Event Detection as Multi-Task Text Generation

by

Ujjwala Anantheswaran

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2022 by the
Graduate Supervisory Committee:

Chitta Baral, Chair
Nakul Gopalan
Hannah Kerner

ARIZONA STATE UNIVERSITY

December 2022

ABSTRACT

Event detection refers to the task of identifying event occurrences in a given natural language text. Event detection comprises two subtasks; recognizing event mention (event identification) and the type of event (event classification). Breaking from the sequence labeling and word classification approaches, this work models event detection, and its constituent subtasks of trigger identification and trigger classification, as independent sequence generation tasks. This work proposes a multi-task generative model trained on event identification, classification, and combined event detection. The model is evaluated on on general-domain and biomedical-domain event detection datasets, achieving state-of-the-art results on the general-domain Roles Across Multiple Sentences (RAMS) dataset, establishing event detection benchmark performance on WikiEvents, and achieving competitive performance on the general-domain Massive Event Detection (MAVEN) dataset and the biomedical-domain Multi-Level Event Extraction (MLEE) dataset.

DEDICATION

This work is dedicated to my loving family and dearest friends, whose unconditional love and unwavering faith makes all that I do possible.

ACKNOWLEDGMENTS

Working on this thesis has been an exciting and enlightening journey, that has helped me grow as a researcher. I would like to acknowledge everyone who contributed to making the culmination of this work possible. Primarily, I would like to convey my deepest gratitude to my advisor, Dr Chitta Baral, for not only the opportunity to gain valuable research experience in this domain, but also his supervision and guidance over the entire span of this work. In addition, I would like to sincerely thank my colleagues and fellow researchers, Mr Mihir Parmar and Mr Kuntal Kumar Pal, whose advice and motivation has been in an invaluable aid in the progress of this work.

I would also like to express my gratitude to all the researchers who are part of the Cogint Lab, Arizona State University, for their part in creating an enriching research environment. Finally, I would like to thank my former colleagues, Ms Shivani Priya, Ms Pankhuri, and Ms Ayushi Shekhar for their unwavering support, without which the completion of this thesis would have been much harder.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Table Page

Chapter 1

INTRODUCTION

Event Detection is a crucial language-based information retrieval task. Event Detection (ED) aims to identify salient words or phrases, known as event triggers, from unstructured texts, and classify them into predefined event types that describe the type of event scenario they invoke. It is central to event understanding (Zhang *et al.*, 2022).

Efficient and accurate event detection is the first step in most larger Event Extraction task for more comprehensive information retrieval from language data. Furthermore, event detection as a standalone task has applications in many downstream tasks, such as information retrieval (Jungermann and Morik, 2008; Kanhabua and Anand, 2016), question answering (Costa *et al.*, 2020), and prediction of implicit arguments from event information (Cheng and Erk, 2018).

1.1 Event Detection

An event is a specific occurrence, something that "happens", and is significant to the subject of the text. It involves participants. An event can often be described as a change of state (Consortium, 2005). An event trigger is the word or phrase that most clearly expresses the event occurrence and intent. The event type is a label from a predefined schema that best describes the characteristics of the event being triggers, and defines the scope of roles involved in understanding the event. These participants are referred to as event arguments, and play a role in creating a comprehensive picture of the event in question.

Both the Event Detection subtasks are evaluated with micro precision, recall, and

F-1 scores.

1.2 ED as a Precursor to EE

Event Detection (ED) is foundational to the more comprehensive Event Extraction (EE) task (Ahn, 2006). Argument extraction, and event extraction as a whole, depends heavily on event detection. While ED can be performed as an independent task and has its own uses, the wider EE task, which does more comprehensive information retrieval, requires extracting arguments and roles, and doing so is impossible without first identifying the event in its entirety. Detection of event triggers helps identify the arguments associated with them, while classifying them into the correct event type allows models to not only assign argument roles, but even better identify arguments, with the knowledge of the detected event type and the expected roles associated with them.

This is especially evident in recent problems which attempt to perform EE in low-resource settings, without expensive annotation schemes or sufficient coverage. Attempting to perform argument extraction in the absence of a thorough schema, or in the absence of a schema entirely, places even more importance on the subtask of ED, as the schema must be inferred from event triggers and their types. Without correct identification and classification, even powerful pretrained models with vast reserves of knowledge cannot leverage the semantic information to perform argument extraction, and by extension end-to-end event extraction, meaningfully.

In light of this dependency, successful event extraction models have time and again implemented pipelined architectures (Ahn, 2006; Si *et al.*, 2021; Liu *et al.*, 2022) which perform ED as a precursor to argument extraction to create a complete EE system. A common problem they face is the problem of error propagation, originating from subpar ED modules that handicap the overall efficacy of their EE systems. Improving

performance on ED is critical to improving existing and future EE frameworks.

1.3 Research value and Contributions

1.3.1 Research Evaluation

We treat this problem as sequence generation. The few existing works that treat ED and EE as sequence generation evaluate their performance in accordance with existing word classification models. In order to evaluate our model performance fairly and accurately in comparison to these, we also follow the word classification paradigm for evaluation. The generated sequences are post-processed and framed as token classifier outputs. We use the common seqeval package used for entity recognition to obtain precision, recall, and micro and macro F-1 scores.

1.3.2 Contributions

1. To the best of our knowledge, this is only the second work to leverage ED subtasks, the first to utilize all of the ED subtasks separately and jointly, and the first one to do so while moving away from the dominant token classification paradigm.

2. This thesis presents the novel technique of multitasking over a single complex ED task by implementing a generative reformulation to its constituent subtasks of both EI and EC, and then creating a single robust multitask model that not only performs the tasks of EI and EC, but also improves performance on our primary task of ED by leveraging its learned knowledge of the subtasks that make up ED.

3. This work adds to the canon of research works in the active domain of using instructional prompts, as well as research works that formulate ED as sequence

generation (of which there are very few)

4. This work establishes new state-of-the-art benchmarks on the general domain dataset RAMS and establishes the benchmark ED performance on the general domain WikiEvents dataset. In addition, we present a more robust and comprehensive model for the general domain MAVEN dataset, which improves on some shortcomings of existing state-of-the-art models.

1.4 Structure of Thesis

This work will begin with enumerating the task definition and the importance of Event Detection to comprehensive information retrieval. Chapter 2 offers an overview of the state of research in this domain and its dominant paradigms. In chapter 3, the details and distinctive features of the datasets used in the development of this work are discussed. Chapter 4 details the novel generative reformulation necessary to conform to the most recent research paradigm. Chapter 5 discusses another novel technique used in this work - prompted multi-tasking over all Event Detection subtasks. This chapter also provides a comprehensive analysis of our method in different settings. The final chapter summarizes key results and findings, while detailing existing limitations and possible avenues of future research.

Chapter 2

LITERATURE REVIEW

This chapter provides an overview of major paradigms in Event Detection, with brief explanations of pioneering and seminal works, as well as the state of the most recent research endeavours in the paradigms. In keeping with the primary task of this thesis work, this literature review will focus on supervised event detection. As many of the works highlighted in this chapter perform Event Extraction (of which Event Detection is a preliminary subtask) in a pipelined fashion, this allows us to observe their performance on the ED task independently. Section 2.1 enumerates the earliest work in this domain - highlighting works that first formulated the problem of Event Detection, and the use of simple classifiers and statistical methods that relied more heavily on engineering complex hand-crafted features with domain expertise. Section 2.2 introduces the subsequent paradigm of using neural network based, where researchers focused with different neural network configurations such as CNNs, RNNs, and GCNs to model complex relations. Recently, successful event detection and event extraction models have leveraged deep learning architectures, especially Transformer-based Pretrained Language Models (PLMs) to learn complex semantic relationships and longer-term contexts. We explore these models in Sections 2.3. Finally, we briefly discuss the most recent breakthrough: performing ED as a sequence generation task, a paradigm which is drastically different from all before it. Research in this avenue is very new and still ongoing, and it is hoped that this work will add to the canon of research in this domain.

2.1 Feature-based Models

The earliest formulation of Event Extraction as a task comes from Consortium (2005), which defined this specific information retrieval task, while also providing the widespread terminology of entities, event triggers, types, arguments, and argument roles. Ahn (2006) was one of the first works to delineate the stages of this task further, providing the now-commonplace subtask decomposition of Event Extraction into Event Detection, i.e. trigger identification and classification, and Argument Extraction. The scope and type of features being used was advanced further by the inclusion of cross-event features (Gupta and Ji, 2009; Liao and Grishman, 2010) and cross-document features (Ji and Grishman, 2008). Subsequent works (Gupta and Ji, 2009; Riedel *et al.*, 2010) used handcrafted features to perform Event Extraction and its adjacent tasks such as temporal inference and relational extraction respectively. Hong *et al.* (2011) leveraged cross-entity reference, proposing fine-grained entity consistency as a key feature to trigger identification and classification (this idea was explored further by Liu *et al.* (2016)). This work combined unsupervised ML techniques such as clustering along with information retrieval by mining and used statistical classifiers to infer trigger existence and types.

The majority of models so far had been pipelined models that performed trigger identification separately from classification. Chen and Ng (2012) proved that training a model on trigger identification and classification jointly was a superior approach to ED. Furthermore, joint training not only allowed knowledge sharing, but also reduced error propagation along the ED pipeline. This denoted a shift from pipelined models (Ji and Grishman, 2008; Gupta and Ji, 2009; Patwardhan and Riloff, 2009; Liao and Grishman, 2011; McClosky *et al.*, 2011; Huang and Riloff, 2021; Li *et al.*, 2013a) to joint training architectures (Riedel and McCallum, 2011b,a; Li *et al.*, 2013b;

Venugopal *et al.*, 2014).

Li *et al.* (2013b) was the first architecture to treat end-to-end EE as structured prediction. It used a set of complex human-designed lexical features, entity information, and syntactic features as local features, in addition to global features (for example, context and dependency path between multiple triggers), over which it trained a structured perceptron. Bronstein *et al.* (2015) adapted this approach to low-resource settings by using Wordnet in conjunction with this architecture to augment annotated data. Liu *et al.* (2016) leveraged global features such as event-event association further, while integrating latent local information such as fine-grained entity type. This also marked the beginning of methods integrating external knowledge sources to improve ED, like Chen *et al.* (2017); Liu *et al.* (2017). These models leveraged information such as argument annotations, and language resources such as Framenet (Baker *et al.*, 1998) and Freebase (Bollacker *et al.*, 2008). Liu *et al.* (2017) used external knowledge and rule-based methods, along with constraints on word features such as POS tags and entity labeling to automatically annotate unlabeled external data, thus generating more samples. However, the choosing features is a labour-intensive process and requires linguistic intuition as well as domain expertise. This makes these models less suited for new application domains and limits the cross-domain adaptability. Furthermore, the resources for feature extraction might involve errors, which may be propagated to the main event detector.

2.2 Neural Network Architectures

The works in this section were among the first to leverage pretrained embeddings (Mikolov *et al.*, 2013b,a) as features, thus reducing the need labour-intensive feature engineering. Nguyen and Grishman (2015); Chen *et al.* (2015) were among the earliest works to formulate ED as a token-classification problem, and use surround-

ing tokens as context. Both these approaches used Convolutional Neural Networks (CNNs), which require a fixed window size for the context. Nguyen and Grishman (2015) proved the domain adaptability of neural models. It used word embeddings, position embeddings, and entity type embedding as features. DMCNN (Chen *et al.*, 2015), a pipelined ED model, used dynamic multi-pooling to retrieve multiple events per sentence. JRNN (Nguyen *et al.*, 2016) used a joint training scheme, in addition to using LSTM (Long Short-Term Memory) networks to improve local contextual information. Recurrent Neural Networks (RNNs) also found use in Ghaeini *et al.* (2016), one of the first methods to attempt to capture multi-word triggers. The tree-structured dbRNN (Sha *et al.*, 2018) also used an RNN, enhanced with dependency bridges to carry syntactically related information and a tensor layer to capture latent interaction between trigger candidates. Another jointly trained model, JMEE (Liu *et al.*, 2018), used a Graph Convolutional Network (GCN) to learn syntactic contextual representations of each node by the leveraging the representative vectors of its immediate neighbors in the graph.

2.3 Transformer-based Models

With the advent of Transformer models (Vaswani *et al.*, ????) that have shown promising results across language tasks, there has been a surge of joint and pipelined architectures that perform ED in diverse settings. The wealth of knowledge behind pretrained language models (PLMs) make them a more powerful tool to leverage for language tasks. Unsurprisingly, frameworks integrating PLMs outstrip previously established baselines on multiple tasks and datasets. However, PLMs are trained on general domain data, making them less suited to successful right-out-of-the-box applications on tasks which require domain-specific knowledge and contexts. Most ED and EE architectures leverage BERT (Devlin *et al.*, 2018) to perform word classifi-

cation. PLMEE (Yang *et al.*, 2019) uses BERT embeddings along with a pipelined structure for EE, where predicted triggers and types are used for argument extraction. This work treats multi-word triggers as separate tokens with the same event type class label assigned. DMBERT (Wang *et al.*, 2019) uses BERT along with convolutional layers to encode word and position embeddings and then uses dynamic multi-pooling. It also employs adversarial training: uses a discriminator to judge the accuracy of annotations, improving performance on noisy data and allowing weakly or semi supervised applications. Lu *et al.* (2019) decouples lexical-specific and lexical-free representations and improves performance on sparse event types. EKD (Tong *et al.*, 2020) and GPTEDOT (Veyseh *et al.*, 2021) use different Transformer-based architectures (BERT and GPT-2 (Radford *et al.*, 2019), respectively) to generate data. GPTEDOT, however, generates data not for ED, but for Event/Trigger Identification, due to relative simplicity and ease of evaluating quality of generated data. It uses these generated samples and the original ED annotations to train a multi-task model with improved trigger identification capabilities. More recent works that use BERT embeddings for word classification are SaliencyED (Liu *et al.*, 2022) which uses ensemble classifiers, and OntoED (Deng *et al.*, 2021) that can perform low-resource and zero-shot event detection. BERT embeddings also find use in graph-based architectures such as DyGIE++ (Wadden *et al.*, 2019) and OneIE (Lin *et al.*, 2020), where event triggers are set as nodes with edges capturing cross-subtask and cross-instance interactions. Finally, APEX (Wang *et al.*, 2022a) augments input with type-specific prompts and seed triggers to perform ED using BERT-large. CLEVE (Wang *et al.*, 2021b) and Yu *et al.* (2021) are two recently proposed architectures that leverage the pretraining paradigm to improve ED on a target dataset.

Many Transformer-based models also leverage the pretrained knowledge of BERT

by framing ED as a question answering task. These include BERT_QA (Du and Cardie, 2020), Boros *et al.* (2021), and RCEE (Liu *et al.*, 2020), which use simple questions to perform trigger extraction. While these models perform ED jointly, MQAEE (Li *et al.*, 2020) performs ED in a pipelined fashion, by using separate questions for EI and EC, and using the answer of the former to form the question for the latter. MQAEE and Wang *et al.* (2021a) treat ED as machine reading comprehension.

With the advent of more powerful sequence-to-sequence models such as T5 (Raffel *et al.*, 2020), there has been an increased interest in formulating a range of language tasks, including event detection and event extraction, as sequence generation tasks. TANL (Paolini *et al.*, 2021) formulates ED and other information retrieval tasks as augmented translation tasks. Text2Event (Lu *et al.*, 2021) uses a complex tree-based structure, while GDAP (Si *et al.*, 2021) uses a relatively simpler pipelined system to perform ED and EE. GDAP, unlike other pipelined ED models in the past, performs EC before EI, and uses predicted event types as prompts for corresponding trigger extraction. With this thesis, we add to the canon of works leveraging the sequence generation and promptng paradigm, and introduce the addition of instructional prompts for ED and its subtasks.

