoding (iii) crossmodal transformer blocks iv) sentence-level transformers v) feature-fusion, and finally vi) target-specific MLPs for prediction.

word-embeddings. We embed each text utterance $t_j \in [t_1, t_2, \cdots, t_M]$ as the arithmetic mean of all its word representations from BERT, and obtain a text encoding $k_j \in \mathbb{R}^{768}$, given as x_T^j = BERT(t_j), $\forall j \in [1, N]$. We thus obtain a sequence of text embeddings $X_T = [x_T^1, x_T^2, \cdots, x_T^N]$.

Audio Features: To encode audio segments, we use wav2vec2 (Baevski et al. 2020), which has shown shown significant potential for extracting audio features for speech language understanding tasks. We embed each audio utterance a_j as the arithmetic mean of the output representation from wav2vec2, to obtain an audio encoding $l_j \in \mathbb{R}^{768}$, given as x_A^j = wav2vec2$(\hat{a}_j)$, $\forall j \in [1, N]$. The sequence of audio embeddings is represented as $X_A = [x_A^1, x_A^2, \cdots, x_A^N]$.

Video Features: We encode the video frames using BEiT (Bao et al. 2022), which is a pre-trained bidirectional transformer based encoder for extracting image representations. BEiT has shown great promise for obtaining pre-trained representations for downstream vision tasks (Hatamizadeh et al. 2022). We embed each frame v_j^k in the video fragment v_j as the arithmetic mean of visual tokens representations of that frame. We then average over all the frames to obtain the aggregated encoding $x_V^j \in \mathbb{R}^{768}$ of the segment v_j, given as $x_V^j = \frac{1}{L}\sum_{k=1}^{L}$ BEiT(v_j^k), $\forall j \in$

$[1, N]$, where L is the number of frames in the segment v_j. The sequence of video embeddings is represented as $X_V = [x_V^1, x_V^2, \cdots, x_V^N]$.

8.5.3 MPCNet: MPC Crossmodal Transformer

Due to the multimodal nature of the data, the model must learn the correlations and inter-dependencies between modalities. The model needs to accurately contrast visual, auditory, and textual information in order to characterize the speaker's affective state (Soleymani, Pantic, and Pun 2011; Chen, Wu, and Jiang 2016; Montacié and Caraty 2018). Hence, we leverage and build upon crossmodal transformers (Tsai et al. 2019; Zadeh et al. 2019), which have shown to be effective for learning fused multimodal representations through latent crossmodal adaptation. Let the set of available modalities be represented as $\mathbb{M} = \{V, A, T\}$, namely Video, Audio and Text respectively. The basic building block of the crossmodal transformer is the crossmodal attention module, which reinforces source modality α with target modality β using their respective locally-enriched feature sequences.

Locally-Aware Positional Encoding (Sawhney et al. 2021a): Given input sequence $X_\alpha \in \mathbb{R}^{L \times 768}$, where $\alpha \in \mathbb{M}$, we first pass this representation through a 1D temporal convolutional layer to capture the local sequence structure (Tsai et al. 2019; Yao et al. 2015). This step produces a locally-aware sequence of features $\hat{X}_\alpha$, given as $\hat{X}_\alpha = \text{Conv1D}(X_\alpha), \forall \alpha \in \mathbb{M}$. To enable the sequences to carry temporal information (Tsai et al. 2019; Vaswani et al. 2017), we augment positional embedding *pos* to locally-aware features $\hat{X}_\alpha$ to yield position enriched

features $\widetilde{X}_\alpha = \hat{X}_\alpha + pos$, where pos is,

$$pos_{j,2l}, pos_{j,2l+1} = \sin\left(\frac{j}{10^{\frac{8l}{d}}}\right), \cos\left(\frac{j}{10^{\frac{8l}{d}}}\right) \tag{8.3}$$

$$\tag{8.4}$$

Crossmodal Attention (Tsai et al. 2019): For two modalities $\alpha, \beta \in \mathbb{M}$ where $\alpha \neq \beta$, the crossmodal attention layer fuses crossmodal information through latent adaptation between α and β (Tsai et al. 2019). Given position-aware features $Z^{i-1}_{\alpha\to\beta}$ and Z^{i-1}_{α} at the $(i-1)^{\text{th}}$ transformer block, the intermediate latent adaption $\hat{Z}^{i}_{\alpha\to\beta}$ is computed as,

$$\tilde{Z}^{i-1}_{\alpha\to\beta} = \text{LN}(Z^{i-1}_{\alpha\to\beta}), \;\; \tilde{Z}^{i-1}_{\alpha} = \text{LN}(Z^{i-1}_{\alpha}) \tag{8.5}$$

$$\hat{Z}^{i}_{\alpha\to\beta} = \text{softmax}\left(\frac{\tilde{Z}^{i-1}_{\alpha\to\beta} W_q W_k^\top (\tilde{Z}^{i-1}_{\alpha})^\top}{\sqrt{d}}\right) \tilde{Z}^{i-1}_{\alpha} W_v + \tilde{Z}^{i-1}_{\alpha\to\beta} \tag{8.6}$$

$W_{(\cdot)}$ are learnable weight matrices, and d is the feature dimension, LN means layer-norm, and $Z^{0}_{\alpha\to\beta} = \widetilde{X}_\beta$. The intermediate latent adaption $\hat{Z}^{i}_{\alpha\to\beta}$ is then passed through a feedforward (FF) layer to yield $Z^{i}_{\alpha\to\beta}$ as $Z^{i}_{\alpha\to\beta} = \text{FF}(\text{LN}(\hat{Z}^{i}_{\alpha\to\beta})) + \text{LN}(\hat{Z}^{i}_{\alpha\to\beta})$.

Sentence-Level Transformer (Tsai et al. 2019): We concatenate $Z_{\alpha\to\beta}$ from the crossmodal transformers sharing the same target modality $\beta \in \mathbb{M}$ to yield modality specific representations $Z_\alpha, \forall\alpha \in \mathbb{M}$, given as $Z_V = [Z_{T\to V}; Z_{A\to V}], \;\; Z_A = [Z_{T\to A}; Z_{V\to A}], \;\; Z_T = [Z_{V\to T}; Z_{A\to T}]$.

Next, these hidden states are passed through self-attention transformers (Tsai et al. 2019; Vaswani et al. 2017) to collect temporal information. The temporal encodings are then concatenated and passed through a feed forward layer to yield the ensembled temporal representation Z.

Modality Specific Attention-Fusion We propose an additional attention fusion mechanism to capture the importance of a specific target modality representation Z_α with respect to sibling representations Z_β ($\alpha \neq \beta$). We first compute attention weights $W_\alpha, \forall\alpha \in \mathbb{M}$ for video, audio

and textual representations respectively, given as,

$$W_\alpha = \frac{\widetilde{W_\alpha}}{\sum_{\alpha' \in \mathbb{M}} \widetilde{W'_\alpha}}, \quad \text{where } \widetilde{W_\alpha} = \text{softmax}(\hat{W}_\alpha Z_\alpha + \hat{b}_\alpha) \tag{8.7}$$

where $\hat{W}_\alpha$ and $\hat{b}_\alpha$ are learnable parameters, and $\alpha \in \mathbb{M}$. We then fuse the attention video, audio and textual features by multiplying the computed weights with their corresponded feature representations to yield the fused temporal representation $Z_{\text{fused}} = \sum_{\alpha \in \mathbb{M}} W_\alpha Z_\alpha$

Final Network and Prediction: Finally, we combine the ensembled temporal representation Z with the fused temporal representation Z_{fused} by using a feed-forward layer with a residual block to yield the final hidden representation h, given as $h = FF(Z_{\text{fused}}) + Z$. The final hidden representation is then passed through $|\mathbb{U}|$ multi-layer perceptrons (MLPs) to yield the prediction $y^u, \forall u \in \mathbb{U}$ as $y^u = \sigma(\text{MLP}^u(h))$, where σ represents the final activation function. We use a linear activation for volatility prediction and a sigmoid for price movement, respectively. We use Mean Squared Error (MSE) and Binary Cross-Entropy (BCE) for these tasks, respectively.

8.6 Experiments

Baselines: We compare `MPCNet` against several modern and traditional baselines across varied domains and modalities as follows:

8.6.0.1 Price-based Baselines

: Utilizing historical price exclusively.

- **HistPrice**: Following (Du and Budescu 2007), we use ARIMA model to perform regression/classification on past 30-days time series.
- **P-SVM** (Chatzis et al. 2018): We apply Support Vector Regression (SVR) and Classi-

	Model	Stock Index (Small)				Stock Index (Large)				Currency Exchange Rate			
		$MSE_1\downarrow$	$MSE_3\downarrow$	$MSE_7\downarrow$	$MSE_{15}\downarrow$	$MSE_1\downarrow$	$MSE_3\downarrow$	$MSE_7\downarrow$	$MSE_{15}\downarrow$	$MSE_1\downarrow$	$MSE_3\downarrow$	$MSE_7\downarrow$	$MSE_{15}\downarrow$
Baselines	HistPrice	2.486	2.234	1.880	1.664	3.397	3.316	2.934	2.972	2.709	3.187	3.127	3.291
	P-SVM Chatzis et al. 2018	2.489	2.220	1.915	1.753	2.568	2.921	1.971	2.012	2.104	2.534	1.921	2.231
	P-LSTM Yu and Li 2018	2.421	2.217	1.845	1.731	2.128	2.194	2.108	1.456	1.424	1.867	1.015	1.569
	MLP	2.524	2.214	1.899	1.680	1.469	1.597	0.937	0.981	1.060	1.441	0.802	1.159
	LSTM Poria et al. 2017	2.290	2.210	1.750	1.680	1.346	1.304	0.724	0.779	1.219	1.296	0.762	0.558
	MMIM Han, Chen, and Poria 2021	2.290	*2.092*	1.779	1.598	1.287	1.133	0.718	0.622	**0.975***	1.081	*0.500*	0.510
	MDRM Qin and Yang 2019a	**2.065**	2.511	*1.748*	*1.597*	*1.281*	1.578	*0.683*	0.612	1.183	1.627	0.769	0.512
	HTML Yang et al. 2020a	2.296	2.133	1.771	1.611	1.302	*1.127*	0.766	*0.609*	*0.988*	1.118	0.588	0.498
	MULT Tsai et al. 2019	*2.073*	2.179	1.768	1.605	1.288	1.133	**0.672***	0.742	1.022	*1.018*	0.549	*0.497*
Ablation	MPCNet (T)	2.599	2.390	1.931	2.278	1.906	1.613	1.122	1.262	1.666	1.943	*1.140*	1.801
	MPCNet (A)	2.345	2.457	*1.770*	2.151	1.732	1.614	1.221	*0.724*	1.507	1.963	1.289	1.791
	MPCNet (V)	2.532	2.285	2.108	2.023	1.904	1.617	1.223	1.247	2.273	1.964	1.746	1.511
	MPCNet (T+A)	2.423	*2.221*	2.135	1.956	1.564	1.637	1.456	1.111	1.234	2.144	1.967	1.578
	MPCNet (A+V)	2.280	2.413	2.026	*1.680*	1.857	*1.572*	1.697	0.864	1.621	*1.904*	1.419	1.463
	MPCNet (V+T)	*2.257*	2.321	2.002	2.108	*1.477*	1.596	*1.195*	1.398	1.087	2.017	1.819	*1.407*
	MPCNet (V+A+T) (Ours)	2.233	**2.089***	**1.732***	**1.594***	**1.269***	**1.046***	0.806	**0.607**	1.176	**1.001**	**0.469***	**0.470***

(a) Stock Indices and Currency Exchange Rate

	Model	Gold Price				10-Year Bond Yield				3-Month Bond Yield			
		$MSE_1\downarrow$	$MSE_3\downarrow$	$MSE_7\downarrow$	$MSE_{15}\downarrow$	$MSE_1\downarrow$	$MSE_3\downarrow$	$MSE_7\downarrow$	$MSE_{15}\downarrow$	$MSE_1\downarrow$	$MSE_3\downarrow$	$MSE_7\downarrow$	$MSE_{15}\downarrow$
Baselines	HistPrice	3.193	3.039	2.675	2.683	4.132	4.020	3.472	3.334	3.899	3.665	3.063	2.913
	P-SVM Chatzis et al. 2018	2.568	2.543	1.967	2.104	3.212	3.589	2.986	3.141	3.235	3.143	2.922	2.874
	P-LSTM Yu and Li 2018	1.965	1.998	1.043	1.764	2.212	1.699	2.340	1.453	3.433	2.909	2.678	2.477
	MLP	1.431	1.654	0.904	0.955	1.811	1.743	1.288	1.382	2.582	2.523	2.239	2.231
	LSTM Poria et al. 2017	1.472	1.484	0.703	0.508	1.735	1.801	1.169	1.235	2.421	2.439	2.044	2.013
	MMIM Han, Chen, and Poria 2021	*1.292*	1.292	*0.565*	0.486	*1.698*	*1.604*	*1.080*	*1.053*	2.345	2.392	1.977	*1.902*
	MDRM Qin and Yang 2019a	1.436	1.843	0.710	*0.483*	1.729	1.699	1.126	1.223	2.406	2.622	2.096	1.993
	HTML Yang et al. 2020a	**1.277***	*1.291*	0.589	0.524	**1.685**	1.612	1.103	1.149	*2.342*	*2.356*	*1.962*	1.998
	MULT Tsai et al. 2019	1.314	1.335	0.579	0.503	2.122	1.837	1.104	**1.037***	**1.174***	2.515	1.973	1.903
Ablation	MPCNet (T)	1.967	1.859	1.122	1.750	1.977	1.928	2.067	1.614	2.774	2.723	2.654	2.602
	MPCNet (A)	*1.573*	1.484	1.617	1.803	2.279	1.940	1.965	1.513	2.754	3.242	2.726	2.536
	MPCNet (V)	2.136	2.028	1.586	1.158	2.318	1.969	1.576	1.674	2.857	2.740	2.630	2.616
	MPCNet (T+A)	1.798	1.567	0.985	1.678	2.067	1.956	1.944	1.865	2.759	2.699	*2.345*	2.613
	MPCNet (A+V)	1.752	*1.403*	1.245	*0.959*	1.996	1.903	1.897	1.700	*2.750*	*2.632*	2.538	*2.527*
	MPCNet (V+T)	1.681	1.959	*0.864*	1.428	1.756	1.874	1.511	1.366	3.135	2.678	2.457	2.564
	MPCNet (V+A+T) (Ours)	1.342	**1.275***	**0.562***	**0.477***	1.767	**1.602***	**0.979***	1.142	2.431	**2.319***	**1.948***	**1.901***

(b) Gold Prices, Long-term (10-Years) and Short-term (3-Months) Bonds

(c) Performance comparison with baselines and ablations for volatility prediction in terms of MSE τ-days after the call ($\tau \in \{1, 3, 7, 15\}$). (T: Text, V: Video, A: Audio). **Bold** denotes best performance performance. Light cyan shows second-best performance. Results are averaged over 5 independent runs. * indicates that the result is statistically significant with respect to state-of-the-art based on Wilcoxon's signed rank test with $p < 0.001$. Our proposed approach outperforms price-based and multimodal baselines.

fiers (SVC) on 30-days historical price data for volatility and price movement prediction, respectively.

- **P-LSTM** (Yu and Li 2018): We use LSTM model to extract predictive patterns from 30-days historical price time-series.

8.6.0.2 Multimodal Baselines

: We present contemporary multimodal methods that utilize visual, vocal, and verbal cues.

- **MLP**: A simple multi-layer perceptron where multimodal features are averaged out along the time series and concatenated before the final prediction layer.
- **LSTM (Poria et al. 2017)**: Multimodal time series are input to individual LSTMs and averaged before final prediction.
- **MMIM (Han, Chen, and Poria 2021)**: Uses LSTMs to encode the video and audio sequence, and BERT for text. The encoded features are passed through a fusion layer for maximizing mutual information between unimodal sequences before prediction.
- **MDRM (Qin and Yang 2019a)**: BiLSTM layers encode unimodal sequences, which are then fused together using another layer of BiLSTM to extract multimodal inter-dependencies.
- **HTML (Yang et al. 2020a)**: HTML is a transformer-based architecture that takes fuses multimodal feature representations before passing through Transformer layers for prediction.
- **MulT (Tsai et al. 2019)**: Uses transformer encoders to align language, facial gestures, and acoustic sequences with variable sampling rates and long-range dependencies.

Experiment Settings: MPCNet uses a hidden dimension $H = 512$, dropout $\delta = 0.1$, number of attention heads $n_h = 2$, and number of transformer blocks $n_b = 2$. We use a learning rate (lr) of $1e^{-3}$ for regression, and $1e^{-4}$ for classification. We use PyTorch for all models, and optimize MPCNet using AdamW optimizer for 30 epochs and apply early stopping with a patience of 10 on a Tesla K80 GPU. **Evaluation Metrics**: Similar to prior work (Qin and Yang 2019a; Yang et al. 2020a), we evaluate predicted volatility using the mean squared error (MSE) and the price movement classification task using F1 score, for $\tau \in \{1, 3, 7, 15\}$.

8.7 Results

	Model	Stock Index (Small)				Stock Index (Large)				Currency Exchange Rate			
		$F1_1\uparrow$	$F1_3\uparrow$	$F1_7\uparrow$	$F1_{15}\uparrow$	$F1_1\uparrow$	$F1_3\uparrow$	$F1_7\uparrow$	$F1_{15}\uparrow$	$F1_1\uparrow$	$F1_3\uparrow$	$F1_7\uparrow$	$F1_{15}\uparrow$
Baselines	HistPrice	0.390	0.470	0.400	0.420	0.430	0.430	0.410	0.420	0.190	0.260	0.210	0.230
	P-SVM Chatzis et al. 2018	0.400	0.480	0.340	0.530	*0.433*	0.490	0.338	0.500	0.190	0.270	0.190	0.370
	P-LSTM Yu and Li 2018	0.410	0.473	0.291	0.546	0.399	0.391	0.421	0.442	0.123	0.232	0.165	0.341
	MLP	0.349	0.435	0.209	0.539	0.267	0.319	0.331	0.351	0.101	0.201	0.124	0.311
	LSTM Poria et al. 2017	0.449	*0.435*	0.269	0.527	0.414	0.596	0.371	0.432	0.137	0.229	0.199	0.369
	MMIM Han, Chen, and Poria 2021	0.435	**0.653***	0.302	0.605	0.392	**0.631**	0.329	0.601	0.296	0.217	0.142	0.385
	MDRM Qin and Yang 2019a	0.449	0.419	0.462	0.355	0.409	0.392	0.494	0.324	0.177	0.161	*0.379*	0.152
	HTML Yang et al. 2020a	0.490	*0.645*	0.458	0.541	0.431	0.504	0.557	0.482	0.484	0.531	0.298	**0.626***
	MULT Tsai et al. 2019	*0.491*	0.630	*0.536*	*0.629*	*0.443*	*0.625*	*0.572*	*0.612*	*0.499*	*0.547*	**0.473***	*0.521*
Ablation	MPCNet (T)	0.393	0.423	0.241	0.263	0.361	0.304	0.419	0.396	0.332	0.215	0.252	0.378
	MPCNet (A)	0.288	0.233	0.182	0.365	0.211	0.315	0.397	0.435	0.410	0.283	0.111	0.331
	MPCNet (V)	*0.437*	0.522	0.340	*0.497*	0.335	0.304	*0.464*	0.443	0.438	0.148	0.254	0.412
	MPCNet (T+A)	*0.437*	0.569	0.289	0.489	0.367	0.312	0.422	0.471	0.404	0.245	0.392	0.466
	MPCNet (A+V)	0.415	0.565	0.290	0.465	0.388	0.321	0.455	*0.463*	0.434	0.186	*0.374*	*0.511*
	MPCNet (V+T)	0.406	*0.573*	*0.342*	0.469	0.359	*0.326*	0.458	0.405	*0.450*	*0.295*	0.350	0.336
	MPCNet (V+A+T) (Ours)	**0.501***	0.590	**0.565***	**0.638***	**0.460***	0.590	**0.559***	**0.620***	***0.520****	***0.570****	0.329	0.450

(a) Stock Indices and Currency Exchange Rate

	Model	Gold				10-Year Bond Yield				3-Month Bond Yield			
		$F1_1\uparrow$	$F1_3\uparrow$	$F1_7\uparrow$	$F1_{15}\uparrow$	$F1_1\uparrow$	$F1_3\uparrow$	$F1_7\uparrow$	$F1_{15}\uparrow$	$F1_1\uparrow$	$F1_3\uparrow$	$F1_7\uparrow$	$F1_{15}\uparrow$
Baselines	HistPrice	0.360	0.390	0.350	0.400	0.31	0.290	0.220	0.390	0.220	0.160	0.340	0.330
	P-SVM Chatzis et al. 2018	0.390	0.420	0.370	0.380	0.34	0.310	0.33	0.33	0.370	0.220	0.310	0.390
	P-LSTM Yu and Li 2018	0.365	0.352	0.371	0.346	0.32	0.291	0.342	0.258	0.377	0.234	0.332	0.314
	MLP	0.243	0.215	0.288	0.315	0.244	0.299	0.234	0.174	0.332	0.157	0.248	0.394
	LSTM Poria et al. 2017	0.361	0.337	0.304	0.345	0.364	0.311	0.255	*0.394*	0.381	0.168	0.382	0.444
	MMIM Han, Chen, and Poria 2021	0.209	0.508	0.412	0.318	0.411	0.318	0.345	0.138	0.417	0.306	0.417	0.379
	MDRM Qin and Yang 2019a	*0.434*	0.383	0.214	0.317	0.287	0.242	0.314	0.149	0.346	0.198	**0.478***	*0.505*
	HTML Yang et al. 2020a	*0.441*	*0.654*	0.379	0.526	*0.529*	0.278	0.466	0.389	0.424	*0.314*	0.397	0.450
	MULT Tsai et al. 2019	0.329	0.590	**0.454**	*0.533*	**0.534**	**0.364***	*0.485*	*0.400*	*0.428*	0.171	*0.466*	0.493
Ablation	MPCNet (T)	0.341	0.317	0.423	0.492	0.242	0.343	0.155	0.592	0.117	0.437	0.310	0.293
	MPCNet (A)	0.292	0.121	0.119	0.589	0.088	0.157	0.186	0.489	0.252	0.386	0.317	0.314
	MPCNet (V)	0.239	0.414	*0.519*	0.595	0.373	0.436	*0.542*	0.610	*0.503*	0.520	0.314	*0.375*
	MPCNet (T+A)	0.414	*0.483*	0.503	*0.616*	0.322	0.434	0.529	0.593	0.476	*0.545*	*0.323*	0.312
	MPCNet (A+V)	*0.423*	0.445	0.414	0.607	0.372	0.416	0.449	*0.617*	*0.503*	0.510	0.309	0.369
	MPCNet (V+T)	0.420	0.472	0.517	0.565	0.471	*0.454*	0.500	0.585	0.485	0.542	0.315	0.347
	MPCNet (V+A+T) (Ours)	**0.444***	**0.668***	*0.413*	**0.637***	0.386	*0.327*	**0.560***	**0.625***	**0.493***	**0.556***	0.374	**0.537***

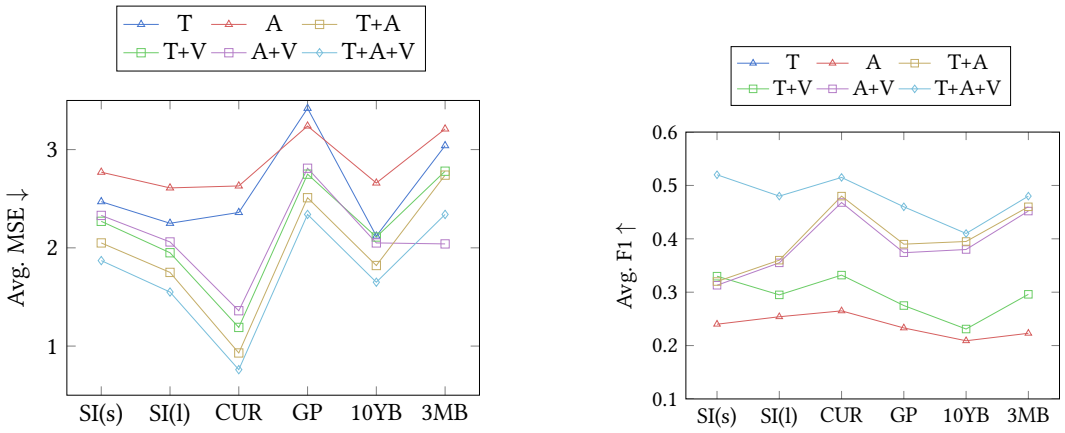
(b) Gold Prices, Long-term (10-Years) and Short-term (3-Months) Bonds

(c) Performance comparison with baselines and ablations for price prediction in terms of F1 score τ-days after the call ($\tau \in \{1, 3, 7, 15\}$). (T: Text, V: Video, A: Audio) **Bold** denotes best performance performance. Light cyan shows second-best performance. Results are averaged over 5 independent runs. * indicates that the result is statistically significant with respect to state-of-the-art based on Wilcoxon's signed rank test with $p < 0.001$. Our proposed approach outperforms price-based and multimodal baselines.

