Causal Relation Extraction from Text using Transformers

Ana Margarida Dias da Silva
Master in Data Science
Department of Computer Science
Faculty of Sciences of the University of Porto
2025

Causal Relation Extraction from Text using Transformers

Ana Margarida Dias da Silva

Dissertation carried out as part of the Master in Data Science

Department of Computer Science

2025

Supervisor

Alípio Jorge, Full Professor,
Faculty of Sciences of the University of Porto,
Institute for Systems and Computer Engineering, Technology and Science (INESC TEC)

Co-supervisor

Maria Silvano, Assistant Professor,
Faculty of Arts and Humanities of the University of Porto,
Institute for Systems and Computer Engineering, Technology and Science (INESC TEC)

Acknowledgements

First, I would like to express my deepest gratitude to my supervisors, Alípio Jorge and Purificação Silvano, for their guidance, stimulating suggestions and constant encouragement.

I am also sincerely thankful to the StorySense project, within which this thesis was developed, for providing the essential resources and support that made this research possible.

In addition, I would also like to acknowledge the support of the *Fundação para a Ciência e a Tecnologia* (FCT) through access to computational resources, namely the Virtual Artificial Intelligence Laboratory (AIvLAB) on the Deucalion platform, granted under the reference 2025.00012.AIVLAB.DEUCALION.

I am especially grateful to Ricardo for his endless patience, love, and care, and to my parents for their unconditional support.

Finally, thank you to my dear friends for bringing joy, laughter, and balance to this journey.

Abstract

Causality plays a vital role in natural language understanding tasks such as event prediction, information retrieval, summarization, sentiment analysis, and question-answering. This importance is reflected in the fact that a significant portion of queries in search engines and conversational systems like ChatGPT are causal in nature. Despite this, identifying causal relations in natural language remains a challenging and largely unresolved problem. Existing research often focuses on detecting individual event head-words as causes or effects, rather than extracting larger, meaningful causal spans, limiting their applicability in real-world scenarios.

This thesis investigates the automatic extraction of causal relations from English and European Portuguese texts, using three different modeling approaches: a start/end pointer-based architecture and a sequence tagging model, both based on RoBERTa, as well as an instruction-tuned large language model API (Gemini).

Two training setups were explored: Single-relation-per-input, which simplifies the task by focusing on one relation at a time but limits handling of multiple relations; and Multi-relation-per-input, which allows extraction of multiple causal relations within a single input. To improve performance in the more complex multi-relation setup, a task-specific warm-start strategy was employed, where the model is first trained on the simpler single-relation task before being fine-tuned on the multi-relation setup.

The Start/End Pointer-based model, enhanced with post-processing techniques such as duplicate removal, thresholding, and overlap limitation, achieved the highest average ranking based on F1-score in the causal span extraction task.

Future work should focus on improving the distinction between causal and non-causal entries to further enhance extraction accuracy.

Keywords: Causality Extraction, RoBERTa, Pointer Approach, Multi-Relation Extraction, Warm-Start Strategy

Resumo

A causalidade desempenha um papel fundamental em tarefas de compreensão de linguagem natural, como previsão de eventos, extração de informação, sumarização, análise de sentimentos e sistemas de resposta a perguntas. Essa importância reflete-se no facto de que uma grande parte das perguntas em motores de pesquisa e sistemas conversacionais, como o ChatGPT, é de natureza causal. Apesar disso, identificar relações causais em linguagem natural continua a ser desafiador e um problema ainda não resolvido. Grande parte dos estudos existentes centram-se na deteção de palavras-chave de eventos, em vez da extração de segmentos causais mais amplos, o que limita a sua aplicabilidade em cenários do mundo real.

Esta tese investiga a extração automática de relações causais a partir de textos em Inglês e Português Europeu, utilizando três abordagens de aprendizagem distintas: uma arquitetura baseada em ponteiros de posição de início/fim e um modelo de *tagging* de sequência, ambos baseados no modelo RoBERTa, bem com uma API de modelo de linguagem ajustada por instruções (Gemini).

Foram exploradas duas configurações de treino: Uma-relação-por-entrada, que simplifica a tarefa ao focar em uma única relação de cada vez, mas limita o tratamento de múltiplas relações; e Múltiplas-relações-por-entrada, que permite a extração de diversas relações causais por entrada. Para melhorar o desempenho nesta configuração mais complexa, foi usada uma estratégia de inicialização específica para a tarefa, na qual o modelo é inicialmente treinado na tarefa mais simples (uma relação por entrada) antes de ser treinado para prever múltiplas relações.

O modelo baseado em ponteiros de início/fim, com técnicas de pós-processamento como remoção de previsões duplicadas, definição de limites de rejeição e limitação de sobreposição de relações, obteve o melhor ranking, com base no F1-score, na tarefa de extração de segmentos causais.

Trabalhos futuros devem-se focar em melhorar a distinção entre entradas causais e não causais, de forma a aumentar ainda mais a precisão da extração.

Palavras-chave: Extração de Causalidade, RoBERTa, Abordagem Ponteiro, Extração de Relações Múltiplas, Estratégias de Inicialização

Table of Contents

List of Tables

List of Figures

List of Acronyms

ALBERT A Lite BERT. 27

BECauSE Bank of Effects and Causes Stated Explicitly. 20, 35, 38, 39, 66, 68, 69

BERT Bidirectional Encoder Representations from Transformers. v, 12– 14, 16, 18, 20– 24, 29– 32, 48

BiLSTM bidirectional LSTM. 10

BIO Begin-Inside-Outside. 8, 15

CASE Challenges and Applications of Automated Extraction of Socio-political Events from Text. iv, 21, 27, 30, 34, 36, 42, 57, 75

CNC Causal News Corpus. iv, v, viii, 2, 3, 21, 31, 33– 38, 40, 47– 50, 60, 65, 67, 70, 75– 80, 82– 85, 87, 88

CNN Convolutional Neural Network. 5, 10, 12, 20

CRF Conditional Random Field. 7– 10, 14, 31, 57, 59

CTB CausalTimeBank. 36, 39, 66, 68, 70, 103

DL Deep Learning. iv, 5, 7, 10, 11, 20

ESL EventStoryLine. 36, 39, 66, 70

FinCausal Financial Document Causality Detection Shared Task. iv, 20, 21, 23, 25– 27, 29, 36

GAT Graph Attention Network. 10

GCN Graph Convolutional Network. 10, 11

GNN Graph Neural Network. 10, 23

GPT Generative Pre-trained Transformer. 12, 20, 32

GRU Gated Recurrent Unit. 5, 10

LLaMA Large Language Model Meta AI. 20

LLM Large Language Model. iv, 11– 13

LSTM Long Short-Term Memory. 5, 10, 11

ML Machine Learning. iv, 5, 7

MLM Masked Language Model. 12, 13

NER Named Entity Recognition. viii, 8, 10, 12, 14, 31

NLP Natural Language Processing. 1– 3, 5, 7, 8, 10, 11, 13, 20, 32

PDTB Penn Discourse TreeBank. 20, 35, 36, 39, 40, 67, 107

POS Part-Of-Speech. 7, 8, 18, 22– 24

QA Question-Answering. iv, 14, 15, 21, 22, 27, 49, 51

RNN Recurrent Neural Network. 10, 12

RoBERTa Robustly Optimized BERT Pre-Training Approach. 21, 22, 24, 25, 31, 49, 51, 54, 65, 66, 87

SpERT Span-based Entity and Relation Transformer. 16, 27, 29

SQuAD Stanford Question Answering Dataset. 14, 15, 49

TED-MDB TED-Multilingual Discourse Bank. 36, 39

1. Introduction

1.1. Motivation and Context

A causal relation represents a relationship between two events, where the occurrence of a cause results in the occurrence of an effect [1]. Language typically does not consist of isolated, unrelated sentences; rather, it is composed of collocated, structured, and coherent groups of sentences. Such coherent and structured groups of sentences are designated as a discourse, wherein spans of text are interconnected through semantic-pragmatic discourse relations, such as causal and temporal [2, 3].

Natural Language Processing (NLP) is a field of artificial intelligence focused on enabling computers to understand, interpret, and generate human language. Within NLP, causal relation extraction benefits applications such as question-answering, event prediction, information retrieval, summarization, and sentiment analysis [1, 4].

The analysis of a dataset containing question-like queries submitted to Yandex in 2012 revealed that, out of approximately 1.5 billion question-like entries, about 81.7 million (5%) were identified as causal questions. Among these, “why” questions were the most prevalent, followed by “what to do if” questions and “what happens if” questions [5]. More recently, a study that randomly selected questions from various types of natural question sources, including search engines and human conversations with ChatGPT, found that 42% of them were causal [6].

The prediction of event sequences shows important significance in areas such as economic forecasting, disaster management, and stock trading. Furthermore, sentiment analysis detects subjectivity in text, often to gauge public mood on stocks, products, or political figures, and has been used for stock market trading strategies. Traditional sentiment analysis faces limitations due to delays in news publication, but causal-sentimental relations, linking causes to sentiment-driven effects, offer a way to predict future sentiment and extend trading horizons. In addition, text summarization leverages causal relationships to maintain the correct order of cause and effect, enhancing summary coherence and ensuring chronological accuracy [4]. Systems that extract causal relations also play an important role in the medical field, especially when medical insights are guided by the structured knowledge within ontologies. This combination enhances the accuracy and

depth of causal understanding, enabling more reliable insights into complex medical relationships [7].

Nevertheless, despite the critical significance of identifying causality within text, the automatic extraction of causal relationships poses considerable challenges for various reasons. Firstly, the context in which causal relations are expressed is often complex and variable and specific background (world) knowledge is required to ascertain causality. Secondly, the availability of datasets is limited and, in general, their sizes are relatively small. Moreover, researchers often develop their datasets using varying rules, which results in a lack of standardized methods for comparing models across different datasets [8].

Finally, the challenge of modeling causality recognition and representation in NLP-focused applications remains an unresolved issue, primarily due to the lack of fair model performance comparisons and inconsistencies in the creation of datasets [9, 10].

1.2. Objective

The main aim of this thesis is to develop an automated process for extracting causal relations, identifying the cause, effect and signal spans (i.e., linguistic markers such as “because” or “due to” that explicitly indicate a causal relationship). To accomplish this objective, we will explore English datasets and a European Portuguese dataset, automatically extracting information by fine-tuning pre-trained models for the specific task.

The input of such a task typically consists of individual sentences or short text segments (Figure 1.1) from which the system must automatically identify the cause, effect, and signal spans. These examples, drawn from the Causal News Corpus (CNC), illustrate the process of Cause-Effect-Signal Span Detection that forms the core focus of this thesis [8, 11].

This thesis was conducted as part of the StorySense project, which aims to enhance the extraction and analysis of narratives in text. By focusing on the identification of causal relationships, the project seeks to improve the coherence and depth of narrative understanding, ultimately facilitating more nuanced interpretations of stories.

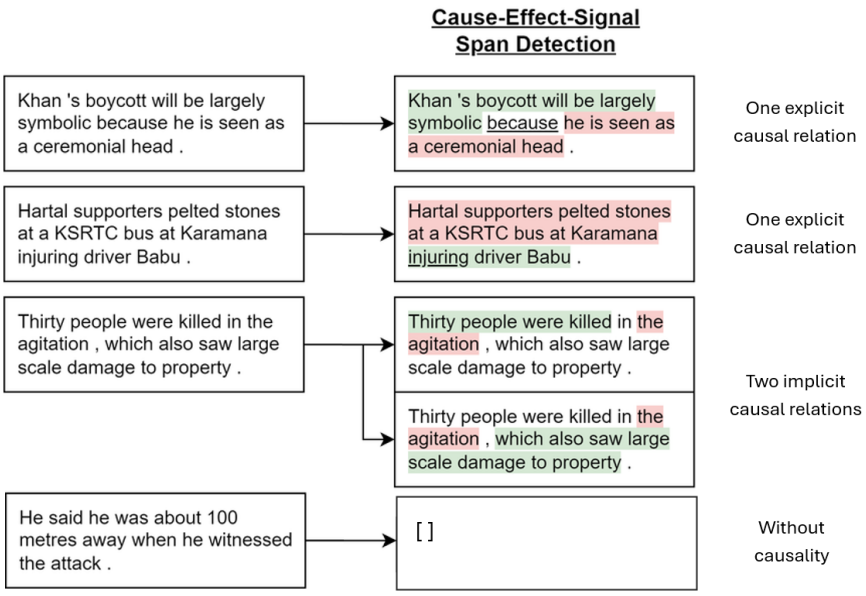


Figure 1.1: Examples from the Causal News Corpus. Cause spans are indicated by Pink, Effect spans are indicated by Green and Signals, if present, are underlined. Adapted from [1].

1.3. Dissertation Structure

This thesis is organized into eight chapters as follows:

- **Chapter 2 - Background:** This chapter provides an extensive review of essential concepts in NLP, with an emphasis on causal relation extraction. It also discusses architectures and techniques relevant to identifying causes, effects, and signal spans in text, laying the foundation for the following chapters.
- **Chapter 3 - Related Work:** This chapter reviews the developments in causal relation extraction, tracing the evolution of approaches over time. State-of-the-art approaches for cause-effect-signal detection, pre-trained models, and fine-tuning strategies are also discussed.
- **Chapter 4 - Data and Methodology for Evaluation:** This chapter presents the datasets used in this study, highlighting the differences between them. It also details the pre-processing steps applied to prepare the data for causal relation extraction. Finally, the methodologies used for evaluating the models are described.
- **Chapter 5 - Causal Relation Extraction on the CNC Dataset:** This chapter focuses on the experimental results obtained when fine-tuning pre-trained models on the CNC dataset. It includes a comparison of different modeling approaches, highlighting performance insights and challenges unique to the CNC dataset.

- **Chapter 6 - Causal Relation Extraction on Benchmark Datasets:** This chapter presents the experimental results obtained by fine-tuning pre-trained models on benchmark datasets for causal relation extraction. It includes cross-dataset comparisons, and an analysis of model generalization, allowing a broader assessment of performance across different data sources.
- **Chapter 7 - Causal vs. Non-Causal Discrimination and Relation Number Calibration:** This chapter focuses on analyzing the model's ability to discriminate between causal and non-causal entries. It also examines the calibration of predicted relation counts, assessing how accurately the models capture the number of causal relations in a given text.
- **Chapter 8 - Conclusions:** The final chapter synthesizes the findings of the research, highlights the contributions to causal relation extraction, and outlines potential directions for future work.

2. Background

This chapter introduces the key theoretical concepts and models that are fundamental to understanding the insights explored in later chapters, covering topics from linguistics and statistics to Deep Learning (DL), with a particular focus on their relevance to causal relation extraction.

Causal relation extraction systems can be broadly categorized into three groups [12]:

- **Knowledge-based approaches**, including rule-based and pattern-based methods, presented in Section 2.1;
- **Statistical Machine Learning (ML)-based approaches**, presented in Section 2.2;
- **DL-based approaches**, encompassing architectures like Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), Graph-based methods, and Pre-trained Models. These are presented in Section 2.3.

Many of the techniques covered in this chapter are widely applied across various NLP tasks.

2.1. Knowledge-based/Linguistic Approaches

These approaches rely on manually crafted rules or lexicons. They're interpretable but not easily scalable. This is the case with techniques based on parse trees or even WordNet. **WordNet** is a lexical database that organizes English words into synonym sets based on their meaning, providing semantic relationships such as synonyms, antonyms, hypernyms, and meronyms [13].

A **parse tree** shows the hierarchical syntactic structure of the input according to the grammar. Parse trees are usually generated by applying Context-Free Grammar rules, which define the syntactic structure of languages and guide the parsing process to build the trees [2].

Figure 2.1 shows a parse tree generated by the Stanford Lexicalized Parser, which produces parses by searching for the most probable tree structure given the learned probabilistic grammar [14, 15].

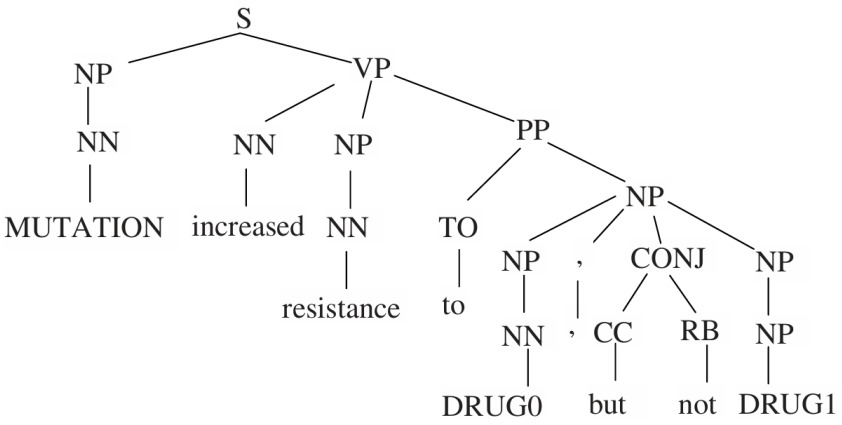


Figure 2.1: Parse tree generated by the Stanford Parser for the sentence *"MUTATION0 increased resistance to DRUG0, but not DRUG1"* [14].

Parse trees can serve as the initial structural analysis that enables the subsequent rule-based identification of relations by providing the necessary grammatical components. In [14], the parse tree provides the raw structural information, which is then interpreted by one set of grammatical relation rules to yield the constituents. Following that, a separate set of custom-defined rules specifically targets these constituents to identify and extract the desired causal relations.

Another important family of grammars used in causality extraction is **dependency grammar**. In dependency formalisms, phrasal constituents are not directly involved. Instead, the syntactic structure of a sentence is represented entirely through directed, binary grammatical relations between words, as shown in Figure 2.2. In Figure 2.2, the relationships between words are shown above the sentence using directed, labeled arcs from heads to dependents. A root node explicitly marks the root of the tree - the head of the entire structure.

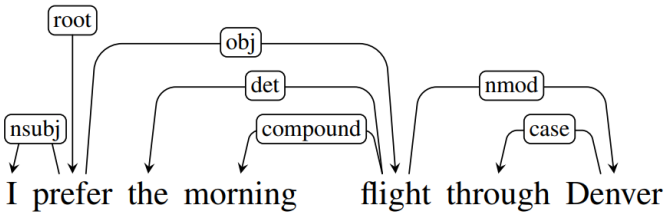


Figure 2.2: An example of a dependency parse [2].

Figure 2.3 presents the dependency analysis from Figure 2.2 visualized as a tree, alongside its corresponding phrase-structure analysis (the parse tree) [2]. It is important to note that, in the literature, this type of dependency information is more commonly used in sophisticated systems, such as DL-based causal extraction approaches, rather than in rule-based approaches.

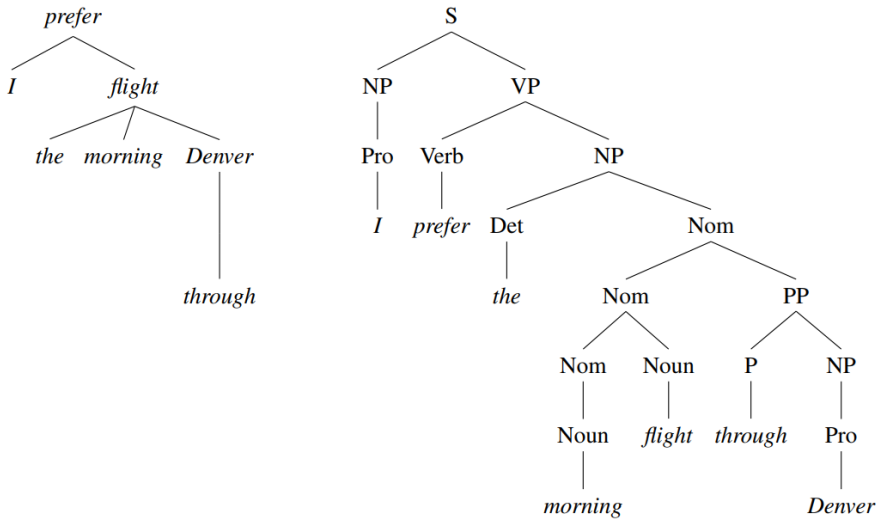


Figure 2.3: Dependency and constituent analyses for *"I prefer the morning flight through Denver"* [2].

In pattern-based approaches, **Part-Of-Speech (POS) tags** are also used to specify the grammatical category of each word in a sentence, such as *Noun* or *Verb*. These tags can be generated using third-party NLP tools such as `Stanford CoreNLP` and `Stanza` [16, 17]. POS tagging refers to the process of assigning these labels to each word in a sequence.

Tagging is a disambiguation task because many words can have multiple possible parts-of-speech. The objective is to assign the correct tag based on the context. For instance, the word *"book"* can function as a *Verb* (e.g., *"book that flight"*) or as a *Noun* (e.g., *"hand me that book"*) [2]. This is a type of sequence labeling task, which will be introduced in the following section, as sequence labeling algorithms are also relevant for modeling causal extraction as a sequence labeling problem.

2.2. Statistical ML-based Approaches

In the context of causal relation extraction, statistical ML models such as **Conditional Random Fields (CRFs)** have proven effective, particularly for tasks that can be formulated as sequence labeling problems [18].

The causal relation extraction can be framed as a sequence labeling problem, similar to Named Entity Recognition (NER). NER is a fundamental task in NLP that involves identifying and classifying spans of text that refer to specific entities such as people, organizations, locations, and more [2].

Unlike POS tagging, where each word is assigned a single label and segmentation is straightforward, NER involves identifying and labeling spans of text. This adds complexity due to segmentation ambiguity - deciding which spans constitute entities and where their boundaries begin and end. Most words in a sentence are not part of any named entity, which further complicates the task [2].

A common approach to span recognition is Begin-Inside-Outside (BIO) tagging, which marks the beginning, continuation, and absence of an entity in a sequence of words. This scheme allows for word-by-word labeling while encoding both entity boundaries and types [2]. Figure 2.4 shows three different tagging schemes.

Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O

Figure 2.4: NER as a sequence model, with IO, BIO, and BIOES taggings [2].

The IO scheme simplifies tagging by removing the B (Begin) tag, though this comes at the cost of losing some boundary information. On the other hand, the BIOES scheme extends the standard BIO format by adding two additional tags: E (End) to mark the final word of a multi-word entity, and S (Single) for entities that consist of only one word. CRFs are trained to assign a tag to each token in a text.

The CRF is a log-linear model that assigns a probability to an entire output sequence Y (e.g., a sequence of tags), given an input sequence X (e.g., a sentence) defined as [2]:

$$p(Y \mid X) = \frac{1}{Z(X)} \exp\left(\sum_k w_k F_k(X, Y)\right) \tag{2.1}$$

where

- $F_k(X, Y)$ is a global feature function;
- w_k is the learned weight associated with the feature function F_k;
- $Z(X)$ is the partition function (normalizer) that sums over all possible output sequences to ensure the probabilities sum to 1.

Each global feature function $F_k(X, Y)$ is defined as the sum of local feature functions over the sequence [2]:

$$F_k(X, Y) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i) \tag{2.2}$$

Equation (2.1) defines the CRF model, and Equation (2.2) shows how global features are constructed from local ones.

In linear-chain CRF, each local feature depends only on the current tag y_i, the previous tag y_{i-1}, the full input sequence X, and the current position i. In traditional CRF, features are hand-crafted from the input using templates based on word identity, capitalization, suffixes, and other linguistic or contextual cues. These local features can be broadly categorized into *emission features*, which capture the compatibility between the input and the current label y_i, and *transition features*, which capture the dependencies between consecutive labels y_{i-1} and y_i.

CRFs use the Viterbi algorithm for inference, while the log-likelihood is computed using a log-space variant of the Forward algorithm for training [2].

Viterbi decoding is a dynamic programming algorithm used to find the most probable sequence of labels (i.e., the path with the highest global probability) [2]. At each step, it computes the best score for reaching a given state, considering both *emission* and *transition* probabilities. The probability of the most likely path ending in state t with observation i at position x is given by [2, 16]:

$$p_t(i, x) = e_t(i) \cdot \max_k \left[p_k(j, x-1) \cdot p_{kt} \right] \tag{2.3}$$

where

- $e_t(i)$ is the emission probability of observing element i in state t;

- $p_k(j, x-1)$ is the probability of the most likely path ending in state k at position $x-1$ with element j;
- p_{kt} is the transition probability from state k to state t.

Instead of relying solely on hand-engineered input features, *emission scores* can also be generated by DL methods that learn rich representations from the data.

2.3. DL-based approaches

In NLP, various deep learning architectures have been employed to model linguistic features and dependencies. Commonly used models include CNNs and Recurrent Neural Networks (RNNs), along with their more advanced variants such as LSTM networks and GRUs. These models are typically categorized as sequential-based architectures, as they process input data in a linear order and capture semantic and syntactic information from local consecutive word sequences.

In contrast, graph-based models such as Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs) represent text as a graph of nodes (e.g., words, entities) and edges (e.g., syntactic or semantic relations), allowing for the modeling of non-sequential and long-range dependencies [12].

Among sequential models, the combination of a **bidirectional LSTM (BiLSTM)** with a **CRF layer** has emerged as a widely adopted architecture for sequence labeling tasks, such as NER. The BiLSTM processes sequences in both forward and backward directions, allowing it to capture contextual information from both past and future tokens, while the CRF layer models label dependencies, improving prediction consistency across sequences [19].

Graph Neural Networks (GNNs) can be effectively applied to extract information from dependency trees by treating these linguistic structures as graphs. In such settings, the implicit relational structure of text is converted into a graph where words serve as nodes and their grammatical dependencies form the edges [20]. The GCN model, *GCNConv*, is specifically designed to operate on direct graphs with labeled edges, incorporating edge-wise gates to regulate the contribution of individual dependency gates during information propagation. These GCNs serve as powerful sentence encoders, learning latent feature representations of words within the sentence [20].

Many traditional GCNs are often transductive, meaning they learn embeddings only for nodes present during training and have limited ability to generalize to unseen nodes. To address this limitation, inductive frameworks like GraphSAGE, *SAGEConv*, have been developed. GraphSAGE learns a generalizable function that can efficiently generate low-dimensional node embeddings for previously unseen nodes or entirely new graphs by sampling and aggregating feature information from a node's local neighborhood [21]. Figure 2.5 illustrates the GraphSAGE approach.

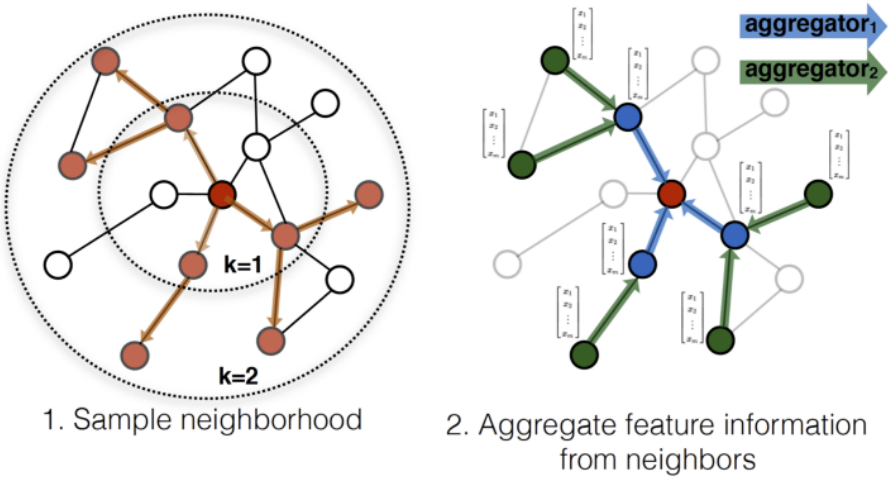


Figure 2.5: Illustration of the GraphSAGE sample and aggregate approaches [21].

GraphSAGE offers flexible aggregator functions, including mean aggregator (which can be seen as an inductive extension of GCN), an LSTM aggregator, and a pooling aggregator, which provide more architectural freedom compared to the fixed convolutional rules of basic GCNs. A key distinction from traditional GCNs is that GraphSAGE typically concatenates the node's current representation with the aggregated neighborhood vector, acting as a "skip connection" [21].

2.3.1 Word Embeddings and Large Language Models (LLMs)

In classical DL, words are represented as independent tokens, without explicit consideration of broader semantic relationships such as synonymy or hyponymy. Consequently, associations between words are learned from the training data, which constrains their scope and generalizability. To address this limitation, the NLP community has developed approaches that leverage large unlabeled corpora to infer lexical relationships. Two predominant methods identified in the literature are word embeddings and language models [4].

Word embeddings represent words as vectors based on their co-occurrence with other words, under the assumption that semantically similar words appear in similar contexts. These vectors can be compared using measures like cosine similarity, enabling models to identify similar meanings - even for unseen words. However, traditional embeddings such as Word2Vec or GloVe (Global Vectors for Word Representation) are static and assign only one vector per word, which limits their ability to capture context and multiple word senses (e.g., *"cause"* as a *noun* vs. a *verb*), potentially hindering causal relation extraction [4].

To overcome these limitations, contextualized word representations have been introduced through modern language models.

Recent advances introduced LLMs, which are typically built on the transformer architecture and trained on massive unlabeled text corpora. The **transformer architecture** is a neural network design introduced in the paper *"Attention is all you need"*, which processes entire sequences in parallel rather than step-by-step as RNNs or CNNs [22].

The transformer's architecture consists of stacked self-attention layers and point-wise fully connected layers in both the encoder and decoder, illustrated in the left and right sides of Figure 2.6, respectively [22]. Each transformer block passes input forward through a residual stream, a running pathway that carries the original signal upward, while adding outputs from a multi-head attention layer and a feedforward layer, both with layer normalization. This preserves earlier information while letting each layer add refinements. Inputs are formed by adding token embeddings to positional encodings. Stacks of these blocks form language models, which use a final unembedding layer and softmax to produce word probabilities. Large transformers can handle very wide context windows - up to hundreds of thousands of tokens - enabling them to leverage extensive context for predictions [2].

Masked Language Models (MLMs) introduced in Bidirectional Encoder Representations from Transformers (BERT), are trained by masking specific words in a sentence and predicting the masked words from the surrounding context. This bidirectional approach allows the model to understand context from both directions, making it highly effective for tasks that require nuanced comprehension, such as sentiment analysis and NER [23].

Generative Pre-trained Transformers (GPTs) models are trained in a unidirectional, autoregressive fashion, predicting the next word based solely on preceding words. This design makes GPTs particularly strong in generating fluent, coherent, and contextually

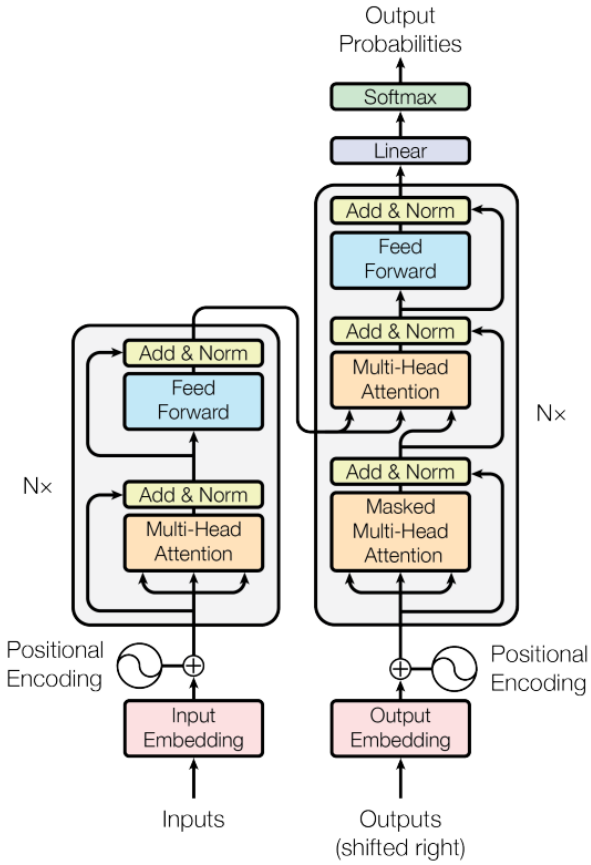


Figure 2.6: Transformer's architecture with the encoder on the left side and the decoder on the right side [22].

relevant text, lending them to applications such as creative writing, conversational agents, and open-ended text generation.

By leveraging these methods, LLMs learn rich contextualized word representations that capture deeper semantic and causal relationships in language, enabling state-of-the-art performance across a wide range of NLP tasks.

The MLMs are often referred to as encoder-only architectures, as they generate contextualized representations (encodings) for each input token but are not typically employed for text generation through decoding or sampling. This distinction is significant: MLMs are generally designed for interpretative or discriminative tasks, rather than for generating coherent sequences of text [2].