Chapter 3

DATASETS

In view of the motivation expressed before: the improvement of ED as a preliminary to EE, the majority of the datasets used in this work are EE datasets, maintaining the scope of possible extensions of the proposed approach to the argument extraction task using similar or identical experimental settings.

The datasets we choose to demonstrate our method on span a range of characteristics, from sentence-level to multi-sentence level, with varying proportions of non-event instances, and include multi-word triggers and multi-class triggers, on which the efficacy of trigger extraction techniques are understudied. In addition to these general domain datasets, we include a domain dataset to illustrate the performance of our method on domain-specific datasets.

3.1 RAMS

RAMS (Ebner *et al.*, 2020) is a dataset created primarily for the task of multi-sentence argument linking. This dataset has 9,124 annotated event triggers across 38 event types and subtypes. The annotated argument roles are in a 5-sentence window around the related event trigger. The official data split in the original paper defines 3,194, 399, and 400 documents for training, development, and testing respectively.

3.2 MAVEN

The MAssive eVENt detection dataset, or MAVEN, (Wang *et al.*, 2020) was proposed with the idea of combating data scarcity and low coverage problem in prevailing general domain event detection datasets. It contains trigger and event type annota-

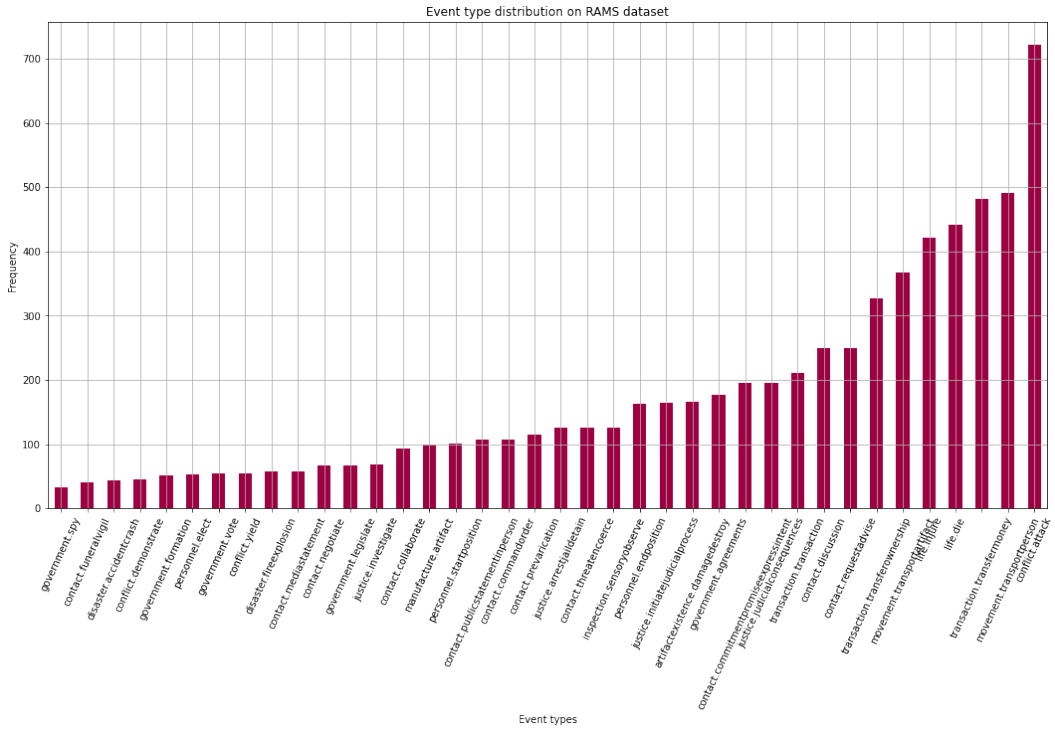

Figure 3.1: Distribution of Event Types in RAMS

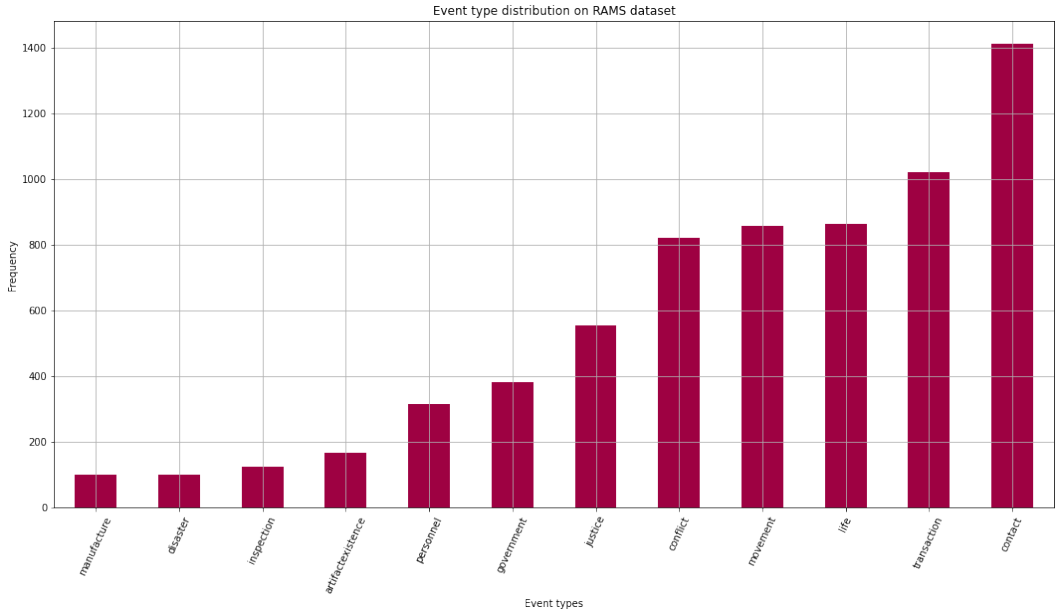

Figure 3.2: Distribution of Macro Event Types in RAMS

Event type	Frequency	Example triggers
conflict.attack	721	massacre, battle, bombing
movement.transportperson	491	smuggling, walked, incarcerate
transaction.transfermoney	482	reimbursed, paid, purchasing
life.die	442	die, murder, assassinating
life.injure	422	surgery, injured, brutalized
movement.transportartifact	367	imported, trafficking, smuggling
transaction.transferownership	327	auction, donated, acquire
contact.requestadvise	250	advocating, recommending, urged
contact.discussion	249	discuss, meet, negotiated
transaction.transaction	211	funded, donated, seized

Table 3.1: Top 10 Event Types in RAMS, along with Example Triggers.

Event type	Frequency	Example triggers
process_start	2468	began, debut, took place
causation	2465	resulted in, caused, prompted
attack	2255	bombing, attacked, struck
hostile_encounter	1987	fought, conflict, battle
motion	1944	fell, pushed, moved
catastrophe	1785	explosion, hurricane, flooded
competition	1534	event, championships, match
killing	1380	killed, murder, massacre
process_end	1323	closing, complete, ended
statement	1269	asserted, proclaimed, said

Table 3.2: Top 10 Event Types in MAVEN, along with Example Triggers.

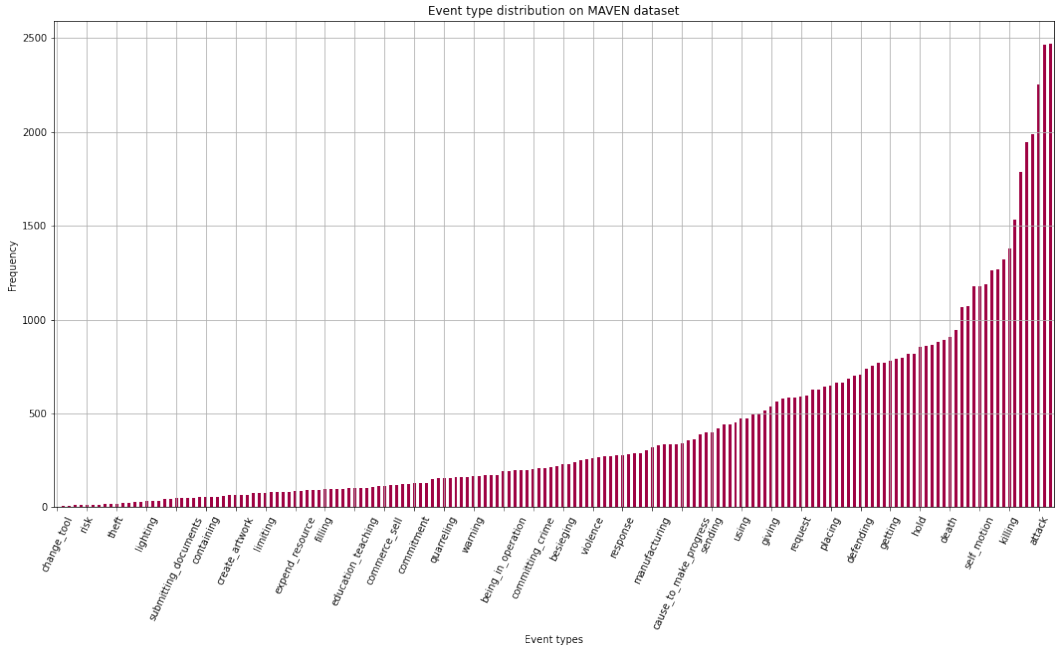

Figure 3.3: Distribution of Event Types in MAVEN

tions on 4,480 documents from Wikipedia, with close to 119K event mentions classified into 168 event types. The event types in MAVEN are derived from the frames defined in the linguistic resource Frame net (Baker *et al.*, 1998). MAVEN is a dataset aimed at event detection, and thus lacks argument role annotations. We use MAVEN, despite the lack of argument annotations, as it provides by far the largest coverage in terms of event diversity, as the statistics will make clear. The high event coverage provided by MAVEN results in more events per sentence on average, as compared to other general domain ED datasets. Another significant feature is the frequency of multi-word triggers, or trigger phrases.

Despite the large scope of coverage offered by MAVEN, out of its 168 types, the majority of documents deal with a subset of all event types. 18% of all event types (for example, *Breathing* and *Change Tool*) have less than 100 annotated instances, making them hard to learn and identify (Zhang *et al.*, 2022). The dataset, reflective of real world data, has a long tail distribution (Wang *et al.*, 2020), as seen in Figure

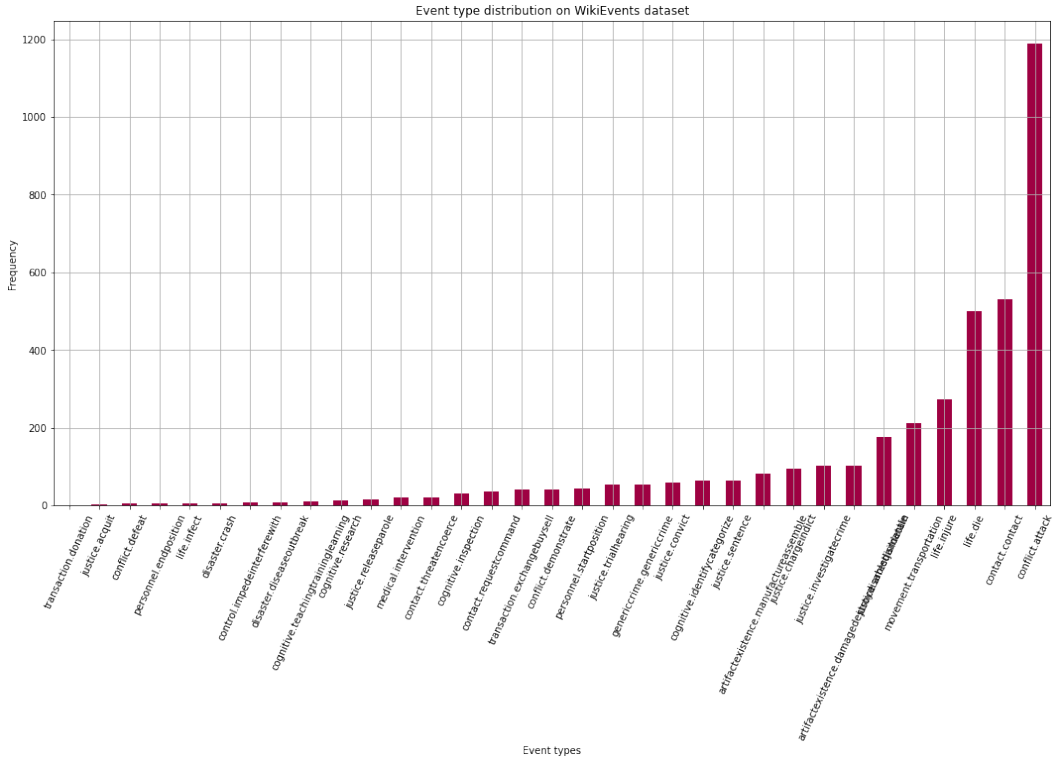

Figure 3.4: Distribution of Event Types in WikiEvents

3.3. Using macro metrics to evaluate performance on this dataset can give us an idea of comparative model performance on sparsely represented event classes.

3.3 WikiEvents

The WikiEvents dataset is a dataset for argument extraction proposed by Li *et al.* (2021). It contains 34 event types in a two-level hierarchy, and 67 event types in a three-level hierarchy. Like MAVEN, this dataset is highly imbalanced, with only 3 event types populated with more than 400 instances across training and testing splits. The most frequent event, $conflict.attack$, has nearly twice as many instances as the next most frequent event type. Furthermore, 54% of the dataset instances are negative instances, i.e. instances with no event occurrences. Existing work on this dataset focuses exclusively on document-level argument extraction and event

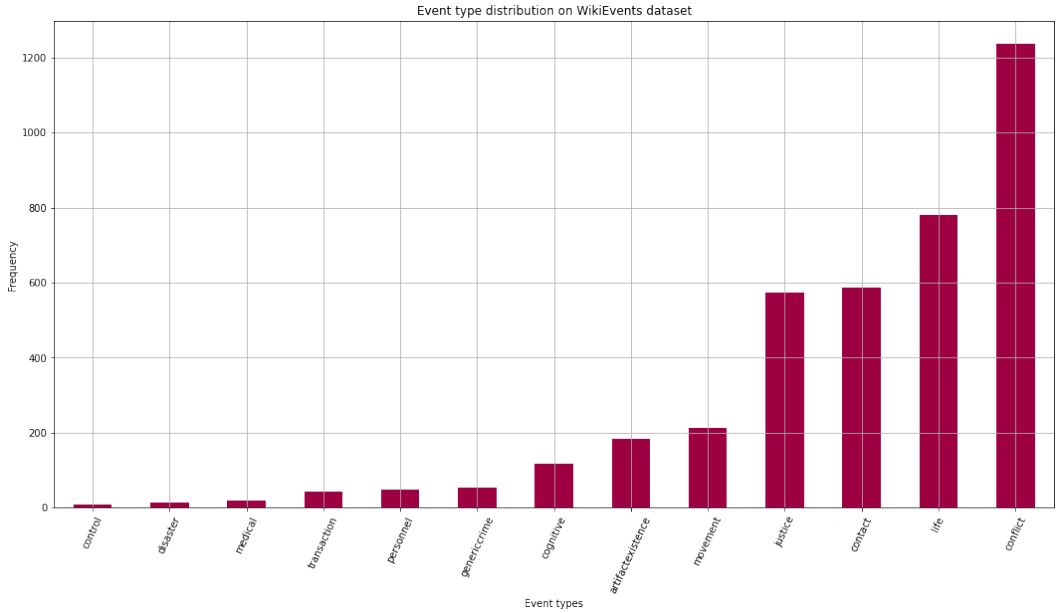

Figure 3.5: Distribution of Macro Event Types in WikiEvents

Event type	**Frequency**	**Example triggers**
conflict.attack	1188	explosion, shot, attack
contact.contact	530	met, said, been in touch
life.die	501	killed, died, shot
life.injure	273	injuring, wounded, maimed
movement.transportation	212	transferred, brought, arrived
justice.arrestjaildetain	176	arrested, capture, caught
artifactexistence.damage destroydisabledismantle	103	damaged, destruction, removed
justice.investigatecrime	102	analysis, discovered, investigation
justice.chargeindict	96	charged, accused, alleged
artifactexistence.manufacture assemble	82	construct, make, build

Table 3.3: Top 10 Event Types in WikiEvents, along with Example Triggers.