Performance Comparison: Tables 8.3c and 8.4c show the comparative results for the volatility and price prediction tasks, respectively. We observe that baselines that use historical price alone significantly underperform across all settings. Simple models like MLP and LSTM are disadvantaged as they require feature aggregation through averaging over long sequences of time series. Sophisticated LSTM models such as MMIM and MDRM struggle on both tasks due to their inability to capture long-range dependencies in hour-long video calls with multiple dialogues.

Combining multimodal context from the visual, vocal, and verbal cues using a transformer encoder (as done in HTML and MuLT) helps improve performance across different settings. Our proposed model achieves significantly better performance across both tasks for multiple financial targets. MPCNet's ability to model the inter-dependencies between the pairs of modalities using cross-model attention and modality-specific attention fusion contributes towards its outperformance compared to contemporary multimodal methods. Moreover, MPCNet performs attention fusion using weights for pairs of the modalities to determine the mutual importance of each modality which helps it improve over the MuLT baseline. However, it can also be observed that there is ample room for improvement in both volatility and price movement prediction. We attribute this to the inherent difficulty of the task and motivate further research by discussing current shortcomings through error analysis in Sec-8.8.

Ablation: Impact of Multimodality: The ablation results of the proposed MPCNet model in Tables 8.3c and 8.4c strongly suggest the potency of multimodal features over unimodal counterparts, for both tasks, across all financial targets. We observe significant gains due to the addition of aligned video features in the MPCNet model. We attribute this to the presence of additional behavioral cues such as facial expressions and body language, aligned with the call transcripts and audio signals through attention mechanisms in the temporal domain. In order to validate the importance of combining visual, vocal and verbal cues, we conduct additional ablation experiments for MDRM, HTML, and MuLT baselines with varying input modalities. Figure 8.4 shows that blending video features (V) with text(T) and audio(A) leads to improvements over the best bimodal model (T+A), evaluated in terms of time-averaged MSE and F1 scores for MPCNet. We see a similar trend for HTML, MDRM, and MuLT, respectively. Moreover, we see that the addition of video (V) modality to each of *A, T, A+T* settings shows favorable gains. This



(a) Volatility

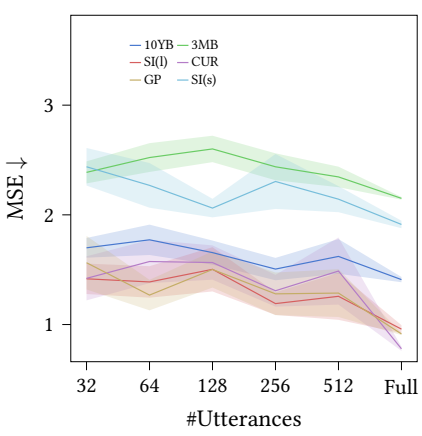
(b) Movement

Figure 8.4: Ablation analysis of modalities in MPCNet for (a) Volatility and (b) Price Movement prediction, averaged over $\tau = \{1, 3, 7, 15\}$. SI(s): Stock Index (Small), SI(l): Stock Index (large), CUR: Currency Exchange Rate, GP: Gold Price, 3MB: 3-Month Bond Yield, 10YB: 10-Year Bond Yield. Addition of video (V) modality to each of *A, T, A+T* settings shows favourable gains (increase in F1 and decrease in MSE).

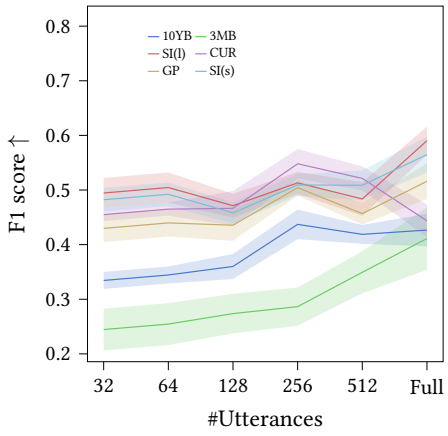
provides strong empirical evidence in support of multimodal fusion of visual, vocal, and verbal modalities for financial prediction tasks on MPC calls.

Impact of Call Length: We probe MPCNet's sensitivity with respect to the input call length by feeding only the first n utterances of the call to the model. As shown in Figure 8.5, we see major performance improvements with increasing call length and achieve best performance on incorporating the full conference call. These observations suggest that the Q&A session is substantially beneficial than just the initial speech by the governor, as the Q&A provides an opportunity to analyze non-verbal cues and answers to questions are not rehearsed beforehand. Our observations reinforce prior studies which have shown the importance of Q&A sessions, which serve as a clarifier of the overall economic outlook (Ehrmann and Fratzscher 2007).

Performance Drift over Time: Results in Table 8.3c and Figure 8.6 show that multimodal models exhibit greater uncertainty in the short term after the MPC call. However, there is a gradual decay in gains of multimodal models for volatility prediction as we move ahead in



(a) Volatility



(b) Movement

Figure 8.5: Performance variation with increasing input call lengths (#utterances) on (a) Volatility and (b) Movement prediction. The results are averaged over $\tau = \{1, 3, 7, 15\}$. Performance improves with increasing call length (reduced MSE and increase in F1), with best results on the full conference calls.

time after the conference call. This trend is not pronounced for price movement prediction which remains consistent throughout as observed from Table 8.4c. Short-term stock volatility prediction is more complex due to the erratic price fluctuations after a MPC call. We attribute the saturation in the volatility prediction performance to the dilution of the market reaction to the MPC calls as we "drift" away from them. These price fluctuations settle as more time elapses, similar to the phenomenon of PEAD (Post Earnings Announcement Drift) (Bernard and Thomas 1989; Sadka 2006).

8.8 Qualitative Analysis

Video 1: Federal Reserve Meeting (2020): Following the MPC call, the SP500 suffered a significant drop within the next 20 days. Studying the call's video frames aligned with text transcripts, we notice in Figure 8.7a that when asked about their plans on interest rate during the Q&A session, the governor's speech had sudden fillers words along with animated hand gestures. Past research (Pérez-Rosas et al. 2015) suggests that increased use of filler words, rapid

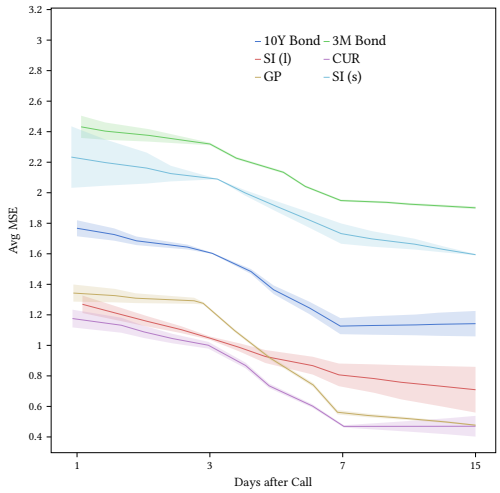
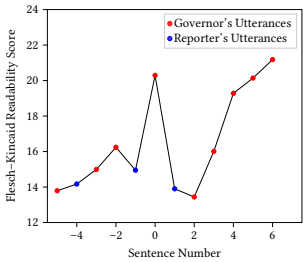


Figure 8.6: Drift in predicted stock volatility over time. The line graph represents the mean MSE of MPCNet SI(s): Stock Index (Small), SI(l): Stock Index (large), CUR: Currency Exchange Rate, GP: Gold Price. As time increases, the MSE decreases due to the PEAD phenomenon (Bernard and Thomas 1989).

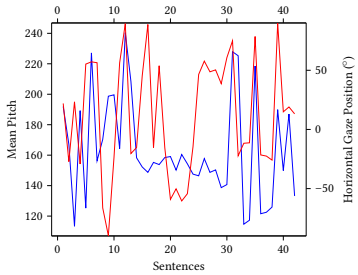
hand movements, and a closed body posture with hands crossed interlocked tightly may indicate a lack of confidence in the speaker. It was later ascertained that the Federal Reserve convened an emergency meeting a week later to announce interest rate cut of 0.50%. We observe how MPCNet successfully predicts the decrease in price of stock index and increase in gold prices for all choices of τ while it's unimodal (A,T,V) and bimodal (text-audio) counterparts fail to do the same each time. Though the text reveals no lack of confidence, the *combination of aligned audio-visual cues* likely allows the model to make a *successful prediction.*

Video 2: European Central Bank (2016): Ten days post monetary policy conference, long-term and short-term bond saw an increase in volatility by 15-25%, respectively. However, the prices of long-term bond yield saw a downward trend contrary to the short-term bond yields. The meeting involved the governor mentioning concerns about trade disruptions and employment reduction due to 'Brexit'. We notice that this call in specific was longer than previous others. Anecdotally, longer conferences are linked with turbulent economic conditions as more time is spent clarifying

(a) Video-1: Chair of the Federal Reserve exhibits closed body language, frequent interlocking of hands, and enhanced use of filler words during Q&A session when asked about rising interest rates. Past research (Abouelenien, Mihalcea, and Burzo 2016; Sen et al. 2020) suggests these non-verbal cues indicate of lack of confidence.



(b) Video-2: Flesch–Kincaid Readability score of the utterances. The governor's utterances become more elongated and difficult to pass when questioned regarding employment reduction. The reporter's sentences are simple and easy to comprehend.



(c) Video-3: Erratic mean pitch of the governor's audio clips (—-) and rapid changes in horizontal gaze position (—-). Randomness in non-verbal cues adds noise, affecting predictions.

Figure 8.7: Qualitative Analysis

journalist questions. We also observe enhanced complexity of text readability due to dense technical discussion in Q&A dialogues (Figure 8.7b). Transformer based models such as HTML, MuLT, and MPCNet were able to capture linguistic complexity and long-range dependencies. Here, we observe that the above three strategies correctly make correct predictions.

Video 3: South African Reserve Bank (2022): We now analyze this MPC call as an *error analysis* where `MPCNet` predicts incorrectly. Here, the price-based LSTM model gains a profit by correctly predicting a 9-12% increase in the currency exchange (ZAR USD rate) for $\tau = 3, 7, 15$. On carefully analyzing the contents of the conference call, we notice (Figure 8.7c) that the governor took a sudden hawkish stance on inflation due to the oil crisis propelled by the Ukraine war and economic sanctions. Moreover, observing the visual and vocal cues, we find a great deal of variance in the mean audio pitch and speaker's erratic eye gaze. We attribute the erroneous performance to the potential overfitting of the model as well as unique information about world knowledge present at test time not seen before in the training set. We believe that future research

in combining knowledge from *alternate sources such as news and social media can benefit prediction performance.*

8.9 Ethical Considerations and Limitations

Examining a speaker's tone and speech in conference calls is a well-studied task in past literature (Qin and Yang 2019a; Yang et al. 2020a). Our work focuses on video conference calls for which government institutions and financial regulatory bodies publicly release call videos, transcripts and audio recordings. The conference call and price data used in our study is open source. We do not collect any personalized data or violate any privacy laws in using, storing or releasing the MPC conference calls data for financial analysis.

Limitations: We acknowledge the presence of gender bias in our study, given the imbalance in the gender ratio of speakers of the calls. We also acknowledge the demographic bias in our study as the central banks studied in our work are restricted to certain geographies and may not directly generalize for other countries. We also limit our study to English-only calls, motivating further studies on other multilingual conference calls.

Potential risks: Our contributions are meant as an exploratory research in the financial domain and no part of the work should be treated as financial advice. All financial investments decisions are subject to market risk and should be made after extensive testing. Practitioners should check for various biases (demographic, gender, modeling, randomness) before attempting to use the provided code/data/methods for real-world purposes.

8.10 Conclusion and Future Work

We present a dataset of Monetary Policy Conference video calls to predict financial risk and price movement. We also present MPCNet, a strong benchmark model that uses cross-modal transformer blocks and modality-specific attention fusion on input time series for financial forecasting on MPC calls. We further analyze the benefits of each modality, evaluate the effect of multi-task setting for joint prediction of financial assets, examine biases due to dataset distribution, and effect of non-verbal behavioral cues extracted from spontaneous Q&A session. We motivate future work to explore several interesting direction including but not limited to conversational dialogue modeling of Q&A sessions, fine-grained multimodal emotion recognition, gaits and posture analysis to identify non-verbal behavioral cues, augmenting video data with external knowledge graphs, etc.

CHAPTER 9

DocFin: Multimodal Financial Prediction and Bias Mitigation using Semi-structured Documents

Abstract

Financial prediction is complex due to the stochastic nature of the stock market. Semi-structured financial documents present comprehensive financial data in tabular formats, such as earnings, profit-loss statements, and balance sheets, and can often contain rich technical analysis along with a textual discussion of corporate history, and management analysis, compliance, and risks. Existing research focuses on the textual and audio modalities of financial disclosures from company conference calls to forecast stock volatility and price movement, but ignores the rich tabular data available in financial reports. Moreover, the economic realm is still plagued

with a severe under-representation of various communities spanning diverse demographics, gender, and native speakers. In this work, we show that combining tabular data from financial semi-structured documents with text transcripts and audio recordings not only improves stock volatility and price movement prediction by 5-12% but also reduces gender bias caused due to audio-based neural networks by over 30%.

9.1 Introduction

Financial risk modeling is of great interest to capital market participants for making sound investment decisions. Earnings calls are quarterly audio conference calls wherein company executives discuss their companies' performance and future prospects with outside analysts and investors (Qin and Yang 2019b). Mergers and Acquisitions (M&As) conference calls are held preceding financial transactions involving two or more entities such that either one of the participant companies takes over the other(s) ("acquisition") or combines with another to become a joint entity ("merger") (Sawhney et al. 2021a). Both kinds of events consist of a prepared speech delivery by company executives on analysis and future expectations followed by a spontaneous analyst-driven question-answer session to seek additional information (Ye, Qin, and Xu 2020). Several past works have utilized the text transcripts and audio recordings from these calls to improve the stock forecasting (**mathur2022monopoly**; **sawhney2020risk**; Yang et al. 2020a; Zhou, Zhang, and Yang 2020; Chen, Huang, and Chen 2020; Ye, Qin, and Xu 2020; Sawhney et al. 2021a; Sawhney, Aggarwal, and Shah 2021). However, most prior works exclusively focus on vocal verbal information, often ignoring information from official financial documents. *Financial semi-structured documents* such as 10-K and 10-Q reports are publicly

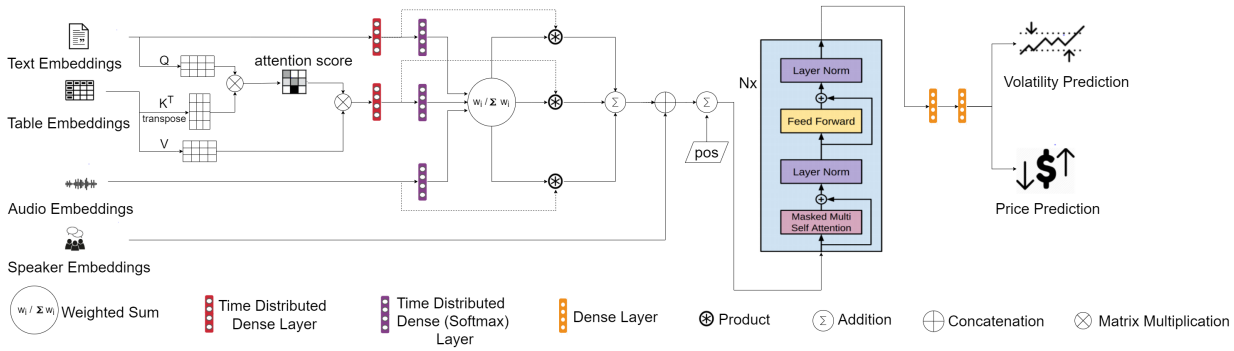


Figure 9.1: We combine the text transcripts and audio call recording input to a neural model with tabular text from semi-structured financial documents. Here we illustrate how M3A (Sawhney et al. 2021a) for volatility and stock price prediction on M&A calls uses dot product attention to extract condensed table representations relevant to text from transcripts, weight averages each modality through softmax layer to obtain a fused embedding, combines the fused embeddings with speaker and positional embedding, and finally passes through the Transformer model for stock price movement and volatility prediction tasks.

available, recurrent mandatory filings made by public companies to disclose their financial performance. These semi-structured financial documents present comprehensive financial data in tabular format, such as earnings, profit and loss statements and balance sheets, and can often contain more than 100's of tables worth technical analysis. Information contained in such financial documents also includes a textual discussion of corporate history, management analysis, compliance, risks, and future plans about new projects relevant for investment decision-making (Kogan et al. 2009b).

Recent studies such as (Sawhney, Aggarwal, and Shah 2021) have highlighted the downside of utilizing audio-based multimodal approaches for financial risk prediction due to the inherent gender bias induced in learning models due to the imbalance of speaker demographics in call recordings . Audio features such as speakers' pitch and intensity can vary greatly across genders. Under-representation of female executives in conference calls is amplified by deep learning models, leading to high error disparity between stock predictions across sensitive attributes.

We combine tabular from financial semi-structured documents input with existing vocal-

verbal information from audio call recordings to improve stock price movement and volatility prediction. We demonstrate that supplementing existing conference calls transcripts with the tabular financial data substantially reduces the unintended gender bias in financial prediction tasks and offers a robust and unbiased alternative to gender-sensitive audio features in cases where under-representation of women speakers (only 7% female speakers in SP 500 Earnings calls dataset (Li et al. 2020b) and 12% in Merger&Acquisition calls (Sawhney et al. 2021a) dataset) in executive positions induces unneeded correlations in model predictions. The novel **contributions** of our work are:

- We combine publicly available earnings calls (MAEC (Li et al. 2020b)) and M&A calls (Sawhney et al. 2021a) datasets with tabular data extracted from SEC-EDGAR 10-Q and 10-K company-filing documents.
- We utilize tabular information from financial semi-structured documents with existing textual and audio modalities to show 8-12% relative improvement in stock volatility and price movement prediction tasks across several baseline and state-of-the-art models.
- We empirically show the extent of induced gender bias due to audio modality in the financial prediction models and demonstrate the usefulness of tabular data extracted from semi-structured financial documents as an alternative to audio modality for reducing gender bias by 30% in audio-based neural networks, without significant performance degradation.

9.2 Methodology

Problem Formulation: We consider an input conference call recording $\chi = [t; a; tab]$, such that each call comprises of multimodal components: N textual utterances $t = [t_1, t_2...t_N]$ aligned with their corresponding audio slices $a = [a_1, a_2...a_N]$, and $tab = [tab_1, tab_2...tab_M]$ corresponding to the M tables extracted from the company filings relevant to the call. Each conference call is associated with speaker information denoted by $s = [s_1, s_2...s_N]$, representing the sequence of speakers for the utterances. We formulate volatility as a regression task (Kogan et al. 2009b) and price movement prediction as a binary classification task (Xu and Cohen 2018b).

Measuring stock volatility : Following (Kogan et al. 2009b), we formulate volatility as a regression task. For a given stock with a close price of p_k on the trading day k, we calculate the average log volatility as the natural log of the standard deviation of return prices r in a window of τ days as.

$$v_{[0,\tau]} = \ln\left(\sqrt{\frac{\sum_{k=1}^{\tau}(r_k - \bar{r})^2}{\tau}}\right) \tag{9.1}$$

where $r_k = \frac{p_k - p_{k-1}}{p_{k-1}}$ is the return price on day k for a given stock, and $\bar{r}$ is the average return price over a period of τ days.