BERT and its variants are pre-trained using not only Masked Language Modeling, but also Next Sentence Prediction (NSP) to enable sentence-level relationship modeling. The model is given pairs of sentences and must predict whether the second sentence follows the first in the original corpus. Inputs are tokenized and augmented with special tokens:

[CLS] at the start, [SEP] between sentences and at the end, and segment embeddings indicating sentence boundaries. The output vector corresponding to [CLS] from the final transformer layer is fed into an NSP classification head to produce a two-class prediction [23].

By leveraging pre-training and fine-tuning strategies, BERT and its variants have significantly advanced performance in tasks such as sequence classification (e.g., causality detection, sentiment analysis, etc.) and sequence labeling (e.g., NER) [2].

Sequence/text classification involves assigning a single label to an entire text sequence, using the special token [CLS]. The final hidden state associated with this token is used to represent the entire sequence. This vector is passed to a classifier head, which produces a probability distribution over the target classes [2].

Sequence labeling tasks, as mentioned before, assign a label to each individual token in a sequence. Each token's final hidden state is passed through a classifier that outputs a probability distribution over possible labels. The predicted label is typically the one with the highest probability. The classifier includes a weight matrix that maps token representations to label scores, and a CRF layer may be used to model dependencies between labels across the sequence [2].

Another important downstream application is **span prediction**, as seen in fine-tuning BERT on the Stanford Question Answering Dataset (SQuAD) [23]. In this setup, the model takes a question and a context passage as input and produces two outputs: start logits, representing the probability distribution over tokens for the beginning of the answer, and end logits, representing the distribution for the end of the answer. The loss is computed based on the difference between the predicted and ground truth start and end positions. This approach frames the problem as identifying the exact span within the context that contains the answer.

2.4. Modeling Paradigms for Causal Relation Extraction

Architectures for causal relation extraction found in the literature are largely inspired by techniques originally developed for tasks such as Question-Answering (QA), NER and general Relation Classification and Extraction. These approaches have inspired several architectural paradigms in causal relation extraction, including:

- **Pointer-based models (Start/end Pointer Models):** These models predict the

start and end positions of argument spans (like traditional QA-style encoders). While affective for pinpointing span boundaries, they typically require additional mechanisms to handle the variable number of causal relations that may appear in a single sentence;

- **Token-centric models (Sequence Tagging):** These approaches adapt sequence labeling schemes (such as BIO or BILOU) to assign role-specific tags (e.g., cause, effect, signal) at the token level. Though straightforward and efficient, they can struggle with overlapping argument spans;
- **Span-centric models:** Instead of predicting labels for individual tokens, span-centric methods represent and score entire spans directly. This enables richer modeling of span-internal structure and inter-span interactions, offering greater flexibility and expressiveness.

2.4.1 More about QA systems

Pointer-based architectures became popular in QA tasks because they extract precise answer spans from a given context. Many QA datasets (like SQuAD) require models to select exact spans from a passage as answers. Pointer mechanisms allow models to point to start and end positions in the input, rather than generating answers word-by-word, leading to fewer hallucinations.

While single-span models perform well on datasets where answers are short and contiguous, they are insufficient when questions require multiple discontinuous spans. The DROP dataset, released in 2019, marked a turning point by allowing answers to be sets of spans, motivating the development of models specifically for Multi-Span Reading Comprehension. The **Multi-Type Multi-Span Network (MTMSN)** directly addressed these challenges by [24]:

- Predicting a distribution over the number of spans required;
- Considering all gold spans during training, even if the predicted span count is wrong, without committing to a fixed alignment.

However, pointer-based designs still rely on token-level start/end scores that must be post-processed into spans, which can be inefficient and suboptimal for multi-span reasoning. This motivated span-centric approaches, where candidate spans are explicitly

represented, scored, and selected without token-to-span conversion. Central to these models is the **SpanClassifier**, which operates over three levels of representation: *(1) token representations* from a contextual encoder, capturing the meaning of each token; *(2) boundary representations* from the start and end tokens, encoding positional cues; and *(3) a span representation* that aggregates the internal tokens - often enhanced with intra-span attention to focus on informative words. This enriched span vector is then scored or labeled. In models like SpanQualifier, it is further optimized jointly with an inter-span interaction layer under a global loss, enabling spans to be evaluated in the context of all others in the passage [25].

2.4.2 Relation Classification with Overlapping Entities

In relation extraction, given a predefined set of target relations and a sentence such as

"Leonardo DiCaprio starred in Christopher Nolan's thriller Inception.",

the objective is to extract relational triplets such as (*"Leonardo DiCaprio"*, Plays-in, *"Inception"*) or (*"Inception"*, Director, *"Christopher Nolan"*). This task involves two key subproblems: identifying entities (entity recognition) and determining the relationships between them (relation classification) [26].

Traditional relation extraction models typically treat entity recognition and relation classification as separate tasks, often relying on token-level sequence labeling for identifying entities. While effective in many settings, these approaches struggle with complex linguistic phenomena such as overlapping or nested entities, where a single token may belong to multiple spans. For example, in the sentence *"Ford's Chicago plant employs 4,000 workers"*, both *"Chicago"* and *"Chicago plant"* are valid entities [26].

To address these challenges, span-centric models like Span-based Entity and Relation Transformer (SpERT) have been proposed, treating entity recognition and relation classification within a unified span framework.

SpERT, illustrated in Figure 2.7, uses BERT to extract entities and relations from text through three main steps [26]:

- *(1) Span Classification:* All possible token spans (up to a fixed length) are encoded using max-pooled BERT embeddings, a learned span-width embedding, and

the [CLS] token for sentence-level context. These combined features are passed through a softmax classifier to predict the span's entity type or label it as non-entity;

- *(2) Span Filtering:* Spans classified as non-entities are discarded, leaving only those likely to represent entities;
- *(3) Relation Classification:* All pairs of remaining entity spans are evaluated for possible relations using their embeddings and a localized context embedding drawn from the text between them. A sigmoid classifier is applied to each pair in both directions (to account for asymmetric relations), and relations are accepted if their confidence score exceeds a set threshold.

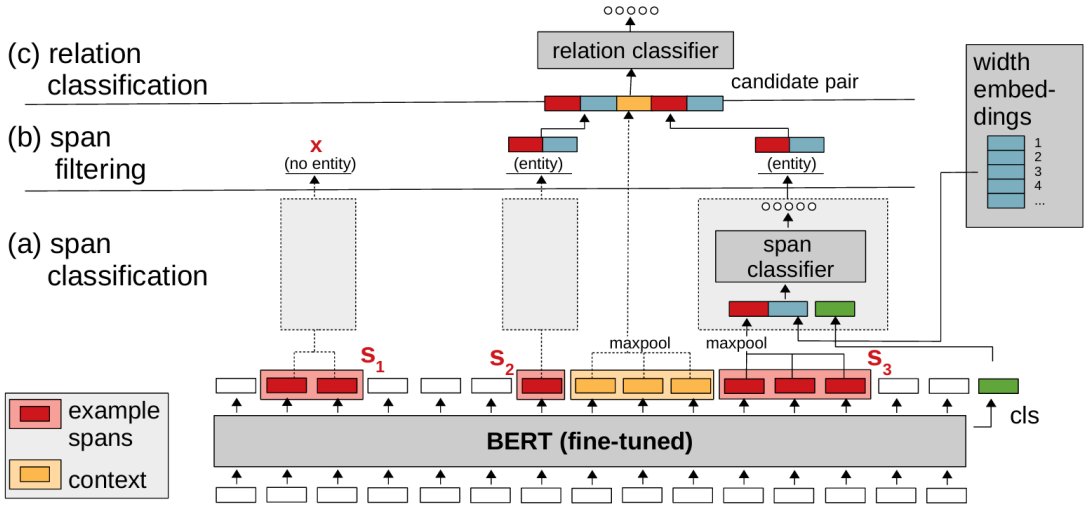


Figure 2.7: SpERT model [26].

2.5. Summary

This chapter presents the theoretical concepts that are essential for understanding the approaches reviewed in Chapter 3 as well as the experiments conducted in this thesis.

Not all approaches discussed will be implemented in this study; only those deemed most promising after a careful evaluation of the state-of-the-art are selected.

3. Related Work

This chapter is structured to provide a comprehensive overview of causal relation extraction. It begins with a section on the evolution of techniques, tracing the progression from traditional linguistic and pattern-based approaches to deep learning techniques. Following this, the chapter discusses relevant competitions, highlighting how pre-trained language models can be adapted for causal relation extraction. Finally, the chapter includes a dedicated section on prompting techniques for few-shot learning models. Although these models were not widely used in past competitions, our study aims to assess their limitations in the context of causal relation extraction experiments and compare them with BERT variants.

3.1. Evolution of techniques for Causal Relation Extraction

Traditional techniques, largely used before 2005, include clue phrases and pattern-based approaches. **Clue Phrases-based approach** utilizes a predefined set of keywords (clues) in a rule-based method, relying on lists of words and phrases, often including causal verbs or prepositions - where causal verbs like *causes* commonly indicate causality. Clues can be hand-curated lists or part of a lexical resources such as WordNet and VerbNet [27].

Pattern-based approach identifies and utilizes frequently occurring linguist patterns to extract causal relations. It relies on recognizing consistent structures that indicate causality, such as the `<Noun-Verb-Noun>` pattern found in sentences like *"Smoking causes cancer"* or *"Rain causes floods"*. These patterns can be identified using POS tags, which classify words into categories such as noun, verb or adjective, as mentioned in the previous chapter. The patterns can also be generalized to other causal expressions and derived from dependency trees or phrase structures [27]. A common example is the `<NP-Verb-NP>` pattern, where noun phrases (NP) represent the cause and effect, and the verb serves as the causal link, as in *"Excessive sun causes sunburn"* or *"Lack of money creates poverty"* [27].

In [28], they developed a linguistic-based approach to detect explicitly expressed causality in text, relying on causative verbs, causal links, and specific grammatical structures while avoiding domain knowledge dependence. Their work focused on extracting causality from

a medical database using predefined graph patterns and syntactic parse trees, identifying cause-and-effect relationships within single sentences.

In [29], the authors also presented a linguistic pattern-based approach. They focus on explicit intra-sentential lexico-syntactic pattern of the form <NP_1-Verb-NP_2>, where the verb is a simple causative - the linking verb refers only to the causal link, most of the time being synonymous with cause. For example, *"Earthquakes generate tidal waves"*. Here the verb *"generate"* is synonymous with *"cause"*. In [29], NP_1 and NP_2 were created from lexical knowledge bases. The patterns are then identified and ranked by imposing constraints on nouns and verbs through WordNet-based course-grained procedure.

Some studies leverage external information by using several large-scale **causal knowledge bases** developed to support research in causal reasoning, containing direct cause-effect event pairs (e.g., CausalNet, CauseNet, CausalBank) [10]. **CausalNet** [30] consists of commonsense causal triplets at the word or phrase level, extracted from large web corpora using statistical association measures. The extracted causal pairs form a directed network representing causal relationships. Figure 3.1 shows a sample fragment of the network containing three terms, where nodes are lemmatized terms, edges show causal links, and edge labels indicate how often each relation occurs in the corpus.

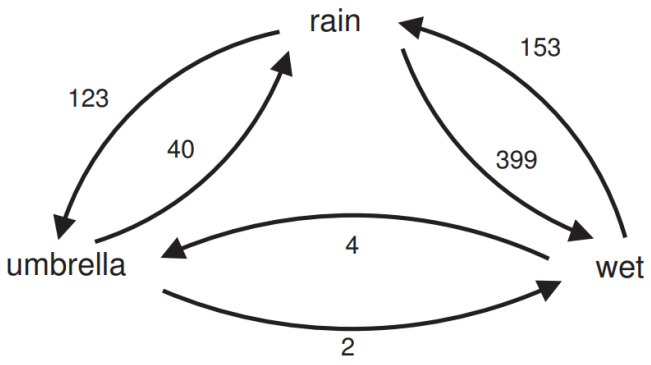


Figure 3.1: A fragment of CausalNet [30].

CauseNet [31] is a concept-level causal graph containing over 11 million relations extracted from web text, organized as directed edges between causal concepts through pattern-based information extraction techniques. **CausalBank** [32] is a massive sentence-level corpus of 314 million cause-effect pairs mined from Common Crawl using lexical pattern matching. From CausalBank, a connected lexical Cause Effect Graph is derived with weighted edges, where weights correspond to frequency and causal strength scores, similar to CausalNet [32].

While these knowledge bases provide valuable repositories of causal relations, most are automatically constructed, often via pattern-based or semi-supervised methods, and may suffer from noise or limited linguistic variation [10].

Recent advances in **DL** have greatly enhanced NLP tasks, including causality extraction, by enabling models to automatically learn rich syntactic and semantic features through word embeddings. Typical causal relation extraction frameworks consist of three components: input representations (e.g., word, sentence, and paragraph-level embeddings, often enriched with prior knowledge or linguistic features), context encoders (such as CNNs and transformers) and classifier decoders [18]. Transformer-based pre-trained language models now represent the state-of-the-art in NLP, with **BERT** and its variants being among the most prominent [10].

In particular, domain-specific pre-trained models demonstrate significant potential in advancing causal extraction tasks. Since the word distributions in general-purpose corpora differ substantially from those in specialized domain corpora, standard pre-trained models often exhibit suboptimal performance in domain-specific applications. In contrast, models pre-trained from scratch on domain-specific corpora, such as **SciBERT** (a language model trained on scientific text) and **BioBERT** (a language model designed for biomedical text mining) can effectively mitigate this limitation [12].

The study in [33] highlights that BERT-valiant models achieved superior and more consistent performance for causality extraction from medical texts compared to GPT-4 with zero-shot prompting and Large Language Model Meta AI (LLaMA)2.

In [34], a study involving causal span extraction across several datasets - such as AltLex, Bank of Effects and Causes Stated Explicitly (BECauSE), Penn Discourse TreeBank (PDTB), Financial Document Causality Detection Shared Task (FinCausal) - found that BERT-variant encoder models achieved superior performance compared to GPT-3.5-turbo and GPT-4, particularly when provided with a sufficient amount of training data. The capabilities of GPT-3.5-turbo and GPT-4 were evaluated using various prompting strategies, including zero-shot and few-shot in-context learning.

3.2. Competitions

This section focus on shared tasks that aimed to extract cause and effect spans (a continuous string of text) given an input causal sentence - FinCausal and Shared Task 3 of

Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE) (that is centered around the CNC).

In both shared tasks, each causal relation consists of exactly one cause span and one effect span. Additionally, in CASE, signals are annotated when present, whereas in FinCausal they are not.

Notably, the vast majority of teams in both competitions used BERT or BERT-variant models.

3.2.1 FinCausal

The FinCausal has had five editions from 2020 through 2025, with each edition from 2020 to 2022 progressively expanding the dataset. FinCausal 2020 featured two tasks: the first focused on identifying the presence of causality in documents (binary classification), while the second aimed at extracting the specific cause and effect of causal statements. Notably, the second task included only causal examples and excluded non-causal ones, similar to the setup of sub-task 2 in Shared Task 3 of CASE. In the 2021 and 2022 editions, the organizers opted to continue only with the task previously defined as Task 2, focusing exclusively on the extraction of causes and effects [35–37]. The 2025 edition introduced a generative QA format across English and Spanish subtasks [38].

In the task 2 of the 2020 to 2022 editions of FinCausal, a document could contain multiple cause/effect pairs to be extracted, similar to sub-task 2 in Shared Task 3 of CASE. However, in FinCausal, each example was duplicated for every cause/effect pair, which implicitly defined the number of predictions to be output. This applied to both the training set and the final test set used to rank the teams in the competition. Teams were able to leverage this information to improve their results. This setup differed from that of sub-task 2 in Shared Task 3 of CASE.

3.2.1.1 FinCausal 2020:

In the 2020 edition, specifically in task 2, the top-performing system (NTUNLP team) used a BERT model and a Viterbi decoder for span optimization. The second-place system, developed by the Gbe team, was based on Robustly Optimized BERT Pre-Training Approach (RoBERTa) for predicting start and end positions and incorporated heuristics for span extraction [35]. The two teams were close in terms of F1 score, but in terms of exact match, the NTUNLP team led by 8.78%.

The NTUNLP team addressed causal relation tagging using a BERT-variant token classifier with various labeling strategies. Four main approaches were tested: (1) simple causal role classification, using only role labels, $\{O, C, E\}$; (2) combining BIO with causal role tags, $\{O, B-C, I-C, B-E, I-E\}$; (3) concatenating the output from BERT's final hidden state with POS tags via one-hot vectors (the POS tags were labeled by the Stanford CoreNLP Stanza toolkit); and (4) introducing a Viterbi decoder to improve sequence prediction using predefined BIO transitions. Results showed that the BIO-causal role tagging scheme underperformed compared to the simpler causal role tagging scheme and POS features offered minimal benefit. However, combining the BIO-causal role tagging scheme with a Viterbi decoder led to the best results [16].

Gbe Team The team that ranked second developed the framework presented in Figure 3.2, as explained in [39]. The final hidden states of the transformer passed through a dense layer to generate logits for the start and end positions of both cause and effect spans. This method is similar to state-of-the-art QA models, but extracts two spans (cause and effect) instead of one.

The Gbe team found that the system's training was more stable when using models pretrained for QA. The final system was based on a RoBERTa span extraction model (similar to QA architecture).

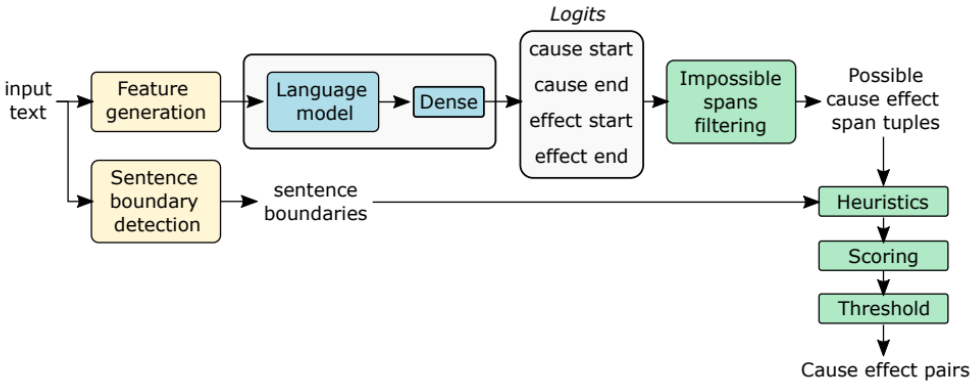


Figure 3.2: System architecture proposed by GBe team at FinCausal 2020, subtask 2 [39].

For post-processing, they used a filtered span enumeration strategy using heuristics, applied in a single pass (within the top-N logits), as demonstrated in Figure 3.2. The candidate cause-effect span pairs were generated by first selecting top-N highest scoring start and end token positions (based on model logits) for both cause and effect. Then, all possible combinations of these start-end pairs were enumerated and passed through

a series of filtering heuristics that eliminated invalid spans - such as overlapping cause/-effect spans, spans that crossed sentence boundaries, exceeded maximum length or fell outside valid token mappings. Only the span pairs that passed all these checks were kept as valid predictions, which were then scored and ranked based on their logits to identify the most likely cause-effect relationship in the text.

The number of cause/effect combinations to return could be determined either externally (by the number of duplicated entries in the shared task dataset, which implicitly defined the number of cause/effect combinations to extract) or dynamically, based on the span probability and a specified threshold, as would occur in a real-world scenario. The team computed the span probability as the product of the individual probabilities for the start and end positions of both the cause and the effect:

$$P(\text{cause}_{\text{start}}) \times P(\text{cause}_{\text{end}}) \times P(\text{effect}_{\text{start}}) \times P(\text{effect}_{\text{end}})$$

The team found that an optimal threshold of 0.99 maximized performance in estimating the number of predictions.

3.2.1.2 FinCausal 2021:

In the 2021 edition, team NUS-IDS achieved the highest F1 score, while team DSC-IITISM and team NVJPFSI followed closely behind. In terms of Exact Match, team DSC-IITISM achieved the highest score, followed closely by NVJPFSI and NUS-IDS [36].

Team NUS-IDS proposed framework is presented in Figure 3.3. They proposed a solution that incorporates dependency relations within a sentence to enhance the identification of cause-effect spans by representing dependency relations using a graph neural network [17]. Each example was modeled as a direct graph, where nodes correspond to tokens enriched with BERT and POS embeddings, and edges represent head-to-tail connections based on dependency parsing.

The model used a two-layer GNN with a SAGEConv operators to pass messages across dependency relations, leveraging node features and neighborhood aggregation.

Figure 3.4 shows some examples, highlighted by their span labels. They noticed a tendency for words with the same cause/effect label to be more connected on these graphs.

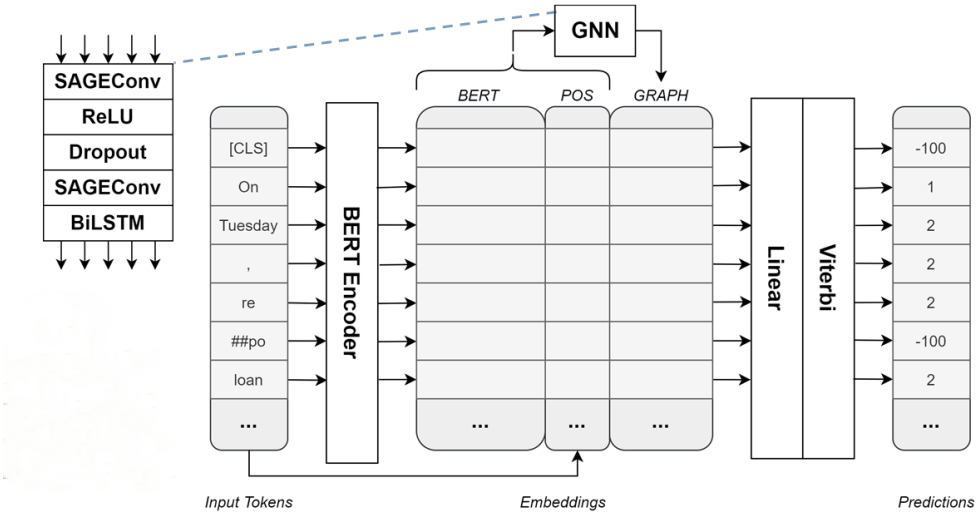


Figure 3.3: Pipeline proposed by Team NUS-IDS at FinCausal 2021 [17].

Therefore, they incorporated this information into the model as features. The Graph Neural Network (GNN) module produced graph representations that were concatenated with the BERT and POS embeddings and fed into a linear layer for token classification.

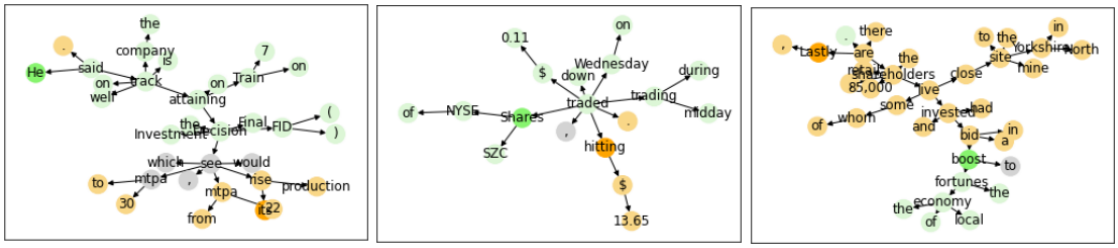


Figure 3.4: Dependency-tree represented as directed graphs, with Cause, Effect, and Other spans highlighted in green, orange and grey respectively [17].

The model extends the work presented by Team NTUNLP at FinCausal 2020, which involved a BERT token classifier with Viterbi decoding, as previously mentioned. The proposed framework slightly improves performance.

Team DSC-IITISM combined POS tags with BIO scheme using ensemble optimization strategy comprising BERT, RoBERTa, XLNet and GPT-2. The mode was calculated to find the most frequently occurring label [40].

To improve Exact Match scores, the post-processing step selected the longest cause-effect pair in cases of multiple causal chains and merged closely aligned pairs separated by few tokens, both strategies aligning with annotation rules of the Shared Task to prioritize comprehensive and unified causal segments.

Team NVJPFSI explained their work in [41] and based their approach on the framework proposed by Team Gbe at FinCausal 2020 (described in [39]). The team employed ensemble learning by combining predictions from five models, using grid search to find the optimal weight combination (summing to 1.) based on exact match scores from training and development sets. For each model, the top-5 predictions were weighted and aggregated to compute final scores. An additional function applied these weights to determine the most likely cause-effect pair for each input. To avoid duplicate outputs, a suffix index was used to ensure distinct predictions when multiple results were allowed for the same input text.

3.2.1.3 FinCausal 2022:

In the 2022 edition, Team SPOCK led in all evaluation metrics, followed closely by team iLab [37].

Team SPOCK explored both span-based and sequence tagging strategies [42]. The sequence tagging model based on RoBERTa-Large achieved better results. They adopted the BIO tagging scheme, where each token in the input sequence was assigned one of the following tags: $\{B-C, I-C, B-E, I-E, O\}$. Furthermore, to differentiate multiple cause-effect pairs within the same example, a numerical prefix was added at the beginning of the instances, as many teams did in the previous editions. Figure 3.5 presents an example containing multiple cause-effect pairs, where each pair is distinguished by a numerical prefix. The corresponding BIO tags are displayed beneath each token.

1	Ceteris	paribus	,	the	fiscal	deficit	this	fiscal	will	widen	to	around
O	B-E	I-E	I-E	I-E	I-E	I-E	I-E	I-E	I-E	I-E	I-E	I-E
4	%	owing	to	the	stimulus	if	extra	transfers	from	RBI	are	counted
I-E	I-E	O	O	B-C	I-C	O	O	O	O	O	O	O
,	the	deficit	's	size	could	be	3.8	%	.			
O	O	O	O	O	O	O	O	O	O			
2	Ceteris	paribus	,	the	fiscal	deficit	this	fiscal	will	widen	to	around
O	O	O	O	O	O	O	O	O	O	O	O	O
4	%	owing	to	the	stimulus	if	extra	transfers	from	RBI	are	counted
O	O	O	O	O	O	O	B-C	I-C	I-C	I-C	I-C	I-C
,	the	deficit	's	size	could	be	3.8	%	.			
O	B-E	I-E	I-E	I-E	I-E	I-E	I-E	I-E	I-E			

Figure 3.5: Example with multiple cause-effect pairs - scheme used by Team SPOCK at FinCausal 2022 [42].

An ensemble model comprising 11 individual models was employed, using a majority voting approach (the same model was trained with different random seeds).

Team iLab built upon the winning FinCausal 2021 system, enhancing it by incorporating external cause-effect knowledge into the graph embeddings [43]. This was achieved by utilizing an external Cause-Effect-Graph specific to the financial domain, built from CausalBank. The paper outlines two strategies for injecting this knowledge:

• *Dependency Tree Structure with Weighted Edges*: Building on the dependency tree approach of the FinCausal 2021 system, this method introduces directed edges between causal token pairs and assigns weights to the connections - presented in image (b) of Figure 3.6;

• *Sentence Chain Structure*: This separates cause and effect spans into two linear chains, linking tokens sequentially within each chain. Tokens sharing causal relations across the chains are connected with weighted, unidirectional edges derived from the external cause-effect graph - presented in image (c) of Figure 3.6. During validation and testing, a single chain connects the whole input sequence, with causal token pairs connected through intra-chain links - presented in image (d) of Figure 3.6.

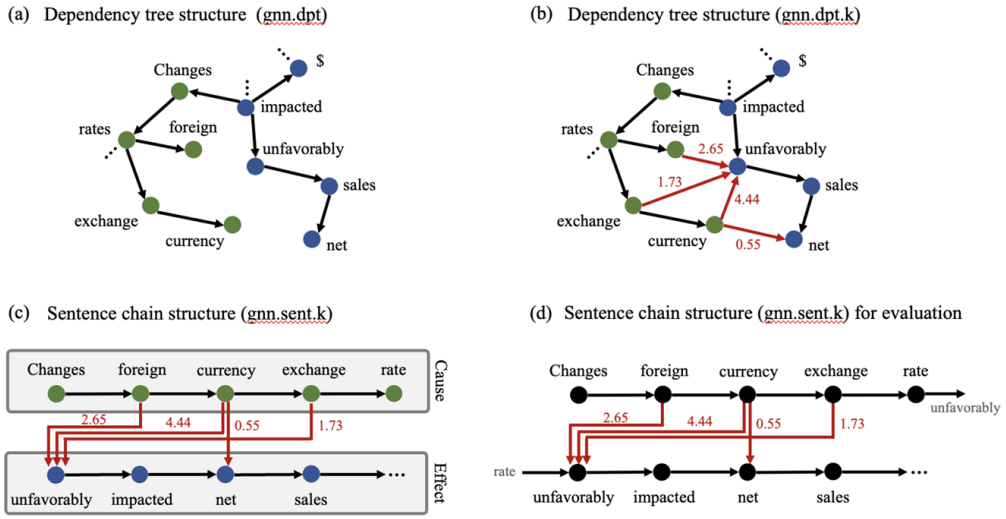


Figure 3.6: Four types of sub-graph generated by iLab's graph builder: (a) baseline dependency tree-based sub-graph; (b) adding edges and weights (in red) for cause-effect relations; (c) sentence chain structure with cause-effect edges for training; (d) a single chain connecting the entire input text for inference. All black edges in (b), (c), and (d) share the same small weight. Green nodes belong to the cause span, while blue nodes belong to the effect span. [43].

The sentence chain structure with cause-effect graph injection, image (c) of Figure 3.6, outperformed both the baseline dependency tree structure, image (a) of Figure 3.6, and its knowledged-injected variant, image (b) of Figure 3.6. Additionally, the knowledge injection proved effective when applied to the proposed sentence chain structure but had limited impact on the dependency tree structure.

3.2.2 Shared Task 3 of CASE

This Shared Task was held in 2022 and 2023, with each edition progressively expanding the dataset. It comprised two tasks, following a similar structure to the 2020-2022 editions of FinCausal. Some concepts from this Shared Task were already introduced earlier in the FinCausal subsection.

3.2.2.1 Shared Task 3 of CASE, 2022 Edition:

In the 2022 edition, specifically in task 2, the top-performing system (1Cademy) used a A Lite BERT (ALBERT) for predicting start and end positions. The second-place system, developed by the IDIAPers team, used the pre-trained T5 model to generate up to four causal relations for each example, conditioning it three times per example. Finally, the third-place system, the SPOCK team, employed a span-based modeling inspired by SpERT [1].

The 1Cademy Team tackled the task using a QA-based approach, as other teams did in FinCausal [44]. They implemented a signal classifier to detect the presence of a signal. Additionally, they introduced beam search strategies as post-processing constraints tailored to the task. The proposed framework is illustrated in Figure 3.7.

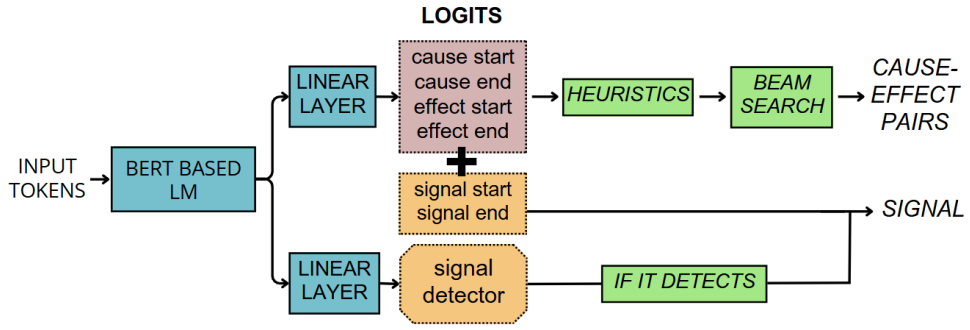


Figure 3.7: Reading Comprehension based Approach via Beam Search-based Position Selector.

As seen before, for this type of approach, post-processing constraints are necessary to ensure that the predicted end position comes after the start position of the same span, and cause and effect spans do not overlap. This team used a masked scoring strategy to enforce either `cause-before-effect` or `cause-after-effect`, discarding any span combinations that violated the *discourse linear order*, and selected the top-k candidates based on their summed scores:

$$\text{P}(\text{cause}_{\text{start}}) + \text{P}(\text{cause}_{\text{end}}) + \text{P}(\text{effect}_{\text{start}}) + \text{P}(\text{effect}_{\text{end}})$$

The beam-search-based span selector is described in detail in Figure 3.8. Constraints based on the *discourse linear order* are crucial for identifying spans that maximize prediction confidence. Since it is unknown beforehand whether the cause appears before the effect in the text or vice versa, the method considers both possibilities separately. For each scenario, `CBeforeE` and `CAfterE`, a masked scoring strategy is applied to ensure span positions follow the respective discourse order. Specifically, when assuming `cause-before-effect`, only span combinations where the cause's start and end positions come before those of the effect in the text are considered; all others are masked out. Similarly, for `cause-after-effect`, the opposite ordering is enforced.