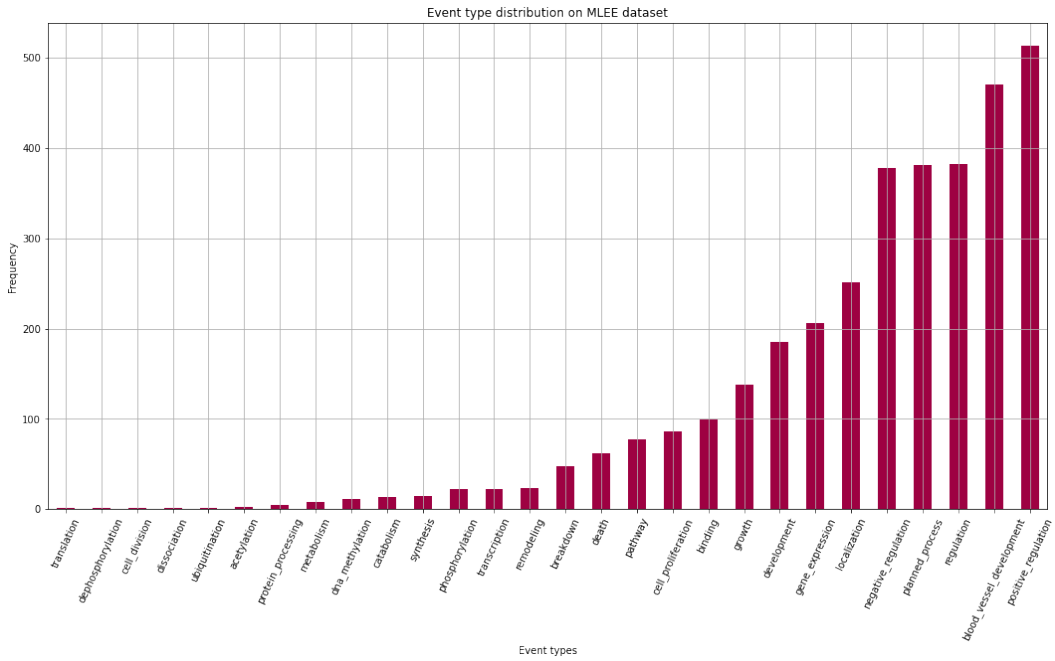

Figure 3.6: Distribution of Event Types in MLEE

extraction.

For our experiments, we use the authors' data split of 206, 20, and 20 documents for training, development, and testing, respectively. In the absence of existing baselines, we establish the benchmark performances on sentence-level ED on this dataset for future researchers.

3.4 MLEE

Our domain dataset is taken from the biomedical domain. The Multi-Level Event Extraction (MLEE) corpus (Pyysalo *et al.*, 2012) is taken from 262 PubMed abstracts centered around tissue-level and organ-level processes. Event trigger tokens are classified over 19 biomedical event types drawn from the GENIA ontology. These events fall under one of 4 high-level event types: molecular, anatomical, general, and planned. The prevailing best models for ED and EE on this dataset use domain-specific pretrained models or embeddings. We follow the original data split of 131,

Category	**Event type**	**Example triggers**
Anatomical	cell_proliferation	proliferation, proliferate, growing
	development	formation, progression, morphogenesis
	blood_vessel_development	angiogenic, angiogenesis
	death	death, apoptosis, survival
	breakdown	dysfunction, disrupting, detachment
	remodeling	remodeling, reconstituted
	growth	proliferation, growth, regrowth
Molecular	synthesis	production, formation, synthesized
	gene_expression	expression, expressed, formation
	transcription	expression, transcription, mRNA
	catabolism	disruption, degradation, depleted
	phosphorylation	phosphorylation
	dephosphorylation	dephosphorylation
General	localization	migration, metastasis, infiltrating
	binding	interactions, bind, aggregation
	regulation	altered, targeting, contribute
	positive_regulation	up-regulation, enhancement, triggered
	negative_regulation	inhibition, decrease, arrests
Planned	planned_process	treatment, therapy, administration

Table 3.4: Event Types in MLEE, along with Example Triggers.

Dataset	Docs			#triggers	#types
	Train	Dev	Test		
MLEE	131	44	87	8014	30
RAMS	3194	399	400	9124	38
MAVEN	2913	710	857	118732	168
WikiEvents	206	20	20	3951	49

Table 3.5: Dataset Statistics (Overview)

Dataset	Neg (%)	Events per row		Types per row		#zs
		Avg	Max	Avg	Max	
MLEE	18.22	2.867	16	2.369	9	3
RAMS	0	1.066	6	1.061	4	0
MAVEN	8.64	2.433	15	2.314	15	0
WikiEvents	54.11	1.671	7	1.429	6	1

Table 3.6: Dataset Statistics (Post-processed). Neg: Instances with No Event Occurrences. #zs: Number of Event Types in Test That Are Not Seen in Train.

44, and 87 abstracts for training, development, and testing, respectively, along with general domain models.

Chapter 4

EVENT DETECTION SUBTASKS AS SEQUENCE GENERATION

Existing methods, which are predominantly discriminative classifiers, cannot easily leverage pretrained semantic knowledge. These models fall short of correctly identifying complex events, such as event triggers associated with multiple event types, or event triggers that are multi-word phrases. Furthermore, they face difficulties in few-shot ED settings. Lastly, these models, once trained, lack cross-domain or cross-task adaptability. In view of these shortcomings, and the rapidly progress development of powerful pretrained sequence-to-sequence models, we formulate ED and its subtasks as sequence generation problems.

4.1 ED on Multi-Sentence Level

Previous works on the RAMS dataset (Veyseh *et al.*, 2021) conduct ED on this dataset on the sentence level. However, for multiple reasons, we do not conform to this paradigm. Firstly, the data , as provided by the original authors, in its native form, is not sentence-level. This is owing to the fact that RAMS is geared towards multi-sentence argument role linking and extraction. The original configuration allows us to test the efficacy of our model on a different setting - the multi-sentence level.

In addition, on converting the data to sentence level, we find that the dataset is highly imbalanced, with 77% of the sentences containing no events. This represents a severely class-imbalanced dataset, which is detrimental to model performance.

Furthermore, any models performing event argument extraction (EAE) on this dataset would work on these multi-sentence instances. Thus, any ED frameworks intended to be preliminary stages in successful end-to-end Event Extraction frame-

works must be likewise able to perform the ED task on the same input format, i.e. on the multi-sentence level.

Lastly, increasing the number of sentences in a single input instance increases the scope for misidentifying the salient event trigger. However, we hypothesize that in this case, the semantic information from the input as well as the presence of relevant arguments would provide necessary context for accurate identification and classification of the salient event trigger.

4.2 Task Decomposition

Event identification (EI) and Event classification (EC) are the subtasks that make up an ED problem (Ahn, 2006). A successful ED model must perform both these tasks accurately. Sequence labeling approaches do these inherently simultaneously, by identifying the correct triggers to label, and then assigning the correct event type as their labels. However, this paradigm does not allow for both of these individual subtasks to be carried out independently and in parallel. However, these subtasks can be framed as text generation tasks with similar generation formats as our proposed output format for ED. We use these constituent subtasks as additional independent tasks to augment our model.

4.3 Input Reformulation

Existing works that leverage sequence generation to perform ED or EE are recent and comparatively scarce. Research on the most efficient sequence representation of this problem is still ongoing. GDAP (Si *et al.*, 2021) uses a highly simplified output sequence format for both identification and classification; however, it performs ED sequentially, causing error propagation. TANL (Paolini *et al.*, 2021) and Text2Event (Lu *et al.*, 2021) use other generation-conducive patterns; however, the former gen-

erates a considerable number of task-irrelevant tokens, while the latter generates a complex output structure that can be difficult to parse at scale.

We propose a relatively simple sequence generation format. The labels, whether they are all event triggers, types of events, or a more comprehensive list of event triggers and their corresponding type annotations, are converted to a delimited string with a single-character delimiter. This creates a consistent pattern that can be learned by the model. In the absence of any events, we use the label NONE.

It is important to note that due to the presence of multi-class triggers, the number of unique event types and unique triggers for an instance might differ. This makes all tasks distinct and independent of one another, as opposed to Event Detection being simply a linear combination of predicted labels for EI and EC.

4.3.1 Event Identification, Classification

Each event for each of these tasks contains a single component, i.e. either the trigger, or the event type. Hence, we can represent the output of each instance as a delimited sequence of labels. For example, an instance with x unique triggers would have the following label representation for the EI task:

$$T_1 \mid T_2 \mid T_3 \ ... \ T_x$$

Where T_i is the i^{th} trigger word or phrase. Similarly, an instance with y unique event types occurring in it would have the following label representation for the EC task:

$$E_1 \mid E_2 \mid E_3 \ ... \ E_y$$

Where E_i is the i^{th} type of event occurring in the instance. We delimit all triggers and types with a pipe (|) symbol.

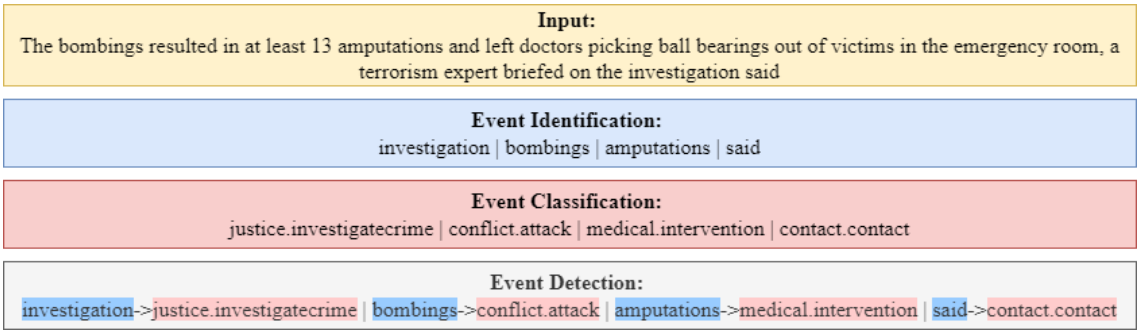

Figure 4.1: Sequence Generation Format for All Event Detection Subtasks

4.3.2 Event Detection

Each event for the event detection task is composed of 2 components: the event trigger, and its corresponding event type. Similar to our sequence formulation for EI and EC, we create a delimited sequence of events for an instance with x events:

$$V_1 \mid V_2 \mid V_3 \ldots V_x$$

Where V_i is the i^{th} unique event occurring in the instance.

The representation of V_i is a combination of the trigger for the particular event, T_i, and its corresponding type, E_i. We use -> as a delimiter between trigger and type, creating a unique format to enumerate the list of events. This allows us to represent multiple events in an instance as follows:

$$T_1 - > E_1 \mid T_2 - > E_2 \mid T_3 - > E_3 \ldots T_x - > E_x$$

For an example instance with our generative reformulations for all subtasks, refer to Figure 4.1.

Chapter 5

EVENT DETECTION AS MULTI-TASK GENERATION

5.1 Background

5.1.1 Multi-Task Learning

Multi-task Learning is a training paradigm in which a machine learning model is trained on multiple separate tasks in order to create a single model that learns shared representations and common ideas between those tasks (Caruana, 1997; Crawshaw, 2020). Across domains, models trained on multiple disparate tasks are better performing and more robust, due to shared learning. Multi-task learning has been leveraged to great effect in (Xie *et al.*, 2022; Lourie *et al.*, 2021), and in specific domains as well (Chen, 2019; Parmar *et al.*, 2022). This paradigm is also the basis of the generative T5 model, which is trained on a diverse set of language-based tasks converted into a text-to-text format (Raffel *et al.*, 2020). TANL (Paolini *et al.*, 2021) carried out multi-task learning experiments over a number of information retrieval tasks, including Event Detection and Argument Extraction. A version of multi-tasking over ED is implemented in GPTEDOT (Veyseh *et al.*, 2021), where generated EI samples are used to augment ED performance. Generation is restricted to EI, as it is easier to evaluate the quality of samples generated. Furthermore, it is not possible to train a discriminative classifier over EC in addition to these 2 subtasks. However, generative models can be used to extend this multi-tasking approach to all ED subtasks, without requiring generation of possibly noisy input samples.

5.1.2 Prompt Engineering

Using prompts and natural language instructions to augment input data and improve model learning is an active research area. The turking test (Efrat and Levy, 2020) was proposed as a method to evaluate how well machine learning models can learn from instructions, akin to humans, on a range of tasks. Later works have investigated how well PLMs gain a semantic understanding of prompts (Webson and Pavlick, 2022; Zhao *et al.*, 2021). The instruction learning paradigm has been investigated in detail (Hase and Bansal, 2021; Ye and Ren, 2021; Mishra *et al.*, 2022), especially in settings such as low-resource or zero-shot settings (Zhong *et al.*, 2021; Sanh *et al.*, 2022; Wei *et al.*, 2022)

Prompt-based models have been used for Event Detection and Event Extraction as well. More recently, GDAP (Si *et al.*, 2021) used predicted labels from earlier in the pipeline as prompts for later stages of trigger identification and argument extraction, while APEX (Wang *et al.*, 2022a), following the example of other works that use prototype event triggers (Wang and Cohen, 2009; Bronstein *et al.*, 2015; Lai and Nguyen, 2019; Lyu *et al.*, 2021; Liu *et al.*, 2020; Zhang *et al.*, 2021) from the dataset, used these triggers as part of tailored prompts for each event type in the schema, which include natural language information such as type definition, example triggers, and common sentence structure for the event types.

5.2 Methods

5.2.1 Multi-Tasking on ED Subtask-level

We leverage this training paradigm for ED, by treating the main task and its independent constituent subtasks as separate tasks of EI, EC, and ED. Our hypothesis is that by explicitly modeling the individual subtasks, we can augment the model's

performance on the combined ED task by creating a robust model that is trained to carry out both identification and classification, and can combine the results of these subtasks intelligently to accomplish ED. We refer to this subtask-based multi-task setting as MTL3 in this work, referring to the number of tasks the model is trained on. By using the original data for multi-tasking, we avoid introducing more noise into the training data. For rarer event types, modeling EC separately enables model to better identify sentences where those event types occur, which improves classification of identified triggers for these event types.

5.2.2 Prompting for Generative ED

Adding natural language prompts have shown promising results in improving performance in PLMs (Liu *et al.*, 2021). Prompt engineering is an active area of research across domains. Unlike previous prompt-based approaches (Wang *et al.*, 2022a), we do not create prompts solely within the scope of the ED task, such as event type-specific prompts. Instead, we use prompting to improve multi-tasking performance, by designing prompts that clearly indicate how to perform event identification, classification, or detection. We do this by employing natural language descriptions of the tasks and expected output. For example, we use the following text to prompt the model to perform ED generatively:

The text given as input discusses ongoing events. An event nugget is a word or phrase that most clearly expresses the event occurrence. Generate output in the format [event nugget− >event type] for all events in the text. If there are no events, generate NONE.