Price movement prediction : Following (Xu and Cohen 2018b), we define price movement $y_{d-\tau,d}$ over a period of τ days as a binary classification task. For a given stock, we employ its close price, which can either rise or fall on a day d compared to a previous day $d - \tau$, to formulate the classification task as:

Year	# of MA Calls	Mean # of Utterances	Mean # of Speakers	Mean # of Tables	
				10-K	10-Q
2016	192	117.421	11.265	217.234	107.093
2017	206	96.825	11.14	216.83	101.961
2018	232	90.517	10.607	231.073	107.525
2019	133	97.413	10.39	228.624	124.248
2020	49	104.897	10.326	216.571	105.53

Table 9.1: Dataset statistics for the M&A dataset

$$y_{[d-\tau,d]} = \begin{cases} 1, p_{d+\tau} > p_d, \\ 0, p_{d+\tau} \leq p_d \end{cases} \tag{9.2}$$

Given a conference call χ, we experiment with several baseline and state-of-the-art multimodal financial prediction models (example M3A (Sawhney et al. 2021a) in Fig. 9.1). We predict the average negative log volatility $v_{[0,\tau]}$ and price movement direction $y_{[0,\tau]}$ using the multimodal call data $\chi = [t; a; tab]$ for $\tau = 3$, 7 and 15-day interval.

Encoding Text Transcript, Audio Call and Speakers: We process text and audio data following earlier works on Earnings Calls (Li et al. 2020b) and M&A calls (Sawhney et al. 2021a). Each text utterance t_i is represented as a 768-dimensional encoding g_i using BERT. Each audio utterance a_i is encoded into its embedding h_i corresponding to the type of conference call. For M&A calls, we extract h_i as a 62-dimensional encoding described in (Eyben et al. 2016) using OpenSMILE[1] and for Earnings Calls as a 29-dimensional low-level audio features encoding using Praat (Boersma and Van Heuven 2001). We extract the list of speakers from the transcripts and assign each speaker s_i a sequential ID in the order of listing and represent the speaker embedding as one-hot encoding.

Encoding Tables from Company Filings: Taking inspiration from past literature (Chen et al. 2020b), we linearize each table tab_i into a sentence representation. For a row i with column

[1] https://pypi.org/project/opensmile/

Year	# of Calls	Mean # of Utterances	Mean # of Speakers	Mean # of Tables	
				10-K	10-Q
2015	632	87.357	1.764	194.781	92.381
2016	1127	87.299	1.747	211.944	98.733
2017	469	109.396	1.886	211.217	92.974
2018	160	154.143	2.018	205.362	94.512

Table 9.2: Dataset statistics for the Earnings Call dataset

Model	Volatility Prediction			Price Prediction		
	$MSE_3 \downarrow$	$MSE_7 \downarrow$	MSE_{15}	$F1_3 \uparrow$	$F1_7 \uparrow$	$F1_{15} \uparrow$
RoBERTa + LSTM (Liu et al. 2019a)	0.78 (0.009)	0.58 (0.009)	0.47 (0.006)	0.57	0.58	0.49
GloVe + LSTM (Pennington, Socher, and Manning 2014)	0.80 (0.005)	0.60 (0.004)	0.48 (0.005)	0.55	0.56	0.42
FinBERT + LSTM + (Aracı 2019)	0.78 (0.008)	0.60 (0.004)	0.47 (0.005)	0.58	0.58	0.48
MDRM (Qin and Yang 2019b)	0.78 (0.005)	0.58 (0.003)	0.46 (0.002)	0.59	0.58	0.46
MDRM + `DocEmbedding`	0.76 (0.006)	0.55 (0.001)	0.43 (0.004)	0.62	0.61	0.49
M3ANet (Sawhney et al. 2021a)	0.79 (0.020)	0.61 (0.012)	0.48 (0.001)	0.61	0.62	0.54
`M3ANet + DocEmbedding`	**0.73*** (0.008)	**0.54*** (0.012)	**0.42*** (0.012)	**0.66***	**0.63***	**0.56***

Table 9.3: Mean τ-day volatility (MSE) and price movement prediction (F1 score) results for **Merger & Acquisition calls** (M&A dataset) across several models. * indicates result is significantly better than the M3ANet under Wilcoxon's Signed Rank test. Adding `DocEmbedding` outperforms base methods across all tasks and intervals.

names c_j and values v_{ij}, the row is represented as 'row i's c_1 is v_{i1}; the c_2 is v_{i2}...'. Each row's representation is concatenated using punctuation to obtain a table representation which is encoded to its 768-dimensional table encoding k_i using BERT.

Combining tabular data with text - audio time series: We provide a generalized method to process, fuse and utilize the tabular data with text-audio modality such that it is extensible across different neural architectures. To this end, we use dot-product attention to allow each text utterance g_i to extract a condensed table representation l_i from the table encoding k_i, such that $l_i = DotProdAttn(g_i, k_i)$. To fuse the encoding, we linearly transform the text and table encoding to the size of the audio encoding and employ the use of multi-headed self-attention (Vaswani et al. 2017). The text, audio and table features are multiplied by their softmax-ed weights ($W' = \sigma(gW_{wt} + b_{wt}) \forall$T,A,TA), summed ($S = W'_T + W'_A + W'_{TA}$), and weighted averaged to get attention weights $W_T, W_A, W_{TA} = \frac{W_T}{S}, \frac{W_A}{S}, \frac{W_{TA}}{S}$, which are added to get the fused embeddings $X_{fused} = gW_T + hW_A + lW_{TA}$. We augment X_{fused} with the speaker information s by concatenation

Model	Volatility Prediction		
	$MSE_3 \downarrow$	$MSE_7 \downarrow$	MSE_{15}
Vpast	2.99	0.83	0.42
LSTM	1.97	0.46	0.32
HAN (GloVe)	1.43	0.46	0.31
MDRM (Qin and Yang 2019b)	1.37	0.42	0.30
MMTFR (Sawhney et al. 2021a)	0.60	0.30	0.18
MMTFR + DocEmbedding	0.58	0.28	0.15
VoLTAGE (Sawhney et al. 2020a)	0.63	0.29	0.17
VoLTAGE + DocEmbedding	0.61	0.28	0.16
M3A (Sawhney et al. 2021a)	0.59	0.29	0.18
M3A + DocEmbedding	**0.57***	**0.27***	**0.15***

Table 9.4: Mean τ-day MSE for stock volatility prediction for **Earnings Calls** (MAEC dataset) across several methods. * indicates result is significantly better than the VoLTAGE under Wilcoxon's Signed Rank test. Our approach of augmenting with DocEmbeddings outperform corresponding base methods across 3,7,15-day intervals

Modality			Merger & Acquisition Calls						Earnings Calls		
			Volatility Prediction			Price Prediction			Volatility Prediction		
Text	Audio	Table	$MSE_3 \downarrow$	$MSE_7 \downarrow$	MSE_{15}	$F1_3 \uparrow$	$F1_7 \uparrow$	$F1_{15} \uparrow$	$MSE_3 \downarrow$	$MSE_7 \downarrow$	MSE_{15}
✓	✗	✗	0.79 (0.003)	0.65 (0.005)	0.49 (0.008)	0.53	0.50	0.46	1.08	0.40	0.20
✗	✓	✗	0.80 (0.003)	0.64 (0.008)	0.56 (0.008)	0.53	0.53	0.44	1.41	0.45	0.38
✗	✗	✓	0.85 (0.002)	0.72 (0.007)	0.63 (0.009)	0.42	0.41	0.40	1.63	0.62	0.56
✓	✓	✗	0.78 (0.004)	0.61 (0.007)	0.46 (0.004)	0.59	0.56	0.49	0.75	0.32	0.21
✓	✗	✓	0.77 (0.010)	0.57 (0.009)	0.47 (0.007)	0.60	0.58	0.48	0.74	0.30	0.20
✗	✓	✓	0.74 (0.010)	0.55 (0.017)	0.42 (0.013)	0.64	0.61	0.51	0.63	0.27	0.19
✓	✓	✓	**0.69 (0.008)**	**0.54 (0.012)**	**0.42 (0.012)**	**0.66**	**0.63**	**0.54**	**0.57**	**0.27**	**0.15**

Table 9.5: Ablation analysis of M3A model augmented with DocEmbedding for each modality for volatility (MSE) and price movement prediction (F1 score) tasks across Earnings Calls and M&A calls datasets (mean and st. dev. of 5 runs for each approach). Combining audio, text and tabular data gives best performance (see **bold**). Green shade highlights that substituting company filings instead of its audio counterpart in conjunction with text transcripts does not significantly deteriorate model performance.

(represented by $\oplus$) and the position embeddings *pos* by addition as $X_{final} = (X_{fused} + pos) \oplus s$. The augmented document features, called DocEmbedding, can be used by an encoder (recurrent, attention-based or Transformer) for processing to produce the task predictions.

9.3 Experiments

Datasets: We train and test several baseline and state-of-the-art models that utilize the multi-modal input on two datasets: Multimodal Aligned Earnings Call (MAEC) Dataset (Li et al. 2020b)

and Multimodal Multi-Speaker Merger&Acquisition Call Financial Forecasting (M3A) Dataset (Sawhney et al. 2021a), both containing aligned text transcripts and audio recordings of their respective types of conference calls. We collect the most recently filed 10-K and 10-Q documents before the date of the call[2] and parse the HTML content to retrieve all tables with at least 10 rows. We describe the dataset statistics in Table 9.1 and Table 9.2. We tune all hyper-parameters using Grid Search and implement all methods with Keras[3]. We use training/validation/testing splits released by respective datasets.

9.4 Results and Discussion

Effect of Tabular Datat on Financial Predictions: Table 9.3 shows the performance of several baseline and SOTA models for predicting price movement and stock volatility for Merger & Acquisition calls on the M&A dataset. Table 9.4 reports the volatility prediction performance on the MAEC dataset. We report average MSE and F1 scores for volatility and price movement prediction, respectively. We observe significant gains (8-12%) in both tasks across attention based (MDRM, VoLTAGE, MMFTR) and Transformer models (M3A) by combining tabular information extracted from financial semi-structured documents with text-audio time series. Past works have mostly been restricted to verbal-vocal cues obtained from the conference call recordings, lacking the context required to verify speaker claims against technical facts as indicated by reports. Our method helps the underlying neural architectures utilize contextualize information related to compliance, risks, and future plans from audio-textual utterances with technical indicators presented in financial reports. In line with previous works (Sawhney et al. 2020a), it can be seen

[2]Using https://github.com/jadchaar/sec-edgar-downloader
[3]`https://keras.io/`

Modality	Earnings Calls			Merger & Acquisition Calls		
	$\Delta G = MSE_F - MSE_M$			$\Delta G = F1_M - F1_F$		
	$\tau = 3$	$\tau = 7$	$\tau = 15$	$\tau = 3$	$\tau = 7$	$\tau = 15$
Text (T)	0.27	0.10	0.14	0.22	0.14	0.11
Audio (A)	0.33	0.15	0.19	0.36	0.27	0.23
Table (Tab)	0.19	0.07	0.09	0.16	0.09	0.06
A + T	0.30	0.12	0.17	0.27	0.17	0.15
A + Tab	0.27	0.13	0.16	0.25	0.12	0.08
A + T + Tab	0.25	0.10	0.14	0.22	0.08	0.11
T + Tab	**0.21**	**0.08**	**0.10**	**0.18**	**0.10**	**0.07**

Table 9.6: Modality specific ΔG i.e. the difference between the MSE and F1 for volatility prediction in Earnings Calls dataset and price prediction in M&A calls, respectively for 3, 7, and 15 days over 5 runs. We use SOTA M3A model for experiments. Here A stands for Audio only, T for Text only and Tab for Tabular modality. We show that tabular information can substitute audio input to reduce gender bias in multimodal financial prediction tasks.

that the performance gain is not symmetric across time intervals and tends to decrease with increasing time delay after the release of company filings and the press release of conference calls.

Ablation Study: Table 9.5 shows ablation across different modalities observed for the SOTA M3A model applied to both datasets to understand the impact of varying modalities and their correlations. Unimodal settings severely underperform across both tasks. The addition of tabular information extracted from company filing data to verbal-vocal cues shows a gain of 10-12% across different settings. Interestingly, utilizing text transcripts with table data from financial documents instead of its audio counterpart does not deteriorate the model performance (Table 9.5, highlighted in green). This has important implications for proposing company filing as an alternative to the audio input as vocal cues are noisy and processing-heavy.

9.4.1 Bias Reduction through Company Filings

We evaluate the gender bias in SOTA M3A model by quantifying the error disparity in MSE/F1 score between male and non-male speakers ($\Delta G = MSE_F - MSE_M / F1_M - F1_F$) for individual text, audio and table inputs and their combinations across 3, 7 and 15-day intervals in Table 9.6. We

observe that the table modality has the least error disparity. Audio modality has consistently higher error individually as well as in combination with either of the other modalities, while it significantly drops when considering just text and table data. The primary reason for the observation tends to be the imbalance in the male and female distribution in speakers of earnings calls. In our case, since female examples are very less in comparison to the male counterparts (only 7% in earnings calls and 12% in M&A calls identify as females), the model discriminates between male and female examples by inferring insufficient information beyond its source and learns imperfect generalizations between the attributes and labels.

9.4.2 Audio vs Tabular Information

While audio input modality certainly improves model performance, it adds unintended model bias due to the differences in acoustic features for males and females. Audio clips require processing-heavy algorithms such as forced alignment (Sakoe and Chiba 1978b) to extract meaningful features from linguistic and acoustic utterances as opposed to semi-structured information in tables that can be utilized with minimal processing. Replacing audio clips with tabular data from company filings leads to a reduction of data processing time and data storage requirements by over 90% and 50%, respectively for both MAEC and M&A datasets. As evident from Table 9.5 and 9.6, tabular information preserves model performance while avoiding unwanted stereotypes arising due to gender-specific audio features such as shimmer and jitter. Hence, we propose to utilize tabular information as an effective substitute for audio input for multimodal financial prediction tasks.

9.5 Conclusion and Future Work

In this work, we show that combining tabular data from financial semi-structured documents with text transcripts and audio recordings improves stock volatility and price movement prediction by 5-12% along with reduction in gender bias learned by audio-based neural networks by over 30%. We empirically show that our approach is generic and extensible to recurrent, attention-based and Transformer models. Future work can utilize advances in document-NLP to extract temporal information extraction (Mathur et al. 2021b), temporal dependency parsing (Mathur et al. 2022d), and NLI (Mathur et al. 2022c) for better contextual understanding of financial reports. Predicting the correct layout can also helps align audio with transcripts (Mathur et al. 2022a).

9.6 Limitations

We acknowledge the presence of gender bias in our study, given the imbalance in the gender ratio of speakers of the calls. We also acknowledge the demographic bias sawhney-etal-2021-empirical in our study as the companies are organizations within the public stock market of the United States of America and may not generalize directly to non-native speakers. At the same time, we extensively study the components causing gender bias and propose ways to fix it in the current contributions.

9.7 Potential risks

Our contributions are meant as exploratory research in the financial domain and no part of the work should be treated as financial advice. All financial investment decisions should be made after extensive testing. Practitioners should check for various biases (demographic, gender,

modeling, randomness) before attempting real-world use cases.

9.8 Ethical Considerations

Examining a speaker's tone and speech in conference calls is a well-studied task in past literature (Qin and Yang 2019b; Chariri 2009). Our work focuses on conference calls for which companies publicly release transcripts and audio recordings. The data used in our study corresponding to M&A and Earnings conference calls is open-sourced and available for download. The company document filings we use to extract tabular data are publicly available, open source and devoid of human intervention at its source. We do not collect any personalized data or violate any privacy laws in using, storing or releasing the company filing data for financial analysis.

CHAPTER 10

PersonaLM: Language Model Personalization via Domain-distributed Span Aggregated K-Nearest N-gram Retrieval Augmentation

Abstract

We introduce Domain-distributed span-Aggregated K-nearest N-gram (PersonaLM) retrieval augmentation to improve language modeling for Automatic Speech Recognition (ASR) personalization. PersonaLM leverages contextually similar n-gram word frequencies for recognizing rare word patterns associated with unseen domains. It aggregates the next-word probability distribution based on the relative importance of different domains to the input query. To achieve this, we propose Span Aggregated Group-Contrastive Neural (SCAN) retriever that learns to

rank external domains/users by utilizing a group-wise contrastive span loss that pulls together span representations belonging to the same group while pushing away spans from unrelated groups in the semantic space. We propose ASAP benchmark for ASR LM personalization that consists of three user-specific speech-to-text tasks for meetings, TED talks, and financial earnings calls. Extensive experiments show that PersonaLM significantly outperforms strong baselines with a 10-16% improvement in perplexity and a 5-8% reduction in Word Error Rates on popular Wikitext-103, UserLibri, and our ASAP dataset. We further demonstrate the usefulness of the SCAN retriever for improving user-personalized text generation and classification by retrieving relevant context for zero-shot prompting and few-shot fine-tuning of LLMs by 7-12% on the LAMP benchmark.

10.1 Introduction

Language modeling is a core task in NLP with important applications in automatic speech recognition (ASR) (Mikolov et al. 2010; Chen et al. 2015; Xu et al. 2018a). Pre-trained LMs (Irie et al. 2019a; Li et al. 2020e) memorize a surprising amount of knowledge from their training corpora in the underlying neural network parameters (Petroni et al. 2019; Jang et al. 2022). However, this makes it difficult to personalize them for text generation, non-streaming ASR re-scoring, and on-device streaming ASR models for unseen users and domains due to the existence of user-preferred rare word patterns, facts, proper names, and other domain-specific tail words not seen frequently in the training data (Schick and Schütze 2019; Maynez et al. 2020; Serai, Sunder, and Fosler-Lussier 2022). Retrieval augmentation (Lewis et al. 2020c) (see Fig. 10.1) can help personalize LMs by explicitly exposing them to external world knowledge during inference

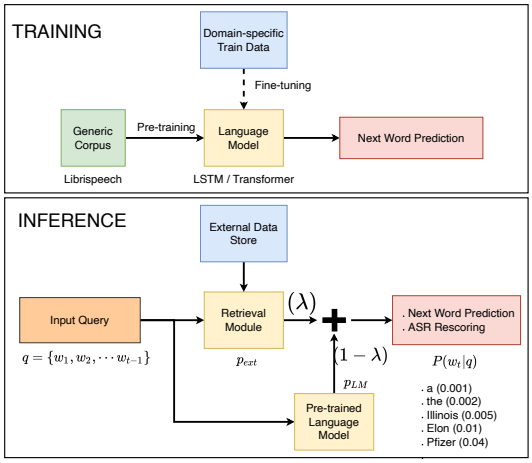


Figure 10.1: ASR LM Personalization: During training, the LM is pre-trained on a generic corpus and optionally fine-tuned on the out-of-domain corpus (see dotted). For query q at inference, LM output p_{LM} is interpolated with the probability distribution p_{ext} retrieved from domain-specific external corpus for next word prediction $p(w_t|q)$ and ASR re-scoring.

(Borgeaud et al. 2022). LMs leverage the retrieval mechanism to select contextually relevant users/domains from an external corpus and then attend over that knowledge to inform their predictions (Liu, Yogatama, and Blunsom 2022).

Prior research has explored kNN-LM memorization (Khandelwal et al. 2020), RETRO (Borgeaud et al. 2022), and attention-based caches (Grave, Joulin, and Usunier 2017). However, these methods give subpar performance as they do not retrieve relevant domains/users prior to context selection from billions of candidates. Recent approaches, notably REALM (Guu et al. 2020) and RAG (Lewis et al. 2020c), incorporate a non-parametric retrieval step during LM pre-training, thus being unable to adapt their context representation for unseen domains.

We address the challenge of capturing rare word patterns associated with specific users / domains by exploiting n-gram word frequencies from underlying domains. Further, we hypothesize that n-gram patterns are domain/user-specific, and augmenting LM predictions with n-gram probabilities from a subset of query-relevant users/domains may lead to better personalization. We anticipate the retrieval augmentation through n-gram frequencies to have additional

advantages of very low computational overhead, efficient caching, and asynchronous updates for newer data without the need for re-computation from scratch.

We propose PersonaLM - Domain-distributed Span-aggregated k-Nearest N-gram Language Model, that aggregates top-k nearest n-gram co-occurrence frequencies from each domain weighted according to the domain's relative importance to the input query, which is augmented with the target word probability distribution for next word prediction and ASR second-pass re-scoring. We utilize a novel Span Aggregated Group-Contrastive Neural (SCAN) retriever that can learn highly discriminative semantic representations to distinguish between text spans from the same group as opposed to random spans using a group-wise contrastive loss. SCAN retriever assigns a relevance score to each textual document/recording from an external corpus based on its semantic similarity with the input query to weigh their contribution to the final prediction.

Our main contributions are:

- **PersonaLM** retrieval augmentation for ASR personalization that leverages group-wise contrastive loss to train **Span Aggregated Group-Contrastive Neural (SCAN) retriever** for ranking query-relevant external domains/users and augments domain-distributed k-nearest n-gram frequencies to improve LM predictions.
- **ASAP - a novel benchmark for ASR LM personalization** consisting of three user-specific ASR tasks in the domains of meetings, TED talks, and financial conference calls. PersonaLM significantly outperforms strong baselines on ASAP benchmark, UserLibri, and Wikitext-103 corpus by $\sim 10-16\%$ perplexity gain and $\sim 5-8\%$ WER reduction.
- **Downstream Application**: SCAN retriever improves context retrieval in personalized text generation and classification via zero-shot prompting and few-shot fine-tuning of LLMs on LaMP corpus by $7-12\%$.

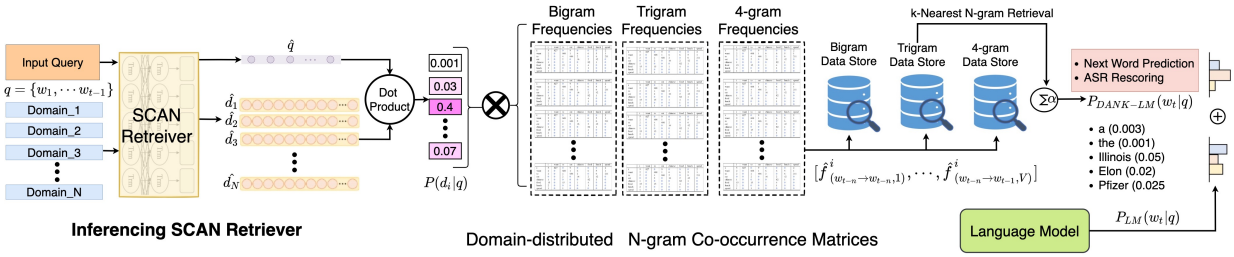


Figure 10.2: PersonaLM: At inference, we compute the relevance score $P(d_i|q)$ between the query and domains d_i as the dot product of their SCAN retriever representations. We construct a data store for each n-gram frequency matrix. k-most similar n-gram contexts wrt to the input query are retrieved and their weighted summation based on the domain's relevance score is computed to get probability distribution over targets $P_{PersonaLM}(w_t|q)$ and interpolated with LM probabilities $P_{LM}(w_t|q)$.