Algorithm 1 beam-search-based span selector

Input: $P_{s_c}, P_{e_c}, P_{s_{ef}}, P_{e_{ef}}, n, k, m.$

Output: $H = \{(s_1, e_1, s_2, e_2, t_i = CBeforeE/CAfterE) : i \leq m\}$

CBeforeE = $\{p^i_{s_c} + p^j_{e_{ef}} : 1 \leq i, j \leq n\}$.

CAfterE = $\{p^i_{s_{ef}} + p^j_{e_c} : 1 \leq i, j \leq n\}$.

Find position pairs with Top-k largest score from both CBeforeE and CAfterE.

Denote the gotten position pairs set as $PS = \{(sp_i, ep_i, t_i = CBeforeE/CAfterE) : sp_i \leq ep_i\}$. t_i implies whether the pair is retrieved from CBeforeE or CAfterE.

Initialize a min heap H.

for $ps_p = (sp_p, ep_p, t_p)$ in PS **do**

  if t_p = $CBeforeE$ **then**

    Find the position pair (i, j) with the largest $p^i_{e_c} + p^j_{s_{ef}}$, which satisfies $sp_p \leq i \leq j \leq ep_p$.

    Calculate $sc_{(sp_p,i,j,ep_p)} = p^{sp_p}_{s_c} + p^i_{e_c} + p^j_{s_{ef}} + p^{ep_p}_{e_{ef}}$.

  else

    Find the position pair (i, j) with the largest $p^i_{e_{ef}} + p^j_{s_c}$, which satisfies $sp_p \leq i \leq j \leq ep_p$.

    Calculate $sc_{(sp_p,i,j,ep_p)} = p^{sp_p}_{s_{ef}} + p^i_{e_{ef}} + p^j_{s_c} + p^{ep_p}_{e_c}$.

  Push $\{(sp_p, i, j, ep_p), t_p, sc_{(sp_p,i,j,ep_p)}\}$ into H.

  if $len(H) > m$ **then**

    $heappop(H)$ based on $sc_{(sp_p,i,j,ep_p)}$.

return H

Figure 3.8: Beam Search-based Position Selector proposed by [44].

Two separate searches are performed in the logits space, each respecting its discourse

order constraint. From each search, the top-k scoring position pairs are selected (where k is the beam size), resulting in a candidate set of span pairs. This beam search step efficiently narrows down the search space to the most promising candidates.

Finally, the algorithm selects the top-m scoring predictions from this candidate set (where m is the number of final predictions to return), allowing it to identify multiple valid cause-effect span pairs in a single sentence. When present, the highest-scoring signal span is always chosen.

IDIAPers Team The system developed by the IDIAPers team was based on the pre-trained T5 language model and framed causal relation extraction as a generative task. It iteratively generated all classes triplets (Cause/Effect/Signal) in a sentence, conditioning each prediction on previously generated triplets. Different generation orders (e.g., `cause->effect->signal` or `effect->cause->signal`) were tested, with better performance for whichever component was generated first. For each triplet, the model was conditioned three times, once per component, using special decoder prefixes (`_cause`, `_effect`, `_signal`) and including prior triplets as history in the encoder input. Missing signals were represented with an `_empty` token [45].

The SPOCK Team, which also competed in FinCausal 2022, built on their previous work and explored both span-based and sequence tagging strategies [46]. Unlike task 2 of FinCausal 2022, they achieved better results with the span-based model.

The span-based model takes a sequence of tokens as input and predicts the cause, effect and signal spans in the sentence by classifying a list of candidate spans of words. The list of candidate spans is generated by selecting all possible spans of words in the sentence up to maximum span length. The model is based on SpERT [26]. The architecture of SpERT was explained in Chapter 2, specifically in Subsection 2.4.2.

The architecture is presented in Figure 3.9. This image was taken from another paper not related to the competition, where the team used this framework to train on different datasets. In that paper, the team also observed that the proposed framework consistently outperformed sequence tagging baselines across all evaluated datasets [47].

For each candidate span, a span representation is created. This involves concatenating:

- The max-pooled BERT embeddings of all tokens within that span;
- A trained width embedding that represents the size (length) of the candidade span;

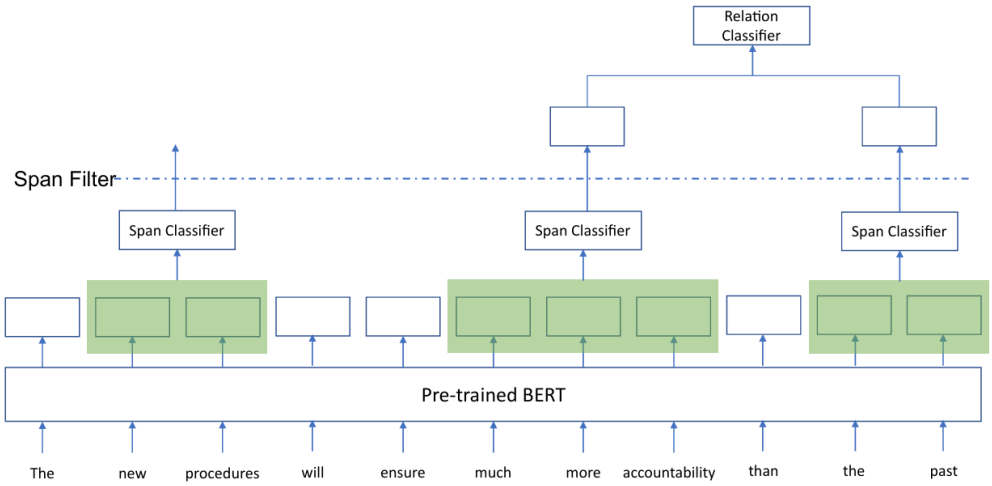


Figure 3.9: Causal SpERT Model [47].

- The [CLS] embedding from BERT.

A span classifier then uses these span embeddings to label each candidate span as either *"Cause"*, *"Effect"*, *"Signal"* or *"Other"*. Following span classification, a relation classifier layer predicts the causal relationship between possible pairs of spans. A pair of spans is represented by concatenating the output embeddings from the span classifier and the max-pooled embeddings of the tokens found between the spans.

For predicting the final cause, effect and signal spans in a text segment, they selected the span with the highest probability in each class.

For sentences containing multiple cause-effect pairs, the SPOCK team treated each distinct pair as a separate training example, ensuring that each instance presented to the model had only one ground-truth cause and one ground-truth effect, as the 1Cademy team also did.

Although the proposed framework was designed to identify multiple cause-effect pairs, the specific format of the training data adopted and the method used to select the final predicted output constrained the reported results to essentially one cause-effect pair per instance.

3.2.2.2 Shared Task 3 of CASE, 2023 Edition:

In this edition, the best results were achieved by **BoschAI Team** [11]. This team treated the task as a sequence tagging problem using the BILOU labeling scheme. To handle

multiple causal relations per sentence, they stacked BILOU labels in three layers separated by pipe operators (|), allowing the model to predict up to three causal relations per sentence. They used only three layers to keep the label space manageable, as the number of sentences with more than three causal relations was very low. The stacked labels observed in training data resulted in approximately 300 distinct three-layer BILOU labels [48].

This stacked labels framework was based on a previous work in the field of NER, in which the authors represent all nested labels for a token as a single concatenated multi-label (using | in examples) [49].

Figure 3.10 shows the modeling technique proposed by BoschAI team, but illustrating only two stacked labels. The phrase *"the clash"* can serve as either the cause of one killing or the effect of a disagreement over the use of a village field. Consequently, this instance contains two causal relations. As shown in Figure 3.10, the word *"clash"* is tagged as *L-ARG0|L-ARG1*, which decodes to the end of a cause span in the first layer and the end of an effect span in the second layer. Since the team used three stacked layers, the final tag for this word would be *L-ARG0|L-ARG1|O*, indicating it is outside any span in the third layer.

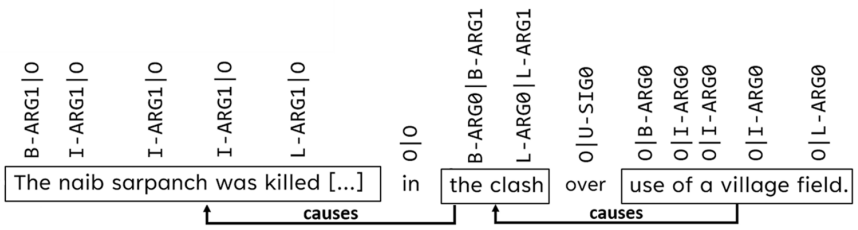


Figure 3.10: Stacked BILOU labels. ARG0 = cause, ARG1 = effect, SIG0 = signal [48].

Since in the CNC dataset, signal spans can be included within either the cause or the effect spans, the team also introduced a notation for these cases, such as *B-ARG0+U-SIG0*.

They used pre-trained BERT-variant models, which generated embeddings that were passed to a linear layer to produce the token-level logits. These logits were then processed by a CRF layer to determine the most likely consistent tag sequence. Notably, they achieved better results with RoBERTa-large than with BERT-large.

During post-processing, a multi-token span was considered valid only if it began with a *B-* tag and ended with an *L-* tag; otherwise, the span was discarded to ensure that only

well-formed spans were predicted. Furthermore, if a layer consisted entirely of *O* tags, no output spans were generated for that case, resulting in fewer than 3 outputs per input in such scenarios.

3.3. Prompting Techniques for Generative AI

Generative AI models, such as GPT-4, have shown remarkable capabilities in a range of NLP tasks. However, as noted in [34] and [33], these models have limitations when applied to causal relation extraction across multiple datasets. Our study aims to evaluate their performance and limitations in the context of causal relation extraction experiments, and to compare them with pre-trained BERT variants.

For this purpose, we will implement modified prompting strategies based on those proposed by [34], adapting them to better suit the causal relation extraction tasks considered in this thesis. Their prompt consists of a causality description, task definition, task name, and output format, as illustrated in Figure 3.11. An additional demonstration component includes instances with their corresponding annotations.

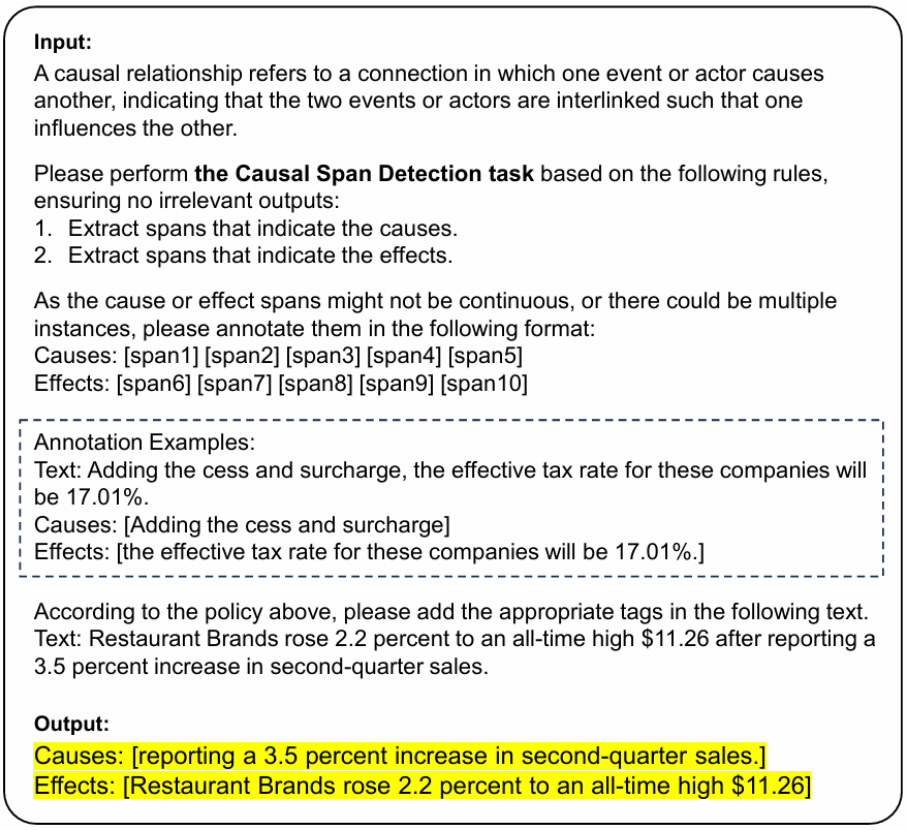


Figure 3.11: Illustrations of prompt for a causal span detection task. Regions outlined with dotted lines indicate demonstrations used in the few-shot setting and are omitted in the zero-shot setting. All text examples are from the FinCausal 2020 dataset [34].

3.4. Summary

Most studies in the literature focus on extracting simple cause-effect pairs, typically using a single-relation-per-input setup, and do not address the extraction of multiple causal relations from the same sequence simultaneously during training. This limitation will be specifically addressed in our experiments.

Not all approaches discussed will be implemented in our study; only those deemed most promising, particularly with respect to the CNC dataset, which serves as the primary dataset for the main development in this thesis, will be adopted.

The frameworks proposed by [44] and [48] will be adopted for this purpose. These frameworks were introduced in the subsection 3.2.2 of the Competitions Section in this chapter.

4. Data and Methodology for Evaluation

The approach to the problem involved the development, training, and evaluation of models using publicly available datasets commonly used by the research community. Some of these datasets had originally been created for competitive challenges. In addition to the datasets, relevant evaluation methodologies and metrics from the literature were employed. This chapter provides a general introduction to these resources and the associated evaluation methodologies.

4.1. Data Resources

Causal relations in text can be either explicit or implicit. A relation is considered explicit when it is marked by a signal, such as a causal connector. However, the explicit relations between arguments can also be expressed through alternative lexicalizations (AltLex), which are phrases or expressions that convey causality without standard connectives and are also usually treated as signals.

Causal relations can also be classified as either intra-sentential or inter-sentential. In intra-sentential causality, the Cause and Effect appear within the same sentence, whereas in inter-sentential causality, they are located in separate sentences [12].

Since causality can be expressed through different representation patterns, it was crucial to train and evaluate the models on a variety of datasets, where causality is not always represented in the same way.

4.1.1 CNC and Benchmark Datasets

The CNC dataset, a relatively recent English-language corpus, was the primary focus of this thesis. As explained in Chapter 3, this dataset is the center of the Shared Task 3 of CASE.

Figure 4.1 illustrates examples of sentences that express causality as well as those that do not. The sentences were taken from the CNC. The first causal example is marked by the explicit causal connector *“due to”*, while the second example indicates causality

through alternative lexical expressions, such as *"created"*. Both expressions are considered causal signals. In the final causal example, an implicit causal relationship is established between *"dissatisfied with the package"* and *"workers staged an all-night sit-in"*. In contrast, sentences that do not express causality lack a cause, an effect, or both [8].

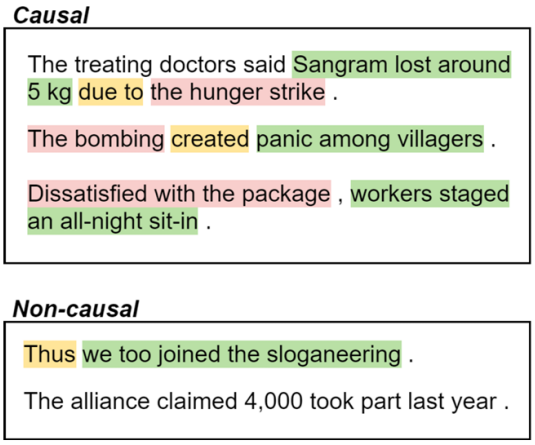


Figure 4.1: Annotated sentences from the CNC. Causes are in pink, Effects in green and Signals in yellow [8].

The CNC corpus includes both explicit and implicit causal relations; however, all relations are restricted to intra-sentential contexts.

In CNC, each sentence contains at least one pair of events, defined as occurrences, actions, or states that hold validity [8]. To annotate causality, the authors adopted the definition of CONTINGENCY from PDTB-3, which applies this label to instances where "one argument provides the reason, explanation, or justification for the situation described by the other" [8].

The PDTB dataset comprises discourse annotations, not limited to causal relationships, on texts from the Wall Street Journal. The PDTB-3 is a large dataset (over 6,000 examples for the CONTINGENCY.CAUSE sense alone) and the authors of CNC believed that it would be advantageous to follow these annotations guidelines [8].

CNC differs from PDTB-3 by focusing on event sentences and permitting more fine-grained arguments that do not necessarily constitute a complete clause. Unlike clauses, which contain a subject and a predicate and can stand as independent or dependent sentences, phrases are smaller units that do not contain both a subject and a predicate and cannot stand alone as complete sentences. This approach of incorporating more varied constructions of causality is similar to the methodology employed in BECauSE 2.0 [8].

Table 4.1 presents the causal datasets used in this project and their annotation coverage, considering factors such as sentence length, linguistic construction, and argument types. Datasets marked “Yes” for inter-sentential annotation also include intra-sentential relations, while those marked “No” contain only intra-sentential annotations.

Table 4.1: Comparison of corpora by inter-sentential annotation, linguistic type, and argument structure.

Corpus	**Inter-sent**	**Linguistic**	**Arguments**
CNC	No	All	Phrases
BECauSE 2.0	No	Explicit	Phrases
AltLex	No	AltLex	Words before/after signal
CausalTimeBank	Yes	All	Event head word(s)
EventStoryLine V1.0	Yes	All	Event head word(s)
PDTB V3.0	Yes	All	Clauses
TED-EN	Yes	All	Clauses
TED-PT	Yes	All	Clauses

In numerous studies on Event Causality Identification, the two datasets used for benchmarking are CausalTimeBank (CTB) and EventStoryLine (ESL). However, in these datasets, arguments are limited to event headwords only, and the complete cause and effect arguments are not fully identified within the sentences.

The TED-Multilingual Discourse Bank (TED-MDB) dataset was also used, specifically its English and Portuguese subsets. TED-MDB is a multilingual resource comprising TED talks annotated at the discourse level in six languages: English, Polish, German, Russian, European Portuguese, and Turkish. The annotations adhere to the objectives and principles established by the PDTB [50].

FinCausal, introduced in the previous chapter, was not used since signals were not annotated.

4.1.2 Pre-processing and Data Analysis

The training and validation sets of the CNC used are from the 2023 edition of Shared Task 3 of CASE [11], with files publicly available on the associated GitHub repository [51]. Specifically, the following files were used:

```
--train_file "/CausalNewsCorpus/data/V2/train_subtask2_grouped.csv"
--validation_file "/CausalNewsCorpus/data/V2/dev_subtask2_grouped.csv"
```

The train file was used for training and development, whereas the validation file was used for evaluation. The annotations for the test set were not available. For clarity, the validation file will be referred to as the test file from now on.

Figure 4.2 illustrates the dataset structure, with the columns defined as follows:

- **corpus**: The name of the dataset or corpus the example belongs to (e.g., `cnc`);
- **doc_id**: Unique identifier for the document within the corpus;
- **sent_id**: Identifier for the sentence within the document;
- **eg_id**: Unique example ID, differentiating multiple examples within the same sentence. In this case, it is always set to 0, as the relations are grouped per entry in the column **causal_text_w_pairs**;
- **index**: A combined unique key string for the example, combining corpus, doc_id, sent_id, and eg_id.
- **text**: The full raw text of the sentence or paragraph containing the causal relation;
- **causal_text_w_pairs**: The annotated text showing causal argument spans marked with `<ARG0>` (Cause), `<ARG1>` (Effect) tags and `<SIG0>` (Signal), if present;
- **num_rs**: The number of causal relations annotated in the example.

corpus	doc_id	sent_id	eg_id	index	text	causal_text_w_pairs	num_rs
cnc	train_01_0	892	0	cnc_train_01_0_892_0	The State alleged they hacked Sabata Petros Chale , 39 , to death in Marikana West , on December 8 , 2016 , allegedly over the allocation of low cost (RDP) houses at Marikana West Extension 2 .	['The State alleged <ARG1>they hacked Sabata Petros Chale , 39 , to death in Marikana West , on December 8 , 2016</ARG1> , allegedly <SIG0>over</SIG0> <ARG0>the allocation of low cost (RDP) houses at Marikana West Extension 2</ARG0> .']	1
cnc	train_01_1	2714	0	cnc_train_01_1_2714_0	Chale was allegedly chased by a group of about 30 people and was hacked to death with pangas , axes and spears .	[]	0
cnc	train_01_100	2680	0	cnc_train_01_100_2680_0	Demonstrators have filed for a permit to hold a rally on Saturday in Yuen Long , the district on the outskirts of Hong Kong where dozens of masked men chased and beat commuters and protesters with wooden poles and metal rods , leaving at least 45 people in hospital .	['<ARG1>Demonstrators have filed for a permit</ARG1> <ARG0><SIG0>to</SIG0> hold a rally on Saturday in Yuen Long</ARG0> , the district on the outskirts of Hong Kong where dozens of masked men chased and beat commuters and protesters with wooden poles and metal rods , leaving at least 45 people in hospital .', 'Demonstrators have filed for a permit to hold a rally on Saturday in Yuen Long , the district on the outskirts of Hong Kong where <ARG0>dozens of masked men chased and beat commuters and protesters with wooden poles and metal rods</ARG0> , <ARG1><SIG0>leaving</SIG0> at least 45 people in hospital</ARG1> .']	2

Figure 4.2: CNC Dataset - with 3 entries.

To test the most promising approaches on the other datasets, the resources were prepared similarly to the files of CNC. The majority of the processing techniques were adopted

text	causal_text_w_pairs	num_rs
The State alleged they hacked Sabata Petros Chale , 39 , to death in Marikana West , on December 8 , 2016 , allegedly over the allocation of low cost (RDP) houses at Marikana West Extension 2 .	['The State alleged <ARG1>they hacked Sabata Petros Chale , 39 , to death in Marikana West , on December 8 , 2016</ARG1> , allegedly <SIG0>over</SIG0> <ARG0>the allocation of low cost (RDP) houses at Marikana West Extension 2</ARG0> .']	1
Chale was allegedly chased by a group of about 30 people and was hacked to death with pangas , axes and spears .	[]	0
Demonstrators have filed for a permit to hold a rally on Saturday in Yuen Long , the district on the outskirts of Hong Kong where dozens of masked men chased and beat commuters and protesters with wooden poles and metal rods , leaving at least 45 people in hospital .	['<ARG1>Demonstrators have filed for a permit</ARG1> <ARG0><SIG0>to</SIG0> hold a rally on Saturday in Yuen Long</ARG0> , the district on the outskirts of Hong Kong where dozens of masked men chased and beat commuters and protesters with wooden poles and metal rods , leaving at least 45 people in hospital .', 'Demonstrators have filed for a permit to hold a rally on Saturday in Yuen Long , the district on the outskirts of Hong Kong where <ARG0>dozens of masked men chased and beat commuters and protesters with wooden poles and metal rods</ARG0> , <ARG1><SIG0>leaving</SIG0> at least 45 people in hospital</ARG1> .']	2

Figure 4.3: CNC Dataset - with 3 entries. Last 3 columns from Figure 4.2 amplified.

from UniCausal [10], with the addition of signal annotations, which were not considered in that repository, as it focused exclusively on cause-effect pairs.

All the datasets, with the exception of CNC, AltLex and BECauSE, contain both intra-sentence and inter-sentence causal relations. In our project, each example was limited to a maximum of 3 sentences, following the approach of [10]. For inter-sentential causal relations, only the sentences containing the Cause and Effect spans were retained, even if they were non-consecutive; extended context was excluded.

With the exception of AltLex, all datasets were randomly split into training and testing sets with an 80/20 ratio. The splits preserved the original distribution of causal and non-causal examples. Discontinuous spans and duplicate entries were removed from the datasets.

AltLex [52] Following [10], the development and gold datasets, annotated by graduate students and crowd workers, respectively, were used in our project. Sentences labeled as REASON (Effect-Signal-Cause) and RESULT (Cause-Signal-Effect) were considered causal, while NONE was treated as non-causal. Based on these templates, causal labels were converted into explicit Cause, Effect and Signal spans. The gold set served as the training set, and the development set served as the test set. After pre-processing,

184 non-causal instances were randomly removed from the test set to match the causal-to-non-causal ratio of the training set.

BECauSE 2.0 [53] In our project, sentences containing an Effect span without a corresponding Cause span were classified as non-causal, since the presence of both spans is necessary for an example to be considered causal. Due to restricted access to the New York Times raw data (subscription required), the annotations from 59 articles were not used.

CTB [9, 54] The pre-processing was very straightforward, except for the limitation on the number of sentences per entry, which was capped at three.

ESL V1.0 [55] Most prior studies treated all `PLOT_LINK` relations as causal. However, the direction of causality is not explicitly indicated in all `PLOT_LINK` instances. Therefore, the flags `CAUSED_BY` and `CAUSES` were used to determine cause and effect in our project. Specifically, if `CAUSED_BY = TRUE`, the `source` is the effect and the `target` is the cause; if `CAUSES = TRUE`, the `source` is the cause and the `target` is the effect. All `PLOT_LINK` relations without either flag set to `TRUE` were excluded. Since some aspects of the ESL annotations remain ambiguous, this dataset was used solely to evaluate performance on causal relations and not on non-causal entries.

PDTB V3.0 [56] and TED-MDB [50] Following the work of [10], the CONTINGENCY sense was treated as causal. Cause and Effect arguments were identified according to the sub-sense:

- The first argument was treated as the Cause span for:
  - CONTINGENCY.CAUSE.RESULT
  - CONTINGENCY.PURPOSE.ARG1-AS-GOAL
  - CONTINGENCY.CONDITION.ARG1-AS-COND
  - CONTINGENCY.NEGATIVE-CONDITION.ARG1-AS-NEGCOND
  - CONTINGENCY.NEGATIVE-CAUSE.NEGRESULT
- The second argument was treated as the Cause span for:

  - CONTINGENCY.CAUSE.REASON
  - CONTINGENCY.PURPOSE.ARG2-AS-GOAL
  - CONTINGENCY.CONDITION..ARG2-AS-COND
  - CONTINGENCY.NEGATIVE-CONDITION.ARG2-AS-NEGCOND

- All other sub-senses were considered non-causal.

Table 4.2: Examples of causal relations from different datasets, showing cause, effect and signal annotations.

Dataset	**Example of causal relation**
BECAUSE	But <ARG1>we must develop innovative, regulatory and oversight responses</ARG1> <SIG0>to</SIG0> <ARG0>keep pace as these market transactions evolve</ARG0>.
CTB	Iraq said it <ARG1>invaded</ARG1> Kuwait <SIG0>because of</SIG0> <ARG0>disputes</ARG0> over oil and money.
ESL	The general <ARG0>strike</ARG0> was <SIG0>staged</SIG0> as a <ARG1>protest</ARG1> against a new round of draconian austerity measures .
AltLex	<ARG0>However, the Italian Socialist Party decided to oppose the war after anti-militarist protestors were killed,</ARG0> <SIG0>resulting in</SIG0> <ARG1>a general strike called Red Week.</ARG1>
PDTB	<ARG1>Philip Morris Cos. posted a 20% jump in third-quarter profit on a 45% revenue increase</ARG1>, <ARG0><SIG0>reflecting</SIG0> strength in the company's cigarette, food and brewing businesses</ARG0>.
TED-EN	<ARG0>It's scattering inside the telescope, creating that very bright image that washes out the planet</ARG0>. <SIG0>So</SIG0> <ARG1>to see the planet, we have to do something about all of that light</ARG1>.

The final counts are provided in Tables 4.3, 4.4, 4.5, 4.6. Table 4.3 shows the number of causal and non-causal entries across datasets, with CNC and PDTB being the two largest datasets.

Table 4.3: Number of causal and non-causal entries across datasets.

	CNC	BEC.	CTB	AltLex	ESL	TED-EN	TED-PT	PDTB
Train - N. Entries								
Non-causal entries	1451	75	774	270	–	47	41	6758
Causal entries	1624	440	221	300	73	73	88	5462
Total	3075	515	995	570	–	120	129	12220
Test - N. Entries								
Non-causal entries	155	20	188	102	–	11	11	1725
Causal entries	185	111	55	115	19	18	22	1366
Total	340	131	243	217	–	29	33	3091

Table 4.4: Breakdown of causal entries into single and multi-relation instances across datasets. “single relation” means exactly one relation; “multi” means more than one relation.

	CNC	BEC.	CTB	AltLex	ESL	TED-EN	TED-PT	PDTB
Train - N. Entries								
With Single Relations	1105	360	188	286	60	66	82	4958
With Multi Relations	519	80	33	14	13	7	6	504
Total	1624	440	221	300	73	73	88	5462
Test - N. Entries								
With Single Relations	133	99	48	111	14	15	21	1247
With Multi Relations	52	12	7	4	5	3	1	119
Total	185	111	55	115	19	18	22	1366

Table 4.5: Number of entries containing 3 or more relations in each dataset.

	CNC	BEC.	CTB	AltLex	ESL	TED-EN	TED-PT	PDTB
Train - N.Entries								
With ≥ 3 relations	101	12	1	1	3	1	1	64
Test - N.Entries								
With ≥ 3 relations	12	0	1	0	0	0	0	16

Table 4.6: Number of causal relations across datasets.

	CNC	BEC.	CTB	AltLex	ESL	TED-EN	TED-PT	PDTB
Train - Total Rel.								
In single-rel entries	1105	360	188	286	60	66	82	4958
In multi-rel entries	1152	175	67	29	31	15	13	1079
Total causal relations	2257	535	255	315	91	81	95	6037
Test - Total Rel.								
In single-rel entries	133	99	48	111	14	15	21	1247
In multi-rel entries	116	24	15	8	10	6	2	255
Total causal relations	249	123	63	119	24	21	23	1502

Tables 4.7 and 4.8 present word count statistics for ARG0 and ARG1 across the training datasets. Overall, ARG0 and ARG1 lengths are relatively short, with most datasets exhibiting median values around 1-10 words. Notably, AltLex and TED-EN show longer tails, as reflected in higher 99th percentile and maximum values. In contrast, CTB and ESL have extremely small spans for both ARG0 and ARG1, with minimal variation. These statistics provide insight into the typical span lengths that the models are required to identify, which is critical for designing appropriate extraction strategies.

Table 4.9 presents word count statistics for SIG0 across the training datasets. Overall, signal spans are very short, with median values of 1-2 words for all datasets.

Table 4.7: Word count statistics for ARG0 (Cause) across training datasets.

Dataset	Min	50th pct	99th pct	Max
CNC	1	8	35	57
BECAUSE	1	6	30	43
CTB	1	1	1	1
AltLex	2	14	62	116
ESL	1	1	6	6
TED-EN	2	9	74	238
TED-PT	1	9	35	42
PDTB	1	10	37	84

Table 4.8: Word count statistics for ARG1 (Effect) across training datasets.

Dataset	Min	50th pct	99th pct	Max
CNC	1	8	33	111
BECAUSE	1	7	29	42
CTB	1	1	1	2
AltLex	2	12	48	88
ESL	1	1	2	3
TED-EN	1	9	38	42
TED-PT	1	9	53	232
PDTB	1	10	35	68

Table 4.9: Word count statistics for SIG0 (Signal) across training datasets.

Dataset	Min	50th pct	99th pct	Max
CNC	1	1	5	8
BECAUSE	1	1	3	4
CTB	1	1	4	4
AltLex	1	2	4	4
ESL	1	1	2	2
TED-EN	1	1	3	3
TED-PT	1	1	4	4
PDTB	1	1	7	25

4.2. Evaluation

The models in this project were evaluated on two tasks: Causal Relation Extraction and Causal vs. Non-Causal Discrimination. The following sections describe the methods used to evaluate each task.

4.2.1 Causal Relation Extraction Task

In this thesis, Precision, Recall, and F1 were computed using the evaluation system from the 2023 edition of Shared Task 3 of CASE, which employs the FairEval implementation of `seqeval`.

Predictions were made on texts where Cause-Effect-Signal spans were directly marked using ARG0, ARG1 and SIG0 start and end boundary markers. Each sentence or paragraph was converted into two tokenized sequences: one representing the token labels for

Cause and Effect, and the other representing the token labels for Signals, since Signals can overlap with Cause or Effect spans.

Evaluation was performed at the relation level. That is, examples containing multiple causal relations were unpacked so that each relation contributed equally to the overall score.