We experiment with a range of example types and configurations, including using

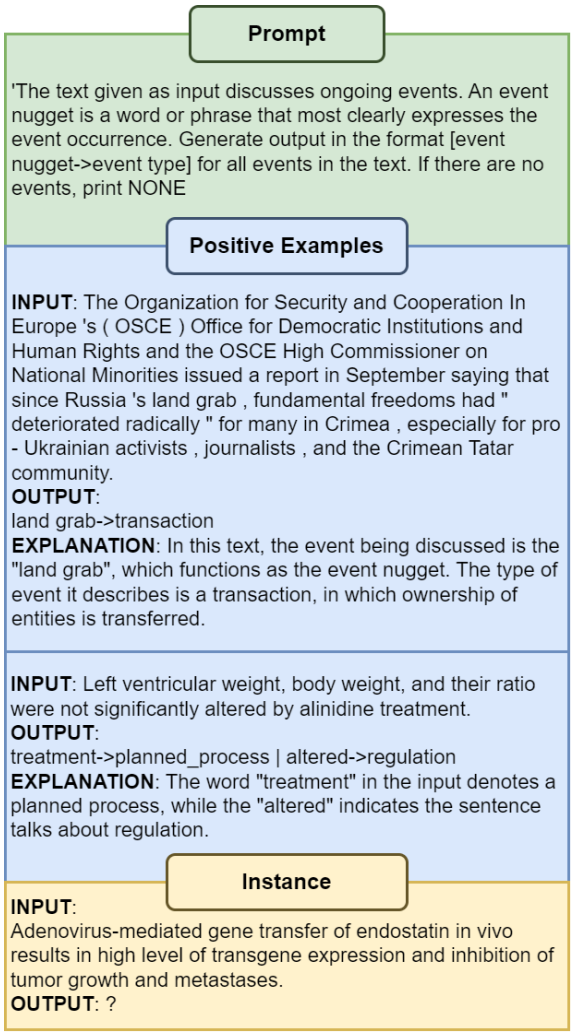

Figure 5.1: Example of an Input Instance for Prompted Generative Event Detection

only general examples, only domain instructions, and a combination of both. We use the following format for integrating examples into our instructional prompt

INPUT: *example input* </s> **OUTPUT**: *example output* </s> **EXPLANATION**: *natural language explanation*

For each subtask, we add 2 examples following the prompt. An example input is shown in Figure 5.1.

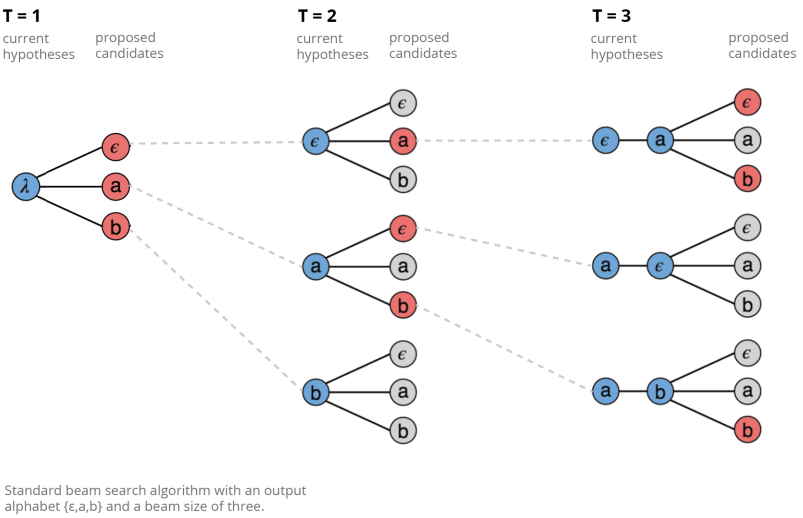

Figure 5.2: Beam Search Decoding with Beam Size 3 over a 3-token Vocabulary. Image from Commons (2021)

5.2.3 Beam Search Decoding

Beam search decoding is the most popular decoding strategy for NLP algorithms like Neural Machine Translation, Image/video captioning, Chatbots, sequence to sequence decoding etc. Beam search considers multiple best options based on beam size using conditional probability, which is better than the sub-optimal Greedy search. Greedy approaches consider the single best token at each step. We obtain only a single sequence, and despite the individual token-to-token transitions being high probability,the final sequence may be suboptimal as a whole. Beam search expands Greedy Search, considering the top k (where k is the beam size) next steps by conditional probability at each point. As we take more timesteps, beam search calculates total sequence probability starting from the best sequences so far. This allows for diversity in sequence length, and allows shorter high probability sequences over longer, less-likely sequences made up of high probability transitions.

5.3 Model Training

The generative approach requires the use of a text-to-text model that can generate free form text. We use the generative T5 model (Raffel *et al.*, 2020), a Transformer-based text-to-text model. The model uses an encoder and decoder stack similar to BERT-base (Devlin *et al.*, 2019). The model is trained using standard maximum likelihood. The model learns to minimize error in predicting the next word in a sequence at the current timestep. The error used is categorical cross-entropy loss, which measures the difference between predicted and actual probability distribution. For language models, the actual probability distribution at the timestep can be expressed as a one-hot encoded vector, where the element corresponding to the ground truth token is 1, and the other are 0. This results in cross-entropy loss being the negative log probability assigned to the next word in the training sequence. The final loss over a sequence is the aggregated loss over all timesteps.

T5 uses teacher forcing (Williams and Zipser, 1989) for network training. Teacher forcing refers to the practice of using predicted token at timestep t to calculate loss, but providing the ground truth sequence up to t tokens in order to predict the token at the next timestep. This allows us to calculate the loss accurately, while improving speed of convergence and model stability.

Optimization is done by using AdaFactor (Shazeer and Stern, 2018) for parameter updates.

5.4 Experimental Setup

For our experiments, we use the T5-base configuration, which has 220 million parameters. It is trained on a multi-task mixture of unsupervised and supervised tasks, such as denoising, sentiment analysis, natural language inference, sentence

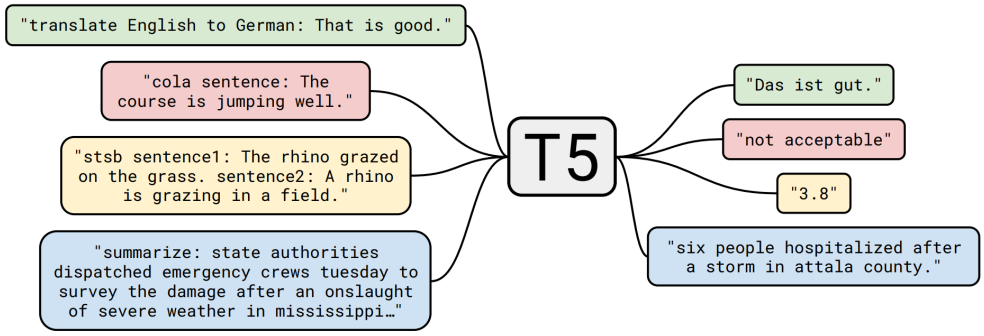

Figure 5.3: Overview of the T5 Model. Image from Raffel *et al.* (2020)

completion, and question answering, among others. We conduct all our experiments using GTX1080 and V100 NVIDIA GPUs. The trained model allows a maximum sequence length of 1024 tokens. All models are trained for 20 to 50 epochs, with a batch size of 1 due to memory constraints. For beam search decoding, we use 50 beams.

The scripts required to create generatively-formatted data, run various task configurations, and evaluate their token-level performance can be found here. [1].

5.5 Results and Discussion

Although we do not perform ED as word classification, in order to compare the efficacy of our method fairly with established baselines, we evaluate our predictions by converting them to token-level labels.

Across the board, we observe an increase of 3-4% in F-1 score over individual task performance on task decomposition and subsequent multi-tasking over the same dataset. This supports our main hypothesis on the efficacy of breaking a complex generation task into its constituent subtasks that can be used to support model learning on the primary task. We observe a further increase of 4% in model performance

[1] `https://github.com/ujjwalaananth/MTL3_EventDetection`

Dataset	Single-task		MTL3 (tags)		MTL3 (instr)	
	All	Pos	All	Pos	All	Pos
MLEE	71.07	72.20	74.57	75.82	**77.09**	**78.45**
RAMS	63.21	63.21	67.66	67.66	**69.53**	**69.53**
MAVEN	58.10	59.18	62.29	63.56	**62.40**	**63.66**
WikiEvents	54.31	58.47	56.77	61.35	**58.71**	**64.31**

Table 5.1: Results on All Datasets, with Greedy Decoding. Single-task: Event Detection Results. MTL3 (Tags): Training with EI and EC Tasks on the Same Dataset. MTL3 (Instr): Incorporating Natural Language Instructions with Examples. Pos Denotes Performance on Only Event Sentences.

Model	P	R	F1
DMBERT (Wang *et al.*, 2019)	62.6	44.0	51.7
GatedGCN (Lai *et al.*, 2020)	66.5	59.0	62.5
GPTEDOT (Veyseh *et al.*, 2021)	55.5	78.6	65.1
Our model	**71.6**	**71.0**	**71.3**

Table 5.2: Results on RAMS. All Previous Models Are Sentence-level BERT-based Models.

on 3 out of 4 datasets by engineering instructional prompts and incorporating them as part of the input. Finally, performing beam search decoding improves prediction performance further by up to 2%.

5.5.1 RAMS

Existing baselines perform ED on sentence-level. We compare our multi-sentence ED performance with DMBERT (Wang *et al.*, 2019), GatedGCN (Lai *et al.*, 2020), and GPTEDOT (Veyseh *et al.*, 2021). All these models are BERT-based. The state-of-the-art model, GPTEDOT, leverages the multi-task learning paradigm similarly, however, owing to the limits of classification-based representations, only uses EI, and

Model	P	R	F-1	WF-1
Single-task	60.0	49.6	54.3	52.1
Our model	**60.8**	**60.6**	**60.7**	**59.4**

Table 5.3: Results on WikiEvents. WF-1: Weighted F-1 %

requires generation of data to augment existing examples. Using only the native data as provided by the authors (Ebner *et al.*, 2020), we achieve 67.66% by using our MTL3 approach. Adding instructional prompts, we achieve 69.53% F-1 score, which surpasses GPTEDOT by 4.4%. Training for more epochs further increases model performance to 71.33%. Furthermore, the difference between precision and recall is drastically lower than the competing discriminative models, indicating that our model is less biased, and more robust.

Additionally, simply reformulating the problem and performing ED generatively on the multi-sentence level also achieves a competitive score of 64.75%. This illustrates that input reformulation along with incorporated document context alone can achieve competitive performance on ED.

5.5.2 WikiEvents

As there are no existing event detection baselines on this dataset, we use single-task ED sequence generation performance as a baseline to contextualize the benefits on our proposed prompted multi-task learning approach. We establish benchmark performance, to the best of our knowledge. Conforming with widespread convention, we evaluate ED on two-level labels, i.e. both predicted event type and event subtype must match the ground truth. On the sentence-level, performing ED generatively achieves 54.31% F-1 score. Leveraging EI and EC and training a single model over the 3 tasks increases the ED performance, and by adding instructional prompts to

Model	P	R	F-1	M-F1
SaliencyED (Liu *et al.*, 2022)	**64.9**	**69.4**	**67.1**	60.3
Our model	60.1	65.5	62.7	59.1

Table 5.4: Results on MAVEN. All Results Are on the Publicly-available Dev Split. M-F1: Macro F-1 %

this multi-task model, this score increases further by nearly 4.5%. Training for more epochs and using beam search decoding allows us to achieve a maximum F-1 score of 60.71%, with a precision and recall of 60.8% and 60.6% respectively, indicating a relatively balanced model. This is the model performance over the entire dataset, including on sentences with no event types, where false positives, i.e incorrect prediction of the existence of triggers, may occur. These sentences make up nearly half of the entire dataset. On evaluating solely over the sentences with at least one event, we observe that the best performance goes up to 65.67%.

5.5.3 MAVEN

While we use the original train split to train our models, the ground truth labels for the test split are unavailable to us. Furthermore, trigger candidates for the test split are provided by the authors, which can be used to constrain the model output space. We follow the example of SaliencyED (Liu *et al.*, 2022) and evaluate our model performance on the development split of the original MAVEN dataset. The state-of-the-art model, (Wang *et al.*, 2022a), leverages the prompting paradigm to perform word classification for ED. This model requires significant prompt engineering for all 168 event types in the schema. However, as its performance is evaluated on the unavailable test split, with access to possible trigger candidates, we do not report its performance as a comparable baseline. Using a generative format allows us to achieve 58.1%. On implementing our MTL3 architecture with instructional prompts,

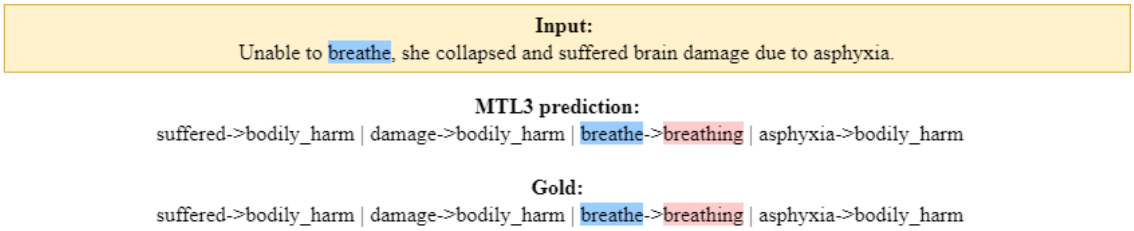

Figure 5.4: Low-resource ED on MAVEN: Breathing

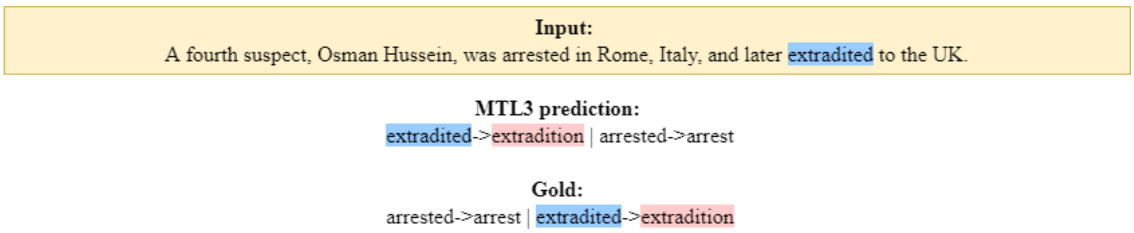

Figure 5.5: Low-resource ED on MAVEN: Extradition

performance increases by 4%. The highest experimental performance occurs when we implement beam search decoding, is 62.66%. While this is below the existing best performance on this dataset, the class imbalance in the MAVEN dataset may contribute to lower micro F-1 score. This is shown by the fact that our model has a comparable macro F-1 score, indicating relatively better performance on the many sparsely populated classes. Furthermore, our model shows significant advantages in performing ED on more complex event instances. An overview of these is discussed in the following sections.

Low-resource ED

As we discussed previously (Chapter 3), MAVEN suffers from the long-tail distribution issue in its dataset. Some event types, such as *Breathing* have very few annotated train examples that the model can learn from. In the face of this imbalance, we find that our model performs admirably in low-resource settings. For examples of successful low-resource event detection, see Figure 5.4. and 5.5. Both these event types, *Breathing* and *Extradition* have less than 20 annotated train instances in more than

8K training sentences (6 and 11 annotated triggers respectively). Despite this, we see the model accurately identifies all triggers in test data that are of these event types, achieving 100% testing precision on both, and 100% and 80% micro F-1 score respectively.

Multi-word Triggers

The multi-task sequence generation model shows significant advantages in performing ED on more complex event instances, specifically, in identifying and correctly classifying multi-class and multi-word event triggers, the former of which occur in 3.42% of the rows and 7.39% of all the triggers in MAVEN, significantly higher than the other datasets. For details, see Table 5.7. However, any mention of this is notably absent in previous works. SaliencyED (Liu *et al.*, 2022) explicitly states that it performs trigger extraction for trigger words, with no mention made of multi-class, or the far more frequent, multi-word triggers (Examples in Figures 5.11 and 5.12.). We explore this in greater detail in the Analysis section of this work.