10.2 Related Work

Language Modeling for Rare Words Prediction: Earliest works explored the use of LSTM with auxiliary pointer networks to predict rare words and long-term dependencies in language modeling (Merity et al. 2017). Neural cache augmentation (Grave, Joulin, and Usunier 2017) stored past hidden activations in cache memory to predict out-of-vocabulary words. Implicit cache memorization (Li, Povey, and Khudanpur 2020) used cache to store past word occurrences as an alternative to the attention-based pointer mechanism. For the ASR re-scoring task, cross-sentence neural LMs proposed to use word usage in preceding sentences to re-rank n-best hypotheses of upcoming sentences (Sun, Zhang, and Woodland 2021; Irie et al. 2019b).

Retrieval Augmentation for Language Modeling: kNN-LM (Khandelwal et al. 2020) interpolated pre-trained LMs with contexts extracted from an external data store using the kNN algorithm. REALM (Guu et al. 2020) proposed a neural retriever to leverage external knowledge during LM pre-training. (Ram et al. 2023) augmented GPT-2 with a large episodic memory for a zero-shot reduction in perplexity. Retrieval-Enhanced Transformer (Retro) (Borgeaud et al.

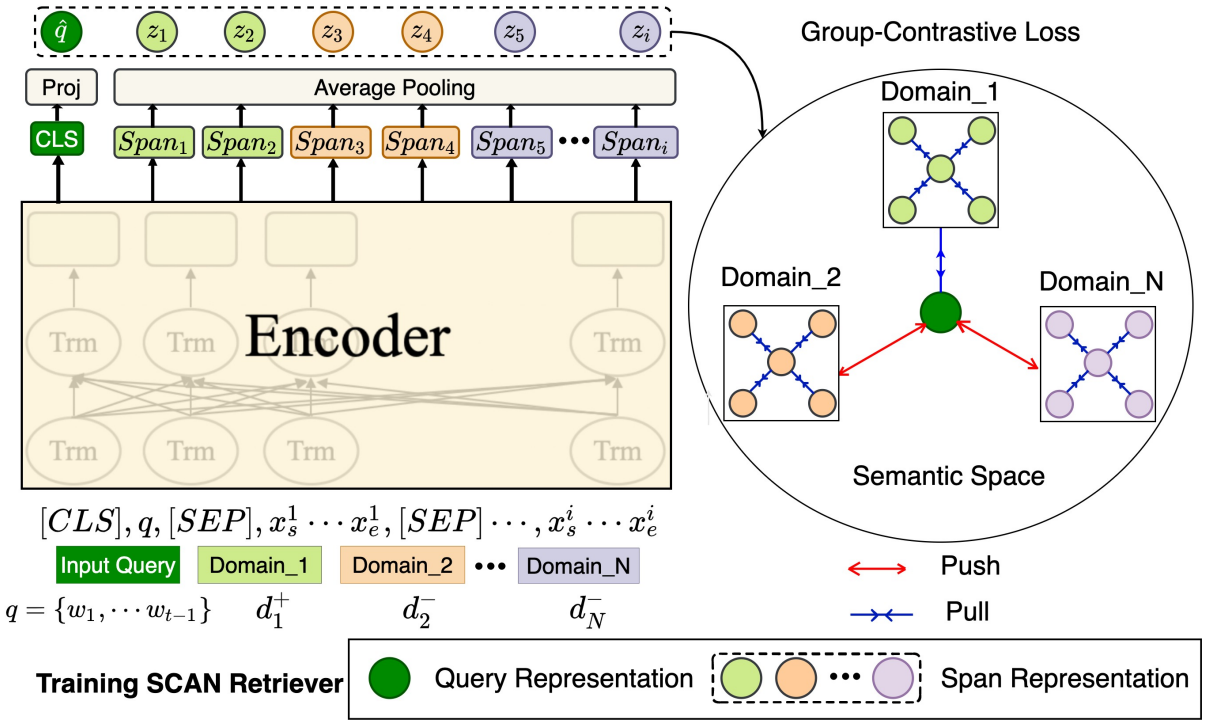


Figure 10.3: SCAN Retriever: Input query q followed by text spans $(x_s, \cdots, x_e)$ from the positive domain (d^+) and $N-1$ negative domains (d^-) separated by [SEP] are passed through the encoder followed by a projector layer and an average pooling layer. SCAN retriever is trained via a group-wise contrastive loss to force the hidden representation of the query $\hat{q}$ close to its own spans z_i, while far away from other groups.

2022) retrieved document chunks similar to preceding tokens using a cross-attention module. Our work is the first to use domain-distributed n-gram representations over document spans to retrieve rare word patterns from the most relevant external knowledge domains.

10.3 PersonaLM Retriever Augmentation

Fig. 10.2 describes our proposed PersonaLM retrieval augmentation approach that biases the predictions from a base LM with the next word probabilities based on the relevance of unseen topic/users to the input query. Given an input query $q = (w_1, \cdots, w_{t-1})$ at inference, autoregressive LMs estimate the probabilty distribution for target token w_t as $P_{LM}(w_t|q)$. To augment the LM output with domain-specific word occurrence information, we calculate the probability distribution of next word prediction over the vocabulary conditioned on the relevance of the underlying domains $(d_1, d_2, \cdots, d_K)$ to the query (q) as:

$$P_{PersonaLM}(w_t|q) = \sum_{i=1}^{K} P(d_i|q) \times P(w_t|d_i, q)$$

10.3.1 SCAN Retriever

Fig. 10.3 shows the architecture of our proposed Span Aggregated Group-Contrastive Neural (SCAN) retriever which is a Transformer encoder pre-trained with Masked Language Modeling (MLM) as well as a novel group-wise contrastive span loss to force the semantic representations of the input query close to its ground truth domain and away from random spans from the different domains. During the training of the SCAN retriever, we first sample spans of varying granularities from multiple domains and encode them using the Transformer encoder. Group-wise contrastive loss is then applied to learn discriminative semantic presentations for enhanced

retrieval performance.

Document Span Sampling: Different granularities of spans capture different properties of the input text. For example, phrase-level spans can capture specific words or entities mentioned in the text while paragraph-level spans can capture more abstract properties of the text such as topic information. In this work, we explicitly sample a set of text spans at the phrase level, sentence level, and paragraph level. We extract T spans for each level of granularity to obtain a total of $3T$ spans corresponding to each input document D. Text spans $(x_s, \cdots, x_e)$ are sampled such that their start position is taken from a uniform distribution $U(1, l-1)$ the span length $l = e - s + 1$ is determined by a beta distribution $B(a, b) \times (l - s)$, where l denotes the number of phrases/sentences/paragraphs in the document with a, b as hyperparameters.

Multi-domain Text Encoding: Formally, let there be a query q with an associated positive domain d^+ and a pool of $N-1$ negative domains (d_i^-). For each domain, we concatenate the input query with multiple spans, add a special [CLS] token before the query text and a [SEP] token between the multiple spans to obtain the concatenated text sequence $t = [CLS], q, [SEP], x_s^1 \cdots x_e^1, [SEP] \cdots, x_s^N \cdots x_e^N$. We encode the input text sequence using a multi-layer Transformer encoder which maps each word to a low-dimensional dense representation $h_0, h_1, \cdots, h_i = Transformer(x_0, x_1, \cdots, x_i)$, where $h_i \in R^H$ with H as the size of hidden dimension. We then pass the encoded representation through a project layer which is a fully-connected layer followed by a non-linear activation $p_i = Tanh(FFN(h_i))$ to prevent representation collapse during contrastive learning. An average pooling operation is applied over the projected word representations to obtain the output representations as $z = AvgPool(p_s \cdots p_e)$.

Group-wise Contrastive Training: We use group-wise contrastive learning that incentives for representations of spans in a group sharing the same semantics to be similar while penalizing

the representations of groups expressing different semantics to be distinguished from each other. It encourages the SCAN retriever to discriminate and score related query-span pairs (from the same domain) higher than unrelated (from different domains) pairs. Given a mini-batch with N domains, the group-wise contrastive loss function L_{GC} is applied over $M = N * (3T + 1)$ spans as:

$$L_{GC} = \frac{-1}{3T} \sum_{i=1}^{N} \sum_{v \in d^+}^{T} \log \frac{\exp(sim(z_i, z_v)/\tau)}{\sum_{j=1}^{M} 1_{i \neq j} \exp(sim(z_i, z_j)/\tau)}$$

where $sim(\cdot)$ refers to the dot product and τ is the temperature parameter.

10.3.2 Retrieving Relevant Domains

At inference, we encode the concatenated query text with the document spans through the SCAN retriever. The relevance score assigned by the retriever model to a particular domain d_i based on the input query q is denoted by $P(d_i|q)$ is computed via the dot product operation between the [CLS] token embedding (z_q) and the average pooled embeddings of the document spans (z_{d_i}) as $P(d_i|q) = sim(z_q, z_{d_i})$.

10.3.3 Constructing k-Nearest N-gram Co-occurrence Matrix

We hypothesize that words that occur together in a specific domain have a high chance to trigger during inference. To exploit the word-level co-occurrence probabilities in text, we construct the n-gram frequencies matrices for each target domain d_i for $n \in [2, 4]$ over the entire vocabulary set V as $f^i_{[(w_{t-n} \rightarrow w_{t-1}), w_v]}$. However, n-gram frequencies tend to get sparser with higher values of n. Moreover, restricting the query n-grams to only exact matches leads to a loss of information due to ignorance of semantically similar n-grams (Li et al. 2017). To overcome these drawbacks, we utilize k-nearest n-grams. We construct a key-value data store with keys

as the Bert embeddings of n-grams and values as their corresponding n-gram co-occurrence frequencies. At inference, PersonaLM uses k-NN with euclidean distance metric to query the datastore for top-k nearest neighbor n-grams based on their BERT representations. The top-k probability distributions obtained from the n-gram datastore are summed over the entire vocabulary to get $\hat{f}^i_{[(w_{t-n} \rightarrow w_{t-1}), w_v]}$. The next word prediction for a selected domain is calculated as a weighted sum of k-nearest bigrams, trigrams, and 4-gram frequencies as $P(w_t|d_i, q) = \sum_{j=2}^{4} \alpha_j * [\hat{f}^i_{[(w_{t-j} \rightarrow w_{t-1}),1]}, \cdots \hat{f}^i_{[(w_{t-j} \rightarrow w_{t-1}),V]}]$, where $\alpha_j \in [0, 1]$ are hyperparameters.

10.3.4 LM Augmentation

We compute the `PersonaLM` retrieved next-word probability $P_{PersonaLM}(w_t|q)$ by summing the normalized k-nearest n-gram co-occurrence probabilities weighted by the relevance score of the selected domain over the target vocabulary as:

$$P_{PersonaLM}(w_t|q) = \sum_{i=0}^{K} sim(z_q, z_{d_i}) \times (\sum_{j=2}^{n} \alpha_j * [\hat{f}^i_{[(w_{t-j} \rightarrow w_{t-1}),1]}, \cdots \hat{f}^i_{[(w_{t-j} \rightarrow w_{t-1}),V]}])$$

Finally, we interpolate the retrieved next-word probability distribution through PersonaLM ($p_{PersonaLM}$) with the base LM output (P_{LM}) using a hyperparameter λ to produce the final next-word probability distribution as:

$$P(w_t|q) = \lambda P_{PersonaLM}(w_t|q) + (1 - \lambda) P_{LM}(w_t|q)$$

10.4 Experiments

10.4.1 Training SCAN retriever

We start with a pre-trained BERT model as the encoder which is further trained using group-wise contrastive learning objective on the following IR benchmarks: (1) MS MARCO Passage Ranking

(MARCO Dev Passage), (2) MS MARCO Document Ranking (MARCO Dev Doc) (Nguyen et al. 2016); (3) TREC 2019 Passage Ranking (TREC2019 Passage) and (4) TREC 2019 Document Ranking (TREC2019 Document) (Craswell et al. 2020).

10.4.2 Datasets

We evaluate the PersonaLM method on our proposed ASAP dataset and the UserLibri corpus.

ASR Language Model Personalization (ASAP) benchmark aims to evaluate the efficacy of LMs for personalized automatic speech recognition based on user/domain-specific information for next word prediction and ASR n-best second-pass rescoring.

(1) Personalized Meeting ASR of user's spoken utterances in professional meetings. This task assesses the model's ability to capture user-preferred dialogue patterns and linguistic characteristics. We leverage the AMI Meeting corpus (Kraaij et al. 2005) by splitting the user-specific recordings to obtain personalized utterance-text pairs.

(2) Personalized TED Talk ASR to convert recorded TED talks delivered by a specific user into text transcript. This task evaluates the LM's capability to capture topics-aware word patterns in the speeches from the TED-LIUM v3 corpus (Hernandez et al. 2018). We split the recorded TED talks temporally with historical utterance-text pairs forming the domain-specific train set.

(3) Personalized Financial Earning Calls ASR: Perform speech-to-text for financial earnings conference calls that include company-specific financial information. The task aims to evaluate a language model's capacity to extract company-specific named entities, abbreviations, facts, and long-tail word patterns. We adopt the conference call-transcript pairs from combined Earnings-21 (Rio et al. 2021) and Earnings-22 (Del Rio et al. 2022) datasets. Additionally, we use Wikitext-103 (Merity et al. 2017) to test the LM domain adaptation in topic-specific documents. **Data Prepro-**

Dataset	Train	Val	Test	Vocab Size	Domain	# Domains
Earnings-21+22	49.6K	7.1K	14.2K	20K	Earning Call	169
AMI Meeting Corpus	17.1K	2.7K	5.8K	11K	Meeting Recording	135
TED-LIUM v3	188.9K	26.6K	9.3K	46K	TED Talk	2351
Wikitext-103	2M	300K	10K	200K	Wikipedia Page	30k
UserLibri	6.3M	700K	10K	10K	Books	107

Table 10.1: Data stats of ASAP, UserLibri, and WikiText-103.

cessing: To study personalization in language modeling, we reformulated all the listed datasets to identify explicit users/domains. For each dataset, we combined the original train/val/test portions and splitted user-based data in the ratio of 70:10:20 such that each user/domain appears only in one of splits. Table 10.1 shows statistics on dataset size and distribution. **UserLibri Dataset** (Breiner et al. 2022a) reformulates the Librispeech corpus into user-specific audio-transcript pairs supplemented with personalized text-only data corresponding to each user with similar vocabulary, character names, and writing styles as the recordings.

10.4.3 Experiments for ASR Personalization

Language Model Architecture: We experiment with both LSTM and Transformer LMs. LSTM model has 2 layers with a 300-d embedding layer and a hidden dimension of 1500. Transformer LM consists of 4 layers of encoder-decoder with 12 heads, 128-d hidden representations, and a feed-forward layer of 3072-d. For generating ASR n-best hypotheses, we use a pre-trained RNN-T ASR Model with Emformer encoder (Shi et al. 2021b), LSTM predictor, and a joiner with 80M parameters.

Pre-training LMs: LSTM and Transformer LMs are pre-trained on Librispeech (Panayotov et al. 2015) train set for 25 epochs with batch size of 256, Adam optimizer and cross-entropy loss for next word prediction. We select model checkpoints with least perplexity on the Librispeech

validation.

Adaptation to Unseen Domains: We evaluate the retrieval augmentation in two settings: (1) *Without fine-tuning*: LM pre-trained on generic corpus; (2) *With fine-tuning*: LM pre-trained on generic corpus and fine-tuned on the entire out-of-domain train corpus. In both cases, evaluation is performed on out-of-domain test set.

Baselines: **(i) LSTM/Transformer**: Language model without any augmentation, **(ii) Neural Cache Model** (Grave, Joulin, and Usunier 2017) augments LM output with continuous a cache memory of previous hidden states. The stored keys are used to retrieve the next word through a dot product-based memory lookup with the query. **(iii) kNN-LM** (Khandelwal et al. 2020): Following (Das et al. 2022), we adopt kNN-LM to memorize context vectors from representations from out-of-domain train set in an external data store. During inference, the k-nearest neighbors of the decoder output representations are interpolated with LM output. (iv) **Unified N-gram Co-occurrence**: N-gram word frequency matrices built from the combined out-of-domain train set of each user/domain are augmented with LM at inference. (v) **PersonaLM $w\backslash$ other retrievers**: Replacing SCAN retriever with DPR (Karpukhin et al. 2020) or Contriever (Izacard et al. 2021).

Ablation Studies: **(i) PersonaLM w\o SCAN retriever**: We use the dot product of query and domain context encoded through pre-trained BERT to compute the weightage of each domain; **(ii) PersonaLM w\o k-Nearest N-grams**: Similar to kNN-LM, we augment the NWP with k-nearest Bert contexts vectors extracted from individual domains. We compute relevance scores from the SCAN retriever to get the weighted sum of the probability distributions from each domain.

Evaluation Metrics: We utilize word-level perplexity scores to evaluate LM performance for

next-word prediction. We also report Word Error Rate (WER) for ASR second-pass re-scoring for ASAP corpus. For UserLibri, we evaluate WER per user for both streaming and non-streaming ASR settings. For each model, we report results with minimal perplexity by iterating the interpolation parameter λ between 0 to 1 in increments of 0.1.

ASR Model Architecture for UserLibri: We utilize separate architectures for streaming and non-streaming ASR. The Conformer Hybrid Autoregressive Transducer (HAT) from (Breiner et al. 2022a) has 86M parameter and consists of 12 encoder layers of 512-d, 4 attention heads, convolution kernel size of 32, and a HAT decoder with a single RNN layer of 640-d. Each label is embedded with 128-d, and inputs are tokenized with a 1k Word-Piece Model trained on the LibriSpeech train set. The models are trained with Adam using group-norm (Wu and He 2018). For streaming ASR evaluation, the Conformer HAT model uses causal convolution, local self-attention, and left-sided context stacking to ensure no look-ahead. The non-streaming version has multi-headed attention. COnformer models in both cases are trained on 960 hours of LibriSpeech audio training set. The LSTM decoder in streaming ASR is a 2-layer RNN of size 1340 with 25M parameters. It uses a similar Word Piece model as the Conformer.

10.4.4 SCAN Retriever Experiments on LaMP

LaMP (Salemi et al. 2023) is a benchmark corpus to evaluate LM personalization on the following user-specific text classification and generation tasks: (1) citation identification, (2) news categorization (3) product rating prediction, (4) news headline generation, (5) scholarly title generation, (6) email subject generation, and (7) tweet paraphrasing. Each data sample contains an input sequence to the model, a target output, and several text samples that encapsulate the user profiles that can be employed for LLM personalization.

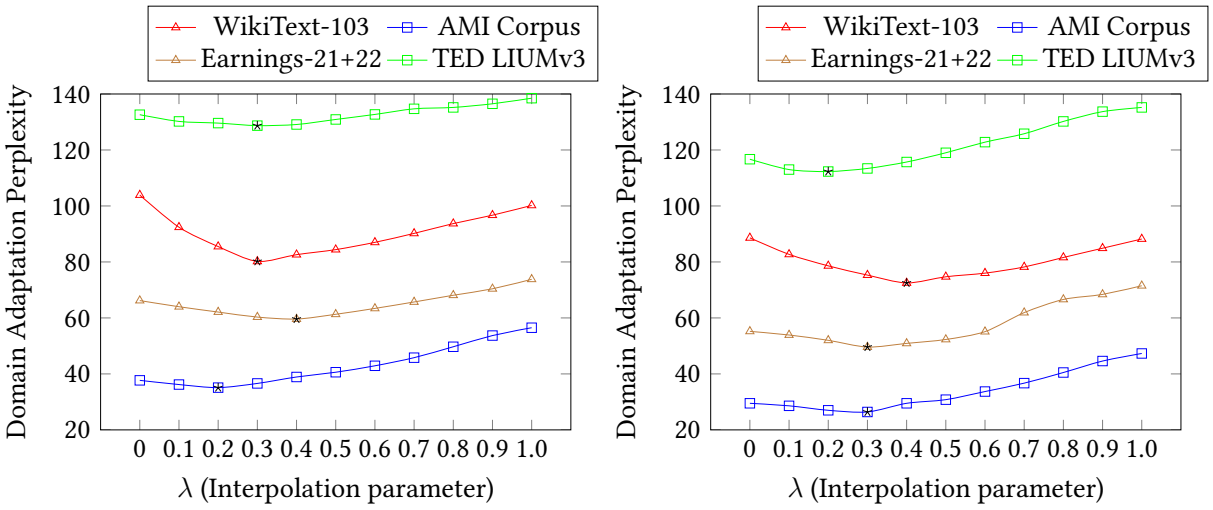
Baselines: Inspired by (Salemi et al. 2023), we compare SCAN retriever with strong baseline retrievers for user-specific context selection: 1) Random, (2) BM25, and (3) Contriever.

Evaluation on LLM Personalization: We evaluate different retrievers for personalized prompt construction in following settings: (a) Zero-shot LLM prompting: Retrieve top-k most relevant user items from external corpus to append in prompts for GPT-3.5[1] and FlanT5-XXL (Chung et al. 2022); (b) Few-shot LM Fine-tuning: Fine-tuning FlanT5-base (Chung et al. 2022) using top-k retrieved items from the user profile.