Before computing the metrics, the number of predicted relations was compared against the number of actual relations in the ground truth. Let n_p denote the number of predicted relations and n_a denote the number of actual relations. The evaluation script handled these cases as follows:

- If $n_p > n_a$, only the first n_a predictions were kept;
- If $n_p < n_a$, the missing predictions (from $n_p + 1$ to n_a) were represented by tokens labeled as `Other` (O).

After this process, the code automatically selected the combination of predictions that yielded the highest F1 score. This is illustrated in Figure 4.4.

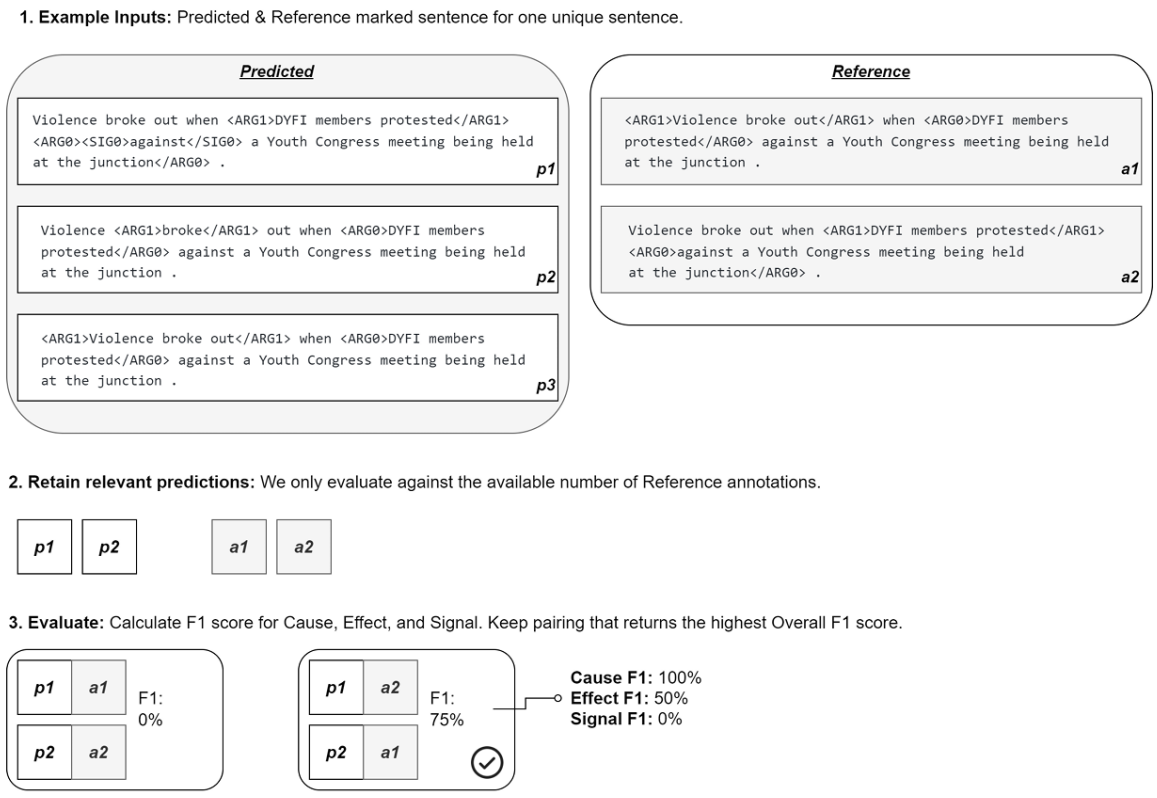


Figure 4.4: An Illustration of how multi-relation examples were processed for sequence evaluation [11].

In the FairEval evaluation system, three additional error types are included. The traditional evaluation considers only true positives (TP), false positives (FP), and false negatives (FN). However, restricting errors to just FPs and FNs does not adequately capture tagging quality when spans overlap between predictions and ground truth. Therefore, in this system, FPs and FNs are reserved exclusively for one-to-zero and zero-to-one mappings. For cases where the prediction overlaps with the ground truth but is not identical, the following errors are applied [57]:

- **LE (labeling error):** identical span, different label;
- **BE (boundary error):** identical label, different (overlapping) span;
- **LBE (labeling-boundary error):** different label, different (overlapping) span.

The formal definitions for traditional and fair modes are given in Equations (4.1), (4.2), (4.3), (4.4) and (4.5).

$$\text{Precision}_{\text{traditional}} = \frac{TP}{TP + FP} \tag{4.1}$$

$$\text{Precision}_{\text{fair}} = \frac{TP}{TP + FP + 0.5 \times (LE + BE + LBE)} \tag{4.2}$$

$$\text{Recall}_{\text{traditional}} = \frac{TP}{TP + FN} \tag{4.3}$$

$$\text{Recall}_{\text{fair}} = \frac{TP}{TP + FN + 0.5 \times (LE + BE + LBE)} \tag{4.4}$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.5}$$

Given sequences of predicted and reference labels, the method first validated label consistency using a BIO tagging scheme. The spans were then converted from standard `seqeval` format into FairEval's span representation, enabling the counting of errors. Finally, the metric computed per-label and overall Precision, Recall, and F1 scores.

To better understand the errors considered in this evaluation, Figure 4.5 shows some examples.

SEQUENCE							
ref_tag	**B-ARG0**	**I-ARG0**	**O**	**O**	**O**	**B-ARG1**	**I-ARG1**
pred_tag	O	O	O	O	O	B-ARG1	I-ARG1

EVALUATION	
ARG0: 1 FN	ARG1: 1 TP

SEQUENCE							
ref_tag	**B-ARG0**	**I-ARG0**	**O**	**O**	**O**	**B-ARG1**	**I-ARG1**
pred_tag	O	O	B-ARG0	I-ARG0	O	B-ARG1	I-ARG1

EVALUATION	
ARG0: 1 FN + 1 FP	ARG1: 1 TP

SEQUENCE							
ref_tag	**B-ARG0**	**I-ARG0**	**O**	**O**	**O**	**B-ARG1**	**I-ARG1**
pred_tag	O	B-ARG0	I-ARG0	I-ARG0	O	B-ARG1	I-ARG1

EVALUATION	
ARG0: 1 BE	ARG1: 1 TP

SEQUENCE							
ref_tag	**B-ARG0**	**I-ARG0**	**O**	**O**	**O**	**B-ARG1**	**I-ARG1**
pred_tag	B-ARG0	I-ARG0	I-ARG0	O	O	B-ARG1	I-ARG1

EVALUATION	
ARG0: 1 BE	ARG1: 1 TP

SEQUENCE							
ref_tag	**B-ARG0**	**I-ARG0**	**O**	**O**	**O**	**B-ARG1**	**I-ARG1**
pred_tag	B-ARG1	I-ARG1	O	O	O	B-ARG0	I-ARG0

EVALUATION	
ARG0: 1 LE	ARG1: 1 LE

SEQUENCE							
ref_tag	**B-ARG0**	**I-ARG0**	**O**	**O**	**O**	**B-ARG1**	**I-ARG1**
pred_tag	B-ARG1	I-ARG1	I-ARG1	O	O	B-ARG0	I-ARG0

EVALUATION	
ARG0: 1 LBE	ARG1: 1 LE

SEQUENCE							
ref_tag	**B-ARG0**	**I-ARG0**	**O**	**O**	**O**	**B-ARG1**	**I-ARG1**
pred_tag	O	O	O	O	O	O	O

EVALUATION	
ARG0: 1 FN	ARG1: 1 FN

Figure 4.5: Examples of errors considered in the evaluation scheme.

As two different evaluation sequences were used (one for Cause-Effect pairs and the other for Signal), the overall Precision, Recall, and F1 are calculated from the totals (summed values of FP, TP, FN, BE, LE, and LBE).

Finally, an exact match score was computed as the proportion of gold relations that were matched exactly by the predicted relations.

4.2.2 Causal vs. Non-Causal Discrimination and Relation Number Calibration

The evaluation described in the previous subsection only considered causal entries. In this subsection, we describe the methods used to evaluate performance on both causal and non-causal entries, using standard classification metrics. Additionally, we outline the approach for measuring the proportion of correct number of causal relations predicted across different test sets. The metrics used are presented below:

- **Non-Causal (Neg) / Causal (Pos):** Class-specific precision, recall, and F1-score for negative (non-causal) and positive (causal) entries.
- **Macro F1:** The average of the F1-scores for both classes, giving equal weight to each class regardless of class imbalance.

- **Balanced Accuracy (BAcc):** The average of recall values for each class, accounting for class imbalance.
- **Proportion of Correct Number of Relations:** The fraction of entries for which the model predicted the correct number of causal relations, regardless of the content of the spans, thereby penalizing only over- or under-prediction.

5. Causal Relation Extraction on the CNC Dataset

This chapter reports the performance of Causal Relation Extraction on the CNC dataset, using only causal entries. As explained in Chapter 4, this thesis focuses on causal relation extraction by adapting pre-trained language models, using the CNC corpus as the main dataset due to its rich annotation of causal relations. This dataset was our reference dataset for development and evaluation, and the findings were later tested on widely used causal corpora.

Three main approaches are presented in this chapter: start/end pointer-based approaches, sequence tagging approaches, and zero-shot/few-shot models. Additionally, two training setups are described for experiments using the pointer-based and sequence tagging approaches:

- **Single-relation-per-input training:** Trains one relation per input.
  The `causal_text_w_pairs` column is expanded to produce a separate input instance for each relation in the text.
- **Multi-relation-per-input training:** Trains on all relations present in a given sentence simultaneously within a single input.
  The `causal_text_w_pairs` column is used in its original form, preserving all relations within the same input.

This distinction is important, since few studies in the literature address the problem using a **multi-relation-per-input training** setup. The objective here is to demonstrate the impact of the multi-relation training setup on improving performance in causal relation extraction.

In the Single-relation-per-input training setting, there were 2257 training entries, corresponding to the total number of causal relations in the training set, as shown in Table 4.6. In the Multi-relation setting, there were 1624 training entries, corresponding to the total number of causal sentences, as seen in Table 4.4.

Each experiment described in the following two sections was run for 10 training epochs.

5.1. Pointer-based Approaches (Start/end Pointer Approaches)

This section presents the results of frameworks that predict start and end positions.

For the single-relation-per-input training setup, the architecture of [44] was used. For the multi-relation-per-input training setup, a new framework was developed within our project. Among the architectures presented in Chapter 3, those employing Pointer-based approaches for causal relation extraction were designed to train only one relation per input.

5.1.1 Single-relation-per-input training

As described, the framework of [44] was used without modification, as it trains only one relation per input. Their framework employs a beam-search algorithm (illustrated in Figure 3.8) for decoding. In our experiments, the beam-search algorithm was configured with a beam size of $k = 5$ and a maximum of $m = 3$ final predictions, with results reported separately for $m = 1$, $m = 2$, and $m = 3$. In CNC, the number of entries containing more than 3 relations is very low.

The top candidate from the beam is always selected, and the predictions are sorted by confidence. Therefore, during evaluation, if predictions need to be discarded (e.g., when the number of predicted relations exceeds the number of true relations), the lowest-confidence predictions are removed first, keeping only the highest-confidence ones up to the number of true relations.

5.1.1.1 Selecting the BERT-variant

First, the dataset was used to evaluate the performance of different BERT-variant models, using $m = 1$. In Table 5.1, the mean and standard deviation of the scores are reported for experiments repeated across five random seeds (42, 123, 777, 999, and 2024). Setting the seed ensures that every step of the pipeline, from pre-processing and data batching to model initialization and GPU computations, produces the same results every run, making training fully reproducible. Results are presented for the best epoch, rather than the last. This was performed using BERT-variant models with pre-trained weights from `HuggingFace` [58–63].

The “sqd2” models are the SQuAD-fine-tuned variants of RoBERTa [59, 61]. The choice of testing these models was motivated by the findings of [39], who observed that architectures predicting start and end positions exhibit greater stability when using models pre-trained for QA, as discussed in Chapter 3.

Table 5.1: Mean and Standard Deviation of F1 scores and Exact Match across seeds using $m = 1$, predicting only one answer per input. “O.” denotes the overall score. “EM” is the Exact Match Ratio.

Model	Cause F1	Effect F1	Signal F1	O. F1	O. EM
roberta-large	0.72 ± 0.01	0.70 ± 0.01	0.70 ± 0.02	0.71 ± 0.01	0.40 ± 0.01
roberta-large-sqd2	0.72 ± 0.01	0.69 ± 0.00	0.72 ± 0.01	0.71 ± 0.01	0.40 ± 0.00
roberta-base	0.72 ± 0.01	0.69 ± 0.01	0.72 ± 0.02	0.70 ± 0.01	0.39 ± 0.01
roberta-base-sqd2	0.70 ± 0.01	0.69 ± 0.01	0.71 ± 0.01	0.70 ± 0.01	0.39 ± 0.01
albert-xxlarge-v2	0.69 ± 0.01	0.70 ± 0.02	0.71 ± 0.02	0.70 ± 0.01	0.40 ± 0.01
bert-large-cased	0.69 ± 0.01	0.70 ± 0.01	0.70 ± 0.01	0.70 ± 0.01	0.37 ± 0.02

From the analysis of Table 5.1, `roberta-large` and `roberta-large-sqd2` achieve the highest Overall F1 scores (0.71 ± 0.01) and also tie for the highest EM values (0.40 ± 0.01 and 0.40 ± 0.00, respectively). Nevertheless, the differences across all models are small, with overlapping standard deviations, indicating that overall performance is very similar.

In the remaining experiments, `roberta-large` is used exclusively.

5.1.1.2 Decoding strategies - Impact of the hyperparameter m

Having selected RoBERTa-large as the BERT variant, the impact of the hyperparameter m (number of output relations) on the CNC will now be assessed. In Tables 5.2 and 5.3, the mean and standard deviation of the scores are reported for different number of predictions m, for experiments repeated across the five random seeds. These results for the CNC test set were obtained from a model trained on the CNC train set.

There is a noticeable decrease in precision for all classes when moving from $m = 1$ to $m = 3$. Conversely, recall improves with higher m values across all classes. This suggests that increasing the number of predictions introduces more false positives, but captures more true positives by allowing multiple predictions per input.

For $m = 2$ and $m = 3$, precision and recall values are more closely aligned, indicating a better balance between the two metrics compared to $m = 1$, where precision is notably higher than recall. For the Cause and Effect spans, the F1 scores remain relatively stable despite the trade-off between precision and recall.

Table 5.2: Mean ± standard deviation of Precision, Recall, and F1 scores across different seeds for different numbers of predictions m, using roberta-large trained on the CNC dataset.

m	Class	Precision	Recall	F1
$m = 1$	**Cause**	0.86 ± 0.01	0.62 ± 0.01	0.72 ± 0.01
	Effect	0.85 ± 0.01	0.60 ± 0.01	0.70 ± 0.01
	Signal	0.82 ± 0.04	0.61 ± 0.02	0.70 ± 0.02
	Overall	0.84 ± 0.01	0.61 ± 0.00	0.71 ± 0.01
$m = 2$	**Cause**	0.75 ± 0.01	0.75 ± 0.01	0.75 ± 0.01
	Effect	0.74 ± 0.01	0.69 ± 0.01	0.71 ± 0.01
	Signal	0.65 ± 0.02	0.64 ± 0.01	0.65 ± 0.01
	Overall	0.72 ± 0.00	0.70 ± 0.01	0.71 ± 0.00
$m = 3$	**Cause**	0.72 ± 0.01	0.77 ± 0.01	0.74 ± 0.01
	Effect	0.71 ± 0.01	0.71 ± 0.01	0.71 ± 0.01
	Signal	0.62 ± 0.02	0.64 ± 0.01	0.63 ± 0.01
	Overall	0.69 ± 0.00	0.71 ± 0.01	0.70 ± 0.00

Table 5.3: Mean ± standard deviation of EM Ratio across different seeds for different numbers of predictions m, using roberta-large trained on the CNC dataset.

Metric	$m = 1$	$m = 2$	$m = 3$
EM	0.40 ± 0.01	0.42 ± 0.01	0.42 ± 0.00

Signal precision decreases noticeably as m increases, dropping from 0.82 ± 0.04 at $m = 1$ to 0.62 ± 0.02 at $m = 3$. This larger decrease compared to other classes suggests that predicting multiple answers introduces more false positives specifically for Signals. Signal recall improves moderately with increasing m, from 0.61 ± 0.02 at $m = 1$ to about 0.64 ± 0.01 at $m = 3$. While recall increases, the gain is relatively small compared to the precision loss.

In summary, the Signal class is more sensitive to increasing the number of predictions, with precision dropping substantially and only marginal recall gains, resulting in a decrease in F1.

It is important to note that, in this architecture, signal detection is performed at the sentence level using the [CLS] token. However, some sentences may contain a signal for one relation but not for others. Additionally, the identified signal span is always the same across all predicted relations within a sentence, even when the true signal spans differ for each relation, because the span with the highest start and end positions is solely selected; no search is performed for the signal class. As a result, the identified signal may not correspond to the true signal for a specific cause-effect pair, representing one of the main limitations of this architecture under a single-relation-per-input training setup.

Finally, signal classification/detection was performed using a separate model rather than a

joint approach (i.e., without sharing weights with the span predictions/extraction). RoBERTa was trained specifically for the signal classification task, and its logits were then used to predict the presence or absence of a signal for a given entry. However, the final results for the Signal span prediction/extraction were worse.

To improve performance and enable the extraction of multiple relations from a single sentence, a different approach is required, since increasing m did not lead to an overall performance gain; therefore, a multi-relation-per-input setup is proposed next.

5.1.2 Multi-relation-per-input training

5.1.2.1 Proposed Framework

Our proposed framework is a transformer-based model that predicts up to a fixed number of causal relations per input, producing start and end token logits for the Cause, Effect, and Signal spans of each relation.

Our proposed framework is inspired by the architecture of [44], discussed in the previous subsection, although that method trains on a single relation per input. Similarly, training in this new setup is performed exclusively on entries containing causal relations, and joint signal detection is performed to flag the presence of a signal.

Among all the architectures introduced in Chapter 3, those employing QA-based approaches for causal relation extraction were designed to train only one relation per input. In contrast, the framework proposed in this thesis is designed to train on all relations present in a given sentence simultaneously within a single input.

While our framework extends MTMSN, introduced in Chapter 2, Subsection 2.4.1, it differs in that it is specifically designed for causal relation extraction with fixed classes (Cause, Effect, and Signal) and a predefined number of relation instances, whereas MTMSN addresses QA with a dynamic number of spans.

To illustrate the proposed approach, Algorithm 1 presents a pseudo-implementation. The setup and key steps are described below:

- **Backbone**: A transformer (e.g., RoBERTa);
- **Fixed number of slots**: $\mathbf{max\_relations}$ is a hyperparameter - the model always allocates exactly these many “relation slots” per example, regardless of how many relations actually occur in the text:

  - If an entry contains fewer than **max_relations** actual relations, the remaining slots are filled with a special position -100;
  - If an entry has more than the maximum allowed, only the first **max_relations** are kept, and the rest are discarded.

- **Gold Signal Flag**: The gold signal flag per sequence is set to 1 if any relation in the sequence has a valid signal span, and 0 otherwise;
- **Classification layers**: The classification layers are initialized as linear layers, specifically the span-position classifier and the signal-flag classifier;
- **Output design - spans**: For each possible causal relation (up to **max_relations**), the Algorithm 1 outputs $z_{\text{span\_positions}}$ that has:
  - start token logits and end token logits of the **cause span**;
  - start token logits and end token logits of the **effect span**;
  - start token logits and end token logits of the **signal span**;
- **Output - Signal Detection**: Additionally, the Algorithm 1 outputs signal-classification logits $z_{\text{sig0\_flag}}$;
- **Mean-Adjusted Loss**: In our approach, for each example in the batch, the loss is computed separately for each relation and class (Cause, Effect, Signal), assigning zero to the loss for positions corresponding to absent spans. Thus, for each example and each class, there is a fixed number of loss values corresponding to the predefined **max_relations**, regardless of how many relations are actually present in the sequence.

  The final loss per-class is then obtained by averaging over all relations and all examples in the batch;
- **Output - Total Loss**: Total loss L_{total} is computed as the mean of the losses for the class spans (ARG0, ARG1, and SIG0) and the signal-classification, ensuring equal weighting for all components.

It is important to note that the total loss formulation described in Algorithm 1 defines the training objective, which is minimized using the AdamW optimizer. Furthermore, before decoding, `log-softmax` is applied to the span-position logits $z_{\text{span\_positions}}$.

Algorithm 1 Pointer-based Multi-relation-per-input setup

Input: Pretrained encoder M, tokenized input x, gold start positions $y_{r,cl}^{(\text{start})}$, gold end positions $y_{r,cl}^{(\text{end})}$, where $r \in [1, max\_relations]$ and $cl \in \{\text{ARG0}, \text{ARG1}, \text{SIG0}\}$, and gold signal-flag per sequence $y_{\text{sig0\_flag}} \in \{0, 1\}$.
Set the number of relations R to the hyperparameter **max_relations**
Set the batch size B to the hyperparameter **batch_size**
Initialize a span-position classifier $f_{\text{span\_positions}}$
Initialize a signal classifier $f_{\text{sig0\_flag}}$
Extract hidden states for the input sequence, $h \leftarrow M(x)$
Compute span-positions logits, $z_{\text{span\_positions}} \leftarrow f_{\text{span\_positions}}(h)$
Extract $z_{r,cl}^{(start)}$ and $z_{r,cl}^{(end)}$ from $z_{\text{span\_positions}}$
Compute binary signal logits per sequence, $z_{\text{sig0\_flag}} \leftarrow f_{\text{sig0\_flag}}(h_{[CLS]})$
for $r = 1 \ldots R$ **do**
    for each class $cl \in \{\text{ARG0}, \text{ARG1}, \text{SIG0}\}$ **do**
        $\ell_{r,cl} = \frac{1}{2}\left[\mathcal{L}_{\text{CE}}(z_{r,cl}^{(start)}, y_{r,cl}^{(\text{start})}) + \mathcal{L}_{\text{CE}}(z_{r,cl}^{(end)}, y_{r,cl}^{(\text{end})})\right]$
    end for
end for
Compute ARG0 scalar loss, $L_{ARG0} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{R}\sum_{r=1}^{R}\ell_{r,arg0}[b]$
Compute ARG1 scalar loss, $L_{ARG1} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{R}\sum_{r=1}^{R}\ell_{r,arg1}[b]$
Compute SIG0 scalar loss, $L_{SIG0} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{R}\sum_{r=1}^{R}\ell_{r,sig0}[b]$
Compute loss for the signal classifier, $L_{\text{CLS}} = \frac{1}{B}\sum_{b=1}^{B}\mathcal{L}_{\text{CE}}(z_{\text{sig0\_flag}}[b], y_{\text{sig0\_flag}}[b])$
Compute total loss, $L_{\text{total}} = \frac{1}{4}(L_{\text{ARG0}} + L_{\text{ARG1}} + L_{\text{SIG0}} + L_{\text{CLS}})$
Output: Span-position logits $z_{\text{span\_positions}}$, signal-flag logits $z_{\text{sig0\_flag}}$, loss L_{total}

Within the proposed framework, `roberta-large` was trained under the Multi-relation-per-input setting. For decoding, we applied beam search (Figure 3.8) with parameters $k = 5$ and $m = 1$. In this case, the model produces up to **max_relations** cause-effect pairs by iterating over relation-slot logits. For each relation index, the beam-search algorithm is applied to the start and end probabilities of cause and effect.

The final score for each cause and effect pair is given by Equation (5.1):

$$\text{score} = \log \text{P}(\text{cause}_{\text{start}}) + \log \text{P}(\text{cause}_{\text{end}}) + \log \text{P}(\text{effect}_{\text{start}}) + \log \text{P}(\text{effect}_{\text{end}}) \quad (5.1)$$

The beam-search algorithm ensures that only one non-overlapping pair of cause and effect spans is decoded per relation-slot.

Before decoding signal spans, the model first determines whether a signal exists in the sequence using the signal-classification logits $z_{\text{sig0\_flag}}$, producing a Boolean-like flag, `has_signal`, which indicates whether signal span decoding should be performed for all relation slots. If `has_signal = 1`, signal span prediction is performed per relation: the start token is chosen as the position with the highest start logit, then candidate end positions are restricted to a fixed window, and finally, the end token with the highest end logit within that window is selected. This procedure enforces a maximum span length and ensures a valid start-end ordering for the predicted signal.

Signal spans are decoded separately from the Cause and Effect spans, as a signal can be included within either argument. For example:

> *"Demonstrators have filed for a permit to hold a rally on Saturday in Yuen Long , the district on the outskirts of Hong Kong where* ***<ARG0>****dozens of masked men chased and beat commuters and protesters with wooden poles and metal rods* ***</ARG0>*** *,* ***<ARG1><SIG0>****leaving* ***</SIG0>*** *at least 45 people in hospital* ***</ARG1>*** *."*

After analyzing the training set distribution, **max_relations** was set to 3 per sentence, and the window for signal span prediction was limited to 5.

5.1.2.2 Initialization Strategies

Additionally, given the impact of initialization on convergence and final performance, two *warm-start* strategies were examined:

- In the *standard warm-start* approach, training is initialized using the pre-trained RoBERTa weights;
- In the *task-specific warm-start* approach, training is initialized from the weights obtained at the last epoch of the Single-relation-per-input training model, leveraging previously learned representations to improve Multi-relation-per-input performance.

Given the added complexity of the Multi-relation-per-input training setup, it was important to leverage the pre-training from the Single-relation-per-input setting, which focuses solely

on learning the patterns of one relation per input. This Single-relation-training phase allows the model to develop a more stable understanding of the task, without being exposed to several relations simultaneously. The results of the different *warm-start* strategies are presented in Tables 5.4 and 5.5.

Table 5.4: Performance (mean ± standard deviation) across different seeds in a multi-relation-per-input training, using a pointer approach.

Configuration	Class	Precision	Recall	F1
Standard warm-start	**Cause**	0.77 ± 0.01	0.78 ± 0.01	0.77 ± 0.00
	Effect	0.75 ± 0.02	0.76 ± 0.02	0.75 ± 0.02
	Signal	0.65 ± 0.02	0.78 ± 0.01	0.71 ± 0.01
	Overall	0.73 ± 0.01	0.77 ± 0.01	0.75 ± 0.01
Task-specific warm-start	**Cause**	0.77 ± 0.01	0.78 ± 0.01	0.78 ± 0.01
	Effect	0.76 ± 0.02	0.76 ± 0.01	0.76 ± 0.01
	Signal	0.67 ± 0.01	0.79 ± 0.01	0.73 ± 0.01
	Overall	0.74 ± 0.01	0.78 ± 0.01	0.76 ± 0.01

Table 5.5: Mean ± standard deviation of EM Ratio across different seeds in a multi-relation-per-input training, using a pointer approach.

Metric	*Standard*	*Task-specific*
EM	0.46 ± 0.02	0.47 ± 0.01

It was observed that one of the limitations of this training set-up was that the predictions for different relations sometimes exhibited a high degree of similarity. The proposed framework outputs three relations per entry independently; that is, relation 2 does not take into account the output of relation 1, and relation 3 does not take into account the outputs of relations 1 and 2. To address this, experiments incorporating an overlap penalization term were conducted. The details and results of these trials are presented in Appendix A. Additionally, this appendix includes a comparison of results obtained using the Standard Loss Averaging approach, which excludes ignored positions during loss computation.

The Mean-Adjusted Loss Approach, which includes the ignored positions (as proposed in our framework), performed slightly better than the Standard Loss Approach, while the overlap penalization term did not lead to performance improvements. These findings indicate that careful handling of ignored positions in the loss is beneficial, whereas explicit penalization of overlapping relations is unnecessary in this setup.

5.1.2.3 Prediction Refinement: Duplicate Removal and Confidence-Based Sorting

Finally, a filtering step was added to the output predictions:

- (1) Since the model is forced to output 3 relations per instance, many outputs contained duplicate relations when fewer than 3 true relations existed. In the test set, after filtering out duplicates, only 31 out of 185 entries retained 3 distinct relations;
- (2) Additionally, predictions were sorted in descending order of their scores, before evaluation, using the scores defined in Equation (5.1). These scores are the model's *confidence scores* for cause-effect pairs. Recall from Chapter 4 that extra predictions beyond the number of actual relations (ground truth) are discarded during evaluation.

An increase in overall mean F1 from 0.76 to 0.78 was observed when moving from non-filtered and non-sorted to filtered and sorted predictions. The exact match ratio also improved, rising from 0.47 to 0.49. In summary, the best-performing configuration employed the *task-specific warm-start* approach, with predictions filtered to remove duplicate relations and sorted prior to evaluation.

5.1.3 Comparison of Training Setups

Tables 5.6 and 5.7 present a comparison between single-relation-per-input and multi-relation-per-input training, with results reported for the best-performing configuration. Evaluation is defined as follows:

- *Single:* Evaluation restricted to entries with a single relation in the ground truth;
- *Multi:* Evaluation restricted to entries with more than one relation in the ground truth;
- *Overall:* Evaluation performed over all entries.

From Tables 5.6 and 5.7, it can be seen that Multi-rel-per-input training setup outperforms Single-rel-per-input training setup on *Multi-Relation* instances, while still preserving strong performance on *Single-Relation* cases. This leads to improved overall F1 and EM. These findings indicate that training with multiple relations per input provides a better balance across both *Single-Relation* and *Multi-Relation* sets, resulting in a more robust model overall.

Focusing only on the Multi-relation-per-input training setup (last 3 columns of Table 5.6), the model shows the highest precision when identifying Causes and Effects, and the highest recall when identifying Signals.

Table 5.6: Performance comparison between single-relation-per-input training and multi-relation-per-input training, using pointer approaches. (*)*Task-specific warm-start* approach, with filtered and sorted predictions.

Class	Single-rel-per-input training, $m = 1$			Multi-rel-per-input training (*)		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Single						
Cause	0.89 ± 0.00	0.89 ± 0.00	0.89 ± 0.00	0.89 ± 0.02	0.87 ± 0.02	0.88 ± 0.02
Effect	0.86 ± 0.01	0.85 ± 0.00	0.85 ± 0.00	0.84 ± 0.01	0.82 ± 0.01	0.83 ± 0.01
Signal	0.83 ± 0.02	0.85 ± 0.01	0.84 ± 0.01	0.82 ± 0.02	0.86 ± 0.01	0.84 ± 0.01
all	0.86 ± 0.01	0.86 ± 0.00	0.86 ± 0.00	0.85 ± 0.01	0.85 ± 0.01	0.85 ± 0.01
Multi						
Cause	0.77 ± 0.03	0.32 ± 0.02	0.45 ± 0.02	0.72 ± 0.03	0.67 ± 0.03	0.69 ± 0.03
Effect	0.82 ± 0.03	0.34 ± 0.02	0.48 ± 0.02	0.76 ± 0.01	0.67 ± 0.01	0.71 ± 0.00
Signal	0.77 ± 0.06	0.36 ± 0.03	0.49 ± 0.04	0.62 ± 0.02	0.74 ± 0.01	0.68 ± 0.01
all	0.79 ± 0.02	0.34 ± 0.01	0.47 ± 0.01	0.70 ± 0.02	0.69 ± 0.01	0.69 ± 0.01
Overall						
Cause	0.86 ± 0.01	0.62 ± 0.01	0.72 ± 0.01	0.82 ± 0.02	0.78 ± 0.01	0.80 ± 0.01
Effect	0.85 ± 0.01	0.60 ± 0.01	0.70 ± 0.01	0.80 ± 0.01	0.75 ± 0.01	0.78 ± 0.01
Signal	0.82 ± 0.04	0.61 ± 0.02	0.70 ± 0.02	0.72 ± 0.01	0.80 ± 0.01	0.76 ± 0.01
all	0.84 ± 0.01	0.61 ± 0.00	0.71 ± 0.01	0.78 ± 0.01	0.77 ± 0.00	0.78 ± 0.01

Table 5.7: EM Ratio scores for single-rel-per-input training vs. multi-rel-per-input training, using pointer approaches. (*)*Task-specific warm-start* approach, with filtered and sorted predictions.

Metric	Single-rel-per-input training, $m = 1$			Multi-rel-per-input training(*)		
	Single	*Multi*	*Overall*	*Single*	*Multi*	*Overall*
EM	0.60 ± 0.01	0.16 ± 0.03	0.40 ± 0.01	0.60 ± 0.03	0.36 ± 0.03	0.49 ± 0.00

5.2. Sequence Tagging Approaches

It is important to note that all the experiments described in the previous section were performed in the Virtual Artificial Intelligence Laboratory (AIvLAB) on the Deucalion platform in a multi-GPU setting. This was possible for the architectures predicting start and end positions. However, when using the sequence tagging architectures, all tests had to be run on Google Colab in a single-GPU environment. The reason for this change in computing resources is explained in the Appendix B.