5.5.4 MLEE

We distinguish between 2 sets of models for biomedical event detection. The former set of models include 2 SVM-based models (Pyysalo *et al.*, 2012; Zhou and Zhong, 2015) and a pipelined two-stage model (He *et al.*, 2018a), which conducts event identification and classification separately. These approaches are comparatively labour-intensive; they require the creation of handcrafted features for these tasks. The second set of models are neural network-based models. These include a CNN with embeddings encoding event type, POS labels and topic representation (Wang *et al.*, 2017), RNN with word and entity embeddings (V S S Patchigolla *et al.*, 2017), and LSTM-based models that integrate other biomedical datasets in order to perform

Model	**P**	**R**	**F-1**
SVM (Pyysalo *et al.*, 2012) *	70.8	81.7	75.8
SVM2 (Zhou and Zhong, 2015) *	72.2	82.3	76.9
Two-stage (He *et al.*, 2018a) *	79.2	80.3	79.8
EANNP (Nie *et al.*, 2015)	71.0	**84.6**	77.2
CNN (Wang *et al.*, 2017)	80.6	74.2	77.8
GRU (V S S Patchigolla *et al.*, 2017)	79.8	78.4	79.1
LSTM (He *et al.*, 2018b)	81.8	77.8	79.7
LSTM + CRF (Chen, 2019) (w/o TL)	81.6	74.3	77.8
LSTM + CRF (Chen, 2019) (w/ TL)	81.8	77.7	79.7
BiLSTM + Att (He *et al.*, 2022)	**82.0**	78.0	**79.9**
Our model	75.9	80.4	78.1

Table 5.5: Results on MLEE dataset. * Indicates Models Which Require Engineering Hand-crafted Features. All Neural-network Based Models in This Table Use Dependency-based Embeddings Specific to Biomedical Texts. w/TL: Results When 4 Biomedical Datasets Are Used for Transfer Learning.

transfer learning Chen (2019).

All existing baselines use pretrained embeddings and other language resources specifically for biomedical texts such as the resources published by (Pyysalo *et al.*, 2013). For example, LSTM (He *et al.*, 2018b) and the state-of-the-art BiLSTM (He *et al.*, 2022), like the majority of existing models, employs Word2vecf (Levy and Goldberg, 2014) to train dependency-based word embeddings. These embeddings are trained on Pubmed abstracts that are parsed using the Gdep parser: a dependency parse tool built for use on biomedical texts. Likewise, EANNP (Nie *et al.*, 2015) undertakes a similar approach to pretraining embeddings, but uses Medline abstracts

instead.

However, in keeping with our experimental settings on the general domain datasets, our approach uses embeddings pretrained on the general domain. This puts us at a disadvantage, given the lack of access to informative, domain-specific embeddings.

The lower precision of our model indicates that domain knowledge is significant to competitive performance on domain-specific datasets, and pretrained general domain knowledge is not sufficient to surpass state-of-the-art benchmarks achieved by domain-specific models on these datasets.

Nevertheless, even when at a disadvantage, our general domain model equals or surpasses (Pyysalo *et al.*, 2012; Zhou and Zhong, 2015; Nie *et al.*, 2015; Wang *et al.*, 2017; Chen, 2019) methods that require labour-intensive handcrafted features, or sophisticated architectures with pretrained embeddings, along with extensive feature extraction and engineering. We observe that out model also has higher recall than the majority of the neural network-based approaches. This strongly suggests that the task decomposition and prompted multi-tasking approach is promising, and by tailoring it to specific domains by integrating domain-specific knowledge, it may be possible to achieve significantly better performances, as well as obtain a model that is robust and able to handle complex event occurrences such as multi-word, multi-class, and overlapped events.

5.6 Additional Experiments

5.6.1 Alternative Model Architectures

For experimental purposes, we use generative architectures other than T5-base to test performance on certain datasets.

T5-large is a variant of the T5 model, but with 770 million parameters, thus

requiring more computational power to train but capable of learning more complex dependencies. On the biomedical dataset MLEE, we observe it achieves a recall of 81% at the same settings as T5-base, which exceeds the best model recall. However, the precision drops, causing overall F-1 score to be slightly below the best T5-base model, indicating that, for domain-specific data, the shortcoming is not in model size, but in the knowledge base it is trained on.

Another alternative to vanilla T5 models is using models trained specifically to follow in-context instructions, similar to the instructional prompts used in MTL3. One of these is T*k*-INSTRUCT (Wang *et al.*, 2022b), which has been shown to outperform other instruction-following models such as Instruct-GPT. However, T5-base outperforms this model on all datasets when task-specific, dataset-agnostic prompts are used.

When using prompts tailored to specific datasets, the overall performance suffers. We observe this on MAVEN, where micro F-1 score is comparable to the best recorded model, macro F-1 score drops by nearly 10% from the best recorded macro F-1 score with task-specific prompts and the T5-base configuration. Dataset-specific prompts are discussed further in Section 5.8.3.

5.6.2 Augmenting Training Data with Output Sequence Permutations

We attempt to make the model robust to the effects of possible perturbations in the training sequences, by augmenting the training data with repeated input instances but with the sequence of events in the output sequence changed. However, this causes a decline in model performance. For the most part, event order does not seem to affect model efficacy. While ground truth sequences generally follow order of occurrence in the input text, there is no particular pattern in the order of events in generated output sequences.

5.7 Alternative Evaluation Metric for Sequence Generation-based ED

The majority of existing works treat this as a multi-class word classification problem, and all baselines, including generative methods, evaluate model results consistent with word classification metrics popularly used for NER tasks (Nakayama, 2018). However we see many multi-label trigger words. Secondly, in real world data the same trigger may function as a trigger for multiple event types, with a different set of arguments corresponding to its role as each event type it triggers. This is more apparent in multi-sentence level inputs. The presence of these complexities makes existing baselines misleading.

As an alternative evaluation scheme, we treat sequence generation based ED as sentence or multi-sentence level multi-label classification, where multi-word triggers are considered distinct labels. For this problem, we treat NONE as a possible label for a given input text.

We calculate the metrics of precision, recall, and F-1 score using conventional formulae:

$$\text{Precision (P)} = \frac{TP}{TP+FP}$$

$$\text{Recall (R)} = \frac{TP}{TP+FN}$$

$$\text{F-1 score} = 2 \times \frac{P \times R}{P+R}$$

where a prediction is counted as true positive only if both trigger span and predicted event type (including subtype) match gold annotations.

In addition to more accurate performance metrics over multi-class triggers, this

Dataset		P	R	F-1
MLEE	All	73.05	76.74	74.85
	Pos	73.97	77.49	75.69
RAMS	All	72.61	71.62	72.11
MAVEN	All	59.01	63.82	61.32
	Pos	60.5	64.67	62.51
WikiEvents	All	61.29	63.33	62.29
	Pos	56.73	57.61	57.17

Table 5.6: Results Using Alternative Evaluation Scheme on All Datasets. All: Multi-label Metrics on All Rows, with None as a Separate Class. Pos: Multi-label Metrics on Instances with at Least One Event. Multi-class and Multi-word Triggers Count as Distinct Labels, with Exact Match Counted as True Positive.

provides a stricter metric to evaluate multi-word triggers, where partial predictions do not contribute to model performance. Using this metric also allows us to evaluate the discriminative performance of an ED model, i.e. the accuracy with which it can identify whether an input text contains an event or not. We implement this evaluation metric based on publicly-available code from another sequence generation model for ED (Si *et al.*, 2021). The results on entire test data as well as event and non-event sentences obtained using this metric are reported in Table 5.6.

5.8 Analysis

5.8.1 Possible Generative Reformulations

We experiment with multiple generative reformulation of the Event Detection task. Apart from the aforementioned trigger-first representation,

$$T_1 - > E_1 \mid T_2 - > E_2 \mid T_3 - > E_3 \ ... \ T_x - > E_x$$

we also explore the type-first formulation for individual events.

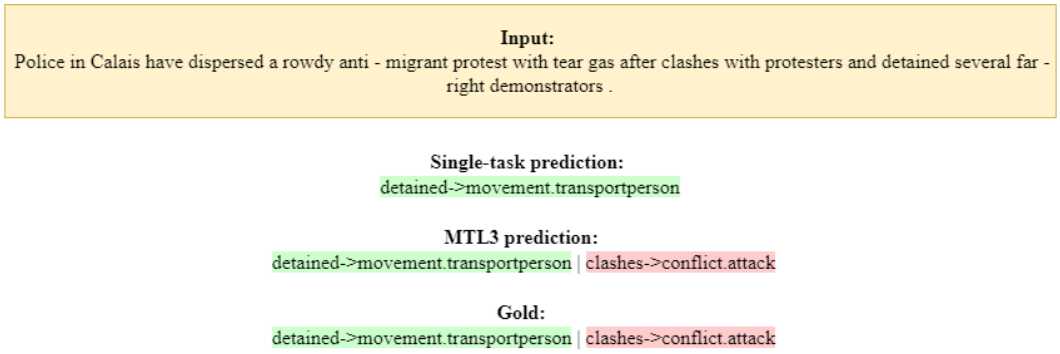

Figure 5.6: Example 1 of MTL3 Improving Prediction over Single-task Setting.

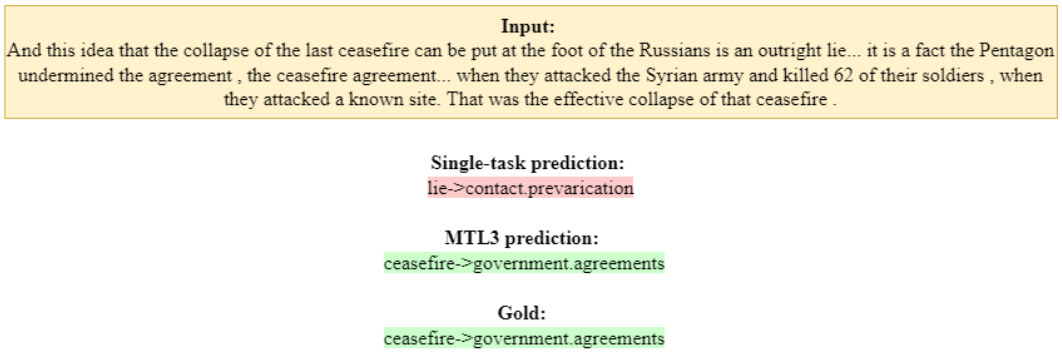

Figure 5.7: Example 2 of MTL3 Improving Prediction over Single-task Setting.

$$E_1 -> T_1 \mid E_2 -> T_2 \mid E_3 -> T_3 \ \ldots \ E_x -> T_x$$

Across datasets, we find that the trigger-first reformulation outperforms type-first reformulation, on both single-task, and multi-task settings.

5.8.2 Efficacy of Subtask-level Multi-tasking

Our hypothesis, that including Event Identification and Event Classification would improve Event Detection performance, is supported by results on all datasets. The inclusion of Event Identification trains the model to be able to identify salient event triggers, which aids in identifying triggers that were not recognized in the single task setting. This task also trains the model to recognize multi-word triggers. Event Classification helps identify the types of events, which helps in more accurate trigger classification, and also in identifying event types that were not recognized in the single

task setting. This extends to event types linked to the same trigger, i.e. multi-class triggers.

For examples of improved ED performance due to multi-tasking, see Figures5.6 and 5.7. More examples and detailed analyses on multi-word and multi-class triggers are documented in the following sections. Single-task and multi-task model metrics over all datasets are documented in Table 5.1.

5.8.3 Efficacy of Instructional Prompts

The combination of the task decomposition and subsequent multi-tasking, and usage of instructional prompts, achieve competitive performances on all the datasets. In this section, we break down the different aspects of our engineered prompts to the increase in performance and attempt to quantify the contribution of individual components in the overall improvement in ED performance.

From Table 5.1, we see the metrics for the MTL3 (tags) model configuration, which does not use instructional prompts at all as part of the input. The tags merely identify the dataset and task (EI, EC, or ED), and do not add any other information about how to perform the task. Nevertheless, this model already improves performance over the single-task by at least 3% for all datasets. This can be attributed to the success of the subtask-level multi-tasking paradigm, with the improved performance due to the knowledge obtained by training the model over event identification and classification in addition to the primary task of event detection.

Our instruction prompts can be divided into the task prompt, and 2 following examples (Figure 5.1). Initially, we experiment with including only the prompt, i.e. a natural language instruction that explains the task, and in the case of event detection, also enumerates the format in which to generate annotated triggers. We discover that including these prompts improves performance over tag-appended input

by up to 1.5%.

In terms of augmenting these prompts with examples to help the model learn the task with illustrative examples, we initially include a single example from the general domain. Not only does the inclusion of even one example improve performance significantly, but surprisingly, it does so even on MLEE, the domain dataset; indicating that the example, even if from a different domain, nevertheless provides some transferable knowledge.

To explore this idea further, we add an example instance from the biomedical domain to the instructional prompt. As anticipated, with the addition of a domain-relevant example, the performance on MLEE improves further to give us the best performance of 77.43% before beam search decoding and 78.09% after beam search decoding.

Interestingly, performance on general domain datasets also increases with the addition of the biomedical example. This indicates that diversity in instructional prompts is the key to successfully leveraging the instructional learning paradigm to best effect.

As an additional experiment, we replace the biomedical example with an instance from a cybersecurity dataset, CysecED (Man Duc Trong *et al.*, 2020), in addition to the general domain task example. We observe that the performance on general datasets is still higher than if we use a single general domain example, supporting our hypothesis on the efficacy of diverse examples despite not being relevant to the target dataset domain.

Using domain specific instructions improves performance, but using multiple cross-domain instructions improves scope of learning and consistently improves F-1 score, creating more robust models.