10.5 Results and Analysis

Perplexity Evaluation: Tables 10.2 and 10.3 compare the perplexity scores of the proposed `PersonaLM` retrieval augmentation against other baselines. We observe that the Neural Cache model (Li, Povey, and Khudanpur 2020) slightly improves over naive LM baselines but struggles due to its inability to handle long-range dependencies through pointer mechanism. Consistent with observations of (Wang et al. 2023), kNN-LM (Khandelwal et al. 2020) reduces perplexity by 5-10% but is still challenged by the non-parametric fuzzy nature of k-nearest Bert context vectors selected amongst billions of stored contexts from a gigantic data store. Similar to (Drozdov et al. 2022), our experiments show that Unified N-gram Co-occurrence shows slight improvement over kNN-LM as n-grams are better at capturing highly domain-specific rare word patterns. However, it still suffers from sub-optimal n-gram retrievals from a mixture of domains as the target probabilities get averaged out when computed over the entire external corpus. Our proposed method achieves SOTA performance and improves the LM perplexity by a significant margin on WikiText-103 (57.1 – 61.8% w\o fine-tuning, 20.0 – 25.1% with fine-tuning), Earnings21+22

[1] https://platform.openai.com/docs/models/gpt-3-5



(a) LSTM

(b) Transformer

Figure 10.4: Plot of λ (interpolation parameter) vs perplexity of PersonaLM with fine-tuned (a) LSTM, (b) Transformer LMs on WikiText-103, Earnings-21+22, AMI Corpus, and TED LIUMv3 datasets. Curves show convex characteristics with the optimal value of λ varying with different settings.

(46.4 – 47.2% w\o fine-tuning, 11.4 – 12.6% with fine-tuning), AMI Meeting Corpus (71.6 – 71.9% w\o fine-tuning, 7.9 – 12.8% with fine-tuning), and TED LIUMv3 (26.6 – 30.2% w\o fine-tuning, 3.8 – 4.7% with fine-tuning) over base LMs, demonstrating that contextually matching query with most relevant domains via SCAN retriever module boosts retrieval performance which reinforces the next word prediction task. Replacing the SCAN retriever in PersonaLM with other baseline retrievers like Dense Passage Retriever (DPR) or Contriever leads to degraded performance. However, the performance does not decrease below kNN-LM or Unified N-gram Co-occurrence Retrieval methods, signifying the marginal benefit of domain-specific retrieval to augment LM predictions.

ASR Rescoring Analysis on ASAP dataset: Table 10.3 shows results of second-pass ASR rescoring on AMI Meetings and TED LIUMv3 datasets where our proposed approach improves WER relatively by $\sim$ 5%. Retrieval-augmented LMs when combined with the n-best hypotheses

produced by the audio model lead to statistically significant WER reduction with respect to both kNN-LM and PersonaLM with Contriever baselines. Combining audio model and PersonaLM allows wins on tail words while avoiding losses on common word occurrences. **Evaluation on UserLibri dataset** (Breiner et al. 2022b) shows that PersonaLM improves WER for both streaming and non-streaming ASR. Compared to fine-tuning the LM on the entire external personalized corpus (p13n LM), PersonaLM can selectively learn user-specific discriminative patterns in speech text and weigh it appropriately for biasing the LM predictions.

Ablation Analysis: Tables 10.2-10.4 highlights in red show the ablation study for PersonaLM. We observe that SCAN retriever is critical in all settings due to its enhanced ability to learn enhanced discriminative document representations that help assign appropriate weights to external domains. Removing the k-Nearest N-grams severely deteriorates the performance as the LM is no longer able to exploit the personalized n-gram probability distribution from different domains. The severe performance drop in WER for speech datasets in the absence of either of the components underscores their significance for personalized ASR tasks.

Adaptation to Unseen Domains: Retrieval augmentation with fine-tuned LMs shows a sustained relative gain of 5-18% across all settings despite having seen the same data during the fine-tuning stage. This observation validates our our hypothesis that despite the benefits of transfer learning for out-of-domain generalization, explicit memorization is needed to effectively learn user-specific linguistiuc patterns not retained during fine-tuning.

Impact of Interpolation Parameter: Figure 10.4 shows that the optimal value of interpolation parameter λ varies in different settings. λ vs perplexity curve shows convex characteristics for all variants of `PersonaLM`. Perplexity scores improve with increasing λ as explicit memorization of rare word patterns mined from matching domains benefits the next word prediction task but

starts to drop monotonically after reaching an inflection point.

Qualitative Examples: Qualitative examples from ASAP, Wikitext, and UserLibri datasets in Table 10.6 along with model predictions. PersonaLM is able to able to correctly predict proper nouns, abbreviations, and homonyms mistaken by fine-tuned LM and kNN-LM baselines, while also fixing the problem of over-prediction of domain-specific frequent words commonly observed in Unified N-gram Co-occurrence Retrieval baseline.

Downstream Application of SCAN Retriever: Table 10.5 shows the application of different retrievers for improving user-personalized text generation and classification on the LaMP dataset. We aim to retrieve the most relevant user profiles that can be augmented with query context for zero-shot prompting or few-shot fine-tuning of LLMs. SCAN retriever outperforms both BM-25 and Contriever baselines and shows significant gains across different metrics compared to non-personalized LMs across all subtasks, except tweet paraphrasing. As opposed to earlier advances where retrieval augmentation was performed during LM training (Lewis et al. 2020c; Guu et al. 2020), the core merits of our proposed SCAN retriever are that it is extensible to any LM (LSTM, Transformer, Generative LLMs like FlanT5 and GPT-3.5), can seamlessly adapt to new users, enable in-context retrieval augmentation without any LM-specific fine-tuning, and requires very small memory footprint with negligible computational overhead.

10.6 Conclusion

We introduce PersonaLM retrieval augmentation for ASR personalization using SCAN - a neural retriever trained via group-contrastive learning to rank textual documents from an external knowledge corpus based on their semantic similarity with the input query. We aggregate the

probability distribution of the next word prediction by utilizing domain-specific n-gram word frequency representations weighted by the relative importance of the external domains to the input query. Experiments on our proposed ASAP benchmark and the UserLibri dataset show that our method achieves SOTA perplexity and WER. We show that SCAN retriever is also useful for in-context LLM augmentation for zero-shot prompting and few-shot fine-tuning. For future work, we intend to extend our work for multilingual ASR and speech style transfer.

	Model	WikiText-103 Perplexity (↓)	Earnings-21+22 Perplexity (↓)
Without Fine-tuning	LSTM	1384.1	757.6
	+ Neural Cache	1325.3	723.8
	+ kNN-LM	1191.6	659.1
	+ Unified N-gram Co-occurrence Retrieval	603.6	477.2
	+ PersonaLM w \DPR	544.3	420.3
	+ PersonaLM w \Contriever	539.8	412.3
	+ PersonaLM	**527.9**	**405.6**
	+ PersonaLM w\o SCAN Retriever	542.8	415.4
	+ PersonaLM w\o k-Nearest N-gram	542.3	415.7
With Fine-tuning	LSTM	103.9	66.2
	+ Neural Cache	97.6	66.0
	+ kNN-LM	91.8	65.7
	+ Unified N-gram Co-occurrence Retrieval	89.2	64.5
	+ PersonaLM w \DPR	84.2	63.0
	+ PersonaLM w \Contriever	80.2	59.6
	+ PersonaLM	**77.8**	**57.8**
	+ PersonaLM w\o SCAN Retriever	82.7	61.5
	+ PersonaLM w\o k-Nearest N-gram	82.3	61.9

(a) LSTM LM

	Model	WikiText-103 Perplexity (↓)	Earnings-21+22 Perplexity (↓)
Without Fine-tuning	Transformer	1322.3	834.2
	+ Neural Cache	1295.3	802.4
	+ kNN-LM	1150.4	717.8
	+ Unified N-gram Co-occurrence Retrieval	585.3	454.8
	+ PersonaLM w \DPR	578.2	452.7
	+ PersonaLM w \Contriever	569.3	446.1
	+ PersonaLM	**567.6***	**440.4***
	+ PersonaLM w\o SCAN Retriever	572.9	448.2
	+ PersonaLM w\o k-Nearest N-gram	572.1	447.9
With Fine-tuning	Transformer	88.6	55.2
	+ Neural Cache	86.8	54.9
	+ kNN-LM	79.3	54.2
	+ Unified N-gram Co-occurrence Retrieval	76.5	53.8
	+ PersonaLM w \DPR	76.1	52.6
	+ PersonaLM w \Contriever	72.5	49.6
	+ PersonaLM	**70.9***	**48.2***
	+ PersonaLM w\o SCAN Retriever	74.8	51.5
	+ PersonaLM w\o k-Nearest N-gram	73.5	50.3

(b) Transformer LM

Table 10.2: Results comparing the performance of `PersonaLM` Retrieval Augmentation for (a) LSTM and (b) Transformer LMs with baselines and ablations (in red) for the **Next Word Prediction** task on WikiText-103 and Earnings-21+22 datasets. PersonaLM achieves the lowest perplexity scores across all settings. * indicates that the result is statistically significant (5 runs) based on Wilcoxon's signed rank test ($p < 0.001$).

	Model	AMI Meeting Corpus		TED LIUMv3	
		Perplexity (↓)	WER (↓)	Perplexity (↓)	WER (↓)
Without Fine-tuning	Audio Model Only (Emformer)	–	32.54	–	17.23
	Audio Model + LSTM	1636.4	31.75	427.7	13.51
	+ Neural Cache	1545.4	31.69	414.5	13.25
	+ kNN-LM	1232.2	31.62	389.7	7.82
	+ Unified N-gram Co-occurrence Retrieval	606.7	31.25	335.4	7.34
	+ PersonaLM w \DPR	490.5	31.22	332.8	7.23
	+ PersonaLM w \Contriever	471.2	31.15	315.0	7.16
	+ PersonaLM	**463.8***	**31.01***	**313.8***	**7.01***
	+ PersonaLM w\o SCAN Retriever	480.2	31.13	320.3	7.15
	+ PersonaLM w\o k-Nearest N-gram	478.9	31.10	318.8	7.14
With Fine-tuning	Audio Model Only (Emformer)	–	32.54	–	17.23
	Audio Model + LSTM	37.7	31.40	132.6	13.27
	+ Neural Cache	37.5	31.36	132.2	13.03
	+ kNN-LM	37.1	31.27	131.5	7.76
	+ Unified N-gram Co-occurrence Retrieval	36.6	31.20	130.3	7.44
	+ PersonaLM w \DPR	36.2	31.16	130.2	7.28
	+ PersonaLM w \Contriever	35.1	31.14	128.7	7.03
	+ PersonaLM	**34.7***	**31.01***	**127.5***	**6.90***
	+ PersonaLM w\o SCAN Retriever	35.9	31.14	129.7	7.10
	+ PersonaLM w\o k-Nearest N-gram	35.7	31.12	129.4	7.07

(a) LSTM LM

	Model	AMI Meeting Corpus		TED LIUMv3	
		Perplexity (↓)	WER (↓)	Perplexity (↓)	WER (↓)
Without Fine-tuning	Audio Model Only (Emformer)	–	32.54	–	17.23
	Audio Model + Transformer	2114.3	32.05	442.0	13.24
	+ Neural Cache	1987.5	32.01	424.5	13.18
	+ kNN-LM	1579.0	31.95	398.6	7.57
	+ Unified N-gram Co-occurrence Retrieval	637.1	31.37	332.3	7.22
	+ PersonaLM w \DPR	624.9	31.33	327.4	7.14
	+ PersonaLM w \Contriever	601.4	31.25	310.1	7.05
	+ PersonaLM	**592.6***	**31.16***	**308.3***	**6.92***
	+ PersonaLM w\o SCAN Retriever	610.3	31.29	316.6	7.15
	+ PersonaLM w\o k-Nearest N-gram	608.5	31.27	315.5	7.05
With Fine-tuning	Audio Model Only (Emformer)	–	32.54	–	17.23
	Audio Model + Transformer	29.5	31.28	116.7	12.98
	+ Neural Cache	29.3	31.24	116.2	12.78
	+ kNN-LM	29.1	31.19	115.6	7.35
	+ Unified N-gram Co-occurrence Retrieval	28.1	31.14	114.0	7.21
	+ PersonaLM w \DPR	27.7	31.10	113.0	7.04
	+ PersonaLM w \Contriever	26.4	31.03	112.3	6.93
	+ PersonaLM	**25.7***	**30.88***	**111.1***	**6.86***
	+ PersonaLM w\o SCAN Retriever	27.0	31.09	112.7	7.00
	+ PersonaLM w\o k-Nearest N-gram	26.9	31.05	112.8	6.98

(b) Transformer LM

Table 10.3: Results comparing the performance of `PersonaLM` Retrieval Augmentation for (a) LSTM and (b) Transformer LMs with baselines and ablations (in red) for the **Next Word Prediction** and **Second-Pass ASR Re-scoring** tasks on AMI Meeting Corpus and TED LIUMv3 datasets. PersonaLM achieves minimum perplexity WER on both datasets. * indicates that the result is statistically significant (5 runs) based on Wilcoxon's signed rank test ($p < 0.001$).

Model	Streaming			Non-Streaming		
	Test-Clean	Test-Other	All	Test-Clean	Test-Other	All
Conformer Transducer (Audio Model Only)	6.0	11.2	8.5	2.5	6.8	4.5
Conformer Transducer + LM (25M)	5.2	9.1	7.1	2.0	5.5	3.7
+ Fine-tuned LM (p13n)	5.2	8.7	6.9	1.9	4.6	3.2
+ Unified N-gram Retrieval	5.1	8.6	6.8	1.9	4.4	3.1
+ PersonaLM $w\backslash$ Contriver	5.0	8.5	6.8	1.8	4.4	3.0
+ PersonaLM	4.8*	8.3*	6.6*	1.6*	4.2*	2.8*
+ PersonaLM w\o SCAN retriever	5.1	8.6	6.9	1.8	4.6	3.0
+ PersonaLM w\o k-Nearest N-gram	5.0	8.5	6.8	1.8	4.5	3.1

Table 10.4: Performance comparison of PersonaLM Retrieval Augmentation with baselines and ablations for personalized (a) streaming ASR and (b) non-streaming ASR on the UserLibri dataset. PersonaLM reduces the WER by 7-12% across all settings. * indicates that the result is statistically significant (5 runs) based on Wilcoxon's signed rank test ($p < 0.001$).

Dataset	Metric	FlanT5-XXL			GPT-3.5			FlanT5-base (fine-tuned)		
		Non-personalized	Contriver	SCAN Retriever	Non-personalized	Contriver	SCAN Retriever	Non-personalized	Contriever	SCAN Retriever
LaMP-1U: Personalized Citation Identification	Accuracy	0.522	0.675	**0.687**	0.510	0.701	**0.715**	0.522	0.731	**0.745**
LaMP-2U: Personalized	Accuracy	0.591	0.598	**0.608**	0.610	0.693	**0.702**	0.730	0.835	**0.843**
News Categorization	F1	0.463	0.471	**0.484**	0.455	0.455	**0.466**	0.504	0.637	**0.648**
LaMP-3U: Personalized	MAE	0.357	0.282	**0.276**	0.699	0.658	**0.644**	0.314	0.258	**0.246**
Product Rating	RMSE	0.666	0.584	**0.565**	0.977	1.102	**0.980**	0.624	0.572	**0.559**
LaMP-4U: Personalized	ROUGE-1	0.164	0.192	**0.211**	0.133	0.160	**0.172**	0.158	0.201	**0.212**
News Headline Generation	ROUGE-L	0.149	0.178	**0.187**	0.118	0.142	**0.155**	0.144	0.185	**0.192**
LaMP-5U: Personalized	ROUGE-1	0.455	0.467	**0.475**	0.395	0.398	**0.409**	0.424	0.453	**0.470**
Scholarly Title Generation	ROUGE-L	0.410	0.424	**0.433**	0.334	0.336	**0.342**	0.382	0.414	**0.425**

Table 10.5: Performance comparison of zero-shot FlanT5-XXL, GPT-3.5, and few-shot fine-tuned FlanT5-base for personalized text classification and generation results on the validation set of LaMP dataset. For all metrics the higher the better, except for RMSE and MAE. Prompting LLMs with user-specific context selected by the SCAN retriever consistently reports the best performance.

Win/Loss	Ground Truth	Fine-tuned LM	kNN-LM	Unified N-gram Retrieval	PersonaLM	Type
Win	king sharr khan	king sir can	king sir khan	king share can	kingsharr khan	Proper Name
Win	murdoch blinked	mr duck winged	mom duck blink	murdock blinked	murdoch blinked	Proper Name
Loss	tied to a woman	tied to a woman	tied too a woman	died to a woman	died to a woman	homonym
Win	lord of baghdad	lord of bag dad	lord of bag dad	lord baghdad	lord of baghdad	homonym
Win	mister beale	mister bell	mister bell	mister be elle	mister beale	Proper Name
Win	thanks izzy	thanks is he	thanks is he	thank is he	thanks izzy	homonym
Win	North American Treaty Alliance	North American Treaty All Hands	North American Treaty Organization	North American Treaty Alliance	North American Treaty Alliance	Abbreviation Term
Loss	utterly RSVP for this our invitation	utter RSVP for this our invitation	utter respect for this our invitation	utterly rest for this our invitation	utterly respectively for this our invitation	Abbreviation Term

Table 10.6: Qualitative examples: Ground truth, baseline predictions, and PersonaLM predictions for a few samples from UserLibri and ASAP eval set. PersonaLM is able to able to correctly predict proper nouns, abbreviations, and homonyms mistaken by fine-tuned LM and kNN-LM baselines, while also fixing the problem of over-prediction of domain-specific frequent words commonly observed in Unified N-gram Co-occurrence Retrieval baseline.

CHAPTER 10

Conclusion and Future Directions

We now take a step back to understand how all our proposed methodologies connect to our bigger picture. Our main contribution has been the development and implementation of automated techniques for document information extraction, document structure understanding, and manipulations. We also saw multiple applications of these methodologies for downstream tasks and adjacent domains. We have delved into both semantic and structural aspects of information processing for document understanding. Our methods show promising abilities to learn from large-scale document datasets which we developed as part of the research.

10.1 Summary of Our Work

We began with the development of techniques that combined the multihop capability of graph neural networks with Transformer models to reason beyond a fixed context length in long-form documents. Departing away from chucking of context followed by merging of reasoning mechanism, we demonstrated that end-to-end methods can benefit from the backpropagation of losses across the Transformer encoders and graph convolutions pipelines. Beyond this, Transformers models encompassing layout information showed remarkable efficiency for spatial understanding tasks over multimodal document representations. Models such as LayerDoc helped solve the grouping, reading order, and hierarchy reconstruction tasks in an end-to-end fashion in a multi-task learning setup. Hierarchical information also aided document editing and position-aware TTS models, further providing evidence in support of structure understanding. We further demonstrated that document-level information extraction and long-context multimodal understanding aid several non-trivial downstream tasks. An important learning from our research has been on the effect of fine-tuning pretrained language models with multi-task losses to customize their applications in non-standard domains.

10.2 Future Work

An important future work that remains to be realized is the extension of the proposed methods to low-resource settings such as low-resource languages, specific domains, and unseen user groups. Such efforts will require extensive data collection as well as recalibration of necessary processing steps which remains to be a non-trivial challenge. With the advent of Large Language Models, few-shot and zero-shot capabilities of the proposed tasks and systems need to be re-

evaluated. LLMs such as ChatGPT and GPT-4 provide promising new directions to extend our work by leveraging stronger Transformer networks for information extraction. Recent works necessitate experimental evaluation to compare how supervised training stands up to few-shot prompting for information extraction tasks. With improvements in context length limitations of pre-trained decoder-only Transformer models, the challenge of attention sparsity in long-context input still remains to be evaluated. Bias and fairness in neural training have been a long-standing problem that necessitates that any deployment of our proposed methodologies be evaluated for possible harm to the under-served communities. As future work, we motivate the community to undertake extensive studies to analyze the new methods that may be made more robust to adversaries. Following the proposed work, we invite researchers to explore methods to incorporate expectant human values into the language modeling paradigms for document information extraction, structure parsing, and user-based document manipulations.

Potential Applications to Legal Documents: Our research finds immense applications for legal work involving information retrieval from large collection of documents Sansone and Sperlí 2022. In the legal domain, many of the documents may be unstructured or semi-strcutured with loosely defined meta-data Sancheti et al. 2022. Such cases are frequent during the information discovery process and for collecting evidences as part of the FOIA (Freedom of Information Act) from previously unreleased document sets controlled by the United States government. Our proposed document selection algorithm - DocInfer an be repurposed for such application use cases where the requested information query can be seen as an NLI task over a collection of long documents. Chunking these documents into a hierarchy of small paragraphs and sentences, we can apply Paragraph selection module and optimal evidence retrieval mechanism to access the most relevant pieces of information for answering the input

query. Such a process will make the lives of lawyers much easier and free them to focus on more important aspects of the job such as legal reasoning and case preparation. Our work in no way advocates the replacement of legal professional, rather pushes AI technologies to make them more efficient and cut down mundane work. However, further research needs to be conducted to make these methodologies more robust and interpretable. One of the current disadvantages of our system is that it is prone to errors based on legal fine-print due to lack of domain-specialized reasoning modules for context understanding. Another relevant use case in this domain is filtering down documents related to a specific case or type Nguyen et al. 2022. Lawyers indulge in challenging work settings where they may have to select a subset of document based on a certain judicial ruling or legal law from hundreds of thousands of past cases Schumann, Meyer, and Gomez 2022. To solve such problems, our proposed SCAN retriever can be best purposed for this task. SCAN retriever is trained to contrastively select a group of documents that share a particular characteristic with a seed document. In this manner, it can retrieve and rank the most relevant subset of documents, significantly reducing the load of manual grunt work for lawyers. One important modification that may be needed to make this application more successful can be to use a specialized Transformer models pre-trained in legal-language to make the SCAN retriever robust to domain shift.

Potential Applications to Enhance Accessibility: The proposals presented as part of this thesis aim to enhance user accessibility to make document consumption, creation and modifications more efficient for people facing mental or physical challenges. DocLayoutTTS is one of the first attempts of its kind that enables screen readers for structured documents such as forms, websites, and academic articles. Prior works in text-to-speech research were limited to converting text to spoken words. Our methodology addresses the pain point of organizing

text in the correct reading order which can then be utilized for speech generation and help people with reading challenges overcome their lack of agency in interacting with legal forms, contracts, web interfaces, posters, and many more (semi-)structured digital documents. Through DocEdit, we showed how verbal requests can be used to automate visual document editing without the need to manually execute the editing process. This workflow is especially useful for people with physical disabilities such as carpal tunnel syndrome, arthritis, neuropathy, and even amputations that may restrict their locomotor movements. Lastly, our work on Language Model personalization takes a step towards helping people achieve complex linguistic tasks using LLMs in a way that the LLM can adapt to a user's style, tone and identity specifics to aid them in their creative workflows. This can be potentially be helpful for patients suffering from partial dementia who may need writing assistance according to their unique needs and preferences (Wood et al. 2023). Further, our line of research sheds light on how PDFs when created from source files often lose the tagging metadata present in the authoring application related to the content type and order (Jembu Rajkumar et al. 2020). Manually tagging elements, extracting the reading order, and repairing tables and structured content loss is very time-consuming for an average content creator (Pradhan et al. 2022). Moreover, this metadata cannot be easily added back due to the complexity of the PDF format (Jembu Rajkumar, Jordan, and Lazar 2020). Our work on LayerDoc provides a comprehensive solution to enrich PDF documents with descriptive metadata such as OCR recovered text along with the logical structure of content, marking heading level tags, organization of the content in pages, sections, and paragraphs. This is an essential step for remediating a PDF document for accessibility for blind people and those with low vision disabilities.