Each experiment was conducted with a single trial, consistently using the same random seed due to computational constraints imposed by running on Google Colab.

The framework proposed in [48] was used in the work described in this section, with some modifications. As outlined in Chapter 3, it utilizes BILOU labeling and a CRF output layer. This framework achieved the highest performance in the 2023 edition of Shared Task 3 of CASE, using a multi-relation-per-input training setup.

`Roberta-large` was used in all experiments presented in this section, as it was the best-performing model in the study explained in [48].

5.2.1 Single-relation-per-input training

The framework of [48] was adapted for this part of the experiments, as it was originally designed to handle training with multiple relations per input. In this adaptation, a total of 34 tags were used (e.g., “O”, “B-ARG0”, “I-ARG0”, “L-ARG0”, “U-ARG0”, “B-ARG1+U-SIG0”), where ARG0 denotes the cause, ARG1 the effect, and SIG0 the signal, as explained before.

5.2.2 Multi-relation-per-input training

The framework of [48], used here without modification, was specifically designed to support training with multiple relations per input. A total of 298 distinct three-layer BILOU labels were used (e.g., “O|O|O”, “B-ARG0|B-ARG0|I-ARG0”, “B-ARG1+B-SIG0|B-ARG0|O”).

For the experiments in Google Colab, the results of the Single-relation-per-input training and Multi-relation-per-input-training are shown in Tables 5.8, 5.9 and 5.10.

Table 5.8: Class-level performance comparison between single-relation-per-input training and multi-relation-per-input training, using sequence tagging approaches.

Config.	Class	Single-rel-per-input training			Multi-rel-per-input training		
		Precision	Recall	F1	Precision	Recall	F1
Stand.	**Cause**	0.88	0.40	0.55	0.84	0.62	0.72
	Effect	0.90	0.51	0.65	0.83	0.61	0.70
	Signal	0.80	0.61	0.70	0.83	0.69	0.75
	Overall	0.86	0.49	0.63	0.83	0.64	0.72
Task-sp.	**Cause**	—	—	—	0.86	0.64	0.73
	Effect	—	—	—	0.84	0.67	0.74
	Signal	—	—	—	0.82	0.71	0.76
	Overall	—	—	—	0.84	0.67	0.74

Table 5.8 confirms that, for sequence tagging approaches, Multi-relation-per-input training with *task-specific initialization* achieves higher F1 scores, consistent with the conclusions observed in the Pointer Approach section.

Table 5.9 provides a more detailed breakdown, revealing that while Single-rel-per-input training performs well on *Single-Relation* evaluation cases, it struggles significantly on *Multi-Relation* cases. In contrast, the *task-specific* Multi-rel-per-input training yields more balanced and higher overall F1 scores. Finally, Table 5.10 confirms this trend at the

Table 5.9: Performance comparison between Single-relation-per-input training and Multi-relation-per-input training, using sequence tagging approaches, broken down by Class-level, Evaluation-set, and overall results. (*)*Task-specific warm-start* approach.

Class	Single-rel-per-input training			Multi-rel-per-input training(*)		
	Precision	Recall	F1	Precision	Recall	F1
Single						
Cause	0.92	0.65	0.76	0.90	0.80	0.85
Effect	0.91	0.74	0.81	0.84	0.77	0.81
Signal	0.79	0.82	0.80	0.81	0.81	0.81
all	0.88	0.72	0.79	0.85	0.79	0.82
Multi						
Cause	0.68	0.12	0.20	0.79	0.45	0.57
Effect	0.88	0.26	0.40	0.84	0.55	0.67
Signal	0.83	0.40	0.54	0.85	0.62	0.72
all	0.82	0.24	0.37	0.83	0.53	0.65
Overall						
Cause	0.88	0.40	0.55	0.86	0.64	0.73
Effect	0.90	0.51	0.65	0.84	0.67	0.74
Signal	0.80	0.61	0.70	0.82	0.71	0.76
all	0.86	0.49	0.63	0.84	0.67	0.74

Table 5.10: EM Ratio for Single-rel-per-input vs. Multi-rel-per-input training, using sequence tagging approaches. (*)*Task-specific warm-start* approach.

Metric	Single-rel-per-input training			Multi-rel-per-input training(*)		
	Single	*Multi*	*Overall*	*Single*	*Multi*	*Overall*
EM	0.43	0.07	0.26	0.53	0.27	0.41

exact-match level, where Multi-rel-per-input training achieves higher EM ratios across both *Single* and *Multi* cases.

In the sequence tagging approach with multi-relation-per-input training, the CRF operates over a stacked tag set (e.g., B-ARG0|I-ARG1|O), and Viterbi decoding yields a single most probable global sequence of such stacked tags. When this sequence is later decomposed into three layers (three outputs/predictions), these layers represent different projections of the same top-1 joint prediction rather than independent predictions. Consequently, the first layer is not more or less confident than the others, but simply a component of the overall most probable sequence. Since the CRF assigns scores only to stacked labels, confidence values cannot be directly interpreted at the per-layer level. So, the predictions can not be sorted as it were for the Pointer Approach.

Additionally, it is worth noting that, unlike the Pointer-based approach, the sequence tagging approach didn't produce duplicate predictions for the same entry.

5.3. Zero-Shot and Few-Shot Models

To further evaluate the performance of our task-specific models, we conducted a comparison with a large generative AI model. This allows us to assess how models trained specifically for causal relation extraction on smaller architectures perform in comparison with a much larger model.

The experiments were conducted using the Gemini Flash API. A key advantage of this API is the availability of a free daily quota, allowing extensive testing across multiple shot settings without incurring costs [64].

We employed a modified version of the prompting strategy proposed by [34], discussed in Chapter 3. The prompt used in our experiments is illustrated in Figure 5.1. The region outlined with purple color and dotted lines indicates demonstrations used in the few-shot setting; these are omitted in the zero-shot setting. All text annotation examples are taken from the CNC dataset.

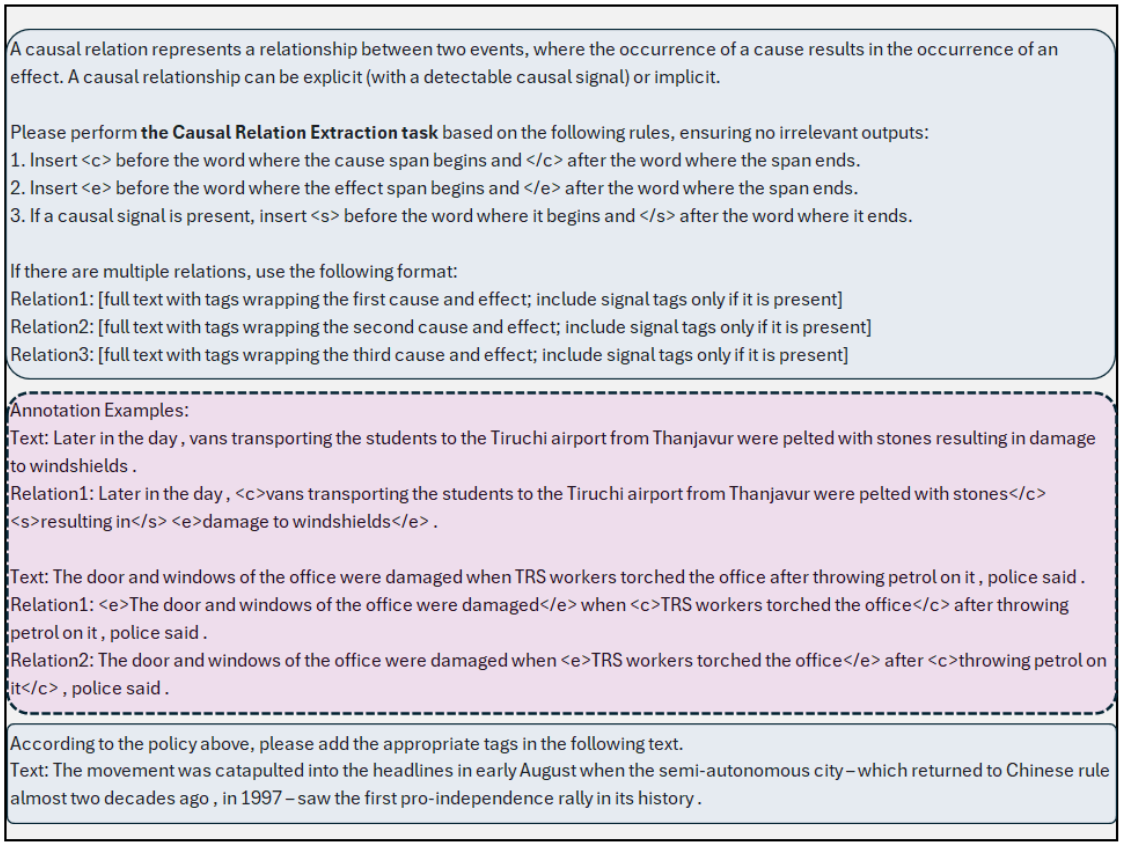


Figure 5.1: Illustration of the prompt used to evaluate Gemini API on the CNC Dataset. The region outlined with purple color and dotted lines indicates demonstrations used in the few-shot setting; these are omitted in the zero-shot setting. All text annotation examples are taken from the CNC dataset.

Table 5.11 summarizes the results, reporting F1 scores for each class as well as Overall F1 and Exact Match (EM) scores for each model.

Table 5.11: Performance comparison of Gemini 2.0 and 2.5 Flash models across different shot settings.

Gemini Model	Class | Metric	0-Shot	2-Shot	10-Shot	30-Shot
2.0 Flash	Cause | F1	—	0.27	0.47	0.52
	Effect | F1	—	0.16	0.39	0.44
	Signal | F1	—	0.42	0.45	0.47
	Overall | F1	—	0.28	0.43	0.48
	Overall | EM	—	0.12	0.16	0.22
2.5 Flash	Cause | F1	0.58	0.63	0.68	0.67
	Effect | F1	0.56	0.65	0.72	0.72
	Signal | F1	0.58	0.59	0.65	0.64
	Overall | F1	0.57	0.63	0.69	0.69
	Overall | EM	0.15	0.21	0.26	0.26

Gemini 2.5 Flash demonstrates substantial gains over Gemini 2.0 Flash across all shot settings and evaluation metrics. Overall, increasing the number of shots boosts performance, but for 2.5 Flash it stabilizes beyond 10.

Figure 5.2 presents a comparison between Gemini 2.0 Flash (30-Shot) and Gemini 2.5 Flash (10-Shot) in their ability to correctly identify causal relationships in text, where each of the three texts contains only a single causal relationship. The causal relationship in the ground truth is marked in the first column.

In the second example shown in Figure 5.2, Gemini 2.0 Flash model mistakenly swapped the cause with the effect, thus misidentifying the causal direction.

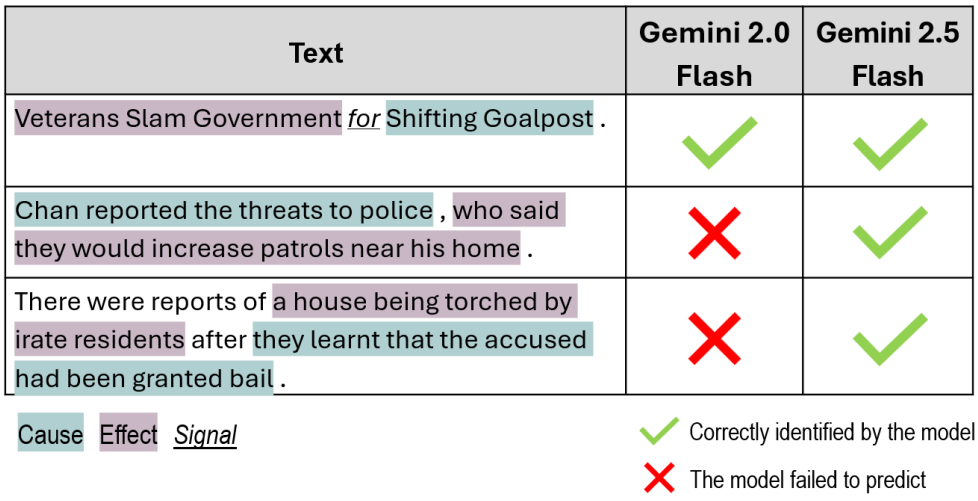


Figure 5.2: Comparing Predictions: Gemini 2.0 Flash (30-Shot) vs Gemini 2.5 Flash (10-Shot).

5.4. Comparison of Models/Approaches

For the experiments in Google Colab, the results of the start/end pointer and sequence tagging approaches are shown in Tables 5.12 (Precision, Recall, and F1) and 5.13 (EM). The Pointer Approach generally outperforms the Sequence Tagging Method across different initialization strategies and classes. However, without filtering and sorting its predictions, the Sequence Tagging Approach achieves slightly higher F1 score for the Signal class. Once predictions are filtered and sorted, the Pointer Approach achieves the highest F1 scores across all classes, demonstrating the effectiveness of the post-processing steps.

Table 5.12: Performance in a multi-relation-per-input training, using pointer and sequence tagging approaches in Google Colab. *Results with filtered and sorted predictions.

Config.	Class	Start/End Pointer Approach			Sequence Tagging Approach		
		Precision	Recall	F1	Precision	Recall	F1
Stand.	**Cause**	0.72	0.75	0.73	0.84	0.62	0.72
	Effect	0.73	0.74	0.73	0.83	0.61	0.70
	Signal	0.68	0.78	0.73	0.83	0.69	0.75
	Overall	0.71	0.75	0.73	0.83	0.64	0.72
Task-sp.	**Cause**	0.75	0.77	0.76	0.86	0.64	0.73
	Effect	0.74	0.76	0.75	0.84	0.67	0.74
	Signal	0.69	0.82	0.75	0.82	0.71	0.76
	Overall	0.73	0.78	0.75	0.84	0.67	0.74
*Task-sp.**	**Cause**	0.79	0.77	0.78	—	—	—
	Effect	0.80	0.76	0.78	—	—	—
	Signal	0.74	0.83	0.78	—	—	—
	Overall	0.78	0.78	0.78	—	—	—

Table 5.13: EM Ratio in a multi-relation-per-input training, using pointer and sequence tagging approaches in Google Colab. *Results with filtered and sorted predictions.

Metric	Start/End Pointer Approach			Sequence Tagging Approach	
	Standard	*Task-specific*	*Task-specific**	*Standard*	*Task-specific*
EM	0.43	0.46	0.49	0.38	0.41

From each of the experiments described in the previous sections (covering pointer-based approaches, sequence tagging approaches, and zero-shot/few-shot models), the model with the highest performance metrics was selected for inclusion in the final overall comparison. Table 5.14 summarizes the results, highlighting differences in performance across models/approaches and demonstrating which approaches are more effective under various evaluation conditions.

As shown in Table 5.14, *Single-Relation* evaluation set highlights `Pointer_3RelTrain_Best` as the top performer, excelling across all metrics. `Gemini_2.5Flash_Best`, on the other

Table 5.14: Performance comparison of different models using Precision, Recall, F1, and EM ratio. Results for the Sequence Tagging and Pointer Approach models are from runs in Google Colab.

Model	Evaluation Set	Precision	Recall	F1	EM
Gemini_2.5Flash_Best	*Single*	0.71	0.75	0.73	0.30
	Multi	0.67	0.60	0.63	0.21
	Overall	0.69	0.68	0.69	0.26
BILOU+CRF_3RelTrain_Best	*Single*	0.85	0.79	0.82	0.53
	Multi	0.83	0.53	0.65	0.27
	Overall	0.84	0.67	0.74	0.41
Pointer_3RelTrain_Best	*Single*	0.87	0.87	0.87	0.62
	Multi	0.68	0.67	0.68	0.34
	Overall	0.78	0.78	0.78	0.49

hand, shows more moderate performance across all metrics. In the *Multi-Relation* evaluation set, `Gemini_2.5Flash_Best` continues to lag behind the others in both F1 and EM. Overall, `Pointer_3RelTrain_Best` stands out with the highest F1 score (0.78) and EM ratio (0.49), offering a balanced performance between Precision and Recall.

Figure 5.3 shows the TP, FP, FN, BE and LE+LBE values for the previously mentioned approaches.

In the Cause and Effect classes, `Pointer_3RelTrain_Best` leads in TP, has the lowest FN, and has a solid balance between FP and FN. `BILOU+CRF_3RelTrain_Best` has the lowest FP, but its high FN is a significant drawback. `Gemini_2.5Flash_Best` performs moderately well in TP and FP, but struggles with higher boundary and labeling errors.

In the Signal class, `Pointer_3RelTrain_Best` again achieves the highest TP and lowest FN. However, the number of FP is nearly twice that of FN. This indicates that signals are frequently predicted in relations where no signal is present. Because the Signal Classification Task is not perfectly accurate, errors propagate to the subsequent stage, causing the model to predict signal spans even when none exist. Additionally, the Signal Classification Task is performed at the entry level, using the [CLS] token, rather than at the relation level. This introduces the same issues seen in the Single-relation-per-input training setup of the Pointer Approach, which reappear in the Multi-relation-per-input training.

Although the Pointer Approach has a high number of FP, it is important to note that the highest FP value in the Signal class actually belongs to Gemini.

Despite these challenges with FP, the Pointer Approach demonstrates strong reliability at the span level: across the top nine signal spans in the ground truth, it nearly matches the

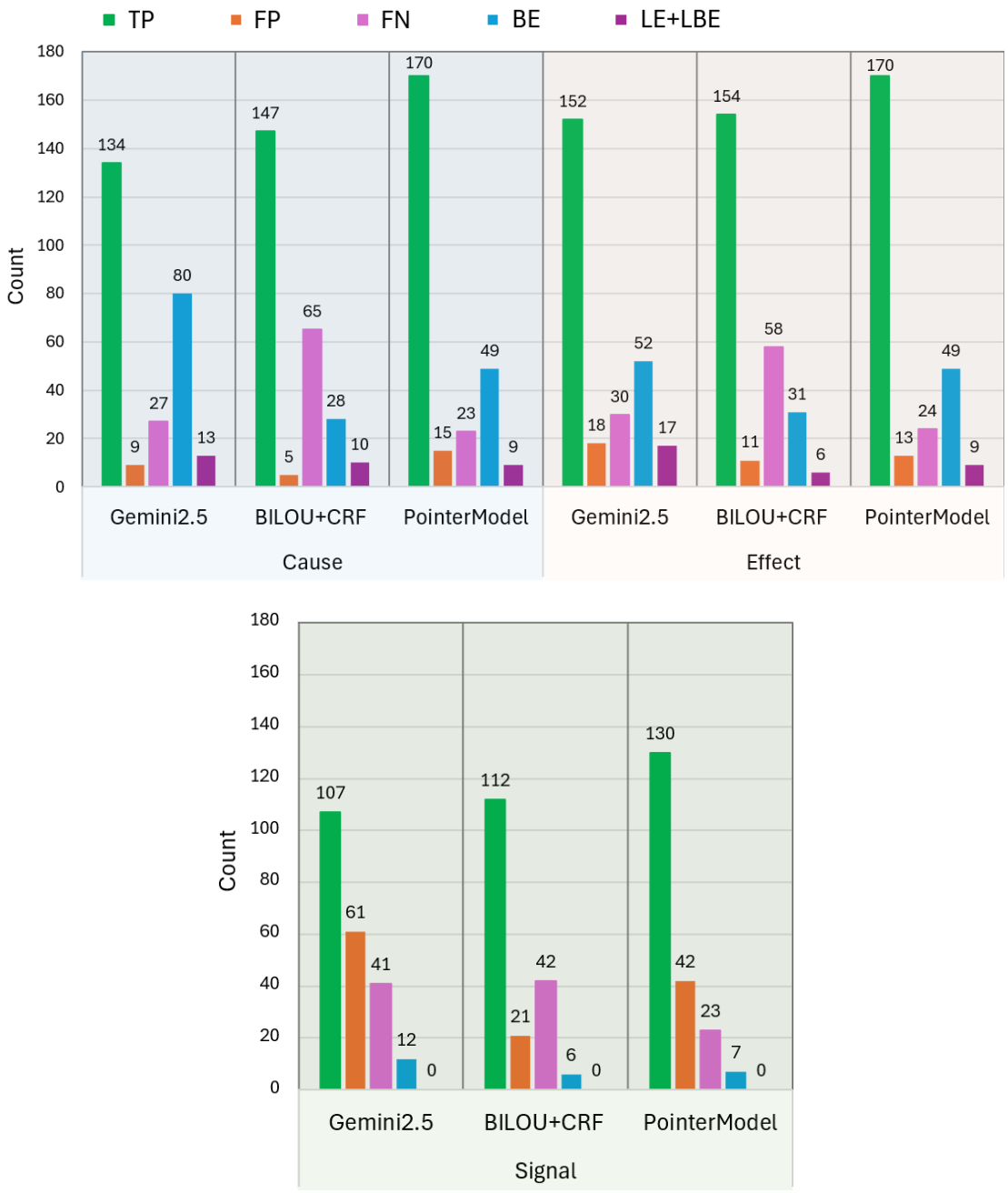


Figure 5.3: TP, FP, FN, BE, and Labeling Errors (LE+LBE) values for the best Pointer Approach, Sequence Tagging Approach and Few-Shot Model, shown for Cause-Effect (top) and Signal (bottom). Results for the Sequence Tagging and Pointer Approach models are from runs in Google Colab. Gemini2.5 = Gemini_2.5Flash_Best; BILOU+CRF = BILOU+CRF_3RelTrain_Best; PointerModel = Pointer_3RelTrain_Best.

counts with only minor underestimation, showing consistently strong alignment (as shown in Table 5.15).

This strong alignment at the span level motivates a closer look at how the presence of signals affects relation extraction performance. Figure 5.4 compares the F1-scores of the `Pointer_3RelTrain_Best` model for cause and effect, distinguishing between Single vs.

Table 5.15: Top 9 Signal Spans in CNC Ground Truth Test Set and TP count for each model (correctly identified by each model).

Signal	GroundTruth Nº	PointerModel TP	BILOU+CRF TP	Gemini2.5 TP
to	32	31	25	27
for	14	13	10	8
if	8	6	6	5
over	8	7	5	3
as	8	7	7	5
demanding	7	7	7	4
by	6	5	5	4
because	4	4	2	4
following	4	4	3	2

Multi evaluation sets and Explicit (with signal) vs. Implicit (without signal) relations. For the `Pointer_3RelTrain_Best`, it can be concluded that the presence of a signal in the relations has a positive impact on extracting the cause and effect spans.

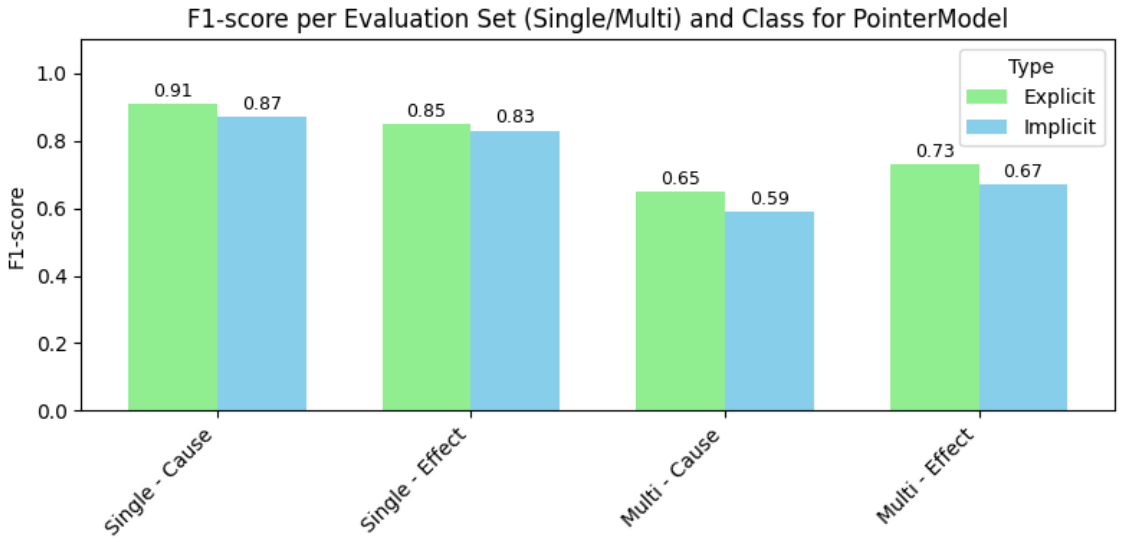


Figure 5.4: Results achieved by the PointerModel = Pointer_3RelTrain_Best.

5.5. Summary

We explored three distinct approaches: a Start/End Pointer-based approach, a Sequence Tagging approach (named BILOU+CRF), and Few-Shot models. RoBERTa-large served as the base model for the first two approaches, while the Gemini Flash API was used for the third.

The proposed Pointer-based approach, with a multi-relation-per-input training setup, outperformed all other models.

6. Causal Relation Extraction on Benchmark Datasets

This chapter presents experiments on causal relation extraction using multiple benchmark datasets (BECauSE, CTB, ESL, AltLex, TED-EN, TED-PT, and PDTB), which encompass diverse annotation schemes. Evaluating the framework across these datasets is crucial for assessing the robustness of our proposed framework described in Chapter 5.

6.1. Pointer-based and Sequence Tagging Approaches

In the experiments described in this section, we used the frameworks described in the first sections of Chapter 5.

An initial experiment was conducted to evaluate the impact of initialization strategies. Eight different models were tested in a Multi-relation-per-input training setup, using Start/End Pointer approaches and BILOU+CRF approaches. These approaches include:

- `PointerModel_3RelTrain_A`;
- `PointerModel_3RelTrain_B`;
- `PointerModel_3RelTrain_C`;
- `PointerModel_3RelTrain_D`;
- `BILOU+CRF_3RelTrain_A`;
- `BILOU+CRF_3RelTrain_B`;
- `BILOU+CRF_3RelTrain_C`;
- `BILOU+CRF_3RelTrain_D`.

where the suffixes A, B, C, and D correspond to different weight initialization strategies. Specifically:

- **A:** Training was initialized with RoBERTa weights (*standard warm-start* approach);
- **B:** Training was initialized with weights from the last epoch of the Single-relation-per-input setup trained specifically on the dataset in question;

- **C:** Training was initialized with weights from the last epoch of the Single-relation-per-input setup trained on the CNC dataset;
- **D:** Training was initialized with weights from the last epoch of the Multi-relation-per-input setup trained on the CNC dataset.

The decision to use the weights of the models trained on the CNC dataset for initialization was motivated by the fact that CNC contains substantially more causal examples in its training set (1624) compared to most benchmark datasets, which include fewer than 500 causal entries.

For each dataset, the BILOU labels occurring in the training sets (single and stacked) were analyzed for use in the sequence tagging approach. The number of distinct tags is reported in Table 6.1.

Table 6.1: Number of distinct BILOU labels on the Train Set of each corpus.

	CNC	BEC.	CTB	AltLex	ESL	TED-EN	TED-PT	PDTB
Single-Flat labels Number of distinct tags	34	18	9	11	12	16	13	31
Three-Layer labels Number of distinct tags	298	153	28	62	25	45	34	256

As observed in Figure 6.1, overall, the Pointer-based Models outperform the Sequence Tagging-based models. For the PDTB dataset, the differences between the 8 approaches are not very large, as this dataset has the most training samples. It has 5462 causal entries, while the others have fewer than 500. Additionally, all approaches achieve their highest F1 scores on the PDTB dataset.

A Friedman test was conducted [65], which rejected the hypothesis that all methods perform equally and indicated statistically significant differences among the models for both evaluation metrics presented in Figure 6.1.

Figure 6.2 presents the results of the Nemenyi post-hoc tests using the data from Figure 6.1. The top lines in the diagrams represent the axis on which the average ranks of the models are plotted, with lower (better) ranks positioned toward the right. Within the Multi-relation-per-input training setup, there are no statistically significant differences among the Pointer-based Models. The same holds for the Sequence Tagging Models (BILOU+CRF).

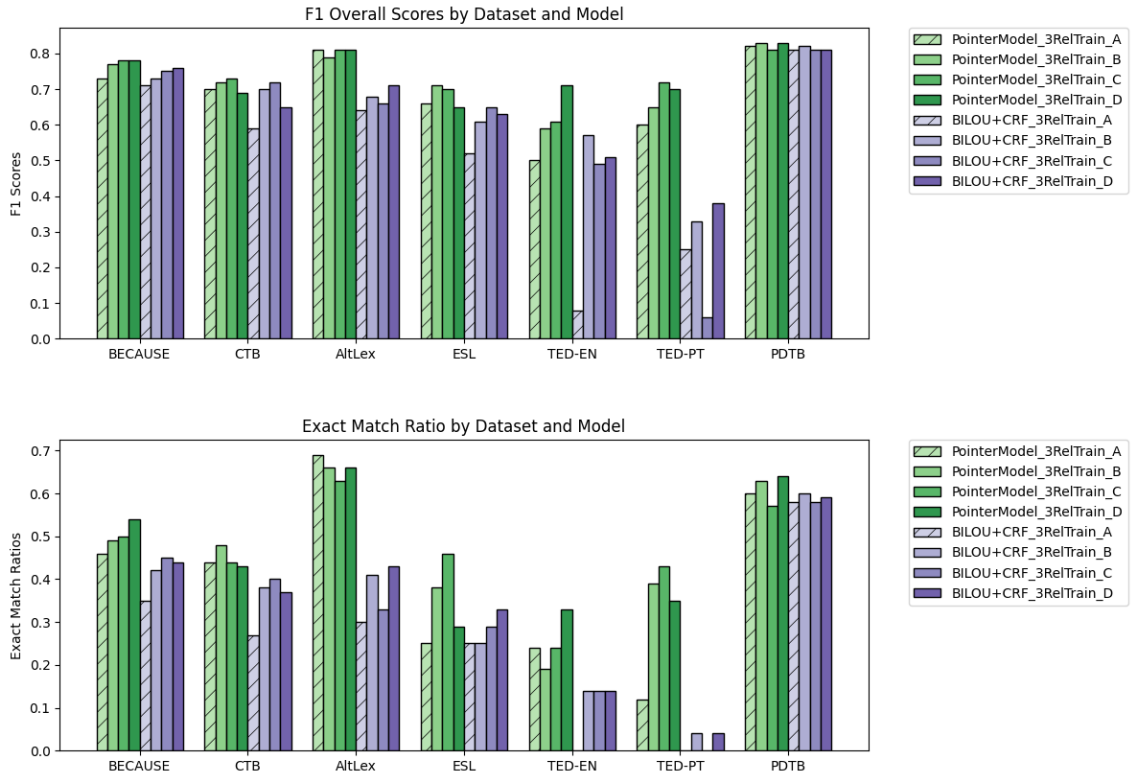


Figure 6.1: Overall F1 Scores (Top) and EM Ratio (Bottom) by Dataset and Model.

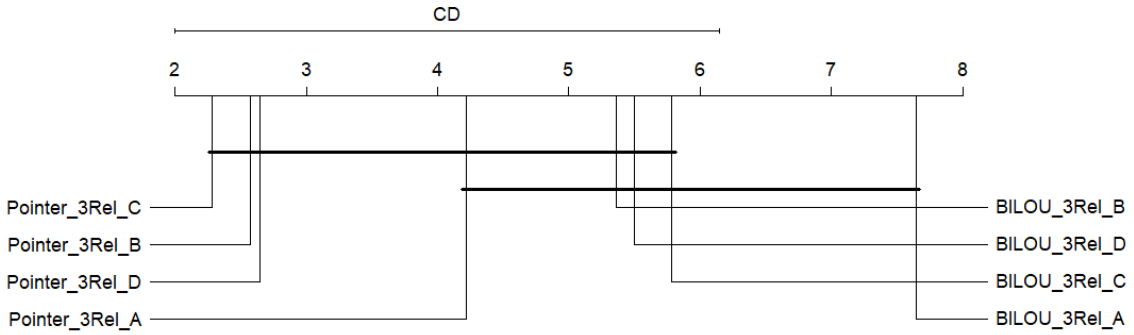
The Nemenyi post-hoc test indicated that `BILOU+CRF_3RelTrain_A` performed significantly worse than `PointerModel_3RelTrain_B`, `C` and `D`.