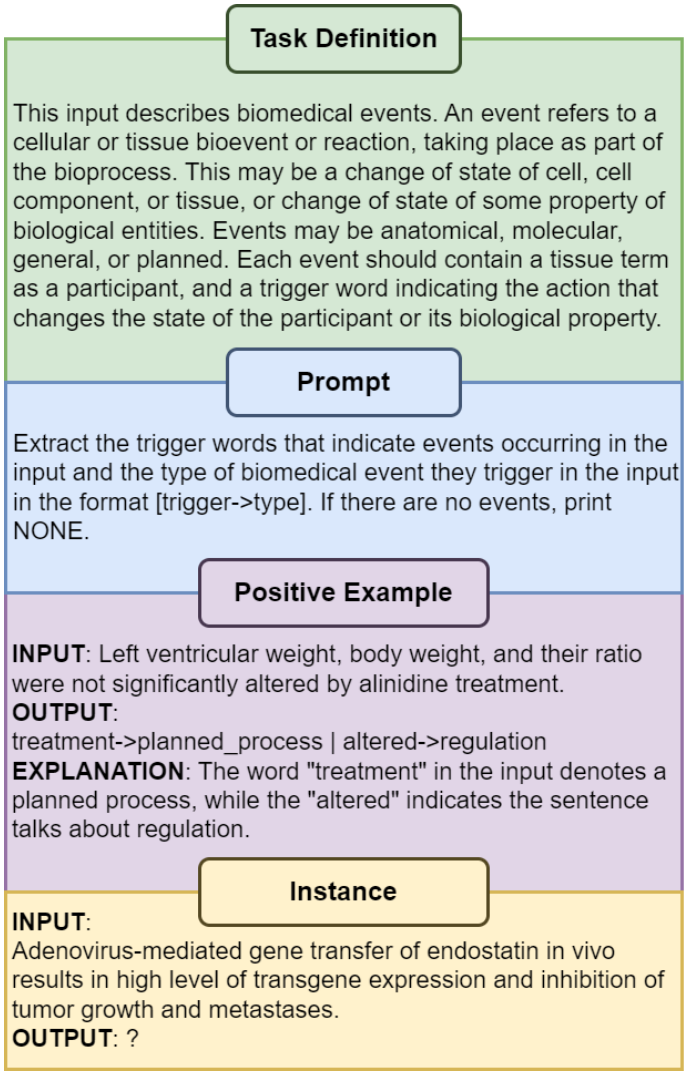

Figure 5.8: Example of an Input Instance for Dataset-specific (MLEE) Prompted Generative Event Detection

Dataset-specific prompts

RAMS and WikiEvents follow the Consortium (2005) guidelines on annotation, while MAVEN uses a tool to automatically extract trigger and type candidates, which are provided to human annotators. MLEE, which requires domain expertise, provides detailed guidelines on the definition of an event in the context of the biomedical event detection task. We can use these provided annotator guidelines to create instructional prompts that are more tailored to the datasets and thus convey the task objective

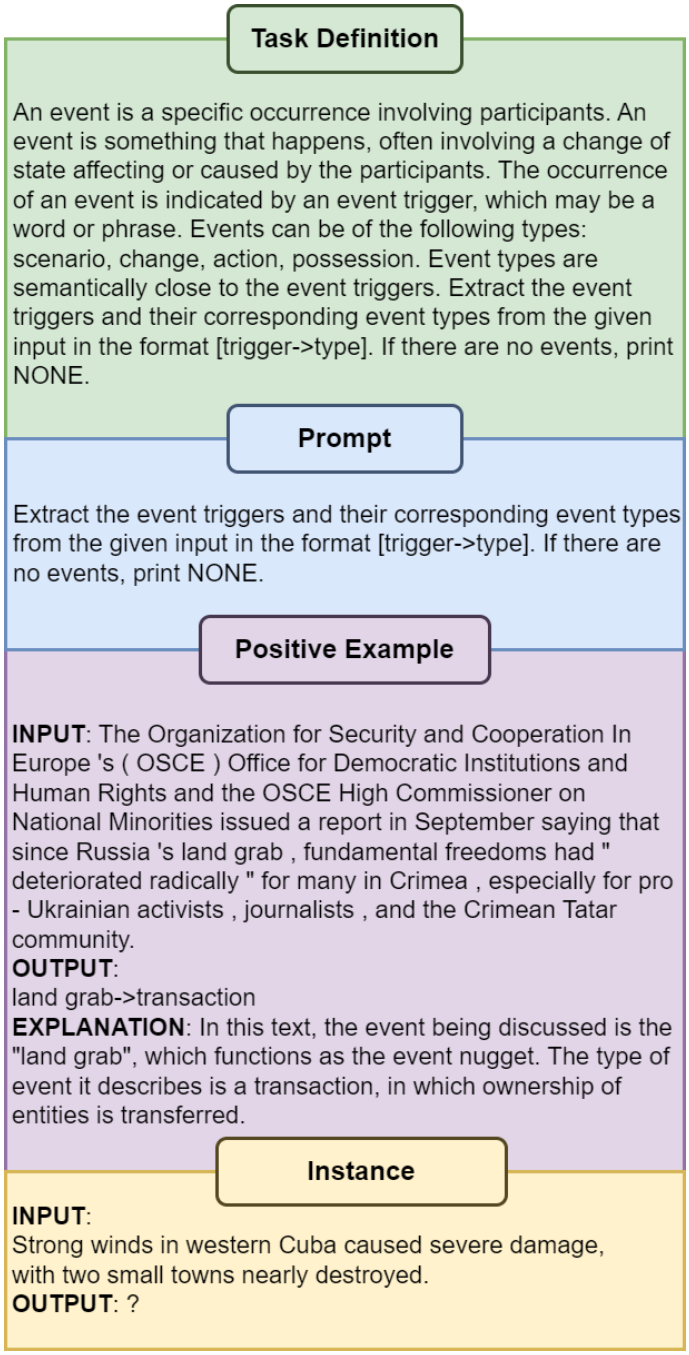

Figure 5.9: Example of an Input Instance for Dataset-specific (MAVEN) Prompted Generative Event Detection

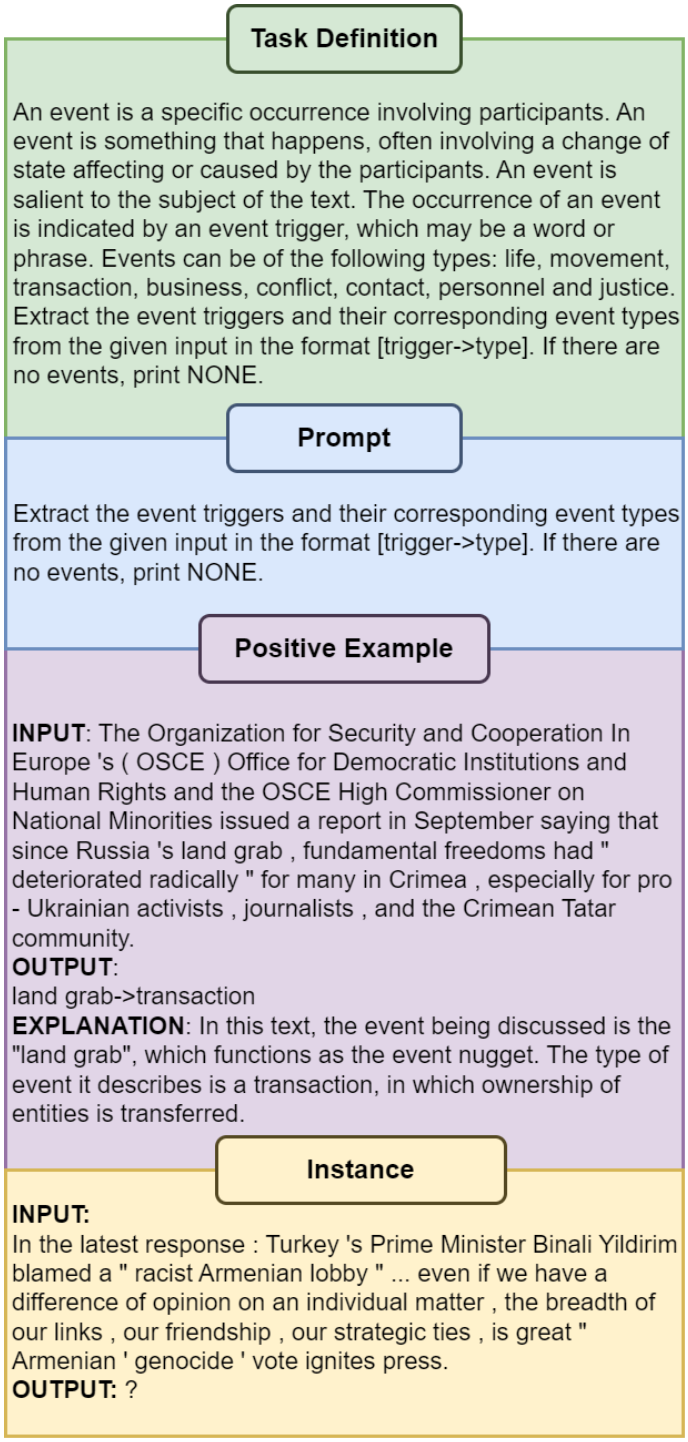

Figure 5.10: Example of an Input Instance for Dataset-specific (RAMS, WikiEvents) Prompted Generative Event Detection

Dataset	Multi-word triggers		Multi-class triggers	
	%instances	%rows	%instances	%rows
MLEE	0	0	0	0
RAMS	3.38	2.89	**3.97**	**3.72**
MAVEN	**3.42**	**7.39**	0.06	0.13
WikiEvents	2.86	2.18	0	0

Table 5.7: Statistics on Multi-word and Multi-class Triggers in All Datasets. %instances: The % of Total Triggers That Are Multi-word or Multi-class. %rows: The % of All Rows That Contain at Least 1 Multi-word or Multi-class Triggers.

more clearly to the model. For example, for MLEE, the dataset-specific instructional prompt for event detection as shown in Figure 5.8.

We observe that for general domain datasets, dataset-specific instructions cause decreased performance, while using task-specific but dataset-agnostic instructions, i.e. the same task-specific generic instructional prompt across datasets, with diverse domain examples, results in better event detection performance.

For MLEE, the tailored instruction contains the task definition which provides important domain context for event definition. In this case, including the general domain example causes a sharp drop in performance due to its misleading nature with respect to the domain-specific prompt. With only a biomedical example, the model performance is competitive (77.8% F-1 score), but still does not exceed performance achieved by using dataset-agnostic prompts with diverse domain examples (78.1% F-1 score).

5.8.4 Performance on Multi-word Triggers

Treating ED on real world datasets as word classification does not accurately measure performance on a significant portion of triggers, which are multi-word phrases.

Dataset	#mwt		EM acc %
	Train	Test	
MAVEN	2442	633	90.84
RAMS	228	20	88.89
WikiEvents	127	18	44.44

Table 5.8: Results on Multi-word Triggers. #mwt: Number of Multi-word Triggers in Testing Data. EM Acc %: Exact Match Accuracy, I.E. Percentage of Multi-word Triggers in Test Data Predicted Accurately by Our Model.

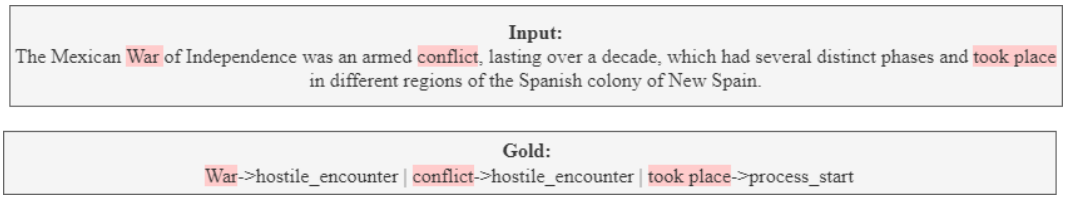

Figure 5.11: Example of an Event with Multi-word Trigger (2 words)

These occur in 7.39% of all sentences and make up 3.42% of all the triggers in MAVEN, and make up 3.38% of all triggers in RAMS (Table 5.7). Treating the classification of multi-word event triggers as word classification can create a misleading estimate, as many triggers only correspond semantically with the event type if the entire phrase is annotated.

For example, for the trigger phrase "took place" in 5.11, labeling only either "took" or "place" would be incorrect, as the individual words are semantically distinct from the meaning of the whole phrase, and would individually denote different event types. Trigger phrases can be up to 4 words long, as in Figure 5.12.

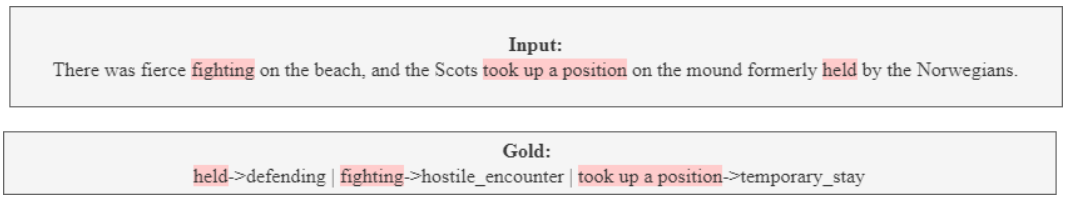

Figure 5.12: Example of an Event with Multi-word Trigger (4 words)

In order to evaluate our model's performance on multi-word trigger phrases, we calculate exact match accuracy for all multi-word triggers. This gives us the percentage of multi-word triggers that are correctly predicted out of all occurrences of multi-word triggers.

Out of all our datasets, the highest number of multi-word triggers is found in MAVEN. Owing to the large number of instances and low rate of negative examples, we have access to a large number of multi-word event triggers. Our model achieves an exact match accuracy of nearly 91%, indicating that our model can accurately extract exact spans for triggers that span from 2 to 4 words. Similarly, our model achieves nearly 89% on multi-word triggers in the RAMS dataset, which has fewer multi-word triggers, which make up a similar proportion of the dataset to MAVEN.

The WikiEvents dataset has fewer multi-word triggers. The exact match accuracy achieved by our model is lower; however, the partially predicted triggers are often semantically similar to the gold annotations. For example, our model extracts "bombing", "assault", and "in touch" in place of "suicide bombing", "the assault", and "been in touch" respectively. For trigger phrases such as "took place" and "set off", where partial predictions are semantically unequal to complete predictions, our model performs respectably. For results, refer to Table 5.8.

5.8.5 Performance on Multi-class Triggers

In real-world event data, we find that the same trigger can function as trigger of different type in context. For example, for **purchasing** in Figure 5.13, this triggers denotes two distinct types of transaction events. This distinction is vital; as, for each type of event it triggers, it has a different set of arguments associated with it in that aspect of that event. $transferownership$ is an event type with roles such as previous and current owner, while $transfermoney$ requires the *amount* event

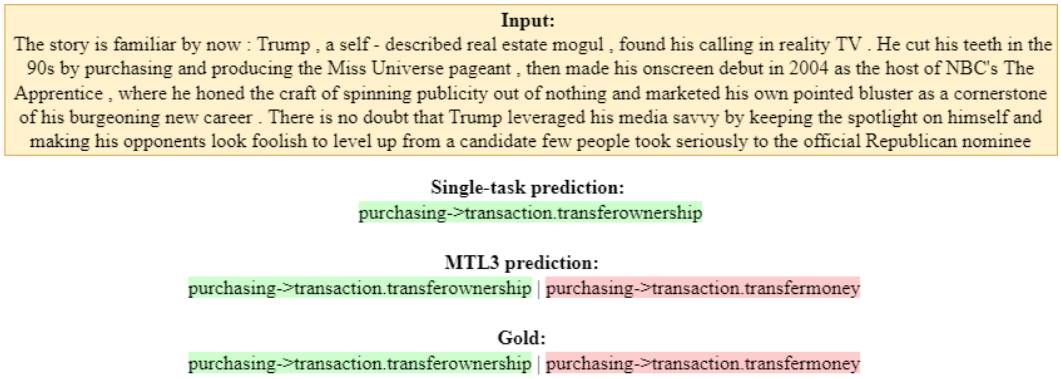

Figure 5.13: Example 1 of MTL3 Improving Prediction on Multi-class Triggers.

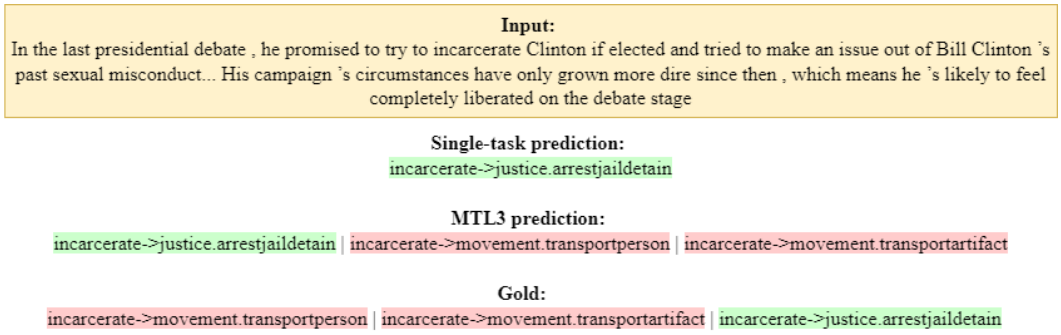

Figure 5.14: Example 2 of MTL3 Improving Prediction on Multi-class Triggers.

information. Another example would be the trigger **murder**, which triggers the *Crime* and *Death*. For complete event understanding, we would need the arguments associated with *Crime*, such as perpetrator and location, as well as the arguments associated with *Death*, such as time. For accurate event detection, it is vital to capture all senses of a particular trigger.

This is difficult to do with existing word classification methods, which perform and evaluate ED as multi-class classification, not multi-label classification, due to the lack of multi-class triggers in popular benchmark datasets. From Table 5.7, however, we can see that a significant of RAMS triggers embody more than one type of event.