Bibliography

Abouelenien, Mohamed, Rada Mihalcea, and Mihai Burzo (2016). "Analyzing thermal and visual clues of deception for a non-contact deception detection approach". In: *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 1–4.

Aggarwal, Milan, Hiresh Gupta, Mausoom Sarkar, and Balaji Krishnamurthy (Nov. 2020a). "Form2Seq : A Framework for Higher-Order Form Structure Extraction". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 3830–3840. DOI: 10.18653/v1/2020.emnlp-main.314.

Aggarwal, Milan, Mausoom Sarkar, Hiresh Gupta, and Balaji Krishnamurthy (2020b). "Multi-Modal Association based Grouping for Form Structure Extraction". In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2064–2073.

Aiello, Marco, A Smeulders, et al. (2003). *Bidimensional relations for reading order detection*. University of Groningen, Johann Bernoulli Institute for Mathematics and ...

Aldawsari, Mohammed, Adrián Pérez, Deya Banisakher, and Mark A. Finlayson (2020). "Distinguishing Between Foreground and Background Events in News". In: *COLING*.

Allen, James F (1983). "Maintaining knowledge about temporal intervals". In: *Communications of the ACM* 26.11, pp. 832–843.

Almazán, Jon, Albert Gordo, Alicia Fornés, and Ernest Valveny (2014). "Word spotting and recognition with embedded attributes". In: *IEEE transactions on pattern analysis and machine intelligence* 36.12, pp. 2552–2566.

Andersen, L (Mar. 2008). "Simple and efficient simulation of the Heston stochastic volatility model". In: *Journal of Computational Finance* 11. DOI: 10.21314/JCF.2008.189.

Appalaraju, Srikar, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha (2021). "DocFormer: End-to-End Transformer for Document Understanding". In: *arXiv preprint arXiv:2106.11539*.

Aracı, Doğu (2019). "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models". In: *ArXiv* abs/1908.10063.

Ariyo, Adebiyi A, Adewumi O Adewumi, and Charles K Ayo (2014). "Stock price prediction using the ARIMA model". In: *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. IEEE, pp. 106–112.

Baevski, Alexei, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli (2020). "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 12449–12460.

Bai, Jiangang, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong (2021). "Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees". In: *arXiv preprint arXiv:2103.04350*.

Bai, Song, Feihu Zhang, and Philip HS Torr (2021). "Hypergraph convolution and hypergraph attention". In: *Pattern Recognition* 110, p. 107637.

Ballesteros, Miguel, Rishita Anubhai, Shuai Wang, Nima Pourdamghani, Yogarshi Vyas, Jie Ma, Parminder Bhatia, K. McKeown, and Yaser Al-Onaizan (2020). "Severing the Edge Between Before and After: Neural Architectures for Temporal Ordering of Events". In: *ArXiv* abs/2004.04295.

Bao, Hangbo, Li Dong, Songhao Piao, and Furu Wei (2022). "BEiT: BERT Pre-Training of Image Transformers". In: *International Conference on Learning Representations*.

Beltagy, Iz, Matthew E Peters, and Arman Cohan (2020). "Longformer: The long-document transformer". In: *arXiv preprint arXiv:2004.05150*.

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston (2009). "Curriculum learning". In: *ICML '09*.

Bento, PMR, JAN Pombo, MRA Calado, and SJPS Mariano (2018). "A bat optimized neural network and wavelet transform approach for short-term price forecasting". In: *Applied energy* 210, pp. 88–97.

Berg, Mark de, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf (1997). "Computational geometry". In: *Computational geometry*. Springer, pp. 1–17.

Bernard, Victor L and Jacob K Thomas (1989). "Post-earnings-announcement drift: delayed price response or risk premium?" In: *Journal of Accounting research* 27, pp. 1–36.

Bethard, Steven, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen (2016). "Semeval-2016 task 12: Clinical tempeval". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 1052–1062.

Bhatia, Parminder, Yangfeng Ji, and Jacob Eisenstein (2015). "Better Document-level Sentiment Analysis from RST Discourse Parsing". In: *ArXiv* abs/1509.01599.

Blei, David M., A. Ng, and Michael I. Jordan (2003). "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3, pp. 993–1022.

Boersma, Paul and Vincent Van Heuven (Jan. 2001). "Speak and unSpeak with PRAAT". In: *Glot Int* 5, pp. 341–347.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). "Enriching word vectors with subword information". In: *Transactions of the association for computational linguistics* 5, pp. 135–146.

Bollerslev, Tim (1986). "Generalized autoregressive conditional heteroskedasticity". In: *Journal of econometrics* 31.3, pp. 307–327.

Borchmann, Łukasz, Dawid Wisniewski, Andrzej Gretkowski, Izabela Kosmala, Dawid Jurkiewicz, Lukasz Szalkiewicz, Gabriela Pałka, Karol Kaczmarek, Agnieszka Kaliska, and Filip Grali'nski

(2020). “Contract Discovery: Dataset and a Few-shot Semantic Retrieval Challenge with Competitive Baselines”. In: *FINDINGS*.

Borgeaud, Sebastian, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. (2022). “Improving language models by retrieving from trillions of tokens”. In: *International conference on machine learning*. PMLR, pp. 2206–2240.

Boukus, Ellyn and Joshua V Rosenberg (2006). “The information content of FOMC minutes”. In: *Available at SSRN 922312*.

Bowman, Samuel R, Gabor Angeli, Christopher Potts, and Christopher D Manning (2015). “A large annotated corpus for learning natural language inference”. In: *arXiv preprint arXiv:1508.05326*.

Breiner, Theresa, Swaroop Ramaswamy, Ehsan Variani, Shefali Garg, Rajiv Mathews, Khe Chai Sim, Kilol Gupta, Mingqing Chen, and Lara McConnaughey (2022a). “UserLibri: A Dataset for ASR Personalization Using Only Text”. In: *arXiv preprint arXiv:2207.00706*.

Breiner, Theresa, Swaroop Indra Ramaswamy, Ehsan Variani, Shefali Garg, Rajiv Mathews, Khe Chai Sim, Kilol Gupta, Mingqing Chen, and Lara McConnaughey (2022b). “UserLibri: A Dataset for ASR Personalization Using Only Text”. In: *Interspeech*.

Cai, Yong, Santiago Camara, and Nicholas Capel (2021). “It’s not always about the money, sometimes it’s about sending a message: Evidence of Informational Content in Monetary Policy Announcements”. In: *arXiv preprint arXiv:2111.06365*.

Cambre, Julia, Jessica Colnago, Jim Maddock, Janice Tsai, and Jofish Kaye (2020). “Choice of voices: A large-scale evaluation of text-to-speech voice quality for long-form content”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13.

Cao, Longbing (2022). “AI in Finance: Challenges, Techniques, and Opportunities”. In: *ACM Computing Surveys (CSUR)* 55.3, pp. 1–38.

Carbonell, Manuel, Pau Riba, Mauricio Villegas, Alicia Fornés, and Josep Lladós (2021). “Named Entity Recognition and Relation Extraction with Graph Neural Networks in Semi Structured Documents”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, pp. 9622–9627.

Carion, Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (2020). “End-to-end object detection with transformers”. In: *European conference on computer vision*. Springer, pp. 213–229.

Cassidy, T., B. McDowell, Nathanael Chambers, and Steven Bethard (2014a). “An Annotation Framework for Dense Event Ordering”. In: *ACL*.

Cassidy, Taylor, Bill McDowell, Nathanel Chambers, and Steven Bethard (2014b). *An annotation framework for dense event ordering*. Tech. rep. Carnegie-Mellon Univ Pittsburgh PA.

Chalkidis, Ilias, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos (2020). “LEGAL-BERT: The muppets straight out of law school”. In: *arXiv preprint arXiv:2010.02559*.

Chambers, Nathanael, T. Cassidy, B. McDowell, and Steven Bethard (2014a). “Dense Event Ordering with a Multi-Pass Architecture”. In: *Transactions of the Association for Computational Linguistics* 2, pp. 273–284.

Chambers, Nathanael, Taylor Cassidy, Bill McDowell, and Steven Bethard (2014b). “Dense event ordering with a multi-pass architecture”. In: *Transactions of the Association for Computational Linguistics* 2, pp. 273–284.

Chariri, Anis (June 2009). "Ethical Culture and Financial Reporting: Understanding Financial Reporting Practice within Javanese Perspective*". In: *Issues In Social And Environmental Accounting* 3.

Chatzis, Sotirios P, Vassilis Siakoulis, Anastasios Petropoulos, Evangelos Stavroulakis, and Nikos Vlachogiannakis (2018). "Forecasting stock market crisis events using deep and statistical machine learning techniques". In: *Expert systems with applications* 112, pp. 353–371.

Chen, Chen, Zuxuan Wu, and Yu-Gang Jiang (2016). "Emotion in context: Deep semantic feature fusion for video emotion recognition". In: *Proceedings of the 24th ACM international conference on Multimedia*, pp. 127–131.

Chen, Chonghao, Fei Cai, Xuejun Hu, Wanyu Chen, and Honghui Chen (2021). "HHGN: A Hierarchical Reasoning-based Heterogeneous Graph Neural Network for fact verification". In: *Information Processing & Management* 58.5, p. 102659.

Chen, Chung-Chi, Hen-Hsen Huang, and Hsin-Hsi Chen (2020). "NLP in FinTech applications: past, present and future". In: *arXiv preprint arXiv:2005.01320*.

Chen, Jianbo, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu (2018). "Language-based image editing with recurrent attentive models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8721–8729.

Chen, Jieshan, Mulong Xie, Zhenchang Xing, Chunyang Chen, Xiwei Xu, Liming Zhu, and Guoqiang Li (2020a). "Object detection for graphical user interface: old fashioned or deep learning or a combination?" In: *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.

Chen, Wenhu, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, SHIYANG LI, Xiyou Zhou, and William Yang Wang (2020b). "TabFact: A Large-scale Dataset for Table-based Fact Verification". In: *ArXiv* abs/1909.02164.

Chen, Wenhu, Xinyi Wang, and William Yang Wang (2021). "A Dataset for Answering Time-Sensitive Questions". In: *ArXiv* abs/2108.06314.

Chen, Xie, Xunying Liu, Mark JF Gales, and Philip C Woodland (2015). "Recurrent neural network language model training with noise contrastive estimation for speech recognition". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5411–5415.

Chen, Yu, Lingfei Wu, and Mohammed Zaki (2020). "Iterative deep graph learning for graph neural networks: Better and robust node embeddings". In: *Advances in Neural Information Processing Systems* 33.

Chen, Zhen-Heng, Siu Cheung Hui, Fuzhen Zhuang, Lejian Liao, Fei Li, Meihuizi Jia, and Jiaqi Li (2022). "EvidenceNet: Evidence Fusion Network for Fact Verification". In: *Proceedings of the ACM Web Conference 2022*.

Cheng, Fei and Yusuke Miyao (2017). "Classifying Temporal Relations by Bidirectional LSTM over Dependency Paths". In: *ACL*.

Chopra, Shivang, Ramit Sawhney, Puneet Mathur, and Rajiv Ratn Shah (2020). "Hindi-english hate speech detection: Author profiling, debiasing, and practical perspectives". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 01, pp. 386–393.

Chung, Hyung Won, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun

Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei (2022). “Scaling Instruction-Finetuned Language Models”. In: *ArXiv* abs/2210.11416.

Clausner, Christian, Stefan Pletschacher, and Apostolos Antonacopoulos (2013). “The significance of reading order in document recognition and its evaluation”. In: *2013 12th International Conference on Document Analysis and Recognition*. IEEE, pp. 688–692.

Cole, Jennifer (2015). “Prosody in context: a review”. In: *Language, Cognition and Neuroscience* 30.1-2, pp. 1–31.

Cong, Jian, Shan Yang, Na Hu, Guangzhi Li, Lei Xie, and Dan Su (2021). “Controllable Context-aware Conversational Speech Synthesis”. In: *arXiv preprint arXiv:2106.10828*.

Conneau, Alexis, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes (2017). “Supervised learning of universal sentence representations from natural language inference data”. In: *arXiv preprint arXiv:1705.02364*.

Craswell, Nick, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees (2020). “Overview of the TREC 2019 deep learning track”. In: *arXiv preprint arXiv:2003.07820*.

Dagan, Ido, Bill Dolan, Bernardo Magnini, and Dan Roth (2010). “Recognizing textual entailment: Rational, evaluation and approaches–erratum”. In: *Natural Language Engineering* 16.1, pp. 105–105.

Dang, Tuan Anh Nguyen, Duc Thanh Hoang, Quang Bach Tran, Chih-Wei Pan, and Thanh Dat Nguyen (2021). “End-to-End Hierarchical Relation Extraction for Generic Form Understanding”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, pp. 5238–5245.

Das, Nilaksh, Duen Horng Chau, Monica Sunkara, Sravan Bodapati, Dhanush Bekal, and Katrin Kirchhoff (2022). “Listen, know and spell: Knowledge-infused subword modeling for improving asr performance of oov named entities”. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Davis, Brian, Bryan Morse, Brian Price, Chris Tensmeyer, and Curtis Wiginton (2021). “Visual FUDGE: Form Understanding via Dynamic Graph Editing”. In: *arXiv preprint arXiv:2105.08194*.

Deka, Biplab, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar (2017). “Rico: A Mobile App Dataset for Building Data-Driven Design Applications”. In: *Proceedings of the 30th Annual Symposium on User Interface Software and Technology*. UIST ’17.

Del Rio, Miguel, Peter Ha, Quinten McNamara, Corey Miller, and Shipra Chandra (2022). “Earnings-22: A Practical Benchmark for Accents in the Wild”. In: *arXiv preprint arXiv:2203.15591*.

Deng, Dan, Haifeng Liu, Xuelong Li, and Deng Cai (2018). “PixelLink: Detecting Scene Text via Instance Segmentation”. In: *ArXiv* abs/1801.01315.

Deng, Jiajun, Zhengyuan Yang, Tianlang Chen, Wen gang Zhou, and Houqiang Li (2021). “TransVG: End-to-End Visual Grounding with Transformers”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1749–1759.

Deng, Jiajun, Zhengyuan Yang, Daqing Liu, Tianlang Chen, Wen gang Zhou, Yanyong Zhang, Houqiang Li, and Wanli Ouyang (2022). “TransVG++: End-to-End Visual Grounding with Language Conditioned Vision Transformer”. In: *ArXiv* abs/2206.06619.

Devlin, J., Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019a). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *NAACL-HLT*.

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

— (June 2019b). "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

Drozdov, Andrew, Shufan Wang, Razieh Rahimi, Andrew McCallum, Hamed Zamani, and Mohit Iyyer (2022). "You can't pick your neighbors, or can you? When and how to rely on retrieval in the kNN-LM". In: *Conference on Empirical Methods in Natural Language Processing*.

Du, Ning and David V Budescu (2007). "Does past volatility affect investors' price forecasts and confidence judgements?" In: *International Journal of Forecasting* 23.3, pp. 497–511.

Ehrmann, Michael and Marcel Fratzscher (2007). "Explaining monetary policy in press conferences". In.

El-Nouby, Alaaeldin, Shikhar Sharma, Hannes Schulz, R. Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W. Taylor (2018). "Keep Drawing It: Iterative language-based image generation and editing". In: *ArXiv* abs/1811.09845.

Eyben, F., K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan, and K. P. Truong (2016). "The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing". In: *IEEE Transactions on Affective Computing* 7.2, pp. 190–202. DOI: 10.1109/TAFFC.2015.2457417.

Fang, Yuwei, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu (2019). "Hierarchical graph network for multi-hop question answering". In: *arXiv preprint arXiv:1911.03631*.

Ganea, Octavian-Eugen, Gary Bécigneul, and Thomas Hofmann (2018). "Hyperbolic neural networks". In: *arXiv preprint arXiv:1805.09112*.

Gao, Hongyang and Shuiwang Ji (2019). "Graph u-nets". In: *international conference on machine learning*. PMLR, pp. 2083–2092.

Gao, Tianyu, Xingcheng Yao, and Danqi Chen (2021). "Simcse: Simple contrastive learning of sentence embeddings". In: *arXiv preprint arXiv:2104.08821*.

Garncarek, Lukasz, Rafal Powalski, Tomasz Stanislawek, Bartosz Topolski, Piotr Halama, Michal P. Turski, and Filip Grali'nski (2021). "LAMBERT: Layout-Aware Language Modeling for Information Extraction". In: *ICDAR*.

Gautam, Akash, Puneet Mathur, Rakesh Gosangi, Debanjan Mahata, Ramit Sawhney, and Rajiv Ratn Shah (2020). "# metooma: Multi-aspect annotations of tweets related to the metoo movement". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 14, pp. 209–216.

Ghoddusi, Hamed, Germán G Creamer, and Nima Rafizadeh (2019). "Machine learning in energy economics and finance: A review". In: *Energy Economics* 81, pp. 709–727.

Gómez-Cram, Roberto and Marco Grotteria (2022). "Real-time price discovery via verbal communication: Method and application to Fedspeak". In: *Journal of Financial Economics* 143.3, pp. 993–1025.

Gorodnichenko, Yuriy, Tho Pham, and Oleksandr Talavera (2021). *The voice of monetary policy*. Tech. rep. National Bureau of Economic Research.

Goyal, Tanya and Greg Durrett (2019). "Embedding time expressions for deep temporal ordering models". In: *ACL*.

Grattarola, Daniele, Daniele Zambon, Filippo Maria Bianchi, and Cesare Alippi (2021). "Understanding Pooling in Graph Neural Networks". In: *ArXiv* abs/2110.05292.

Grave, Edouard, Armand Joulin, and Nicolas Usunier (2017). "Improving Neural Language Models with a Continuous Cache". In.

Gupta, Aditya, Anuj Kumar, Mayank, Vishwa Nath Tripathi, and Sashikala Tapaswi (2007). "Mobile web: web manipulation for small displays using multi-level hierarchy page segmentation". In: *Mobility '07*.

Guu, Kelvin, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang (2020). "Retrieval augmented language model pre-training". In: *International conference on machine learning*. PMLR, pp. 3929–3938.

Ha, Jaekyu, Robert M. Haralick, and Ihsin T. Phillips (1995). "Document page decomposition by the bounding-box project". In: *Proceedings of 3rd International Conference on Document Analysis and Recognition* 2, 1119–1122 vol.2.

Han, Rujun, I-Hung Hsu, Mu Yang, A. Galstyan, R. Weischedel, and Nanyun Peng (2019a). "Deep Structured Neural Network for Event Temporal Relation Extraction". In: *CoNLL*.

Han, Rujun, Mengyue Liang, Bashar Alhafni, and Nanyun Peng (2019b). "Contextualized Word Embeddings Enhanced Event Temporal Relation Extraction for Story Understanding". In: *ArXiv* abs/1904.11942.

Han, Rujun, Qiang Ning, and Nanyun Peng (2019). "Joint Event and Temporal Relation Extraction with Shared Representations and Structured Prediction". In: *EMNLP/IJCNLP*.

Han, Rujun, X. Ren, and Nanyun Peng (2020). "DEER: A Data Efficient Language Model for Event Temporal Reasoning". In: *ArXiv* abs/2012.15283.

Han, Rujun, Yichao Zhou, and Nanyun Peng (2020). "Domain Knowledge Empowered Structured Neural Net for End-to-End Event Temporal Relation Extraction". In: *ArXiv* abs/2009.07373.

Han, Wei, Hui Chen, and Soujanya Poria (Nov. 2021). "Improving Multimodal Fusion with Hierarchical Mutual Information Maximization for Multimodal Sentiment Analysis". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 9180–9192. DOI: 10.18653/v1/2021.emnlp-main.723.

Hanselowski, Andreas, H. Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych (2018). "UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification". In: *ArXiv* abs/1809.01479.

Harley, Adam W, Alex Ufkes, and Konstantinos G Derpanis (2015). "Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval". In: *International Conference on Document Analysis and Recognition (ICDAR)*.

Hatamizadeh, Ali, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu (2022). "Unetr: Transformers for 3d medical image segmentation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 574–584.

He, Dafang, Scott D. Cohen, Brian L. Price, Daniel Kifer, and C. Lee Giles (2017). "Multi-Scale Multi-Task FCN for Semantic Page Segmentation and Table Detection". In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* 01, pp. 254–261.

He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen (2020). "Deberta: Decoding-enhanced bert with disentangled attention". In: *arXiv preprint arXiv:2006.03654*.

Hendrycks, Dan, Collin Burns, Anya Chen, and Spencer Ball (2021). "Cuad: An expert-annotated nlp dataset for legal contract review". In: *arXiv preprint arXiv:2103.06268*.

Hernandez, François, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Esteve (2018). "TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation". In: *Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20*. Springer.

Holzenberger, Nils, Andrew Blair-Stanek, and Benjamin Van Durme (2020). "A dataset for statutory reasoning in tax law entailment and question answering". In: *arXiv preprint arXiv:2005.05257*.

Hong, Teakgyu, DongHyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park (2020). "BROS: A Pre-trained Language Model for Understanding Texts in Document". In.

Hong, Teakgyu, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park (2021). "BROS: A Pre-trained Language Model Focusing on Text and Layout for Better Key Information Extraction from Documents". In: *arXiv preprint arXiv:2108.04539*.

Hono, Yukiya, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda (2021). "Sinsy: A deep neural network-based singing voice synthesis system". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29, pp. 2803–2815.

Hu, Ronghang, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach (2020). "Iterative answer prediction with pointer-augmented multimodal transformers for textvqa". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9992–10002.

Hwang, Sung-Woong and Joon-Hyuk Chang (2021). "Document-Level Neural TTS Using Curriculum Learning and Attention Masking". In: *IEEE Access* 9, pp. 8954–8960.