Additionally, Figure 6.2 shows that, for F1 Overall performance, the top three models are all Pointer-based, with `PointerModel_3RelTrain_C` achieving the best average rank, followed closely by `PointerModel_3RelTrain_B` and `D`. In contrast, the Sequence Tagging models occupy the bottom ranks, with `BILOU+CRF_3RelTrain_A` performing the worst. A similar pattern is observed in the EM ratio rankings.

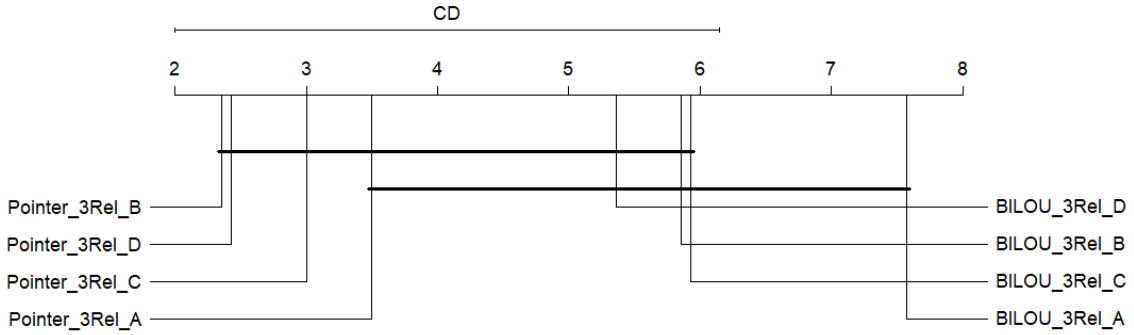
These results highlight the critical role of *task-specific initialization*, consistent with earlier conclusions.

As no single "*task-specific initialization* strategy" (B, C, or D) proved optimal across all datasets for either the Pointer-based or Sequence Tagging models, model selection was performed on a per-dataset basis for the subsequent analysis. For example, on the BECauSE dataset, `PointerModel_3RelTrain_D` achieved the highest F1 score among the Pointer-based models, while `PointerModel_3RelTrain_C` achieved the highest F1 score on the CTB dataset.

The best-performing models for each dataset were identified and are reported in Table 6.2.



(a) Comparison of all models against each other using F1 Overall.



(b) Comparison of all models against each other using EM.

Figure 6.2: Visualization of Nemenyi post-hoc tests for data from Figure 6.1. Groups of models that are not significantly different (at p=0.05) are connected. CD is the critical difference.

These models will later be used in the final section of this chapter for comparison with the Gemini API.

Table 6.2: Best Initialization Strategy (by F1 score) for each dataset.

Dataset	Pointer_3RelTrain Models	BILOU+CRF_3RelTrain Models
BECAUSE	D	D
CTB	C	C
AltLex	D	D
ESL	B	C
TED-EN	D	B
TED-PT	C	D
PDTB	D	B

6.2. Few-Shot Models

The experiments to evaluate this type of approach were once again conducted using the Gemini 2.5 Flash API, as in Chapter 5. Following the findings of the previous chapter, a 10-shot setup was employed.

For the BECauSE dataset, the prompt illustrated in Figure 5.1 was used, with the annotation examples replaced by training texts from the BECauSE dataset. Since the BECauSE

and CNC datasets are very similar in terms of causality representation and patterns, no further adaptations to the prompt were necessary beyond updating the annotation examples.

In fact, for all datasets, the annotation examples were drawn specifically from each respective dataset.

For the CTB and ESL datasets, the following instructions were added to the prompt: “This is an Event Headword Causal Relation Extraction task. In this task, only the headword of the cause and effect events must be identified, not the entire span.”. Without these specific instructions, the model’s performance was very poor, as it tended to identify larger spans (such as phrases) rather than only the event headwords. The prompt used in our experiments is illustrated in Figure 6.3.

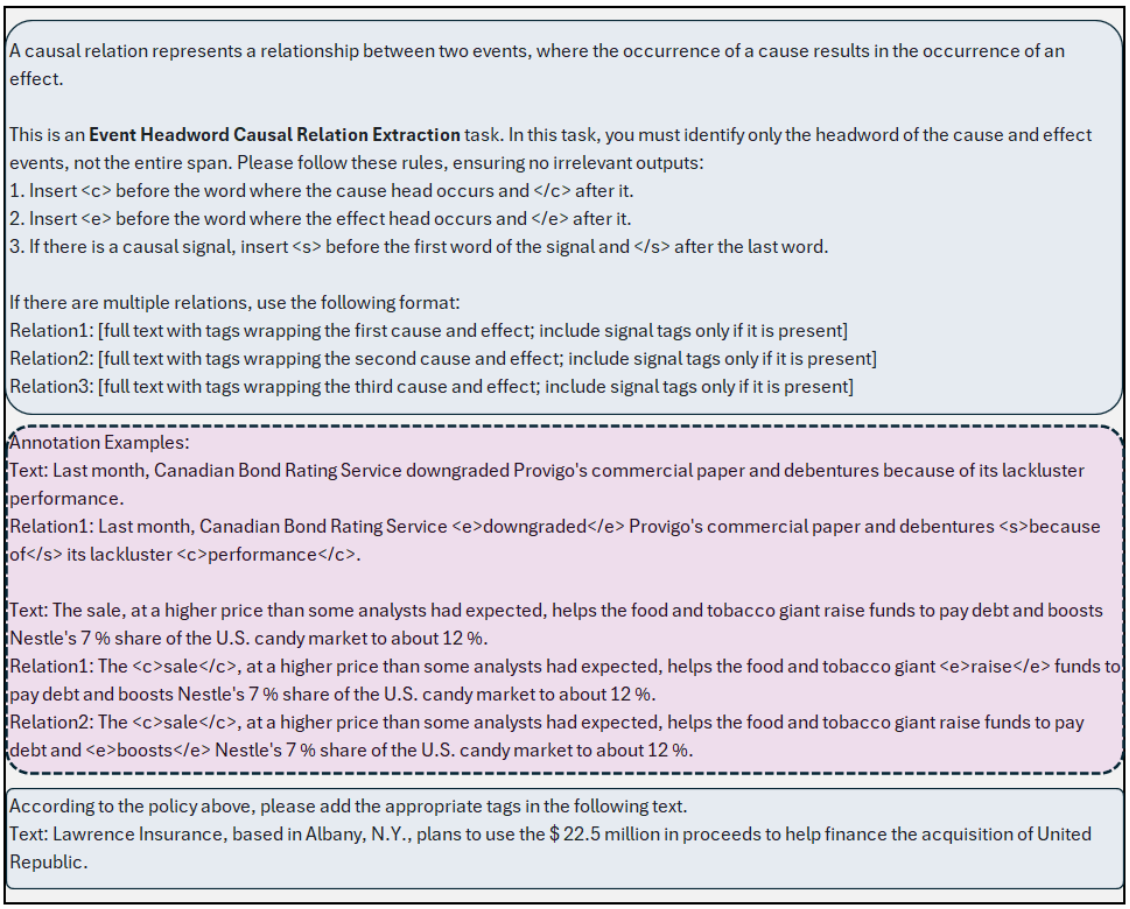


Figure 6.3: Illustration of the prompt used to evaluate Gemini API on the CTB Dataset. All text annotation examples are taken from the CTB dataset.

For the TED-EN, TED-PT, and PDTB datasets, the following instructions were added to the prompt: “In this task, both cause and effect spans should consist of clauses or full sentences, and should not be phrases or single words.”. This addition was motivated by

the observation that the model tended to identify causes and effects as phrases, which resulted in underperformance on datasets such as PDTB.

The results will be presented in the next section.

6.3. Final Comparison Of Models/Approaches

Regarding the Pointer-based approaches, in the multi-relation-per-input setup, the additional filtering step was incorporated for the final comparison, as it had not been applied initially (in the experiments described in the first section). This step removes duplicate predictions and sorts the remaining ones in descending order of confidence scores. For each dataset, the best-performing model was selected for the application of this filtering (Table 6.2). The resulting models, with filtering applied, are indicated with an asterisk in the following results (`PointerModel_3RelTrain_*`).

Regarding the Sequence Tagging approaches, in the multi-relation-per-input setup, the best-performing models described in the first section (presented in Table 6.2) are used in this final comparison.

Additionally, the single-relation-per-input training setup is included in this comparison, as it represents the state-of-the-art for pointer-based causal relation extraction.

In Figure 6.4, the performance of the Single-relation-per-input training setups (1RelTrain), the best Multi-relation-per-input training setups (3RelTrain), and Gemini 2.5 Flash model (with prompts adjusted for each dataset) are compared.

Once again, the Friedman test rejected the hypothesis that all methods perform equally, revealing statistically significant differences among the models for both Overall F1 and EM ratio.

Figure 6.5 presents the results of the Nemenyi post-hoc tests using the data from Figure 6.4. These results indicated that `PointerModel_3RelTrain_*` is significantly better than `BILOU+CRF_1RelTrain` and `Gemini2.5Flash_*`.

Additionally, Figure 6.5 shows that, `PointerModel_3RelTrain_*` achieves the best average ranking in both the F1 and EM Ratio scores.

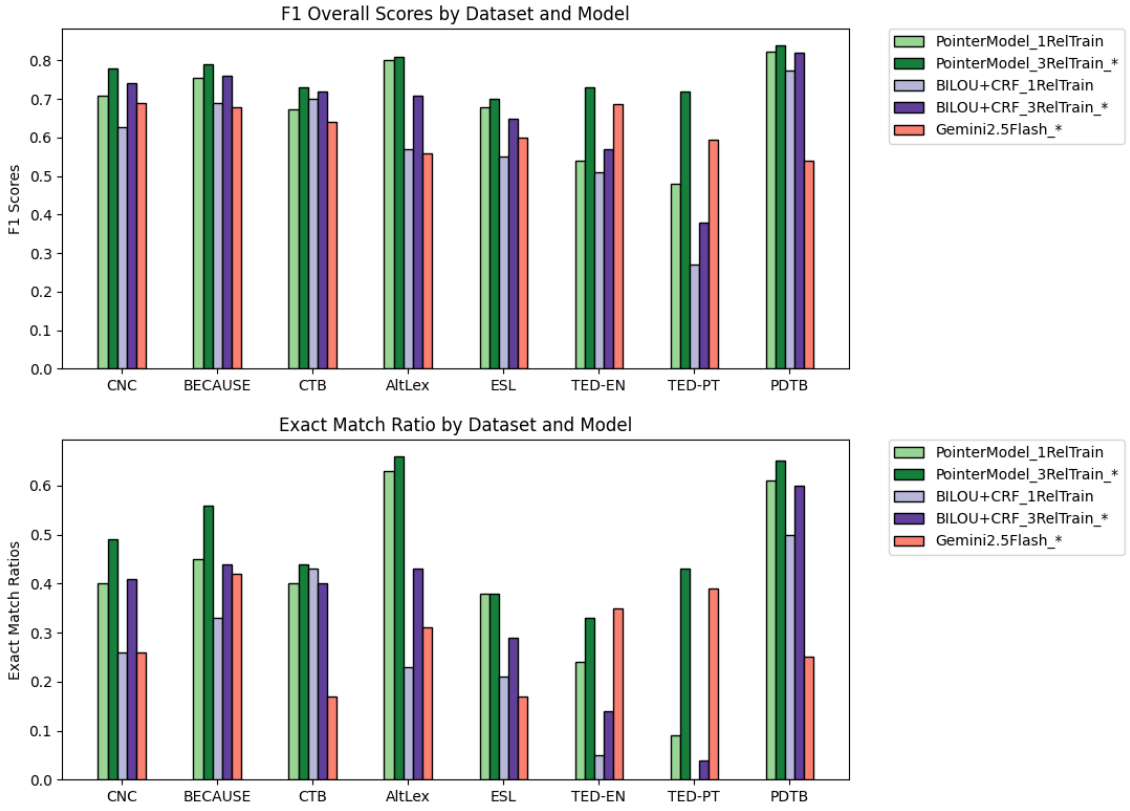
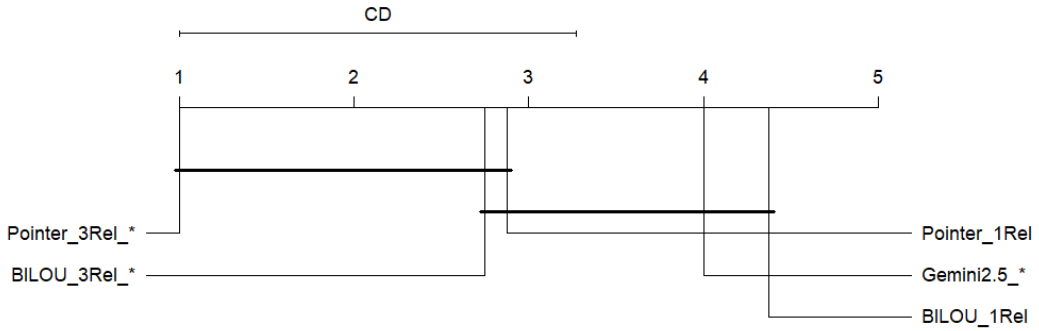
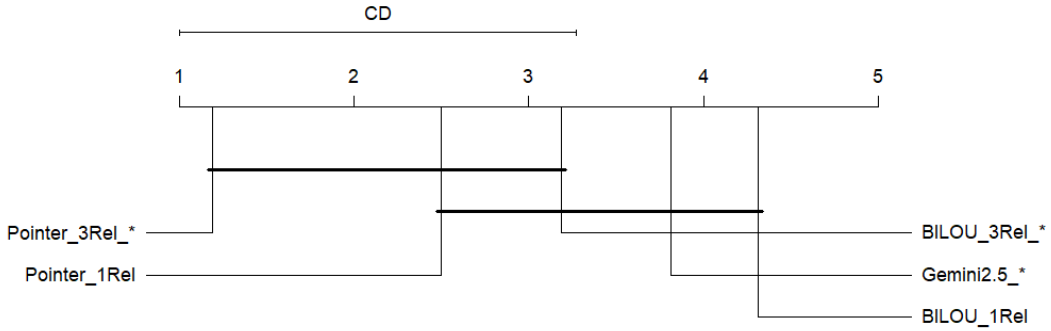


Figure 6.4: Overall F1 Scores (Top) and EM Ratio (Bottom) by Dataset and Model (Best Approach in Each Group).



(a) Comparison of all models against each other using F1 Overall.



(b) Comparison of all models against each other using EM.

Figure 6.5: Visualization of Nemenyi post-hoc tests for data from Figure 6.4. Groups of models that are not significantly different (at p=0.05) are connected. CD is the critical difference.

6.4. Summary

For the Causal Relation Extraction task, the proposed Pointer-based model achieved the highest average ranking among the evaluated models, based on both F1-score and Exact Match Ratio. A Friedman test showed that these differences were statistically significant.

7. Causal vs. Non-Causal Discrimination and Relation Number Calibration

In the experiments described in the previous chapters, only causal entries were used to train the model, and the evaluation was conducted exclusively on causal entries (following the evaluation system from the 2023 edition of Shared Task 3 of CASE). In this chapter, the performance on both causal and non-causal entries is reported.

Additionally, for our proposed framework, post-processing techniques, such as thresholding and overlap limitation, are applied to optimize the number of predicted relations and prevent overprediction. Since this was not penalized by the evaluation system in the 2023 edition of Shared Task 3 of CASE, it is addressed here.

In the experiments described in this chapter, the best models identified in the previous chapters for the Causal Relation Extraction Task were used for each dataset: *Pointer_3RelTrain_Best*, *BILOU+CRF_3RelTrain_Best*, and *Gemini_2.5Flash_Best*. Consequently, single-relation-per-input training setups were not considered. It should be noted that the best-performing model on each dataset was specifically trained on that dataset.

7.1. CNC experiments

7.1.1 Causal vs. Non-Causal Discrimination

The aim of this task is to discriminate between causal and non-causal sentences. There are different ways to approach this, for example by training a classifier. However, in our project, the approach leverages the causal relation extraction model and attempts to use its `confidence score` to determine whether a sentence is causal or not.

For the Pointer-based approach, trials were performed using thresholds based on the relation scores defined in Equation (5.1). The Pointer Models always return at least one relation, since the post-processing steps applied in the previous chapters only filter out duplicate predictions. Equation (5.1), defined in the Chapter 5, is a log-probability aggregation over the span start/end positions of the cause and effect. The higher the value, the more confident the model is that those token spans correspond to a valid causal relation. The range of values of the `relation_score` are:

$$\text{relation\_score} \in (-\infty, 0] \tag{7.1}$$

In our project, for the Pointer-based model, causality was defined by applying a threshold to the relation scores predicted by the model (defined in Equation (5.1)). For each instance, the maximum relation score was compared to $threshold_1$. If the score exceeded this threshold, the instance was classified as causal; otherwise, it was classified as non-causal:

$$\hat{y} = \begin{cases} 1 & \text{if } \max(\text{relation\_scores}) \geq threshold_1, \\ 0 & \text{otherwise.} \end{cases} \tag{7.2}$$

To determine the optimal threshold, the following was first defined for each entry:

$$y_{\text{score}} = \max(\text{relation\_scores}) \tag{7.3}$$

Next, a subset of the training set, randomly selected to match the size of the test set, was used as a validation set. The validation set was then split into two groups according to the ground-truth labels, yielding distributions of scores for causal and non-causal entries, respectively. To analyze these distributions, the empirical cumulative distribution function (CDF) was computed for each group, as shown in Figure 7.1. The steeper slope of the blue curve suggests less variance in maximum scores of causal entries compared to maximum scores of non-causal entries.

Approximately 99% of causal instances have a maximum score above –2.15, whereas about 62% of non-causal instances also exceed this threshold. Because the non-causal CDF increases gradually over a broader range of scores, substantial overlap exists between the two distributions, making it challenging for the model to reliably distinguish causal from non-causal instances based on this score alone.

Analysis of Figure 7.2a indicates that, for the Causal vs. Non-Causal Discrimination Task, Macro F1 and Accuracy reach their highest values between thresholds of −1.50 and −1.17. In contrast, Figure 7.2b shows that, for the Causal Relation Extraction Task in the validation set of CNC, F1 and EM ratios start decreasing at a threshold of −2.30.

In order to preserve the performance of the Causal Relation Extraction Task, a threshold

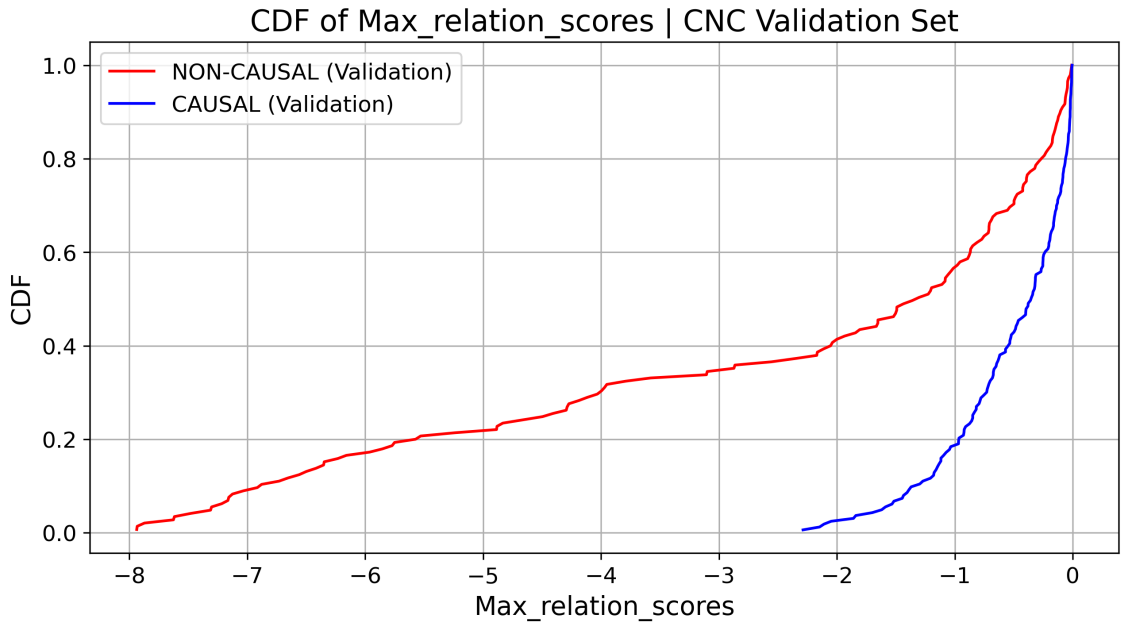


Figure 7.1: CDF of maximum relation scores on the Validation Set of CNC.

of −2.30 was selected and used for evaluation on the test set. The results are presented on Table 7.1. Additional, the results for the *BILOU+CRF_3RelTrain_Best* and the *Gemini_2.5Flash_Best* model are also presented. It can be concluded that the Gemini 2.5 Flash model outperforms the others, achieving the highest macro F1 score and balanced accuracy.

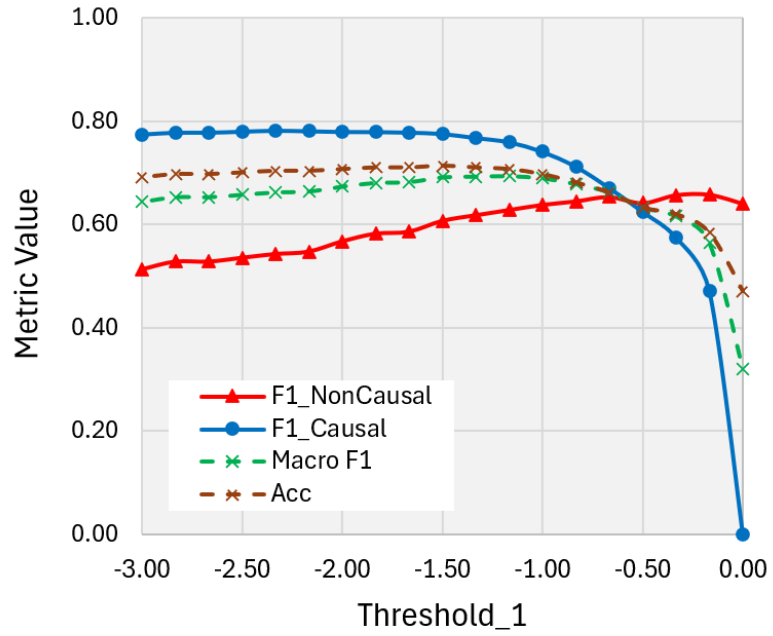
For the Gemini 2.5 Flash API, using the prompts developed for each dataset, a new instruction was added: "If a causal relation is present, return the text with the appropriate tags. If no causal relation is present in the text, return that text unchanged (without tags)."

For the BILOU+CRF model and Gemini 2.5 Flash, an instance was considered causal if at least one relation contained at least one class (cause or effect). Relations containing only a single class (either cause or effect) were included in the evaluation used in the previous chapters. Therefore, for consistency, these entries were defined as causal.

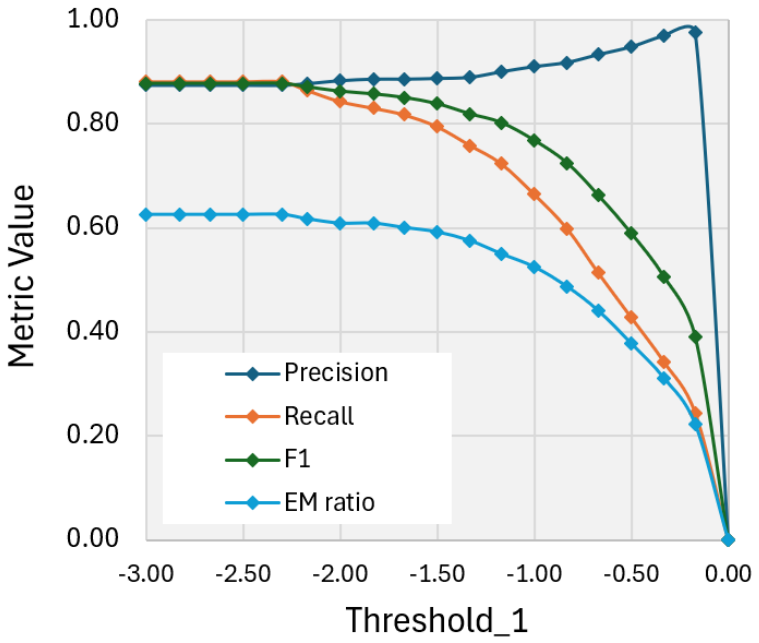
Table 7.1: Causal vs. Non-Causal Discrimination - CNC Test Set Results. BAcc = Balanced Accuracy

Model	**Non-Causal (Neg)**			**Causal (Pos)**			**Macro F1**	**BAcc**
	P	**R**	**F1**	**P**	**R**	**F1**		
PointerModel (tr= -2.30)	0.95	0.34	0.50	0.64	0.98	0.78	0.64	0.66
BILOU+CRF	0.97	0.19	0.31	0.59	0.99	0.74	0.53	0.59
Gemini_2.5Flash	0.97	0.57	0.71	0.73	0.98	0.84	0.78	0.78

It is important to note that the PointerModel and BILOU+CRF were trained exclusively on causal entries, meaning that they were not explicitly trained to distinguish between



(a) Causal vs. Non-Causal Discrimination



(b) Causal Relation Extraction Task

Figure 7.2: Performance on CNC Validation Set based on threshold.

causal and non-causal instances. Therefore, the results of the Causal vs. Non-Causal Discrimination Task for these models are based solely on the confidence scores of their predictions, rather than direct supervision for causal vs. non-causal discrimination.

7.1.2 Relation Number Calibration

The previous mentioned threshold ($threshold_1$) is applied at the entry level: if an input entry contains at least one relation with a score above $threshold_1$, the entry is classified

as causal, and all of its relations are retained (even those with scores below $threshold_1$).

However, to optimize the number of predicted relations and prevent the model from over-predicting, a second threshold is applied to the relations with scores below $threshold_1$. For the CNC dataset, $threshold_2$ = -4 is found to be optimal, as it does not compromise the performance on the validation set for the Causal Relation Extraction Task.

In summary:

- $threshold_1$: entry-level threshold to distinguish between causal and non-causal entries;
- $threshold_2$: per-relation filter applied to less confident predictions within entries defined as causal by the model;

Additionally, although duplicate predictions were previously removed, the relations for the same entry still exhibit a high degree of similarity. To define a maximum allowable overlap, the 99th percentile of overlap for each pair of relations was computed using the training set. The cumulative distribution function (CDF) of the intersection-over-union (IoU) values is shown in Figure 7.3. The orange curve shows the CDF for the IoU of the causal spans (cause and effect spans taken together) between the first relation and the second relation in the training set. The blue curve represents this overlap between the first relation and the third relation; the green curve represents the overlap between the second relation and the third relation in the training set.

The figure indicates that the most similar relations in the ground-truth training set are typically the first two (REL1_vs_REL2). The orange curve indicates that the 99th percentile of their IoU value (overlap) is around 0.5. The others curves have a 99th percentile very low.

For the CNC, the maximum allowed overlap was set to 0.5. This threshold was applied to the test set to further filter the predictions.

In summary, the post-processing steps are: removing duplicate predictions and sorting predictions (introduced in the previous chapters), applying score thresholds, and limiting overlap between relations.

Figure 7.4 shows the matrix of predicted versus actual relation counts for the Pointer Model on the CNC test set, after removing duplicate predictions, applying thresholds, and limiting overlap between relations. Although the model is not explicitly trained to predict

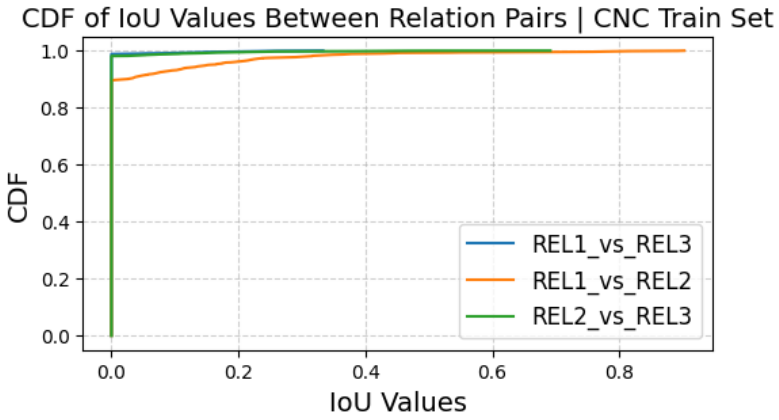


Figure 7.3: CDF of intersection-over-union (IoU) values between relation pairs in the ground-truth of the CNC training set.

the number of relations, it captures the overall trend in the data reasonably well. The primary limitation lies in accurately identifying cases with no relations, where the model frequently over-predicts and assigns at least one relation.

For the remaining models, the corresponding matrices are presented in Figure 7.5.

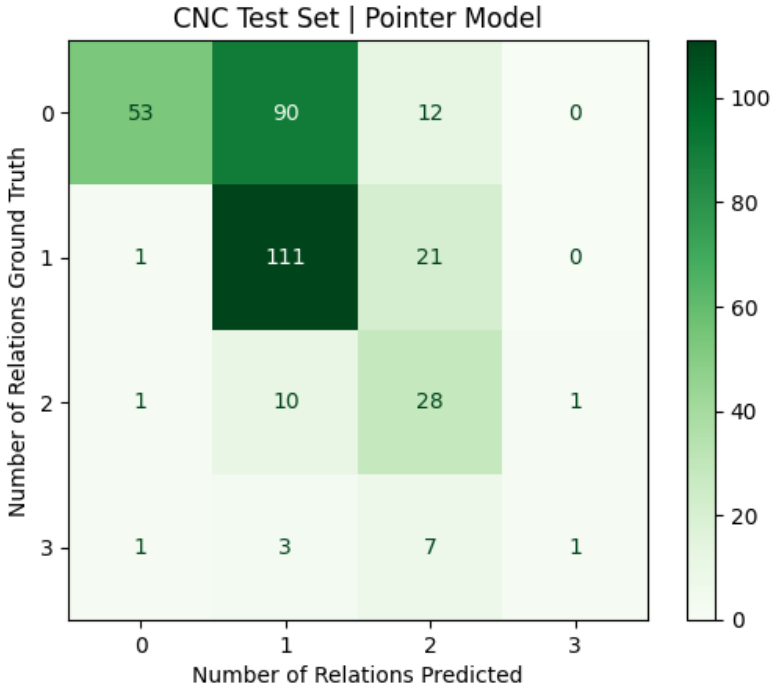


Figure 7.4: Matrix of Predicted vs. Actual Relation Counts for the Pointer Model on the CNC Test Set, after removing duplicate predictions, applying thresholds, and limiting overlap between relations.

Table 7.2 presents the proportion of correct number of relations predicted by the three models. While `Gemini_2.5Flash_Adap*` performs best with 67%, the `PointerModel_3RelTrain_T_0*`

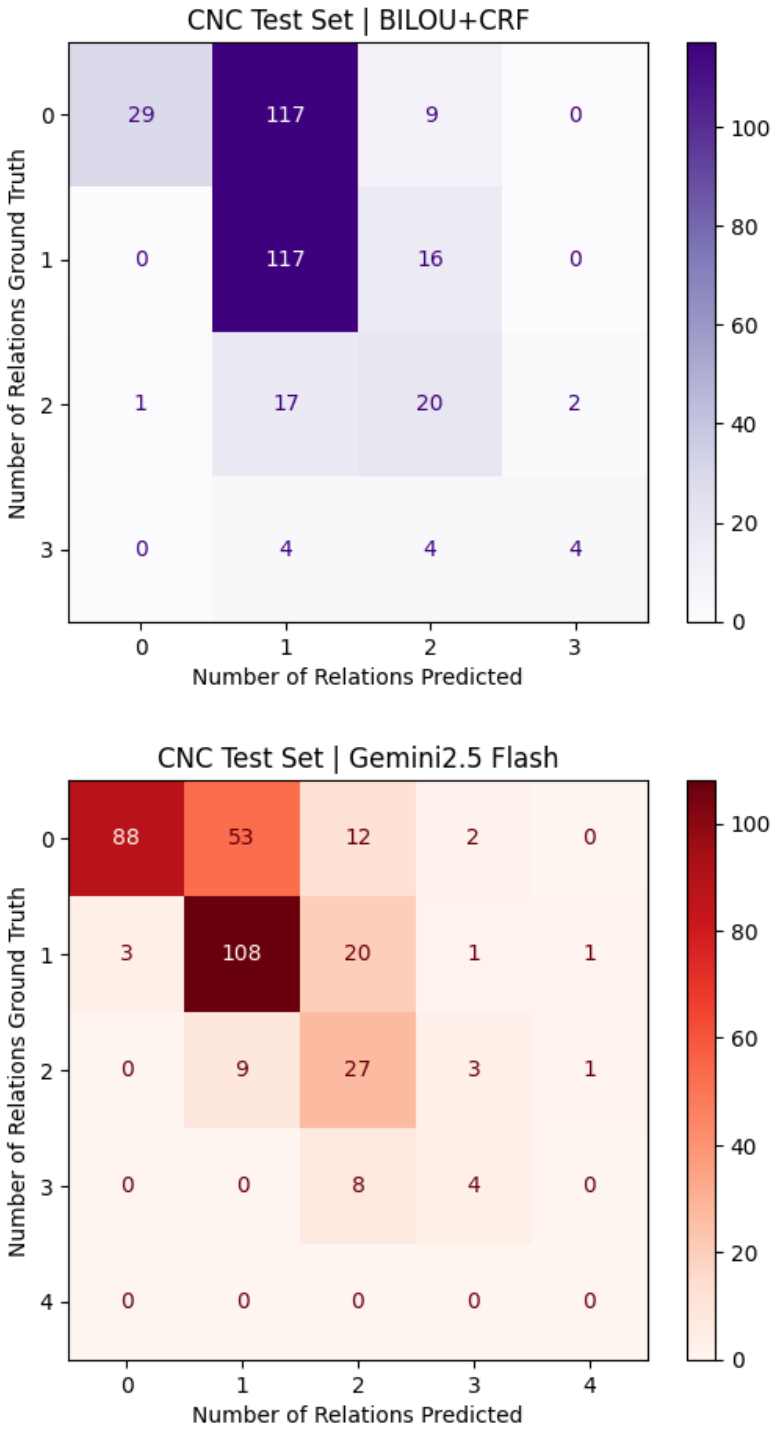


Figure 7.5: Predicted vs. Actual Relation Count Matrices Comparison on CNC Test Set.

achieves a solid 57% and the `BILOU+CRF_3RelTrain_*` model trails behind, correctly predicting the relation count in only 50% of instances.