Our generative technique offers more flexibility in predicting and evaluating multi-class triggers. Furthermore, multi-tasking over subtasks, specifically, training the model to be able to extract all possible event types in an instance, improves perfor-

mance on multi-class triggers, as evidenced by Figures 5.13 and 5.14.

In order to evaluate our model's performance on multi-class triggers, we evaluate the accuracy of prediction over multi-class triggers, i.e. we estimate an accuracy of 50% over a particular event trigger if we predict one of two event types that it triggers in that input instance.

Due to the extremely low number of multi-class triggers in MAVEN (less than 10), the results are inconclusive. On RAMS, we find the model accuracy on average is close to 61%. This indicates, that for event triggers with 2 to 3 event types, our model can capture most of the senses the trigger functions in.

5.8.6 Effect of Negative Examples

From the dataset statistics in 3.6, we see that while MAVEN, RAMS, and MLEE have less than 20% of their dataset instances as non-event, or negative instances, in contrast, the WikiEvents dataset has close to 54% instances that have no major events. Thus, negative examples make up more than half of the dataset. We observe that this composition has a detrimental effect on model performance, as the large proportion of negative examples leaves much fewer examples to train the task on. This detracts from the model's ability to discern relevant events and their types from a set of possible events, and instead, places more importance on the binary classification task of recognizing whether a sentence contains an event or not.

We analyse the effect of negative examples further using F-1 score (Table 5.1). While the results on positive examples are consistently higher than on the entire dataset, the difference between both metrics is stark in the case of WikiEvents. This consistent trend of higher Pos scores indicates that our model is better at identifying events accurately, than identifying without context whether standalone sentences contain relevant events or not. Furthermore, despite the fact that MAVEN has 168

event types and WikiEvents has only 49 (for reference see Table 3.5), the overall performance on MAVEN (62.67%) is higher than on WikiEvents (63.85%). This indicates that rather than the complexity of the ED task, the distribution of event and non-event sentences may hamper the model's ability to perform the task.

We observe a sharp increase in performance (60.71% to 65.67%) over WikiEvents, which is significantly higher than what we observe on other datasets. We attribute this to the much higher share of negative examples in this dataset. From further analysis we find that training on only positive examples improves the ED performance on event sentences by nearly 5%. The performance drops over non-event sentences as the model may predict event occurrence based on salient events in the sentence, that are important in the context of the sentence alone but are divorced from the subject of the document, and therefore annotated as non-events. We explore this further in our discussion of the need for multi-sentence context, which may be a way to counter the negative impact of a high proportion of non-event sentences on our ED model.

5.9 Chapter Summary

In this chapter we explored our novel subtask-level multi-tasking model that performs ED and both its subtasks. We introduce the instructional prompts we use to improve model understanding. Next, we explore the results of this training paradigm and subsequent beam search decoding on all target datasets. Finally, we conduct a fine-grained analysis of the model predictions in different experimental settings and conduct case studies on more complex event instances.

Chapter 6

CONCLUSION

This chapter presents a summary of this thesis work, and enumerates both, the limitations of this current work, and future research directions in the domain of Event Detection and Event Extraction.

6.1 Summary

This work presents a generative reformulation of the Event Detection information retrieval task, along with a novel multi-tasking method that leverages both its constituent subtasks. It compares the results of this model on 4 separate datasets, differing in scope, scale, and domain, against existing state-of-the-art information retrieval sentences. The model significantly outperforms existing methods on one dataset, sets a benchmark on another, and performs comparably on the remaining, including on a domain dataset without domain-specific knowledge. This work also conducts a fine-grained analysis of model performance over different settings and case studies.

6.2 Limitations and Future Scope

1. **Limited Task Scope**: Our work is limited to demonstrating a prompted and sequence generation-based model on a single task, Event Detection. However, this approach is flexible and can easily be adapted to other information retrieval tasks, as demonstrated by Paolini *et al.* (2021). The generative prompted approach that we demonstrate in this work can be extended to the subsequent task of Argument Extraction, as arguments and argument roles can be reformulated

generatively in a similar format to our proposed format for event detection. This would aid in the creation of a generative, instructional prompt-based framework for end-to-end Event Extraction. We leave this to be explored in a future work.

2. **Relative difficulty in distinguishing non-event sentences**: As we observe in the WikiEvents dataset and the sentence-level configuratin of the RAMS dataset, a major hurdle for the model is distinguishing sentences with events and those whose main action is auxiliary to the event, i.e. negative examples. A possible remedy would be a binary classification system to distinguish between event and non-event sentences could improve performance over datasets with a significant proportion of negative examples. This would also improve the performance of the Event Detection module, as it could learn mappings specific to only extracting the salient events from given input.

3. **Lack of Syntactic Information**: We conduct limited preprocessing on the actual instance text. In the manner of early feature-based models, inclusion of syntactic information such as entity types, or Part-of-Speech tagging could help identify trigger candidates better.

4. **Improved Decoding Scheme**: Another possible research direction could be an improved decoding scheme. We find that including predictions from top 2 sequences generated during beam search increases recall significantly but reduces precision. Further experimentation on score-based thresholding could yield a better decoding scheme that would greatly improve recall without negatively impacting precision significantly. .

5. **Improvement in prompt quality**: Another branch of relevant future research could be further prompt engineering and an analysis of the number and scope, as

well as level of detail in examples required to achieve the best possible prompted performance.

6. **Lack of Domain Knowledge**: While the model architecture is adaptable across domains, the model performance suffers from lack of any domain knowledge. For specific domains, having background knowledge could help identify domain-relevant events, as well as improve event type classification, helping our model could achieve more competitive performances. Integrating domain knowledge could be done by utilizing domain-specific tools, such as pretrained domain embeddings, external language-based resources, or pretrained large language models for that domain. We leave domain-specific modeling and subtask-level multitasking to future researchers aiming to leverage this technique in their respective domains.

7. **Need for Multi-sentence Context**: Consider the following examples from the WikiEvents dataset:

   Example 1: The whole building has **collapsed**.
   Example 2: He chose **destruction**.

   For Example 1, the model extracts the token in bold as a relevant event trigger, and assigns it as event of the type of $artifactexistence$ with the subtype $damagedestroydisabledismantle$. For Example 2, the model annotates the sentence as NONE, indicating no salient event was found.

   For Example 1, from the given sentence as standalone examples, it is a reasonable assumption that this trigger is a major event and therefore should be annotated. However, we find that Example 1 is from a document that is mainly

concerned with events of the type *conflict.attack*, with **bombing** and **explosion** being the annotated event triggers. Hence, **collapsed** merely indicates an auxiliary event, not the main event, and thus, the model should have predicted the events as NONE.

In contrast, Example 2 is from a document where the aforementioned destruction is the focus of the text. The context provided by its following sentences makes it clear that destruction is, in fact, the salient event in this case. This is indicated by the gold annotation, denoting **destruction** to be a trigger of event type of *artifactexistence* with the subtype *damagedestroydisabledismantle*

This shows us that sentences tagged NONE may nevertheless have salient events predicted by model, but are tagged NONE because in the original multi-sentence context, the salient event in the sentence is not relevant/less important than events that are the subject of the passage. It is difficult for our model to judge the saliency of an event without the semantic context of its document, and the presence of other events in its vicinity to compare its importance with. This is why it is vital to include multi-sentence or document-level context, as sentence-level information can be misleading in the broader context. As we demonstrate on RAMS, multi-sentence or document-level context can be invaluable to better event detection, by providing necessary context for event type classification. However, multi-sentence inputs increase the output space due to more possibilities. Many existing word classification and sequence labeling models have experimented with including relevant arguments or document information for event detection and extraction. A similar encoding of document information for sentence-level generation is a possible avenue of research.

Ahn, D., "The stages of event extraction", in "Proceedings of the Workshop on Annotating and Reasoning about Time and Events", pp. 1–8 (Association for Computational Linguistics, Sydney, Australia, 2006), URL https://aclanthology.org/W06-0901.

Baker, C. F., C. J. Fillmore and J. B. Lowe, "The Berkeley FrameNet project", in "36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1", pp. 86–90 (Association for Computational Linguistics, Montreal, Quebec, Canada, 1998), URL https://aclanthology.org/P98-1013.

Bollacker, K., C. Evans, P. Paritosh, T. Sturge and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge", in "Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data", SIGMOD '08, p. 1247–1250 (Association for Computing Machinery, New York, NY, USA, 2008), URL https://doi.org/10.1145/1376616.1376746.

Boros, E., J. G. Moreno and A. Doucet, "Event detection as question answering with entity information", CoRR **abs/2104.06969**, URL https://arxiv.org/abs/2104.06969 (2021).

Bronstein, O., I. Dagan, Q. Li, H. Ji and A. Frank, "Seed-based event trigger labeling: How far can event descriptions get us?", in "Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)", pp. 372–376 (Association for Computational Linguistics, Beijing, China, 2015), URL https://aclanthology.org/P15-2061.

Caruana, R., "Multitask learning", Mach. Learn. **28**, 1, 41–75, URL https://doi.org/10.1023/A:1007379606734 (1997).

Chen, C. and V. Ng, "Joint modeling for chinese event extraction with rich linguistic features", pp. 529–544 (2012).

Chen, Y., "Multiple-level biomedical event trigger recognition with transfer learning", BMC Bioinformatics **20** (2019).

Chen, Y., S. Liu, X. Zhang, K. Liu and J. Zhao, "Automatically labeled data generation for large scale event extraction", in "Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)", pp. 409–419 (Association for Computational Linguistics, Vancouver, Canada, 2017), URL https://aclanthology.org/P17-1038.

Chen, Y., L. Xu, K. Liu, D. Zeng and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks", in "Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)", pp.

167–176 (Association for Computational Linguistics, Beijing, China, 2015), URL https://aclanthology.org/P15-1017.

Cheng, P. and K. Erk, "Implicit argument prediction with event knowledge", in "Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)", pp. 831–840 (Association for Computational Linguistics, New Orleans, Louisiana, 2018), URL https://aclanthology.org/N18-1076.

Commons, W., "Standard beam search algorithm with beam sie of 3", (2021).

Costa, T. S., S. Gottschalk and E. Demidova, "Event-qa: A dataset for event-centric question answering over knowledge graphs", CoRR **abs/2004.11861**, URL https://arxiv.org/abs/2004.11861 (2020).

Crawshaw, M., "Multi-task learning with deep neural networks: A survey", URL https://arxiv.org/abs/2009.09796 (2020).

Deng, S., N. Zhang, L. Li, C. Hui, T. Huaixiao, M. Chen, F. Huang and H. Chen, "OntoED: Low-resource event detection with ontology embedding", in "Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)", pp. 2828–2839 (Association for Computational Linguistics, Online, 2021), URL https://aclanthology.org/2021.acl-long.220.

Devlin, J., M. Chang, K. Lee and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding", CoRR **abs/1810.04805**, URL http://arxiv.org/abs/1810.04805 (2018).

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding", in "Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)", pp. 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019), URL https://aclanthology.org/N19-1423.

Du, X. and C. Cardie, "Event extraction by answering (almost) natural questions", in "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)", pp. 671–683 (Association for Computational Linguistics, Online, 2020), URL https://aclanthology.org/2020.emnlp-main.49.

Ebner, S., P. Xia, R. Culkin, K. Rawlins and B. Van Durme, "Multi-sentence argument linking", in "Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics", pp. 8057–8077 (Association for Computational Linguistics, Online, 2020), URL https://aclanthology.org/2020.acl-main.718.

Efrat, A. and O. Levy, "The turking test: Can language models understand instructions?", URL https://arxiv.org/abs/2010.11982 (2020).

Ghaeini, R., X. Fern, L. Huang and P. Tadepalli, "Event nugget detection with forward-backward recurrent neural networks", in "Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)", pp. 369–373 (Association for Computational Linguistics, Berlin, Germany, 2016), URL https://aclanthology.org/P16-2060.

Gupta, P. and H. Ji, "Predicting unknown time arguments based on cross-event propagation", in "Proceedings of the ACL-IJCNLP 2009 Conference Short Papers", pp. 369–372 (Association for Computational Linguistics, Suntec, Singapore, 2009), URL https://aclanthology.org/P09-2093.

Hase, P. and M. Bansal, "When can models learn from explanations? a formal framework for understanding the roles of explanation data", (2021).

He, X., L. Li, Y. Liu, X. Yu and J. Meng, "A two-stage biomedical event trigger detection method integrating feature selection and word embeddings", IEEE/ACM Transactions on Computational Biology and Bioinformatics **15**, 4, 1325–1332 (2018a).

He, X., L. Li, J. Wan, D. Song, J. Meng and Z. Wang, "Biomedical event trigger detection based on bilstm integrating attention mechanism and sentence vector", in "2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)", pp. 651–654 (2018b).

He, X., P. Tai, H. Lu, X. Huang and Y. Ren, "A biomedical event extraction method based on fine-grained and attention mechanism", BMC Bioinformatics **23**, 1, 1–17 (2022).

Hong, Y., J. Zhang, B. Ma, J. Yao, G. Zhou and Q. Zhu, "Using cross-entity inference to improve event extraction", in "Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies", pp. 1127–1136 (Association for Computational Linguistics, Portland, Oregon, USA, 2011), URL https://aclanthology.org/P11-1113.

Huang, R. and E. Riloff, "Modeling textual cohesion for event extraction", Proceedings of the AAAI Conference on Artificial Intelligence **26**, 1, 1664–1670, URL https://ojs.aaai.org/index.php/AAAI/article/view/8354 (2021).

Ji, H. and R. Grishman, "Refining event extraction through cross-document inference", in "Proceedings of ACL-08: HLT", pp. 254–262 (Association for Computational Linguistics, Columbus, Ohio, 2008), URL https://aclanthology.org/P08-1030.

Jungermann, F. and K. Morik, "Enhanced services for targeted information retrieval by event extraction and data mining", in "LWA", (2008).

Kanhabua, N. and A. Anand, "Temporal information retrieval", in "Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval", SIGIR '16, p. 1235–1238 (Association for Computing Machinery, New York, NY, USA, 2016), URL https://doi.org/10.1145/2911451.2914805.

Lai, V. D. and T. H. Nguyen, "Extending event detection to new types with learning from keywords", CoRR **abs/1910.11368**, URL http://arxiv.org/abs/1910.11368 (2019).

Lai, V. D., T. N. Nguyen and T. H. Nguyen, "Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks", in "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)", pp. 5405–5411 (Association for Computational Linguistics, Online, 2020), URL https://aclanthology.org/2020.emnlp-main.435.

Levy, O. and Y. Goldberg, "Dependency-based word embeddings", in "Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)", pp. 302–308 (Association for Computational Linguistics, Baltimore, Maryland, 2014), URL https://aclanthology.org/P14-2050.

Li, F., W. Peng, Y. Chen, Q. Wang, L. Pan, Y. Lyu and Y. Zhu, "Event extraction as multi-turn question answering", in "Findings of the Association for Computational Linguistics: EMNLP 2020", pp. 829–838 (Association for Computational Linguistics, Online, 2020), URL https://aclanthology.org/2020.findings-emnlp.73.

Li, P., Q. Zhu and G. Zhou, "Argument inference from relevant event mentions in Chinese argument extraction", in "Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)", pp. 1477–1487 (Association for Computational Linguistics, Sofia, Bulgaria, 2013a), URL https://aclanthology.org/P13-1145.

Li, Q., H. Ji and L. Huang, "Joint event extraction via structured prediction with global features", in "Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)", pp. 73–82 (Association for Computational Linguistics, Sofia, Bulgaria, 2013b), URL https://aclanthology.org/P13-1008.

Li, S., H. Ji and J. Han, "Document-level event argument extraction by conditional generation", ArXiv **abs/2104.05919** (2021).

Liao, S. and R. Grishman, "Using document level cross-event inference to improve event extraction", in "Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics", pp. 789–797 (Association for Computational Linguistics, Uppsala, Sweden, 2010), URL https://aclanthology.org/P10-1081.