Hwang, Wonseok, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo (2020). "Spatial Dependency Parsing for Semi-Structured Document Information Extraction". In: *arXiv preprint arXiv:2005.00642*.

Irie, Kazuki, Albert Zeyer, Ralf Schlüter, and Hermann Ney (2019a). "Language Modeling with Deep Transformers". In: *Interspeech*.

— (2019b). "Training Language Models for Long-Span Cross-Sentence Evaluation". In: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 419–426.

Iyyer, Mohit, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III (2015). "Deep unordered composition rivals syntactic methods for text classification". In: *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pp. 1681–1691.

Izacard, Gautier, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave (2021). "Unsupervised Dense Information Retrieval with Contrastive Learning". In: *Trans. Mach. Learn. Res.* 2022.

Izacard, Gautier and Edouard Grave (2020). "Leveraging passage retrieval with generative models for open domain question answering". In: *arXiv preprint arXiv:2007.01282*.

— (2021). "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering". In: *EACL*.

Jacquier, Eric, Nicholas G Polson, and Peter E Rossi (2002). "Bayesian analysis of stochastic volatility models". In: *Journal of Business & Economic Statistics* 20.1, pp. 69–87.

Jang, Joel, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo (2022). "Towards Continual Knowledge Learning of Language

Models". In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jaume, Guillaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran (2019). "Funsd: A dataset for form understanding in noisy scanned documents". In: *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*. Vol. 2. IEEE, pp. 1–6.

Jembu Rajkumar, A, JB Jordan, and J Lazar (2020). "Improving PDF accessibility tools for content developers: looking towards the future". In: *Designing for Inclusion: Inclusive Design: Looking Towards the Future*. Springer, pp. 173–181.

Jembu Rajkumar, Aravind, Jonathan Lazar, J Bern Jordan, Alireza Darvishy, and Hans-Peter Hutter (2020). "PDF accessibility of research papers: What tools are needed for assessment and remediation?" In.

Ji, Yangfeng and Jacob Eisenstein (2014). "Representation Learning for Text-level Discourse Parsing". In: *ACL*.

Jiang, Kelvin, Ronak Pradeep, Jimmy J. Lin, and David R. Cheriton (2021a). "Exploring Listwise Evidence Reasoning with T5 for Fact Verification". In: *ACL*.

Jiang, Weiwei (2021). "Applications of deep learning in stock market prediction: recent progress". In: *Expert Systems with Applications* 184, p. 115537.

Jiang, Wentao, Ning Xu, Jia-Yeh Wang, Chen Gao, Jing Shi, Zhe L. Lin, and Sishuo Liu (2021b). "Language-Guided Global Image Editing via Cross-Modal Cyclic Mechanism". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2095–2104.

Jiang, Yuming, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu (2021c). "Talk-to-Edit: Fine-Grained Facial Editing via Dialog". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13779–13788.

Jin, Woojeong, Rahul Khanna, Suji Kim, Dong-Ho Lee, Fred Morstatter, Aram Galstyan, and Xiang Ren (2020). "ForecastQA: A Question Answering Challenge for Event Forecasting with Temporal Text Data". In: *arXiv preprint arXiv:2005.00792*.

Kamruzzaman, Joarder and Ruhul A Sarker (2003). "Forecasting of currency exchange rates using ANN: A case study". In: *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*. Vol. 1. IEEE, pp. 793–797.

Kang, Xiaomian, Yang Zhao, Jiajun Zhang, and Chengqing Zong (2020). "Dynamic Context Selection for Document-level Neural Machine Translation via Reinforcement Learning". In: *ArXiv* abs/2010.04314.

Karpukhin, Vladimir, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih (2020). "Dense Passage Retrieval for Open-Domain Question Answering". In: *Conference on Empirical Methods in Natural Language Processing*.

Khandelwal, Urvashi, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis (2020). "Generalization through Memorization: Nearest Neighbor Language Models". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Khemakhem, Mohamed, Axel Herold, and Laurent Romary (2018). "Enhancing Usability for Automatically Structuring Digitised Dictionaries". In.

Kim, Hyounghun, Doo Soon Kim, Seunghyun Yoon, Franck Dernoncourt, Trung Bui, and Mohit Bansal (2022). "CAISE: Conversational Agent for Image Search and Editing". In: *AAAI*.

Kim, Raehyun, Chan Ho So, Minbyul Jeong, Sanghoon Lee, Jinkyu Kim, and Jaewoo Kang (2019). "Hats: A hierarchical graph attention network for stock movement prediction". In: *arXiv preprint arXiv:1908.07999*.

Kirkpatrick, David G and John D Radke (1985). "A framework for computational morphology". In: *Machine Intelligence and Pattern Recognition*. Vol. 2. Elsevier, pp. 217–248.

Klimkov, Viacheslav, Adam Nadolski, Alexis Moinet, Bartosz Putrycz, Roberto Barra-Chicote, Thomas Merritt, and Thomas Drugman (2017). "Phrase Break Prediction for Long-Form Reading TTS: Exploiting Text Structure Information." In: *Interspeech*, pp. 1064–1068.

Kogan, Shimon, Dimitry Levin, Bryan R Routledge, Jacob S Sagi, and Noah A Smith (2009a). "Predicting risk from financial reports with regression". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 272–280.

Kogan, Shimon, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith (2009b). "Predicting Risk from Financial Reports with Regression". In: *NAACL*.

Kolomiyets, Oleksandr, Steven Bethard, and Marie Francine Moens (2012). "Extracting narrative timelines as temporal dependency structures". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 88–97.

Kong, Yajing, Liu Liu, Jun Wang, and Dacheng Tao (2021). "Adaptive Curriculum Learning". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5047–5056.

Koreeda, Yuta and Christopher Manning (Nov. 2021). "ContractNLI: A Dataset for Document-level Natural Language Inference for Contracts". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 1907–1919. DOI: 10.18653/v1/2021.findings-emnlp.164.

Kraaij, Wessel, Thomas Hain, Mike Lincoln, and Wilfried Post (2005). "The AMI meeting corpus". In.

Kudo, Taku and John Richardson (2018). "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing". In: *arXiv preprint arXiv:1808.06226*.

Lebourgeois, Franck, Zbigniew Bublinski, and Hubert Emptoz (1992). "A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents". In: *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*, pp. 272–276.

Lee, Chen-Yu, Chun-Liang Li, Chu Wang, Renshen Wang, Yasuhisa Fujii, Siyang Qin, Ashok Popat, and Tomas Pfister (Aug. 2021). "ROPE: Reading Order Equivariant Positional Encoding for Graph-based Document Information Extraction". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, pp. 314–321. DOI: 10.18653/v1/2021.acl-short.41.

Lee, Junhyun, Inyeop Lee, and Jaewoo Kang (2019). "Self-attention graph pooling". In: *International conference on machine learning*. PMLR, pp. 3734–3743.

Lee, Kenton, Luheng He, Mike Lewis, and Luke Zettlemoyer (Sept. 2017). "End-to-end Neural Coreference Resolution". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 188–197. DOI: 10.18653/v1/D17-1018.

Leeuwenberg, A. and Marie-Francine Moens (2018). "Temporal Information Extraction by Predicting Relative Time-lines". In: *ArXiv* abs/1808.09401.

Leeuwenberg, A. and Marie-Francine Moens (2019). "A Survey on Temporal Reasoning for Temporal Information Extraction from Text". In: *ArXiv* abs/2005.06527.

Lewellen, Katharina (2006). "Financing decisions when managers are risk averse". In: *Journal of Financial Economics* 82.3, pp. 551–589.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (2020a). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *ACL*.

— (2020b). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *ACL*.

Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. (2020c). "Retrieval-augmented generation for knowledge-intensive nlp tasks". In: *Advances in Neural Information Processing Systems* 33, pp. 9459–9474.

Li, Bofang, Tao Liu, Zhe Zhao, Puwei Wang, and Xiaoyong Du (2017). "Neural bag-of-ngrams". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1.

Li, Bowen, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H. S. Torr (2020a). "ManiGAN: Text-Guided Image Manipulation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7877–7886.

Li, Jiazheng, Linyi Yang, Barry Smyth, and Ruihai Dong (2020b). "MAEC: A Multimodal Aligned Earnings Conference Call Dataset for Financial Risk Prediction". In: *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*. CIKM '20. Virtual Event, Ireland: Association for Computing Machinery, 3063–3070. ISBN: 9781450368599. DOI: 10.1145/3340531.3412879.

Li, Junlong, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei (2022). "Dit: Self-supervised pre-training for document image transformer". In: *ACM Multimedia 2022*.

Li, K., Curtis Wigington, Chris Tensmeyer, Handong Zhao, Nikolaos Barmpalios, Vlad I. Morariu, Varun Manjunatha, Tong Sun, and Yun Raymond Fu (2020c). "Cross-Domain Document Object Detection: Benchmark Suite and Method". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12912–12921.

Li, Kai, Curtis Wigington, Chris Tensmeyer, Handong Zhao, Nikolaos Barmpalios, Vlad I Morariu, Varun Manjunatha, Tong Sun, and Yun Fu (2020d). "Cross-domain document object detection: Benchmark suite and method". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12915–12924.

Li, Ke, Zhe Liu, Tianxing He, Hongzhao Huang, Fuchun Peng, Daniel Povey, and Sanjeev Khudanpur (2020e). "An empirical study of transformer-based neural language model adaptation". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 7934–7938.

Li, Ke, Daniel Povey, and Sanjeev Khudanpur (2020). "Neural Language Modeling with Implicit Cache Pointers". In: *Interspeech*.

Li, Liangcheng, Feiyu Gao, Jiajun Bu, Yongpan Wang, Zhi Yu, and Qi Zheng (2020f). "An End-to-End OCR Text Re-organization Sequence Learning for Rich-Text Detail Image Comprehension". In: *ECCV*.

Li, Minghao, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou (Dec. 2020g). "DocBank: A Benchmark Dataset for Document Layout Analysis". In: *Proceedings*

of the 28th International Conference on Computational Linguistics. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 949–960. DOI: 10.18653/v1/2020.coling-main.82.

Li, Naihan, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu (2019). "Neural speech synthesis with transformer network". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 6706–6713.

Li, Ronghan, Lifang Wang, Shengli Wang, and Zejun Jiang (2021a). "Asynchronous Multi-grained Graph Network For Interpretable Multi-hop Reading Comprehension." In: *IJCAI*, pp. 3857–3863.

Li, Yelin, Junjie Wu, and Hui Bu (2016). "When quantitative trading meets machine learning: A pilot survey". In: *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*. IEEE, pp. 1–6.

Li, Yulin, Yuxi Qian, Yuechen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding (2021b). "StrucTexT: Structured Text Understanding with Multi-Modal Transformers". In: *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1912–1920.

Liao, Minghui, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu (2017). "TextBoxes: A Fast Text Detector with a Single Deep Neural Network". In: *AAAI*.

Lin, Chin-Yew (2004). "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*, pp. 74–81.

Lin, Hongbin and Xianghua Fu (2022). "Heterogeneous-Graph Reasoning and Fine-Grained Aggregation for Fact Checking". In: *Proceedings of the Fifth Fact Extraction and VERification Workshop (FEVER)*.

Lin, Tzu-Hsiang, Trung Bui, Doo Soon Kim, and Jean Oh (2020a). "A Multimodal Dialogue System for Conversational Image Editing". In: *ArXiv* abs/2002.06484.

Lin, Tzu-Hsiang, Alexander Rudnicky, Trung Bui, Doo Soon Kim, and Jean Oh (2020b). "Adjusting image attributes of localized regions with low-level dialogue". In: *arXiv preprint arXiv:2002.04678*.

Lin, Tzu-Hsiang, Alexander I. Rudnicky, Trung Bui, Doo Soon Kim, and Jean Oh (2020c). "Adjusting Image Attributes of Localized Regions with Low-level Dialogue". In: *LREC*.

Lin, Yuxiao, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu (2021). "BertGCN: Transductive Text Classification by Combining GCN and BERT". In: *arXiv preprint arXiv:2105.05727*.

Lioma, C., Birger Larsen, and Wei Lu (2012). "Rhetorical relations for information retrieval". In: *ArXiv* abs/1704.01599.

Liu, Hanmeng, Leyang Cui, Jian Liu, and Yue Zhang (2021a). "Natural language inference in context-investigating contextual reasoning over long texts". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 15, pp. 13388–13396.

Liu, Qi, Dani Yogatama, and Phil Blunsom (2022). "Relational Memory-Augmented Language Models". In: *Transactions of the Association for Computational Linguistics* 10, pp. 555–572.

Liu, Y., Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019a). "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *ArXiv* abs/1907.11692.

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

Liu, Ze, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo (2021b). "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022.

Liu, Zhenghao, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu (2019c). "Fine-grained fact verification with kernel graph attention network". In: *arXiv preprint arXiv:1910.09796*.

Liu, Zhengyuan, Ke Shi, and Nancy F Chen (2021). "Coreference-Aware Dialogue Summarization". In: *arXiv preprint arXiv:2106.08556*.

Luo, Rui, Weinan Zhang, Xiaojun Xu, and Jun Wang (2018). "A neural stochastic volatility model". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Ma, Mingyu Derek, J. Sun, M. Yang, Kung-Hsiang Huang, N. Wen, Shikhar Singh, Rujun Han, and Nanyun Peng (2021). "EventPlus: A Temporal Event Understanding Pipeline". In: *ArXiv* abs/2101.04922.

Malerba, Donato, Michelangelo Ceci, and Margherita Berardi (2008). "Machine Learning for Reading Order Detection in Document Image Understanding". In: *Machine Learning in Document Analysis and Recognition*.

Mann, W. (1987). "RHETORICAL STRUCTURE THEORY: A THEORY OF TEXT ORGANIZATION". In.

Mann, W. and S. A. Thompson (1988). "Rhetorical Structure Theory: Toward a functional theory of text organization". In: *Text Talk* 8, pp. 243 –281.

Manuvinakurike, Ramesh, Jacqueline Brixey, Trung Bui, Walter Chang, Ron Artstein, and Kallirroi Georgila (2018a). "Dialedit: Annotations for spoken conversational image editing". In: *Proceedings 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pp. 1–9.

Manuvinakurike, Ramesh Radhakrishna, Jacqueline Brixey, Trung Bui, W. Chang, Doo Soon Kim, Ron Artstein, and Kallirroi Georgila (2018b). "Edit me: A Corpus and a Framework for Understanding Natural Language Image Editing". In: *LREC*.

Manuvinakurike, Ramesh Radhakrishna, Trung Bui, W. Chang, and Kallirroi Georgila (2018c). "Conversational Image Editing: Incremental Intent Identification in a New Dialogue Task". In: *SIGDIAL Conference*.

Marchal, Alexis (2021). "Risk and Returns Around FOMC Press Conferences: A Novel Perspective from Computer Vision". In: *Proceedings of SAI Intelligent Systems Conference*. Springer, pp. 724–735.

Mathur, Puneet, Meghna Ayyar, Sahil Chopra, Simra Shahid, Laiba Mehnaz, and Rajiv Shah (2018a). "Identification of emergency blood donation request on twitter". In: *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task*, pp. 27–31.

Mathur, Puneet, Franck Dernoncourt, Quan Hung Tran, Jiuxiang Gu, Ani Nenkova, Vlad Morariu, Rajiv Jain, and Dinesh Manocha (2022a). "DocLayoutTTS: Dataset and Baselines for Layout-informed Document-level Neural Speech Synthesis". In: *Proc. Interspeech 2022*, pp. 451–455.

Mathur, Puneet, Mihir Goyal, Ramit Sawhney, Ritik Mathur, Jochen Leidner, Franck Dernoncourt, and Dinesh Manocha (2022b). "DocFin: Multimodal Financial Prediction and Bias Mitigation using Semi-structured Documents". In: *Proceedings of the Findings of 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mathur, Puneet, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha (Aug. 2021a). "TIMERS: Document-level Temporal Relation Extraction". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, pp. 524–533. DOI: 10.18653/v1/2021.acl-short.67.

— (2021b). "TIMERS: Document-level Temporal Relation Extraction". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 524–533.

Mathur, Puneet, Gautam Kunapuli, Riyaz Ahmad Bhat, Manish Shrivastava, Dinesh Manocha, and Maneesh Singh (2022c). "DocInfer: Document-level Natural Language Inference using Optimal Evidence Selection". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mathur, Puneet, Vlad Morariu, Verena Kaynig-Fittkau, Jiuxiang Gu, Franck Dernoncourt, Quan Hung Tran, Ani Nenkova, Dinesh Manocha, and Rajiv Jain (2022d). "DocTime: A Document-level Temporal Dependency Graph Parser". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 993–1009.

Mathur, Puneet, Ramit Sawhney, Shivang Chopra, Maitree Leekha, and Rajiv Ratn Shah (2020). "Utilizing temporal psycholinguistic cues for suicidal intent estimation". In: *European Conference on Information Retrieval*. Springer, pp. 265–271.

Mathur, Puneet, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata (2018b). "Detecting offensive tweets in hindi-english code-switched language". In: *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pp. 18–26.

Maynez, Joshua, Shashi Narayan, Bernd Bohnet, and Ryan McDonald (July 2020). "On Faithfulness and Factuality in Abstractive Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 1906–1919.

Meng, Yuanliang, Anna Rumshisky, and Alexey Romanov (2017). "Temporal Information Extraction for Question Answering Using Syntactic Dependencies in an LSTM-based Architecture". In: *ArXiv* abs/1703.05851.

Merity, Stephen, Caiming Xiong, James Bradbury, and Richard Socher (2017). "Pointer Sentinel Mixture Models". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Mikolov, Tomas, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur (2010). "Recurrent neural network based language model." In: *Interspeech*. Vol. 2. 3. Makuhari, pp. 1045–1048.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119.

Mishra, Rohan, Pradyumn Prakhar Sinha, Ramit Sawhney, Debanjan Mahata, Puneet Mathur, and Rajiv Ratn Shah (2019). "SNAP-BATNET: Cascading author profiling and social network graphs for suicide ideation detection on social media". In: *Proceedings of the 2019 conference of the North American Chapter of the association for computational linguistics: student research workshop*, pp. 147–156.

Mittal, Trisha, Puneet Mathur, Aniket Bera, and Dinesh Manocha (2021). "Affect2mm: Affective analysis of multimedia content using emotion causality". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5661–5671.

Mokady, Ron, Amir Hertz, Amit H. Bermano, and . (2021). "ClipCap: CLIP Prefix for Image Captioning". In: *ArXiv* abs/2111.09734.

Montacié, Claude and Marie-José Caraty (2018). "Vocalic, Lexical and Prosodic Cues for the INTERSPEECH 2018 Self-Assessed Affect Challenge". In: *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*. Ed. by B. Yegnanarayana. ISCA, pp. 541–545. DOI: 10.21437/Interspeech.2018-1331.

Murray, Tom (1999). "Authoring Intelligent Tutoring Systems: An analysis of the state of the art". In.

Naik, Aakanksha, Luke Breitfeller, and C. Rosé (2019). "TDDiscourse: A Dataset for Discourse-Level Temporal Ordering of Events". In: *SIGdial*.

Neumann, A. (2015). "discoursegraphs: A graph-based merging tool and converter for multilayer annotated corpora". In: *NODALIDA*.

Ngo, Nghia Trung, Tuan Ngo Nguyen, and Thien Huu Nguyen (2020). "Learning to select important context words for event detection". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 756–768.

Nguyen, Ha-Thanh, Manh-Kien Phi, Xuan-Bach Ngo, Vu Tran, Le-Minh Nguyen, and Minh-Phuong Tu (2022). "Attentive deep neural networks for legal document retrieval". In: *Artificial Intelligence and Law*, pp. 1–30.

Nguyen, Tri, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng (2016). "MS MARCO: A human generated machine reading comprehension dataset". In: *choice* 2640, p. 660.

Nie, Yixin, Haonan Chen, and Mohit Bansal (2019). "Combining fact extraction and verification with neural semantic matching networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 6859–6866.

Nikou, Mahla, Gholamreza Mansourfar, and Jamshid Bagherzadeh (2019). "Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms". In: *Intelligent Systems in Accounting, Finance and Management* 26.4, pp. 164–174.

Ning, Qiang, Z. Feng, and D. Roth (2017). "A Structured Learning Approach to Temporal Relation Extraction". In: *EMNLP*.

Ning, Qiang, Sanjay Subramanian, and D. Roth (2019a). "An Improved Neural Baseline for Temporal Relation Extraction". In: *EMNLP/IJCNLP*.

Ning, Qiang, Sanjay Subramanian, and Dan Roth (2019b). "An improved neural baseline for temporal relation extraction". In: *arXiv preprint arXiv:1909.00429*.

Ning, Qiang, H. Wu, and D. Roth (2018a). "A Multi-Axis Annotation Scheme for Event Temporal Relations". In: *ArXiv* abs/1804.07828.

Ning, Qiang, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth (2020a). "TORQUE: A Reading Comprehension Dataset of Temporal Ordering Questions". In: *EMNLP*.

— (2020b). "TORQUE: A reading comprehension dataset of temporal ordering questions". In: *arXiv preprint arXiv:2005.00242*.

Ning, Qiang, Hao Wu, and Dan Roth (July 2018b). "A Multi-Axis Annotation Scheme for Event Temporal Relations". In: *ACL*.

Ning, Qiang, Ben Zhou, Z. Feng, H. Peng, and D. Roth (2018). "CogCompTime: A Tool for Understanding Time in Natural Language". In: *ArXiv* abs/1906.04940.

Noh, Yunseok, Yong-Min Shin, Junmo Park, A.-Yeong Kim, Su Jeong Choi, Hyun-Je Song, Seongbae Park, and Seyoung Park (2020). "WIRE: An Automated Report Generation System using Topical and Temporal Summarization". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Panayotov, Vassil, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur (2015). "Librispeech: An ASR corpus based on public domain audio books". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. DOI: 10.1109/ICASSP.2015.7178964.

Peng, Xiaobao, Zhijian Yin, and Zhen Yang (2020). "Deeplab_v3_plus-net for Image Semantic Segmentation with Channel Compression". In: *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, pp. 1320–1324.

Peng, Yaohao, Pedro Henrique Melo Albuquerque, Jader Martins Camboim de Sá, Ana Julia Akaishi Padula, and Mariana Rosa Montenegro (2018). "The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression". In: *Expert Systems with Applications* 97, pp. 177–192.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

Pérez-Rosas, Verónica, Mohamed Abouelenien, Rada Mihalcea, Yao Xiao, CJ Linton, and Mihai Burzo (2015). "Verbal and nonverbal clues for real-life deception detection". In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 2336–2346.