Note that this proportion measures the fraction of entries for which the model predicted the correct number of causal relations, regardless of the content of the spans, thereby penalizing only over- or under-prediction.

Table 7.2: Proportion of Correct Number of Relations - CNC Test Set Results. "PointerModel_3RelTrain_T_O*" is the model with all post-processing steps including duplicate removal, thresholding, and overlap limitation.

Model	Proportion of Correct Nº of Relations
*PointerModel_3RelTrain_T_O**	0.57
*BILOU+CRF_3RelTrain_**	0.50
*Gemini_2.5Flash_Adap**	0.67

Table 7.3 compares the causal relation extraction performance of three models on the CNC test set. `PointerModel_3RelTrain_T_O*` continues to demonstrate the best performance across all evaluated metrics for the Causal Relation Extraction Task. These results indicate that the inclusion of additional post-processing steps did not compromise the model's performance on this task.

Table 7.3: Final Evaluation Measures for Different Models on CNC Causal Relation Extraction Task (Test Set). "O." denotes the overall score. "PointerModel_3RelTrain_T_O*" is the model with all post-processing steps including duplicate removal, thresholding, and overlap limitation.

Model	Cause F1	Effect F1	Signal F1	O.F1	O.EM
*PointerModel_3RelTrain_T_O**	0.78	0.78	0.79	0.78	0.49
*BILOU+CRF_3RelTrain_**	0.73	0.74	0.76	0.74	0.41
*Gemini_2.5Flash_Adap**	0.64	0.72	0.62	0.66	0.21

7.2. Benchmark Datasets

7.2.1 Causal vs. Non-Causal Discrimination

The CDF curves of non-causal and causal entries of all validation sets are presented in Appendix C. The optimal $threshold_1$ values found are as follows:

- BECAUSE validation set: -0.75;
- CTB validation dataset: -1.60;
- AltLex validation set: -3.50;
- TED-EN validation dataset: -1.10;
- TED-PT validation set: -1.25;
- PDTB validation dataset: -1.20.

These values were determined similarly to the procedure used for the CNC dataset.

Table 7.4 compares the causality discrimination performance of the three models on all benchmark test sets.

Table 7.4: Causal vs. Non-Causal Discrimination – Results on All Benchmark Test Sets.

Dataset	Model	Non-Causal (Neg)			Causal (Pos)			Mac.F1	BAcc
		P	R	F1	P	R	F1		
BEC.	PointerModel	0.36	0.20	0.26	0.87	0.94	0.90	0.58	0.57
	BILOU+CRF	0.67	0.30	0.41	0.89	0.97	0.93	0.67	0.64
	Gemini2.5	0.50	0.40	0.44	0.90	0.93	0.91	0.69	0.66
CTB	PointerModel	0.99	0.45	0.62	0.34	0.98	0.51	0.56	0.72
	BILOU+CRF	1.00	0.02	0.04	0.23	1.00	0.37	0.21	0.51
	Gemini2.5	0.96	0.51	0.66	0.35	0.93	0.51	0.59	0.72
AltLex	PointerModel	0.84	0.60	0.70	0.72	0.90	0.80	0.75	0.75
	BILOU+CRF	0.73	0.44	0.55	0.63	0.85	0.73	0.64	0.65
	Gemini2.5	0.70	0.54	0.61	0.66	0.79	0.72	0.66	0.67
TED-EN	PointerModel	1.00	0.64	0.78	0.82	1.00	0.90	0.84	0.82
	BILOU+CRF	0.71	0.91	0.80	0.93	0.78	0.85	0.82	0.84
	Gemini2.5	0.86	0.55	0.67	0.77	0.94	0.85	0.76	0.74
TED-PT	PointerModel	0.50	0.27	0.35	0.70	0.86	0.78	0.56	0.57
	BILOU+CRF	0.50	0.73	0.59	0.82	0.64	0.72	0.66	0.68
	Gemini2.5	0.78	0.64	0.70	0.83	0.91	0.87	0.78	0.77
PDTB	PointerModel	0.86	0.28	0.43	0.51	0.94	0.66	0.54	0.61
	BILOU+CRF	0.87	0.07	0.14	0.46	0.99	0.62	0.38	0.53
	Gemini2.5	0.93	0.36	0.51	0.54	0.97	0.70	0.61	0.66

The Friedman test indicates that the differences among the models listed in Table 7.4 are not statistically significant for either evaluation metric-Macro F1 and balanced accuracy. The results on the CNC were included in this analysis.

The models' average ranks, according to Macro F1 for the Causal vs. Non-Causal discrimination task, are:

- 1) `Gemini_2.5Flash_Adapted*` with 1.43
- 2) `PointerModel_3RelTrain_T*` with 2.00
- 3) `BILOU+CRF_3RelTrain_*` with 2.57

7.2.2 Relation Number Calibration

The optimal $threshold_2$ values were first determined. These values were obtained using a procedure similar to that applied for the CNC dataset. The optimal $threshold_2$ values found are as follows:

- BECAUSE validation set: -0.75;
- CTB validation dataset: -2.40;
- AltLex validation set: -3.50;

- TED-EN validation dataset: -1.30;
- TED-PT validation set: -3.40;
- PDTB validation dataset: -3.00.

Additionally, to define a maximum allowable overlap, the 99th percentile of overlap for each pair of relations was computed using the training set of each dataset.

To recap, the post-processing steps are: removing duplicate predictions and sorting them (as described in previous chapters), applying score thresholds, and limiting overlap between relations.

The matrices for all test sets after performing all post-processing steps are presented in Appendix D.

Table 7.5 compares the proportion of correct number of relations predicted by the three models on all test sets.

Table 7.5: Proportion of Correct Number of Relations across all test sets.

Dataset	PointerModel	BILOU+CRF	Gemini
CNC	0.57	0.50	0.67
BECAUSE	0.75	0.77	0.72
CTB	0.53	0.19	0.55
AltLex	0.65	0.65	0.65
TED-EN	0.76	0.72	0.69
TED-PT	0.64	0.64	0.76
PDTB	0.50	0.44	0.53

The Friedman test indicates that the differences among the models listed in Table 7.5 are not statistically significant. The results on the CNC were included in this analysis.

The models' average ranks, based on correct relation number prediction, are:

- 1) `Gemini2.5Flash_Adapted*` with 1.64
- 2) `PointerModel_3RelTrain_T_0*` with 1.86
- 3) `BILOU+CRF_3RelTrain_*` with 2.50

7.3. Assessing the Impact of Post-Processing Steps on the Causal Relation Extraction Task

Table 7.6 presents the final evaluation measures on the Causal Relation Extraction Task across all test sets.

Table 7.6: Final Evaluation Measures for Different Models on the Causal Relation Extraction Task across all test sets.

Dataset	Model	Cause F1	Effect F1	Signal F1	O.F1	O.EM
CNC	PointerModel_3_T_O*	0.78	0.78	0.79	0.78	0.49
	BILOU+CRF_3_*	0.73	0.74	0.76	0.74	0.41
	Gemini_2.5Flash_Ad*	0.64	0.72	0.62	0.66	0.21
BEC.	PointerModel_3_T_O*	0.76	0.75	0.83	0.77	0.54
	BILOU+CRF_3_*	0.73	0.75	0.79	0.76	0.44
	Gemini_2.5Flash_Ad*	0.66	0.70	0.67	0.68	0.41
CTB	PointerModel_3_T_O*	0.76	0.70	0.75	0.74	0.44
	BILOU+CRF_3_*	0.74	0.68	0.74	0.72	0.40
	Gemini_2.5Flash_Ad*	0.70	0.67	0.50	0.64	0.25
AltLex	PointerModel_3_T_O*	0.76	0.77	0.76	0.77	0.60
	BILOU+CRF_3_*	0.69	0.66	0.77	0.71	0.43
	Gemini_2.5Flash_Ad*	0.54	0.51	0.63	0.56	0.31
TED-EN	PointerModel_3_T_O*	0.78	0.65	0.73	0.72	0.33
	BILOU+CRF_3_*	0.44	0.55	0.71	0.57	0.14
	Gemini_2.5Flash_Ad*	0.69	0.67	0.71	0.69	0.35
TED-PT	PointerModel_3_T_O*	0.72	0.69	0.79	0.73	0.43
	BILOU+CRF_3_*	0.30	0.36	0.50	0.38	0.04
	Gemini_2.5Flash_Ad*	0.63	0.47	0.69	0.59	0.39
PDTB	PointerModel_3_T_O*	0.84	0.82	0.83	0.83	0.63
	BILOU+CRF_3_*	0.82	0.80	0.85	0.82	0.60
	Gemini_2.5Flash_Ad*	0.56	0.48	0.60	0.54	0.25

Once again, the Friedman test revealed statistically significant differences among the models for both Overall F1 and EM ratio on the Causal Relation Extraction Task. The results on the CNC were included in this analysis.

The results of the Nemenyi post-hoc tests using the data from Table 7.6, indicated that `PointerModel_3RelTrain_T_O*` is significantly better than `Gemini_2.5Flash_Adapted*` (similar to the conclusions described in the previous chapter). Therefore, it can be concluded that the additional post-processing steps did not compromise the Pointer Model's performance on the Causal Relation Extraction Task.

7.4. Summary

For the causal vs. non-causal discrimination task, the Gemini 2.5 Flash API achieved the highest average ranking among the evaluated models. However, a Friedman test showed that these differences were not statistically significant. Additionally, the average ranking results indicated that the Sequence Tagging model (BILOU+CRF) performed the lowest on this task.

The post-processing steps required for the Pointer Model did not affect its performance on the Causal Relation task.

8. Conclusions

The main objective of this work was to develop an automated process for extracting causal relations, using the CNC dataset as the main reference for development. Three distinct approaches were explored: a Start/End Pointer-based approach, a Sequence Tagging model and Few-Shot Models. For the first two approaches, RoBERTa-large was used as the base model, and for the third, Gemini Flash API was used. These models were evaluated across both causal relation extraction and causality discrimination tasks.

For the causal relation extraction task, statistical analysis using the Friedman test and the Nemenyi post-hoc test indicated that the proposed Pointer-based framework (with a Multi-relation-per-input training setup) is significantly better than Gemini 2.5 Flash. The proposed Pointer-based framework achieved the highest average ranking in the Causal Span Extraction Task, based on both Overall F1-score and Exact Match Ratio.

Within the Multi-relation-per-input-setup, the Sequence Tagging model required more time to train due to the complexity of the CRF layer and tag dependencies, but benefit from faster inference. Conversely, the pointer-based model involved more complex decoding steps during inference, making it slower at prediction time, although the training was relatively faster.

This work also highlights the critical importance of task-specific initialization. In light of the added complexity introduced by the Multi-relation-per-input training setup, it demonstrated the advantage of leveraging pre-training from the Single-relation-per-input setting, which enables the model to initially focus on learning the patterns associated with a single relation per input. This initialization provided a more stable foundation for subsequently handling the increased complexity of multiple relations within the same input.

The Pointer-based and Sequence Tagging models demonstrated particularly strong performance on larger datasets, suggesting that their effectiveness increases with the availability of more training data.

With respect to causal vs. non-causal discrimination task, the Gemini 2.5 Flash API obtained the highest average ranking among the evaluated models. However, statistical analysis using the Friedman test indicated that these differences were not statistically

significant. The average ranking results also revealed that the Sequence Tagging model ranked the lowest in this task.

8.1. Contributions

This thesis demonstrated the feasibility of extracting causality-related information from text using a variety of modeling approaches. A novel framework was developed that outperformed existing state-of-the-art methods in this domain.

In addition, the incorporation of post-processing strategies significantly improved the consistency of number of relations predicted, while preserving the performance of causal relation extraction.

8.2. Limitations and Future Work

A potential improvement would involve performing the Signal Classification Task at the relation level rather than the entry level. By moving beyond the [CLS]-based entry-level classification, future models could better capture relation-specific signals and mitigate the issues observed.

Using a single model for both causality discrimination and causal relation extraction, without explicitly training it to differentiate between causal and non-causal instances, has proven to be suboptimal. As future work, a two-stage approach is recommended. In the first stage, a dedicated model would be developed to identify entries containing causal relations. In the second stage, the proposed Pointer-based framework, trained under a Multi-relation-per-input setup, could then be applied to extract causal relations from the previously identified causal entries.

The proposed framework, with the hyperparameter max_relations set to 3, achieved optimal performance on the CNC dataset. However, this parameter was not evaluated on the other datasets and was instead fixed at the same value. As a result, in the Multi-relation-per-input setup, all models were trained to predict a maximum of 3 relations per input. Given that the proportion of entries with more than two relations varies considerably across datasets, further investigation into the impact of max_relations is necessary.

References

[1] F. A. Tan, H. Hettiarachchi, A. Hürriyetoğlu, T. Caselli, O. Uca, F. F. Liza, and N. Oostdijk, “Event causality identification with causal news corpus - shared task 3, CASE 2022,” in *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, A. Hürriyetoğlu, H. Tanev, V. Zavarella, and E. Yörük, Eds. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 195–208. [Online]. Available: https://aclanthology.org/2022.case-1.28 [Cited on pages 1, 3, and 27.]

[2] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd ed., 2024, online manuscript released August 20, 2024. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/ [Cited on pages 1, 5, 6, 7, 8, 9, 12, 13, and 14.]

[3] C. Braud, Y. J. Liu, E. Metheniti, P. Muller, L. Rivière, A. Rutherford, and A. Zeldes, “The DISRPT 2023 shared task on elementary discourse unit segmentation, connective detection, and relation classification,” in *Proceedings of the 3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, C. Braud, Y. J. Liu, E. Metheniti, P. Muller, L. Rivière, A. Rutherford, and A. Zeldes, Eds. Toronto, Canada: The Association for Computational Linguistics, Jul. 2023, pp. 1–21. [Online]. Available: https://aclanthology.org/2023.disrpt-1.1 [Cited on page 1.]

[4] B. Drury, H. G. Oliveira, and A. de Andrade Lopes, “A survey of the extraction and applications of causal relations,” *Natural Language Engineering*, vol. 28, pp. 361 – 400, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:246133706 [Cited on pages 1, 11, and 12.]

[5] A. Bondarenko, M. Wolska, S. Heindorf, L. Blübaum, A.-C. Ngonga Ngomo, B. Stein, P. Braslavski, M. Hagen, and M. Potthast, “CausalQA: A benchmark for causal question answering,” in *Proceedings of the 29th International Conference on Computational Linguistics*, N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond,

and S.-H. Na, Eds. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 3296–3308. [Online]. Available: https://aclanthology.org/2022.coling-1.291 [Cited on page 1.]

[6] R. Ceraolo, D. Kharlapenko, A. Khan, A. Reymond, R. Mihalcea, B. Schölkopf, M. Sachan, and Z. Jin, "Analyzing human questioning behavior and causal curiosity through natural queries," 2024. [Online]. Available: https://arxiv.org/abs/2405.20318 [Cited on page 1.]

[7] U. Jaimini, C. A. Henson, and A. P. Sheth, "An ontology design pattern for representing causality," in *WOP@ISWC*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:267682748 [Cited on page 2.]

[8] F. A. Tan, A. Hürriyetoğlu, T. Caselli, N. Oostdijk, T. Nomoto, H. Hettiarachchi, I. Ameer, O. Uca, F. F. Liza, and T. Hu, "The causal news corpus: Annotating causal relations in event sentences from news," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 2298–2310. [Online]. Available: https://aclanthology.org/2022.lrec-1.246 [Cited on pages 2 and 35.]

[9] P. Mirza and S. Tonelli, "An analysis of causality between events and its relation to temporal information," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, J. Tsujii and J. Hajic, Eds. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 2097–2106. [Online]. Available: https://aclanthology.org/C14-1198 [Cited on pages 2 and 39.]

[10] F. A. Tan, X. Zuo, and S.-K. Ng, "Unicausal: Unified benchmark and repository for causal text mining," 2023. [Online]. Available: https://arxiv.org/abs/2208.09163 [Cited on pages 2, 19, 20, 38, and 39.]

[11] F. A. Tan, H. Hettiarachchi, A. Hürriyetoğlu, N. Oostdijk, O. Uca, S. Thapa, and F. F. Liza, "Event causality identification - shared task 3, CASE 2023," in *Proceedings of the 6th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text*, A. Hürriyetoğlu, H. Tanev, V. Zavarella, R. Yeniterzi, E. Yörük, and M. Slavcheva, Eds. Varna, Bulgaria:

INCOMA Ltd., Shoumen, Bulgaria, Sep. 2023, pp. 144–150. [Online]. Available: https://aclanthology.org/2023.case-1.19/ [Cited on pages 2, 30, 36, and 43.]

[12] J. Yang, S. C. Han, and J. Poon, "A survey on extraction of causal relations from natural language text," 2021. [Online]. Available: https://arxiv.org/abs/2101.06426 [Cited on pages 5, 10, 20, and 34.]

[13] G. A. Miller, "Wordnet: a lexical database for english," *Commun. ACM*, vol. 38, no. 11, p. 39–41, Nov. 1995. [Online]. Available: https://doi.org/10.1145/219717.219748 [Cited on page 5.]

[14] Q. Bui, B. Nuallain, C. Boucher, and P. Sloot, "Extracting causal relations on hiv drug resistance from literature," *BMC Bioinformatics*, vol. 11, 2010. [Cited on page 6.]

[15] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, Jul. 2003, pp. 423–430. [Online]. Available: https://aclanthology.org/P03-1054/ [Cited on page 6.]

[16] P.-W. Kao, C.-C. Chen, H.-H. Huang, and H.-H. Chen, "NTUNLPL at FinCausal 2020, task 2:improving causality detection using Viterbi decoder," in *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, D. M. El-Haj, D. V. Athanasakou, D. S. Ferradans, D. C. Salzedo, D. A. Elhag, D. H. Bouamor, D. M. Litvak, D. P. Rayson, D. G. Giannakopoulos, and N. Pittaras, Eds. Barcelona, Spain (Online): COLING, Dec. 2020, pp. 69–73. [Online]. Available: https://aclanthology.org/2020.fnp-1.11/ [Cited on pages 7, 9, and 22.]

[17] F. A. Tan and S.-K. Ng, "NUS-IDS at FinCausal 2021: Dependency tree in graph neural network for better cause-effect span detection," in *Proceedings of the 3rd Financial Narrative Processing Workshop*, M. El-Haj, P. Rayson, and N. Zmandar, Eds. Lancaster, United Kingdom: Association for Computational Linguistics, 15-16 Sep. 2021, pp. 37–43. [Online]. Available: https://aclanthology.org/2021.fnp-1.6/ [Cited on pages 7, 23, and 24.]

[18] W. Ali, W. Zuo, W. Ying, R. Ali, G. Rahman, and I. Ullah, "Causality extraction: A comprehensive survey and new perspective," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 7, p. 101593, 2023. [Online].

Available: https://www.sciencedirect.com/science/article/pii/S1319157823001477 [Cited on pages 7 and 20.]

[19] Z. Li, Q. Li, X. Zou, and J. Ren, "Causality extraction based on self-attentive bilstm-crf with transferred embeddings," *Neurocomputing*, vol. 423, p. 207–219, Jan. 2021. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2020.08.078 [Cited on page 10.]

[20] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651021000012 [Cited on page 10.]

[21] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2018. [Online]. Available: https://arxiv.org/abs/1706.02216 [Cited on page 11.]

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762 [Cited on pages 12 and 13.]

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805 [Cited on pages 12 and 14.]

[24] M. Hu, Y. Peng, Z. Huang, and D. Li, "A multi-type multi-span network for reading comprehension that requires discrete reasoning," 2019. [Online]. Available: https://arxiv.org/abs/1908.05514 [Cited on page 15.]

[25] Z. Huang, J. Zhou, C. Niu, and G. Cheng, "Spans, not tokens: A span-centric model for multi-span reading comprehension," 10 2023, pp. 874–884. [Cited on page 16.]

[26] E. Markus and U. Adrian, *Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training*. IOS Press, 2020. [Online]. Available: http://dx.doi.org/10.3233/FAIA200321 [Cited on pages 16, 17, and 29.]

[27] B. Drury, H. Gonçalo Oliveira, and A. Lopes, "A survey of the extraction and applications of causal relations," *Natural Language Engineering*, vol. 28, pp. 1–40, 01 2022. [Cited on page 18.]

[28] C. S. G. Khoo, S. Chan, and Y. Niu, “Extracting causal knowledge from a medical database using graphical patterns,” in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. Hong Kong: Association for Computational Linguistics, Oct. 2000, pp. 336–343. [Online]. Available: https://aclanthology.org/P00-1043/ [Cited on page 18.]

[29] R. Girju and D. Moldovan, “Text mining for causal relations,” in *Proceedings of the FLAIRS Conference*, 2002, pp. 360–364. [Cited on page 19.]

[30] Z. Luo, Y. Sha, K. Q. Zhu, S. won Hwang, and Z. Wang, “Commonsense causal reasoning between short texts,” in *International Conference on Principles of Knowledge Representation and Reasoning*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:27965578 [Cited on page 19.]

[31] S. Heindorf, Y. Scholten, H. Wachsmuth, A.-C. Ngonga Ngomo, and M. Potthast, “Causenet: Towards a causality graph extracted from the web,” in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3023–3030. [Online]. Available: https://doi.org/10.1145/3340531.3412763 [Cited on page 19.]

[32] Z. Li, X. Ding, T. Liu, J. E. Hu, and B. Van Durme, “Guided generation of cause and effect,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, ser. IJCAI-PRICAI-2020. International Joint Conferences on Artificial Intelligence Organization, Jul. 2020, p. 3629–3636. [Online]. Available: http://dx.doi.org/10.24963/ijcai.2020/502 [Cited on page 19.]

[33] S. Gopalakrishnan, L. Garbayo, and W. Zadrozny, “Causality extraction from medical text using large language models (llms),” 2024. [Online]. Available: https://arxiv.org/abs/2407.10020 [Cited on pages 20 and 32.]

[34] T. Takayanagi, M. Suzuki, R. Kobayashi, H. Sakaji, and K. Izumi, “Is chatgpt the future of causal text mining? a comprehensive evaluation and analysis,” in *2024 IEEE International Conference on Big Data (BigData)*, 2024, pp. 6651–6660. [Cited on pages 20, 32, and 60.]

[35] D. Mariko, H. Abi-Akl, E. Labidurie, S. Durfort, H. De Mazancourt, and M. El-Haj, “The financial document causality detection shared task (FinCausal 2020),” in *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and*

MultiLing Financial Summarisation, D. M. El-Haj, D. V. Athanasakou, D. S. Ferradans, D. C. Salzedo, D. A. Elhag, D. H. Bouamor, D. M. Litvak, D. P. Rayson, D. G. Giannakopoulos, and N. Pittaras, Eds. Barcelona, Spain (Online): COLING, Dec. 2020, pp. 23–32. [Online]. Available: https://aclanthology.org/2020.fnp-1.3/ [Cited on page 21.]

[36] D. Mariko, H. A. Akl, E. Labidurie, S. Durfort, H. de Mazancourt, and M. El-Haj, "The financial document causality detection shared task (FinCausal 2021)," in *Proceedings of the 3rd Financial Narrative Processing Workshop*, M. El-Haj, P. Rayson, and N. Zmandar, Eds. Lancaster, United Kingdom: Association for Computational Linguistics, 15-16 Sep. 2021, pp. 58–60. [Online]. Available: https://aclanthology.org/2021.fnp-1.10/ [Cited on page 23.]

[37] D. Mariko, H. Abi-Akl, K. Trottier, and M. El-Haj, "The financial causality extraction shared task (FinCausal 2022)," in *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, M. El-Haj, P. Rayson, and N. Zmandar, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 105–107. [Online]. Available: https://aclanthology.org/2022.fnp-1.16/ [Cited on pages 21 and 25.]

[38] A. Moreno-Sandoval, J. Porta, B. Carbajo-Coronado, Y. Torterolo, and D. Samy, "The financial document causality detection shared task (FinCausal 2025)," in *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, C.-C. Chen, A. Moreno-Sandoval, J. Huang, Q. Xie, S. Ananiadou, and H.-H. Chen, Eds. Abu Dhabi, UAE: Association for Computational Linguistics, Jan. 2025, pp. 214–221. [Online]. Available: https://aclanthology.org/2025.finnlp-1.21/ [Cited on page 21.]

[39] G. Becquin, "GBe at FinCausal 2020, task 2: Span-based causality extraction for financial documents," in *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, D. M. El-Haj, D. V. Athanasakou, D. S. Ferradans, D. C. Salzedo, D. A. Elhag, D. H. Bouamor, D. M. Litvak, D. P. Rayson, D. G. Giannakopoulos, and N. Pittaras, Eds. Barcelona, Spain (Online): COLING, Dec. 2020, pp. 40–44. [Online]. Available: https://aclanthology.org/2020.fnp-1.5/ [Cited on pages 22, 25, and 49.]

[40] G. Haldar, A. Mittal, and P. Gupta, “DSC-IITISM at FinCausal 2021: Combining POS tagging with attention-based contextual representations for identifying causal relationships in financial documents,” in *Proceedings of the 3rd Financial Narrative Processing Workshop*, M. El-Haj, P. Rayson, and N. Zmandar, Eds. Lancaster, United Kingdom: Association for Computational Linguistics, 15-16 Sep. 2021, pp. 49–53. [Online]. Available: https://aclanthology.org/2021.fnp-1.8/ [Cited on page 24.]

[41] X. Wu, “NVJPFSI at FinCausal 2021 span-based causality extraction task,” in *Proceedings of the 3rd Financial Narrative Processing Workshop*, M. El-Haj, P. Rayson, and N. Zmandar, Eds. Lancaster, United Kingdom: Association for Computational Linguistics, 15-16 Sep. 2021, pp. 44–48. [Online]. Available: https://aclanthology.org/2021.fnp-1.7/ [Cited on page 25.]

[42] A. Saha, J. Ni, O. Hassanzadeh, A. Gittens, K. Srinivas, and B. Yener, “SPOCK at FinCausal 2022: Causal information extraction using span-based and sequence tagging models,” in *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, M. El-Haj, P. Rayson, and N. Zmandar, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 108–111. [Online]. Available: https://aclanthology.org/2022.fnp-1.17/ [Cited on page 25.]

[43] Z. Xu, R. Nararatwong, N. Kertkeidkachorn, and R. Ichise, “iLab at FinCausal 2022: Enhancing causality detection with an external cause-effect knowledge graph,” in *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, M. El-Haj, P. Rayson, and N. Zmandar, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 124–127. [Online]. Available: https://aclanthology.org/2022.fnp-1.21/ [Cited on page 26.]

[44] X. Chen, G. Zhang, A. Nik, M. Li, and J. Fu, “1Cademy @ causal news corpus 2022: Enhance causal span detection via beam-search-based position selector,” in *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, A. Hürriyetoğlu, H. Tanev, V. Zavarella, and E. Yörük, Eds. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 100–105. [Online]. Available: https://aclanthology.org/2022.case-1.14/ [Cited on pages viii, 27, 28, 33, 48, and 51.]

[45] M. Fajcik, M. Singh, J. P. Zuluaga-gomez, E. Villatoro-tello, S. Burdisso, P. Motlicek, and P. Smrz, “IDIAPers @ causal news corpus 2022: Extracting cause-effect-signal triplets via pre-trained autoregressive language model,” in *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, A. Hürriyetoğlu, H. Tanev, V. Zavarella, and E. Yörük, Eds. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 70–78. [Online]. Available: https://aclanthology.org/2022.case-1.10/ [Cited on page 29.]

[46] A. Saha, A. Gittens, J. Ni, O. Hassanzadeh, B. Yener, and K. Srinivas, “SPOCK @ causal news corpus 2022: Cause-effect-signal span detection using span-based and sequence tagging models,” in *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, A. Hürriyetoğlu, H. Tanev, V. Zavarella, and E. Yörük, Eds. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 133–137. [Online]. Available: https://aclanthology.org/2022.case-1.18/ [Cited on page 29.]

[47] A. Saha, O. Hassanzadeh, A. Gittens, J. Ni, K. Srinivas, and B. Yener, “A cross-domain evaluation of approaches for causal knowledge extraction,” 2023. [Online]. Available: https://arxiv.org/abs/2308.03891 [Cited on pages 29 and 30.]

[48] T. P. Schrader, S. Razniewski, L. Lange, and A. Friedrich, “BoschAI @ causal news corpus 2023: Robust cause-effect span extraction using multi-layer sequence tagging and data augmentation,” in *Proceedings of the 6th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text*, A. Hürriyetoğlu, H. Tanev, V. Zavarella, R. Yeniterzi, E. Yörük, and M. Slavcheva, Eds. Varna, Bulgaria: INCOMA Ltd., Shoumen, Bulgaria, Sep. 2023, pp. 38–43. [Online]. Available: https://aclanthology.org/2023.case-1.5/ [Cited on pages 31, 33, 57, and 58.]

[49] J. Straková, M. Straka, and J. Hajic, “Neural architectures for nested NER through linearization,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5326–5331. [Online]. Available: https://aclanthology.org/P19-1527/ [Cited on page 31.]

[50] D. Zeyrek, A. Mendes, Y. Grishina, M. Kurfalı, S. Gibbon, and M. Ogrodniczuk, "Ted multilingual discourse bank (ted-mdb): a parallel corpus annotated in the pdtb style," *Language Resources and Evaluation*, vol. 54, 04 2019. [Cited on pages 36 and 39.]

[51] F. Tan, "Causal news corpus," https://github.com/tanfiona/CausalNewsCorpus, 2022. [Cited on page 36.]

[52] C. Hidey and K. McKeown, "Identifying causal relations using parallel Wikipedia articles," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1424–1433. [Online]. Available: https://aclanthology.org/P16-1135/ [Cited on page 38.]

[53] J. Dunietz, L. Levin, and J. Carbonell, "The BECauSE corpus 2.0: Annotating causality and overlapping relations," in *Proceedings of the 11th Linguistic Annotation Workshop*, N. Schneider and N. Xue, Eds. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 95–104. [Online]. Available: https://aclanthology.org/W17-0812/ [Cited on page 39.]

[54] P. Mirza, R. Sprugnoli, S. Tonelli, and M. Speranza, "Annotating causality in the TempEval-3 corpus," in *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, O. Kolomiyets, M.-F. Moens, M. Palmer, J. Pustejovsky, and S. Bethard, Eds. Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014, pp. 10–19. [Online]. Available: https://aclanthology.org/W14-0702/ [Cited on page 39.]

[55] T. Caselli and P. Vossen, "The event StoryLine corpus: A new benchmark for causal and temporal relation extraction," in *Proceedings of the Events and Stories in the News Workshop*, T. Caselli, B. Miller, M. van Erp, P. Vossen, M. Palmer, E. Hovy, T. Mitamura, and D. Caswell, Eds. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 77–86. [Online]. Available: https://aclanthology.org/W17-2711/ [Cited on page 39.]