Petroni, Fabio, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller (2019). "Language Models as Knowledge Bases?" 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, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, pp. 2463–2473.

Poria, Soujanya, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency (2017). "Context-dependent sentiment analysis in user-generated videos". In: *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 873–883.

Powalski, Rafal, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michal Pietruszka, and Gabriela Pałka (2021). "Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer". In: *ICDAR*.

Pradhan, Debashish, Tripti Rajput, Aravind Jembu Rajkumar, Jonathan Lazar, Rajiv Jain, Vlad I Morariu, and Varun Manjunatha (2022). "Development and Evaluation of a Tool for Assisting Content Creators in Making PDF Files More Accessible". In: *ACM Transactions on Accessible Computing (TACCESS)* 15.1, pp. 1–52.

Pustejovsky, J., José M. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, G. Katz, and Dragomir R. Radev (2003a). "TimeML: Robust Specification of Event and Temporal Expressions in Text". In: *New Directions in Question Answering*.

Pustejovsky, James, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. (2003b). "The timebank corpus". In: *Corpus linguistics*. Vol. 2003. Lancaster, UK., p. 40.

Pustejovsky, James, Patrick Hanks, Roser Saurí, Andrew See, Rob Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo (Jan. 2003c). "The TimeBank corpus". In: *Proceedings of Corpus Linguistics*.

Pustejovsky, James and Amber Stubbs (2011). "Increasing informativeness in temporal annotation". In: *Proceedings of the 5th Linguistic Annotation Workshop*, pp. 152–160.

Qin, Lianhui, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui (2021). "TIMEDIAL: Temporal Commonsense Reasoning in Dialog". In: *arXiv preprint arXiv:2106.04571*.

Qin, Yu and Yi Yang (July 2019a). "What You Say and How You Say It Matters: Predicting Stock Volatility Using Verbal and Vocal Cues". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 390–401. DOI: 10.18653/v1/P19-1038.

— (July 2019b). "What You Say and How You Say It Matters: Predicting Stock Volatility Using Verbal and Vocal Cues". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 390–401. DOI: 10.18653/v1/P19-1038.

Rabelo, Juliano, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh (2020). "COLIEE 2020: methods for legal document retrieval and entailment". In: *JSAI International Symposium on Artificial Intelligence*. Springer, pp. 196–210.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. (2021). "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR, pp. 8748–8763.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu (2019a). "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *arXiv preprint arXiv:1910.10683*.

— (2019b). "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *arXiv preprint arXiv:1910.10683*.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer." In: *J. Mach. Learn. Res.* 21.140, pp. 1–67.

Ram, Ori, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham (2023). "In-Context Retrieval-Augmented Language Models". In: *arXiv preprint arXiv:2302.00083*.

Ramachandran, Harish Gandhi and Dan DeRose Jr (2018). "A text analysis of federal reserve meeting minutes". In: *arXiv preprint arXiv:1805.07851*.

Rand, W. M. (1971). "Objective Criteria for the Evaluation of Clustering Methods". In: *Journal of the American Statistical Association* 66, pp. 846–850.

Reimers, Nils, N. Dehghani, and Iryna Gurevych (2016). "Temporal Anchoring of Events for the TimeBank Corpus". In: *ACL*.

Reimers, Nils and Iryna Gurevych (2019). "Sentence-bert: Sentence embeddings using siamese bert-networks". In: *arXiv preprint arXiv:1908.10084*.

Ren, Shaoqing, Kaiming He, Ross B. Girshick, and Jian Sun (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, pp. 1137–1149.

Rezatofighi, Hamid, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese (2019). "Generalized intersection over union: A metric and a loss for bounding box regression". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 658–666.

Richardson, Matthew, Christopher JC Burges, and Erin Renshaw (2013). "Mctest: A challenge dataset for the open-domain machine comprehension of text". In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 193–203.

Rio, Miguel Del, Natalie Delworth, Ryan Westerman, Michelle Huang, Nishchal Bhandari, Joseph Palakapilly, Quinten McNamara, Joshua Dong, Piotr Zelasko, and Miguel Jette (2021). *Earnings-21: A Practical Benchmark for ASR in the Wild*. arXiv: 2104.11348 [cs.CL].

Rosa, Carlo (2013). "The financial market effect of FOMC minutes". In: *Available at SSRN 2378398*.

Ross, Hayley, Jonathon Cai, and Bonan Min (2020). "Exploring Contextualized Neural Language Models for Temporal Dependency Parsing". In: *arXiv preprint arXiv:2004.14577*.

Rothe, Sascha, Shashi Narayan, and Aliaksei Severyn (2020). "Leveraging Pre-trained Checkpoints for Sequence Generation Tasks". In: *Transactions of the Association for Computational Linguistics* 8, pp. 264–280.

Rundo, Francesco, Francesca Trenta, Agatino Luigi di Stallo, and Sebastiano Battiato (2019). "Machine learning for quantitative finance applications: A survey". In: *Applied Sciences* 9.24, p. 5574.

Sadka, Ronnie (2006). "Momentum and post-earnings-announcement drift anomalies: The role of liquidity risk". In: *Journal of Financial Economics* 80.2, pp. 309–349.

Sakoe, Hiroaki and Seibi Chiba (1978a). "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE transactions on acoustics, speech, and signal processing* 26.1, pp. 43–49.

— (1978b). "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, pp. 159–165.

Salemi, Alireza, Sheshera Mysore, Michael Bendersky, and Hamed Zamani (2023). "LaMP: When Large Language Models Meet Personalization". In: *arXiv preprint arXiv:2304.11406*.

Sancheti, Abhilasha, Aparna Garimella, Balaji Vasan Srinivasan, and Rachel Rudinger (2022). "What to Read in a Contract? Party-Specific Summarization of Important Obligations, Entitlements, and Prohibitions in Legal Documents". In: *arXiv preprint arXiv:2212.09825*.

Sansone, Carlo and Giancarlo Sperlí (2022). "Legal Information Retrieval systems: State-of-the-art and open issues". In: *Information Systems* 106, p. 101967.

Sawhney, Ramit, Arshiya Aggarwal, and Rajiv Ratn Shah (June 2021). "An Empirical Investigation of Bias in the Multimodal Analysis of Financial Earnings Calls". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 3751–3757.

Sawhney, Ramit, Mihir Goyal, Prakhar Goel, Puneet Mathur, and Rajiv Shah (2021a). "Multimodal Multi-Speaker Merger & Acquisition Financial Modeling: A New Task, Dataset, and Neural Baselines". In: *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 6751–6762.

Sawhney, Ramit, Piyush Khanna, Arshiya Aggarwal, Taru Jain, Puneet Mathur, and Rajiv Ratn Shah (2020a). "VolTAGE: Volatility Forecasting via Text-Audio Fusion with Graph Convolution Networks for Earnings Calls". In: *EMNLP*.

Sawhney, Ramit, Puneet Mathur, Taru Jain, Akash Kumar Gautam, and Rajiv Shah (2021b). "Multitask Learning for Emotionally Analyzing Sexual Abuse Disclosures". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4881–4892.

Sawhney, Ramit, Puneet Mathur, Ayush Mangal, Piyush Khanna, Rajiv Ratn Shah, and Roger Zimmermann (2020b). "Multimodal multi-task financial risk forecasting". In: *Proceedings of the 28th ACM international conference on multimedia*, pp. 456–465.

Sawhney, Ramit, Arnav Wadhwa, Shivam Agarwal, and Rajiv Shah (2021c). "FAST: Financial News and Tweet Based Time Aware Network for Stock Trading". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 2164–2175.

Schick, Timo and Hinrich Schütze (2019). "Rare Words: A Major Problem for Contextualized Embeddings And How to Fix it by Attentive Mimicking". In: *AAAI Conference on Artificial Intelligence*.

Schlichtkrull, M., Thomas Kipf, P. Bloem, R. V. Berg, Ivan Titov, and M. Welling (2018). "Modeling Relational Data with Graph Convolutional Networks". In: *ArXiv* abs/1703.06103.

Schumann, Gerrit, Katharina Meyer, and Jorge Marx Gomez (2022). "Query-Based Retrieval of German Regulatory Documents for Internal Auditing Purposes". In: *2022 5th International Conference on Data Science and Information Technology (DSIT)*. IEEE, pp. 01–10.

Sen, Umut Mehmet, Veronica Perez-Rosas, Berrin Yanikoglu, Mohamed Abouelenien, Mihai Burzo, and Rada Mihalcea (2020). "Multimodal deception detection using real-life trial data". In: *IEEE Transactions on Affective Computing*.

Serai, Prashant, Vishal Sunder, and Eric Fosler-Lussier (2022). "Hallucination of speech recognition errors with sequence to sequence learning". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30, pp. 890–900.

Shapiro, Adam Hale and Daniel Wilson (2021). "Taking the fed at its word: A new approach to estimating central bank objectives using text analysis". In: Federal Reserve Bank of San Francisco.

Shen, Jonathan, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. (2018). "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions". In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp. 4779–4783.

Shi, Jing, Ning Xu, Trung Bui, Franck Dernoncourt, Zheng Wen, and Chenliang Xu (2020). "A Benchmark and Baseline for Language-Driven Image Editing". In: *Proceedings of the Asian Conference on Computer Vision*.

Shi, Jing, Ning Xu, Yihang Xu, Trung Bui, Franck Dernoncourt, and Chenliang Xu (2021a). "Learning by Planning: Language-Guided Global Image Editing". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13590–13599.

Shi, Jing, Ning Xu, Haitian Zheng, Alex Smith, Jiebo Luo, and Chenliang Xu (2022). "SpaceEdit: Learning a Unified Editing Space for Open-Domain Image Color Editing". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19730–19739.

Shi, Peng and Jimmy Lin (2019). "Simple BERT Models for Relation Extraction and Semantic Role Labeling". In: *ArXiv* abs/1904.05255.

Shi, Yangyang, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer (2021b). "Emformer: Efficient Memory Transformer Based Acoustic Model for Low Latency Streaming Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*. IEEE.

Si, Jiasheng, Deyu Zhou, Tongzhe Li, Xingyu Shi, and Yulan He (Aug. 2021). "Topic-Aware Evidence Reasoning and Stance-Aware Aggregation for Fact Verification". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics.

Simon, Anikó, Jean-Christophe Pret, and A. Peter Johnson (1997). "A Fast Algorithm for Bottom-Up Document Layout Analysis". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 19, pp. 273–277.

Singh, Amanpreet, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach (2019). "Towards vqa models that can read". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326.

Soleymani, Mohammad, Maja Pantic, and Thierry Pun (2011). "Multimodal emotion recognition in response to videos". In: *IEEE transactions on affective computing* 3.2, pp. 211–223.

Stenetorp, Pontus, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii (2012). "BRAT: a web-based tool for NLP-assisted text annotation". In: *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 102–107.

Strötgen, Jannik and Michael Gertz (2013). "Multilingual and cross-domain temporal tagging". In: *Language Resources and Evaluation* 47.2, pp. 269–298.

Subramanian, Shyam and Kyumin Lee (2020). "Hierarchical Evidence Set Modeling for Automated Fact Extraction and Verification". In: *EMNLP*.

Sun, G., C. Zhang, and P. C. Woodland (2021). "Transformer Language Models with LSTM-Based Cross-Utterance Information Representation". In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7363–7367.

Tadle, Raul Cruz (2022). "FOMC minutes sentiments and their impact on financial markets". In: *Journal of Economics and Business* 118, p. 106021.

Tarjan, Robert Endre and Anthony E Trojanowski (1977). "Finding a maximum independent set". In: *SIAM Journal on Computing* 6.3, pp. 537–546.

Thorne, James, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal (2018). "Fever: a large-scale dataset for fact extraction and verification". In: *arXiv preprint arXiv:1803.05355*.

Trong, Hieu Man Duc, Nghia Ngo Trung, Linh Van Ngo, and Thien Huu Nguyen (2022). "Selecting Optimal Context Sentences for Event-Event Relation Extraction". In.

Tsai, Yao-Hung Hubert, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov (2019). "Multimodal transformer for unaligned multimodal language sequences". In: *Proceedings of the conference. Association for Computational Linguistics. Meeting*. Vol. 2019. NIH Public Access, p. 6558.

Vashishtha, Siddharth, Benjamin Van Durme, and A. White (2019). "Fine-Grained Temporal Relation Extraction". In: *ACL*.

Vashishtha, Siddharth, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White (Nov. 2020). "Temporal Reasoning in Natural Language Inference". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 4070–4078. DOI: 10.18653/v1/2020.findings-emnlp.363.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio (2017). "Graph attention networks". In: *arXiv preprint arXiv:1710.10903*.

Veyseh, Amir Pouran Ben, Minh Van Nguyen, Nghia Ngo Trung, Bonan Min, and Thien Huu Nguyen (2021). "Modeling document-level context for event detection via important context selection". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5403–5413.

Walczak, Steven (2001). "An empirical analysis of data requirements for financial forecasting with neural networks". In: *Journal of management information systems* 17.4, pp. 203–222.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman (2018). "GLUE: A multi-task benchmark and analysis platform for natural language understanding". In: *arXiv preprint arXiv:1804.07461*.

Wang, Haoyu, Muhao Chen, Hongming Zhang, and D. Roth (2020a). "Joint Constrained Learning for Event-Event Relation Extraction". In: *EMNLP*.

Wang, Jianan, Guansong Lu, Hang Xu, Zhenguo Li, Chunjing Xu, and Yanwei Fu (2022). "ManiTrans: Entity-Level Text-Guided Image Manipulation via Token-wise Semantic Alignment and Generation". In: *ArXiv* abs/2204.04428.

Wang, Shufan, Yixiao Song, Andrew Drozdov, Aparna Garimella, Varun Manjunatha, and Mohit Iyyer (2023). "KNN-LM Does Not Improve Open-ended Text Generation". In: *ArXiv* abs/2305.14625.

Wang, Yuxuan, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Z. Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Robert A. J. Clark, and Rif A. Saurous (2017). "Tacotron: Towards End-to-End Speech Synthesis". In: *INTERSPEECH*.

Wang, Zilong, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei (Nov. 2021a). "LayoutReader: Pre-training of Text and Layout for Reading Order Detection". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 4735–4744. DOI: 10.18653/v1/2021.emnlp-main.389.

Wang, Zilong, Mingjie Zhan, Xuebo Liu, and Ding Liang (Nov. 2020b). "DocStruct: A Multimodal Method to Extract Hierarchy Structure in Document for General Form Understanding". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 898–908. DOI: 10.18653/v1/2020.findings-emnlp.80.

Wang, Zilong, Mingjie Zhan, Houxing Ren, Zhaohui Hou, Yuwei Wu, Xingyan Zhang, and Ding Liang (2021b). "GroupLink: An End-to-end Multitask Method for Word Grouping and Relation Extraction in Form Understanding". In: *ArXiv* abs/2105.04650.

Weiss, Jerry (2011). *Ekman, P.(2009) Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage. New York: Norton.*

Williams, Adina, Nikita Nangia, and Samuel Bowman (June 2018). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1112–1122. DOI: 10.18653/v1/N18-1101.

Williams, Adina, Nikita Nangia, and Samuel R Bowman (2017). "A broad-coverage challenge corpus for sentence understanding through inference". In: *arXiv preprint arXiv:1704.05426*.

Williams, Ronald J (1992). "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8.3, pp. 229–256.

Williams, Ronald J. and David Zipser (June 1989a). "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". In: *Neural Comput.* 1.2, 270–280. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.2.270.

Williams, Ronald J and David Zipser (1989b). "A learning algorithm for continually running fully recurrent neural networks". In: *Neural computation* 1.2, pp. 270–280.

Wood, Rachel, Emma Dixon, Salma Elsayed-Ali, Ekta Shokeen, Amanda Lazar, and Jonathan Lazar (2023). "Exploring Future Personalization Opportunities in Technologies used by Older Adults with Mild to Moderate Dementia". In.

Wu, Yuxin and Kaiming He (2018). "Group Normalization". In: *International Journal of Computer Vision* 128, pp. 742–755.

Wu, Yuxin, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick (2019). *Detectron2*. https://github.com/facebookresearch/detectron2.

Xu, Hainan, Ke Li, Yiming Wang, Jian Wang, Shiyin Kang, Xie Chen, Daniel Povey, and Sanjeev Khudanpur (2018a). "Neural network language modeling with letter-based features and importance sampling". In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp. 6109–6113.

Xu, Keyulu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka (2018b). "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826*.

Xu, Yang, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei A. F. Florêncio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou (2021). "LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding". In: *ACL/IJCNLP*.

Xu, Yiheng, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou (2020). "Layoutlm: Pre-training of text and layout for document image understanding". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1192–1200.

Xu, Yumo and Shay B. Cohen (July 2018a). "Stock Movement Prediction from Tweets and Historical Prices". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1970–1979. DOI: 10.18653/v1/P18-1183.

Xu, Yumo and Shay B Cohen (2018b). "Stock movement prediction from tweets and historical prices". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1970–1979.

Yang, Linyi, Tin Lok James Ng, Barry Smyth, and Riuhai Dong (2020a). "HTML: Hierarchical Transformer-Based Multi-Task Learning for Volatility Prediction". In: *Proceedings of The Web*

Conference 2020. New York, NY, USA: Association for Computing Machinery, 441–451. ISBN: 9781450370233.

Yang, Xiao, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C. Lee Giles (2017). "Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4342–4351.

Yang, Zhengyuan, Tianlang Chen, Liwei Wang, and Jiebo Luo (2020b). "Improving One-stage Visual Grounding by Recursive Sub-query Construction". In: *ArXiv* abs/2008.01059.

Yao, Jiarui, Haoling Qiu, Bonan Min, and Nianwen Xue (2020). "Annotating Temporal Dependency Graphs via Crowdsourcing". In: *EMNLP*.

Yao, Li, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville (2015). "Describing videos by exploiting temporal structure". In: *Proceedings of the IEEE international conference on computer vision*, pp. 4507–4515.

Yao, Ting, Yingwei Pan, Yehao Li, and Tao Mei (2018). "Exploring visual relationship for image captioning". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 684–699.

Ye, Zhen, Yu Qin, and Wei Xu (2020). "Financial Risk Prediction with Multi-Round Q&A Attention Network." In: *IJCAI*, pp. 4576–4582.

Yin, Wenpeng, Dragomir Radev, and Caiming Xiong (2021). "DocNLI: A Large-scale Dataset for Document-level Natural Language Inference". In: *arXiv preprint arXiv:2106.09449*.

Yu, ShuiLing and Zhe Li (2018). "Forecasting stock price index volatility with LSTM deep neural network". In: *Recent developments in data science and business analytics*. Springer, pp. 265–272.

Zadeh, Amir, Chengfeng Mao, Kelly Shi, Yiwei Zhang, Paul Pu Liang, Soujanya Poria, and Louis-Philippe Morency (2019). "Factorized multimodal transformer for multimodal sequential learning". In: *arXiv preprint arXiv:1911.09826*.

Zaheer, Manzil, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed (2020a). "Big Bird: Transformers for Longer Sequences". In: *ArXiv* abs/2007.14062.

— (2020b). "Big Bird: Transformers for Longer Sequences". In: *ArXiv* abs/2007.14062.

Zaheer, Manzil, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. (2020c). "Big Bird: Transformers for Longer Sequences." In: *NeurIPS*.

Zhang, Min, Feng Li, Yang Wang, Zequn Zhang, Yanhai Zhou, and Xiaoyu Li (2020a). "Coarse and fine granularity graph reasoning for interpretable multi-hop question answering". In: *IEEE Access* 8, pp. 56755–56765.

Zhang, Xiaoyi, Lilian de Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, et al. (2021a). "Screen Recognition: Creating Accessibility Metadata for Mobile Applications from Pixels". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15.

Zhang, Yuchen and Nianwen Xue (2018a). "Neural ranking models for temporal dependency structure parsing". In: *arXiv preprint arXiv:1809.00370*.

— (2018b). "Structured interpretation of temporal relations". In: *arXiv preprint arXiv:1808.07599*.

— (2019). "Acquiring Structured Temporal Representation via Crowdsourcing: A Feasibility Study". In: **SEMEVAL*.

Zhang, Yue, Bo Zhang, Rui Wang, Junjie Cao, Chen Li, and Zuyi Bao (2021b). "Entity Relation Extraction as Dependency Parsing in Visually Rich Documents". In: *arXiv preprint arXiv:2110.09915*.

Zhang, Zhenyu, Bowen Yu, Xiaobo Shu, Tingwen Liu, Hengzhu Tang, Wang Yubin, and Li Guo (2020b). "Document-level Relation Extraction with Dual-tier Heterogeneous Graph". In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1630–1641.

Zhao, Chen, Chenyan Xiong, Corby Rosset, Xia Song, Paul N. Bennett, and Saurabh Tiwary (2020). "Transformer-XH: Multi-Evidence Reasoning with eXtra Hop Attention". In: *ICLR*.

Zhao, Xinyu, Shih ting Lin, and Greg Durrett (2020a). "Effective Distant Supervision for Temporal Relation Extraction". In: *ArXiv* abs/2010.12755.

Zhao, Xinyu, Shih-ting Lin, and Greg Durrett (2020b). "Effective Distant Supervision for Temporal Relation Extraction". In: *arXiv preprint arXiv:2010.12755*.

Zheng, Lucia, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho (2021). "When does pretraining help? assessing self-supervised learning for law and the CaseHOLD dataset of 53,000+ legal holdings". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pp. 159–168.

Zhong, Wanjun, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin (July 2020). "Reasoning Over Semantic-Level Graph for Fact Checking". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.549.

Zhou, Fan, Shengming Zhang, and Yi Yang (2020). "Interpretable operational risk classification with semi-supervised variational autoencoder". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 846–852.

Zhou, Jie, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun (2019). "GEAR: Graph-based evidence aggregating and reasoning for fact verification". In: *arXiv preprint arXiv:1908.01843*.

Zhou, Yichao, Yu Yan, Rujun Han, J. Caufield, Kai-Wei Chang, Y. Sun, P. Ping, and W. Wang (2020). "Clinical Temporal Relation Extraction with Probabilistic Soft Logic Regularization and Global Inference". In: *ArXiv* abs/2012.08790.