[56] B. Webber, R. Prasad, A. Lee, and A. Joshi, *The Penn Discourse Treebank 3.0 Annotation Manual*, University of Pennsylvania, Philadelphia, 2019. [Cited on page 39.]

[57] K. Ortmann, "Fine-grained error analysis and fair evaluation of labeled spans," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*,

N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 1400–1407. [Online]. Available: https://aclanthology.org/2022.lrec-1.150/ [Cited on page 44.]

[58] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta-large," https://huggingface.co/FacebookAI/roberta-large, 2019, pretrained weights available on Hugging Face. [Cited on page 48.]

[59] B. Chan, T. Möller, M. Pietsch, and T. Soni, "Roberta-large for extractive question answering," https://huggingface.co/deepset/roberta-large-squad2, pretrained weights available on Hugging Face. [Cited on page 49.]

[60] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta-base," https://huggingface.co/FacebookAI/roberta-base, 2019, pretrained weights available on Hugging Face.

[61] B. Chan, T. Möller, M. Pietsch, and T. Soni, "Roberta-base for extractive question answering," https://huggingface.co/deepset/roberta-base-squad2, pretrained weights available on Hugging Face. [Cited on page 49.]

[62] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert xxlarge v2," https://huggingface.co/albert/albert-xxlarge-v2, 2019, pretrained weights available on Hugging Face.

[63] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert large model (cased)," https://huggingface.co/google-bert/bert-large-cased, 2019, pretrained weights available on Hugging Face. [Cited on page 48.]

[64] Google, "Gemini api," https://ai.google.dev/gemini-api/docs?hl=pt-br, accessed: 2025-09-01. [Cited on page 60.]

[65] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, p. 1–30, Dec. 2006. [Cited on page 67.]

A. Ablation Study - Causal News Corpus

The trials described in this Appendix, mentioned in Chapter 5, were conducted on a T4 GPU in Google Colab, using multi-rel-per-input training, a pointer architecture, and a *standard warm-start* approach. Three configurations were considered:

- **Mean-Adjusted Loss Approach - baseline:** this setup computes the loss over all positions, assigning zero to ignored positions, and then averages across the entire tensor;
- **Standard Loss Averaging Approach:** this approach excludes ignored positions during loss computation, following the default behavior of the loss function;
- **Overlap Penalization:** builds upon the baseline by incorporating an additional Jensen-Shannon Divergence regularization term, designed to penalize overlapping relation predictions.

As part of the experimental setup, a Jensen-Shannon (JS) Divergence regularization term was incorporated to encourage the model to output distinct and non-overlapping relations. This approach targets cases in which the model produces similar span distributions for multiple relations within the same input. For each predicted relation, softmax probabilities were computed over the start and end logits for ARG0, ARG1, and Signal components. These were concatenated to form a composite representation of each relation. Pairwise JS Divergence was then calculated between all relation pairs within the same sample. A penalty was applied when the divergence fell below a predefined margin, indicating insufficient distinction between predicted relations. This penalty was added directly to the total training loss, alongside the standard cross-entropy losses used for span prediction.

Table A.1: Performance comparison across different configurations.

Class	Mean-Adj. Loss			Standard Loss			Loss w/ Overlap.		
	P	R	F1	P	R	F1	P	R	F1
Cause	0.75	0.79	0.77	0.75	0.77	0.76	0.77	0.78	0.78
Effect	0.74	0.74	0.74	0.75	0.74	0.75	0.76	0.75	0.75
Signal	0.67	0.80	0.73	0.66	0.77	0.71	0.65	0.77	0.71
Overall	0.73	0.77	0.75	0.72	0.76	0.74	0.73	0.77	0.75

B. Loss Convergence: MultiGPU vs. SingleGPU

As explained in Chapter 5, we were able to run the experiments with the architectures predicting start and end positions in a multi-GPU setting in the Virtual Artificial Intelligence Laboratory (AIvLAB) on the Deucalion platform. However, when using the sequence tagging architectures, all tests had to be run on Google Colab in a single-GPU environment.

The results for Cause Relation Extraction using the sequence tagging architectures on Deucalion were very poor. We observed that during training on Deucalion (multi-GPU), the loss stabilized at a much higher value (around 60), whereas on Colab (single GPU), it converged properly, dropping close to zero. This inconsistency did not occur in the other approach. The graphs below show this difference clearly. We used Hugging Face's Accelerate for both single-GPU and multi-GPU training, along with PyTorch's Distributed-DataParallel (DDP) in the distributed setup.

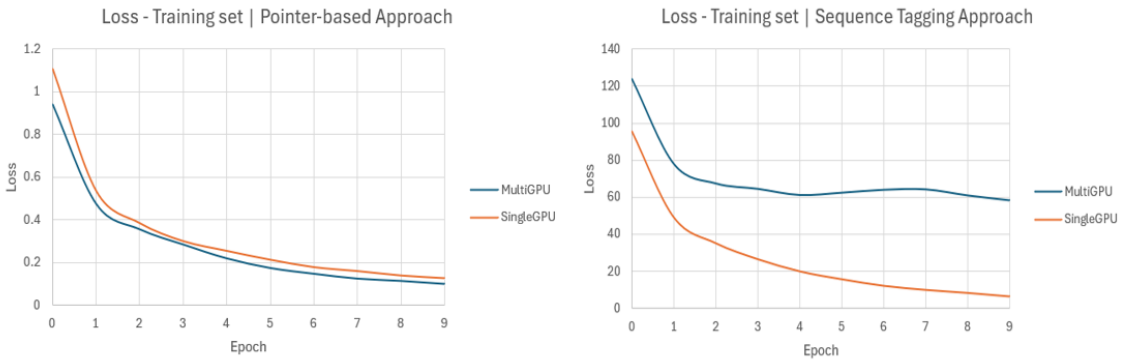


Figure B.1: Loss Differences - MultiGPU vs SingleGPU.

C. CDF of maximum relation scores on Benchmark Datasets

The CDF curves of non-causal and causal entries of all validation benchmarks sets are presented below. This was important to define threshold_1 as mentioned in Chapter 7.

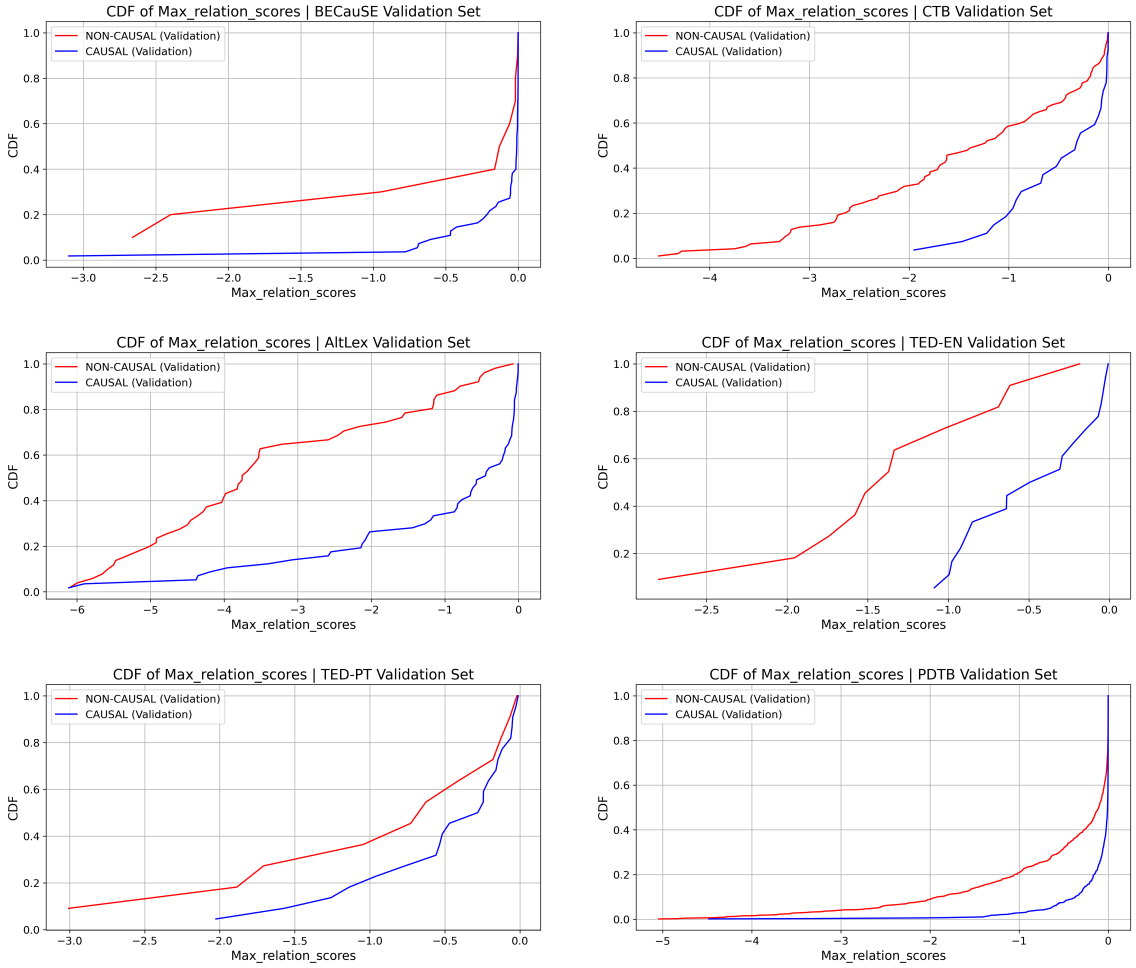


Figure C.1: CDF of maximum relation scores on benchmark validation sets.

D. Comparison of Predicted vs. Actual Relation Counts on Benchmark Datasets

As mentioned in Chapter 7, the matrices for all test sets after performing all post-processing steps are presented in this Appendix.

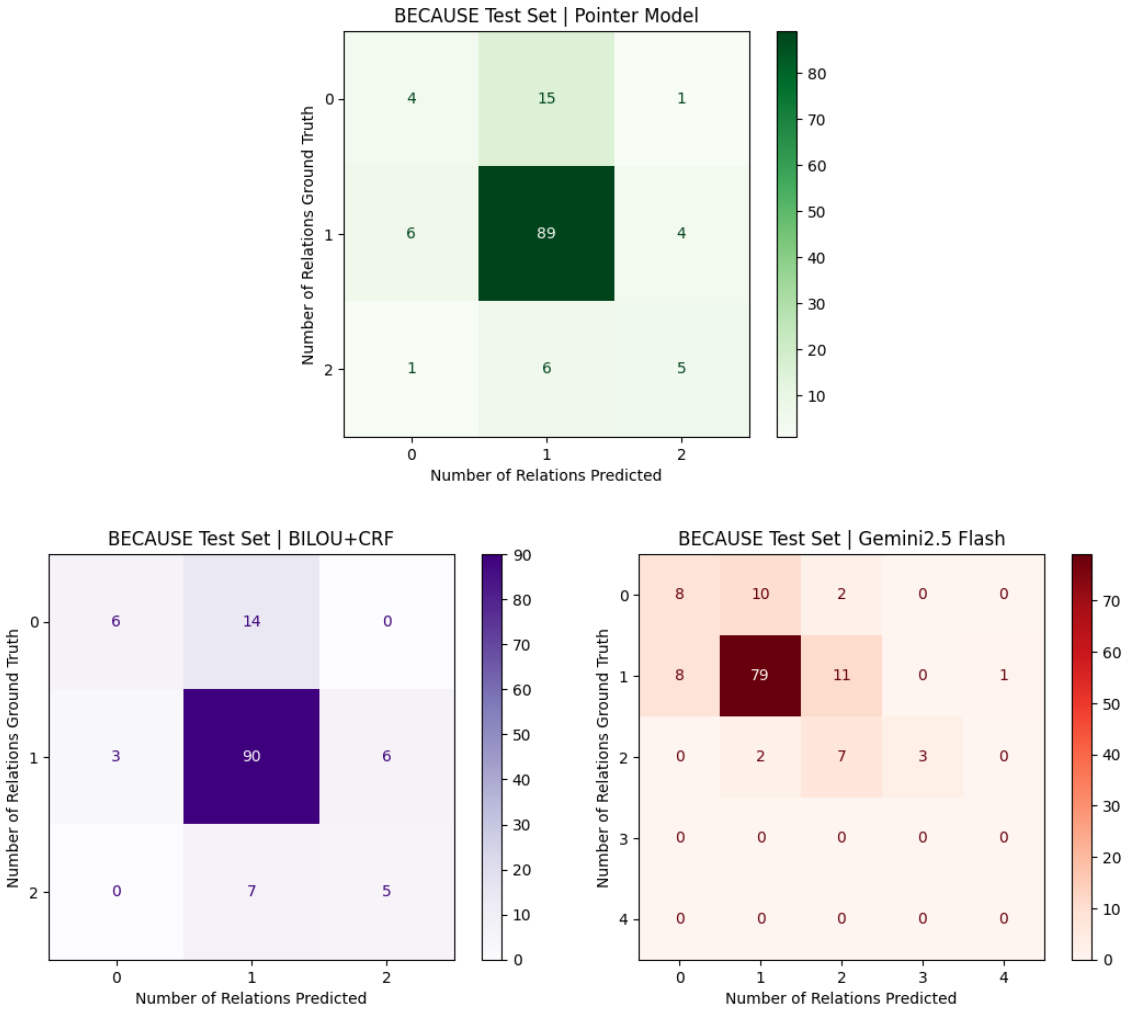


Figure D.1: Relation Count Matrices on BECAUSE Test Set. Top: Pointer Model; Bottom: BILOU+CRF (left) and Gemini2.5 Flash (right).

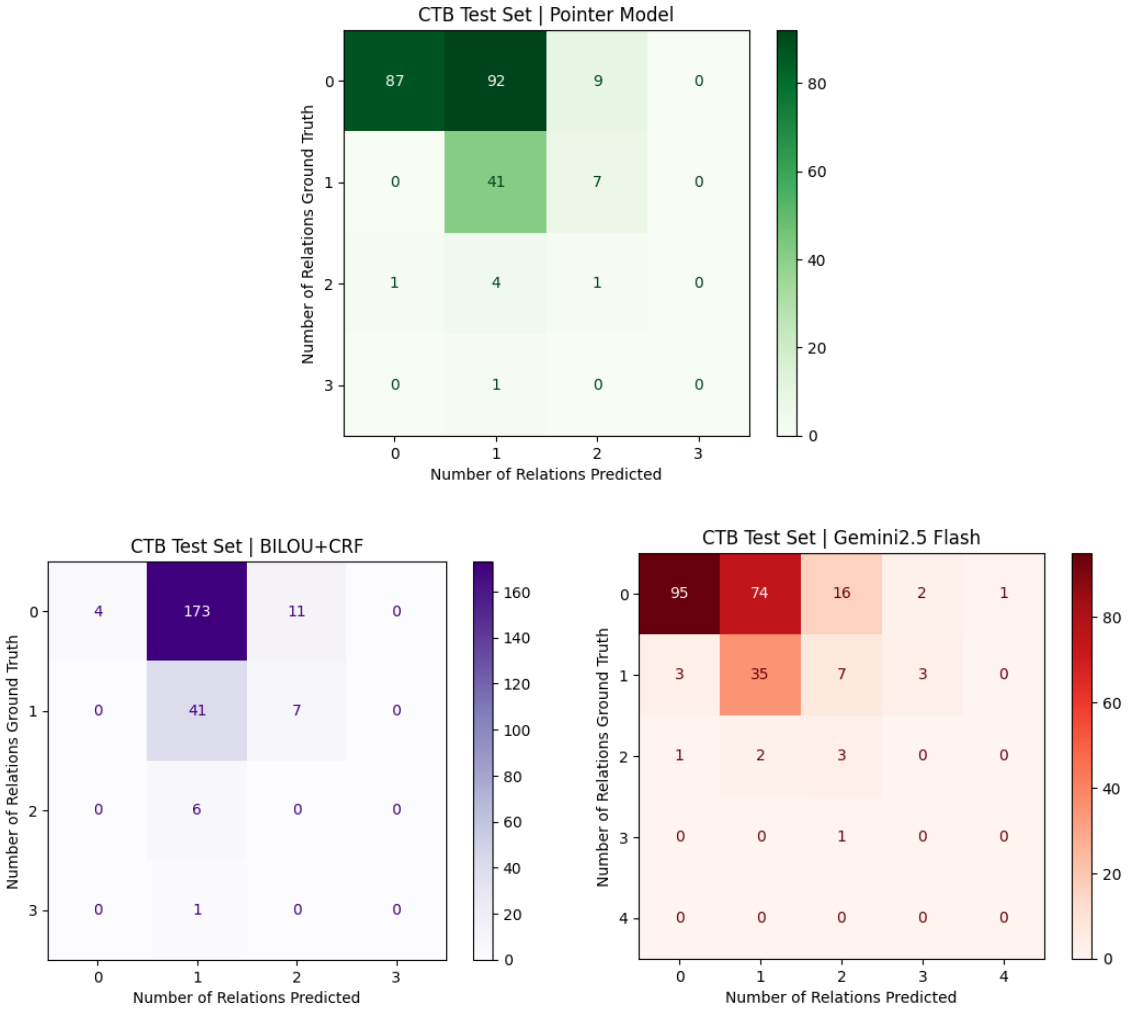


Figure D.2: Relation Count Matrices on CTB Test Set. Top: Pointer Model; Bottom: BILOU+CRF (left) and Gemini2.5 Flash (right).

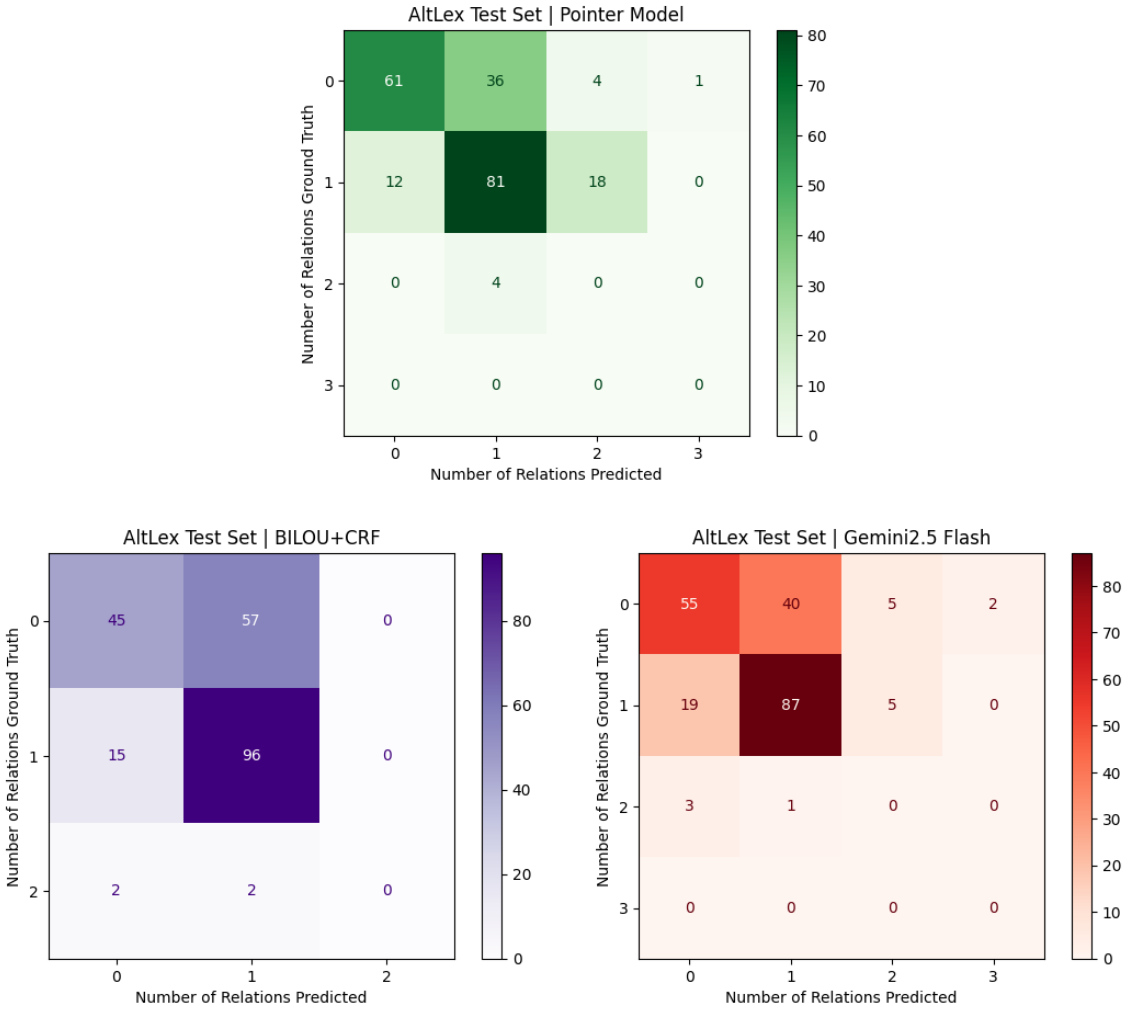


Figure D.3: Relation Count Matrices on AltLex Test Set. Top: Pointer Model; Bottom: BILOU+CRF (left) and Gemini2.5 Flash (right).

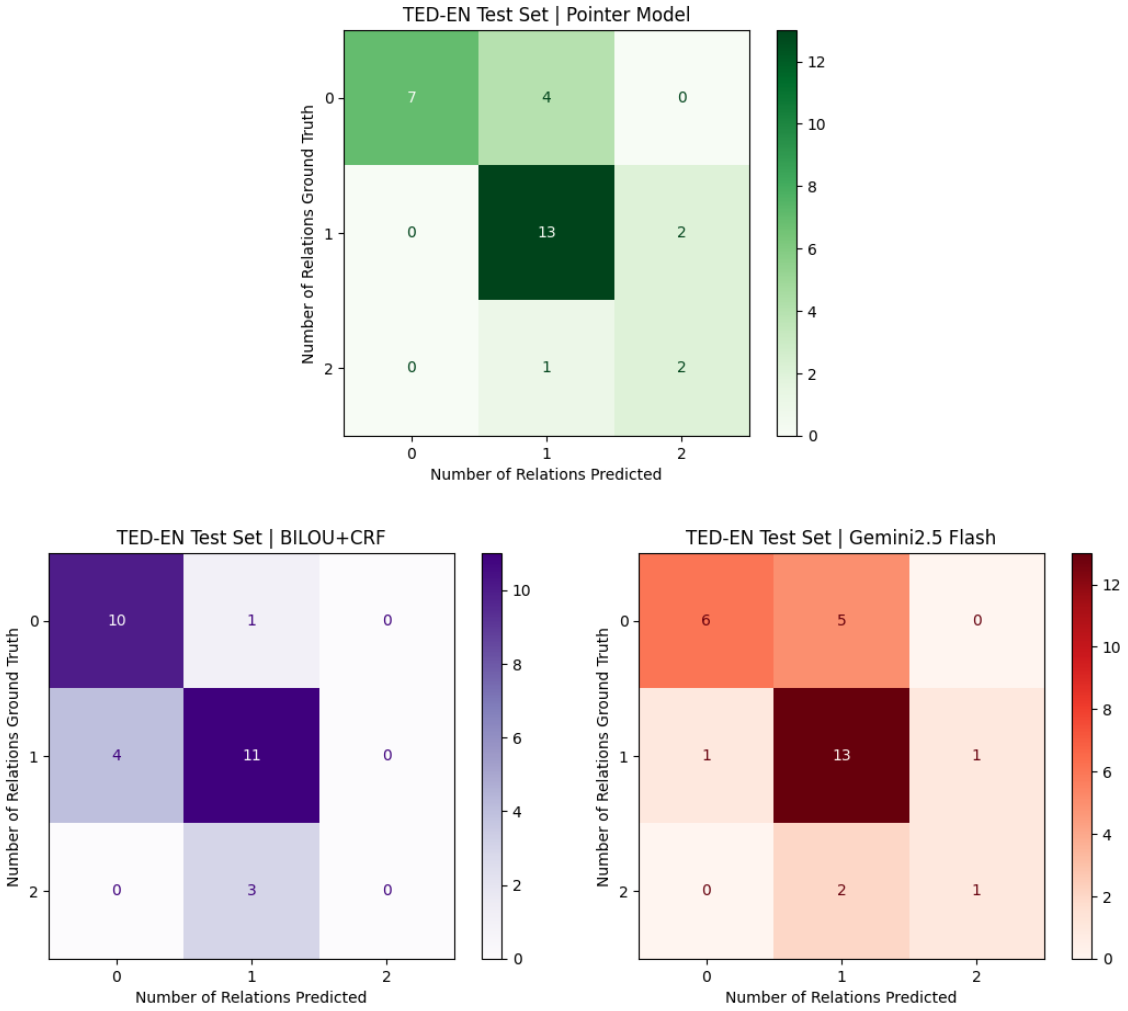


Figure D.4: Relation Count Matrices on TED-EN Test Set. Top: Pointer Model; Bottom: BILOU+CRF (left) and Gemini2.5 Flash (right).

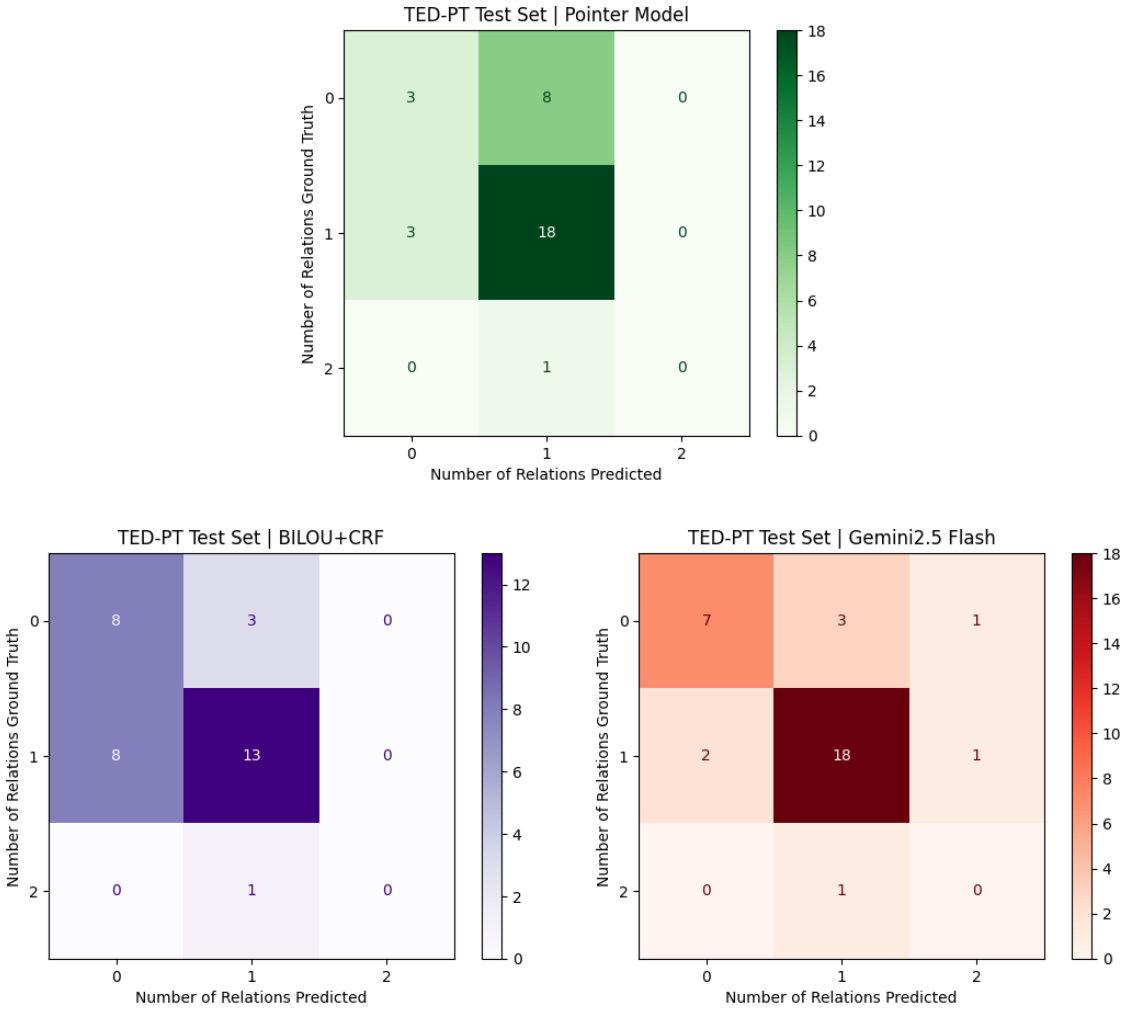


Figure D.5: Relation Count Matrices on TED-PT Test Set. Top: Pointer Model; Bottom: BILOU+CRF (left) and Gemini2.5 Flash (right).

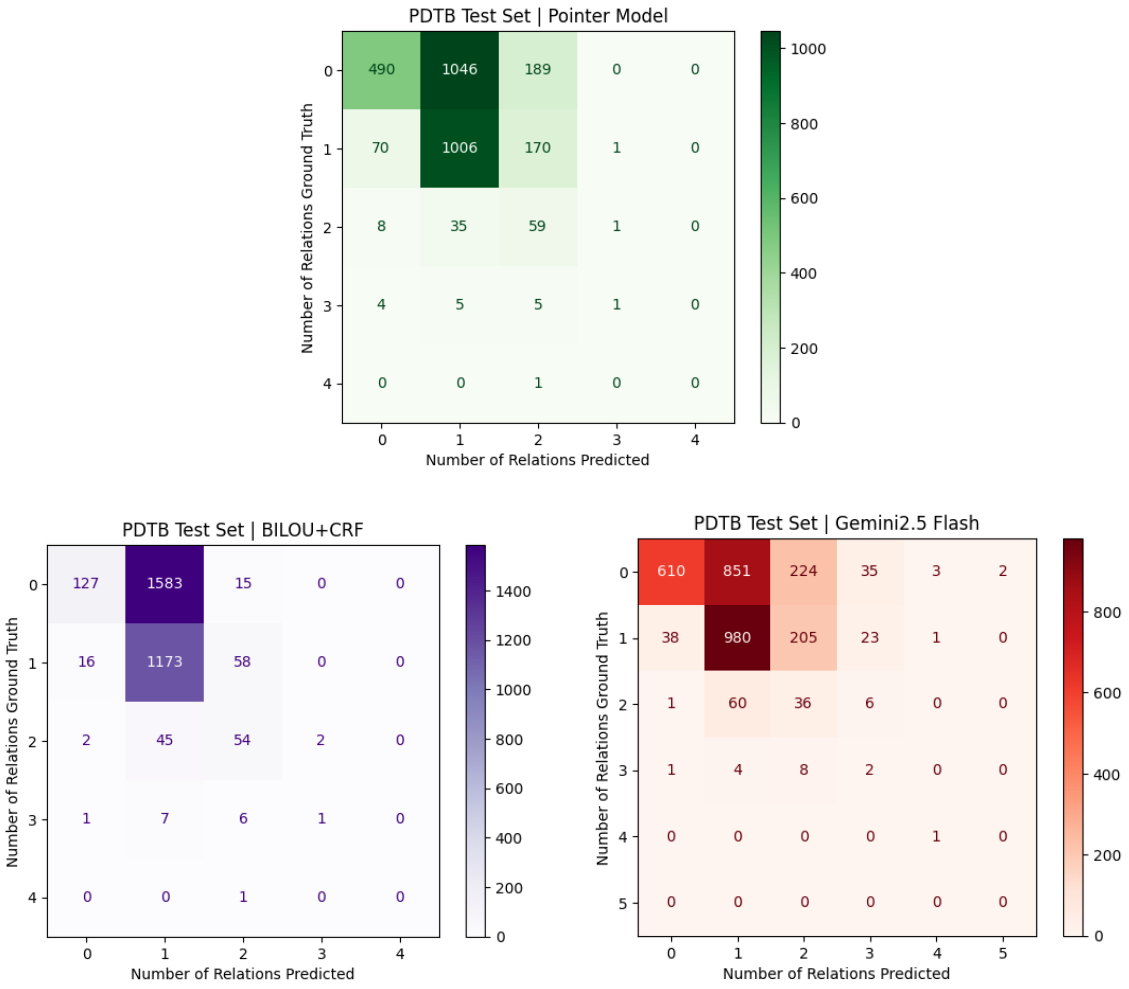


Figure D.6: Relation Count Matrices on PDTB Test Set. Top: Pointer Model; Bottom: BILOU+CRF (left) and Gemini2.5 Flash (right).