Liao, S. and R. Grishman, "Acquiring topic features to improve event extraction: in pre-selected and balanced collections", in "Proceedings of the International Conference Recent Advances in Natural Language Processing 2011", pp. 9–16 (Association for Computational Linguistics, Hissar, Bulgaria, 2011), URL https://aclanthology.org/R11-1002.

Lin, Y., H. Ji, F. Huang and L. Wu, "A joint neural model for information extraction with global features", in "Proceedings of the 58th Annual Meeting of the

Association for Computational Linguistics", pp. 7999–8009 (Association for Computational Linguistics, Online, 2020), URL https://aclanthology.org/2020.acl-main.713.

Liu, J., Y. Chen, K. Liu, W. Bi and X. Liu, "Event extraction as machine reading comprehension", in "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)", pp. 1641–1651 (Association for Computational Linguistics, Online, 2020), URL https://aclanthology.org/2020.emnlp-main.128.

Liu, J., Y. Chen and J. Xu, "Saliency as evidence: Event detection with trigger saliency attribution", in "Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)", pp. 4573–4585 (Association for Computational Linguistics, Dublin, Ireland, 2022), URL https://aclanthology.org/2022.acl-long.313.

Liu, P., W. Yuan, J. Fu, Z. Jiang, H. Hayashi and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing", CoRR **abs/2107.13586**, URL https://arxiv.org/abs/2107.13586 (2021).

Liu, S., Y. Chen, K. Liu and J. Zhao, "Exploiting argument information to improve event detection via supervised attention mechanisms", in "Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)", pp. 1789–1798 (Association for Computational Linguistics, Vancouver, Canada, 2017), URL https://aclanthology.org/P17-1164.

Liu, S., K. Liu, S. He and J. Zhao, "A probabilistic soft logic based approach to exploiting latent and global information in event classification", URL https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11990 (2016).

Liu, X., Z. Luo and H. Huang, "Jointly multiple events extraction via attention-based graph information aggregation", in "Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing", pp. 1247–1256 (Association for Computational Linguistics, Brussels, Belgium, 2018), URL https://aclanthology.org/D18-1156.

Lourie, N., R. L. Bras, C. Bhagavatula and Y. Choi, "Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark", (2021).

Lu, Y., H. Lin, X. Han and L. Sun, "Distilling discrimination and generalization knowledge for event detection via delta-representation learning", in "Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics", pp. 4366–4376 (Association for Computational Linguistics, Florence, Italy, 2019), URL https://aclanthology.org/P19-1429.

Lu, Y., H. Lin, J. Xu, X. Han, J. Tang, A. Li, L. Sun, M. Liao and S. Chen, "Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction", in "Proceedings of the 59th Annual Meeting of the Association for

Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)", pp. 2795–2806 (Association for Computational Linguistics, Online, 2021), URL https://aclanthology.org/2021.acl-long.217.

Lyu, Q., H. Zhang, E. Sulem and D. Roth, "Zero-shot event extraction via transfer learning: Challenges and insights", in "ACL/IJCNLP (2)", pp. 322–332 (2021), URL https://doi.org/10.18653/v1/2021.acl-short.42.

Man Duc Trong, H., D. Trong Le, A. Pouran Ben Veyseh, T. Nguyen and T. H. Nguyen, "Introducing a new dataset for event detection in cybersecurity texts", in "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)", pp. 5381–5390 (Association for Computational Linguistics, Online, 2020), URL https://aclanthology.org/2020.emnlp-main.433.

McClosky, D., M. Surdeanu and C. Manning, "Event extraction as dependency parsing", in "Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies", pp. 1626–1635 (Association for Computational Linguistics, Portland, Oregon, USA, 2011), URL https://aclanthology.org/P11-1163.

Mikolov, T., K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space", in "1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings", edited by Y. Bengio and Y. LeCun (2013a), URL http://arxiv.org/abs/1301.3781.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality", CoRR **abs/1310.4546**, URL http://arxiv.org/abs/1310.4546 (2013b).

Mishra, S., D. Khashabi, C. Baral, Y. Choi and H. Hajishirzi, "Reframing instructional prompts to gptk's language", (2022).

Nakayama, H., "seqeval: A python framework for sequence labeling evaluation", URL https://github.com/chakki-works/seqeval, software available from https://github.com/chakki-works/seqeval (2018).

Nguyen, T. H., K. Cho and R. Grishman, "Joint event extraction via recurrent neural networks", in "Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies", pp. 300–309 (Association for Computational Linguistics, San Diego, California, 2016), URL https://aclanthology.org/N16-1034.

Nguyen, T. H. and R. Grishman, "Event detection and domain adaptation with convolutional neural networks", in "Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)", pp. 365–371 (Association for Computational Linguistics, Beijing, China, 2015), URL https://aclanthology.org/P15-2060.

Nie, Y., W. Rong, Y. Zhang, Y. Ouyang and Z. Xiong, "Embedding assisted prediction architecture for event trigger identification", Journal of bioinformatics and computational biology **13 3**, 1541001 (2015).

Paolini, G., B. Athiwaratkun, J. Krone, J. Ma, A. Achille, R. Anubhai, C. N. dos Santos, B. Xiang and S. Soatto, "Structured prediction as translation between augmented natural languages", CoRR **abs/2101.05779**, URL `https://arxiv.org/abs/2101.05779` (2021).

Parmar, M., S. Mishra, M. Purohit, M. Luo, M. H. Murad and C. Baral, "In-boxbart: Get instructions into biomedical multi-task learning", (2022).

Patwardhan, S. and E. Riloff, "A unified model of phrasal and sentential evidence for information extraction", in "Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing", pp. 151–160 (Association for Computational Linguistics, Singapore, 2009), URL `https://aclanthology.org/D09-1016`.

Pyysalo, S., F. Ginter, H. Moen, T. Salakoski and S. Ananiadou, "Distributional semantics resources for biomedical text processing", (2013).

Pyysalo, S., T. Ohta, M. Miwa, H.-C. Cho, J. Tsujii and S. Ananiadou, "Event extraction across multiple levels of biological organization", Bioinformatics **28**, 18, i575–i581 (2012).

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language models are unsupervised multitask learners", (2019).

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer", (2020).

Riedel, S. and A. McCallum, "Fast and robust joint models for biomedical event extraction", in "Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing", pp. 1–12 (Association for Computational Linguistics, Edinburgh, Scotland, UK., 2011a), URL `https://aclanthology.org/D11-1001`.

Riedel, S. and A. McCallum, "Robust biomedical event extraction with dual decomposition and minimal domain adaptation", in "Proceedings of BioNLP Shared Task 2011 Workshop", pp. 46–50 (Association for Computational Linguistics, Portland, Oregon, USA, 2011b), URL `https://aclanthology.org/W11-1807`.

Riedel, S., L. Yao and A. McCallum, "Modeling relations and their mentions without labeled text", in "Machine Learning and Knowledge Discovery in Databases", edited by J. L. Balcázar, F. Bonchi, A. Gionis and M. Sebag, pp. 148–163 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010).

Sanh, V., A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang,

M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, T. Bers, S. Biderman, L. Gao, T. Wolf and A. M. Rush, "Multitask prompted training enables zero-shot task generalization", (2022).

Sha, L., F. Qian, B. Chang and Z. Sui, "Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction", Proceedings of the AAAI Conference on Artificial Intelligence **32**, 1, URL `https://ojs.aaai.org/index.php/AAAI/article/view/12034` (2018).

Shazeer, N. and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost", CoRR **abs/1804.04235**, URL `http://arxiv.org/abs/1804.04235` (2018).

Si, J., X. Peng, C. Li, H. Xu and J. Li, "Generating disentangled arguments with prompts: A simple event extraction framework that works", CoRR **abs/2110.04525**, URL `https://arxiv.org/abs/2110.04525` (2021).

Tong, M., B. Xu, S. Wang, Y. Cao, L. Hou, J. Li and J. Xie, "Improving event detection via open-domain trigger knowledge", in "Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics", pp. 5887–5897 (Association for Computational Linguistics, Online, 2020), URL `https://aclanthology.org/2020.acl-main.522`.

V S S Patchigolla, R., S. Sahu and A. Anand, "Biomedical event trigger identification using bidirectional recurrent neural network based models", in "BioNLP 2017", pp. 316–321 (Association for Computational Linguistics, Vancouver, Canada,, 2017), URL `https://aclanthology.org/W17-2340`.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, "Attention is all you need", in "Advances in Neural Information Processing Systems", edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett (Curran Associates, Inc., ????).

Venugopal, D., C. Chen, V. Gogate and V. Ng, "Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features", in "Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)", pp. 831–843 (Association for Computational Linguistics, Doha, Qatar, 2014), URL `https://aclanthology.org/D14-1090`.

Veyseh, A. P. B., V. D. Lai, F. Dernoncourt and T. H. Nguyen, "Unleash gpt-2 power for event detection", in "ACL", (2021).

Wadden, D., U. Wennberg, Y. Luan and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations", in "Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)", pp. 5784–5789 (Association for Computational Linguistics, Hong Kong, China, 2019), URL `https://aclanthology.org/D19-1585`.

Wang, A., J. Wang, H. Lin, J. Zhang, Z. Yang and K. Xu, "A multiple distributed representation method based on neural network for biomedical event extraction", BMC Medical Informatics and Decision Making **17** (2017).

Wang, R. C. and W. W. Cohen, "Character-level analysis of semi-structured documents for set expansion", in "Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing", pp. 1503–1512 (Association for Computational Linguistics, Singapore, 2009), URL https://aclanthology.org/D09-1156.

Wang, S., M. Yu, S. Chang, L. Sun and L. Huang, "Query and extract: Refining event extraction as type-oriented binary decoding", CoRR **abs/2110.07476**, URL https://arxiv.org/abs/2110.07476 (2021a).

Wang, S., M. Yu and L. Huang, "The art of prompting: Event detection based on type specific prompts", URL https://arxiv.org/abs/2204.07241 (2022a).

Wang, X., X. Han, Z. Liu, M. Sun and P. Li, "Adversarial training for weakly supervised event detection", in "Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)", pp. 998–1008 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019), URL https://aclanthology.org/N19-1105.

Wang, X., Z. Wang, X. Han, W. Jiang, R. Han, Z. Liu, J. Li, P. Li, Y. Lin and J. Zhou, "MAVEN: A Massive General Domain Event Detection Dataset", in "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)", pp. 1652–1671 (Association for Computational Linguistics, Online, 2020), URL https://aclanthology.org/2020.emnlp-main.129.

Wang, Y., S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Arunkumar, A. Ashok, A. S. Dhanasekaran, A. Naik, D. Stap, E. Pathak, G. Karamanolakis, H. G. Lai, I. Purohit, I. Mondal, J. Anderson, K. Kuznia, K. Doshi, M. Patel, K. K. Pal, M. Moradshahi, M. Parmar, M. Purohit, N. Varshney, P. R. Kaza, P. Verma, R. S. Puri, R. Karia, S. K. Sampat, S. Doshi, S. Mishra, S. Reddy, S. Patro, T. Dixit, X. Shen, C. Baral, Y. Choi, N. A. Smith, H. Hajishirzi and D. Khashabi, "Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks", URL https://arxiv.org/abs/2204.07705 (2022b).

Wang, Z., X. Wang, X. Han, Y. Lin, L. Hou, Z. Liu, P. Li, J. Li and J. Zhou, "Cleve: Contrastive pre-training for event extraction", URL https://arxiv.org/abs/2105.14485 (2021b).

Webson, A. and E. Pavlick, "Do prompt-based models really understand the meaning of their prompts?", (2022).

Wei, J., M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai and Q. V. Le, "Finetuned language models are zero-shot learners", (2022).

Williams, R. J. and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks", Neural Computation **1**, 2, 270–280 (1989).

Xie, T., C. H. Wu, P. Shi, R. Zhong, T. Scholak, M. Yasunaga, C.-S. Wu, M. Zhong, P. Yin, S. I. Wang, V. Zhong, B. Wang, C. Li, C. Boyle, A. Ni, Z. Yao, D. Radev, C. Xiong, L. Kong, R. Zhang, N. A. Smith, L. Zettlemoyer and T. Yu, "Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models", (2022).

Yang, S., D. Feng, L. Qiao, Z. Kan and D. Li, "Exploring pre-trained language models for event extraction and generation", in "Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics", pp. 5284–5294 (Association for Computational Linguistics, Florence, Italy, 2019), URL `https://aclanthology.org/P19-1522`.

Ye, Q. and X. Ren, "Learning to generate task-specific adapters from task description", (2021).

Yu, P., H. Ji and P. Natarajan, "Lifelong event detection with knowledge transfer", in "Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing", pp. 5278–5290 (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021), URL `https://aclanthology.org/2021.emnlp-main.428`.

Zhang, H., H. Wang and D. Roth, "Zero-shot Label-aware Event Trigger and Argument Classification", in "Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021", pp. 1331–1340 (Association for Computational Linguistics, Online, 2021), URL `https://aclanthology.org/2021.findings-acl.114`.

Zhang, W., B. Ingale, H. Shabir, T. Li, T. Shi and P. Wang, "Event detection explorer: An interactive tool for event detection exploration", URL `https://arxiv.org/abs/2204.12456` (2022).

Zhao, J., D. Khashabi, T. Khot, A. Sabharwal and K.-W. Chang, "Ethical-advice taker: Do language models understand natural language interventions?", (2021).

Zhong, R., K. Lee, Z. Zhang and D. Klein, "Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections", (2021).

Zhou, D. and D. Zhong, "A semi-supervised learning framework for biomedical event extraction based on hidden topics", Artificial intelligence in medicine **64 1**, 51–8 (2015).

APPENDIX A

DATA

Input:
In April 1916, Irish republicans launched the Easter Rising against British rule and proclaimed an Irish Republic.

Gold:
launched->process_start | rising->**change_of_leadership** | proclaimed->statement

Predicted:
Rising->**change_of_leadership** | launched->process_start | proclaimed->statement

Figure A.1: Example of Case Error in Annotation.

Event annotations are case sensitive, as a different case can refer to a different event trigger altogether. For example, *Hurricane* vs *hurricane* can be an important distinction, especially if the former refers to the occurrence of a specific instance of a hurricane. However, there may be errors in existing annotations due to improper extraction or human error.

This model extracts text terms correctly with case-sensitivity, which can help identify such errors in annotation. For example, A.1

WikiEvents	Common	RAMS
conflict.defeat, medical.intervention, disaster.diseaseoutbreak, justice.releaseparole, movement.transportation, cognitive.inspection, justice.acquit, justice.sentence, transaction.exchangebuysell, justice.trialhearing, cognitive.identifycategorize, justice.convict, artifactexistence.damagedestroydisabledismantle, genericcrime.genericcrime, artifactexistence.manufactureassemble, contact.requestcommand, control.impedeinterferewith, justice.investigatecrime, justice.chargeindict, cognitive.teachingtraininglearning, transaction.donation, cognitive.research, life.infect, disaster.crash, contact.contact	justice.arrestjaildetain, personnel.startposition, personnel.endposition, conflict.attack, conflict.demonstrate, life.injure, contact.threatencoerce, life.die	contact.collaborate, justice.investigate, contact.commitmentpromiseexpressintent, justice.judicialconsequences, contact.mediastatement, contact.commandorder, manufacture.artifact, contact.negotiate, transaction.transaction, government.legislate, contact.publicstatementinperson, contact.funeralvigil, disaster.fireexplosion, artifactexistence.damagedestroy, government.formation, justice.initiatejudicialprocess, government.agreements, personnel.elect, movement.transportperson, transaction.transferownership, conflict.yield, inspection.sensoryobserve, government.spy, government.vote, transaction.transfermoney, movement.transportartifact, disaster.accidentcrash, contact.discussion, contact.requestadvise, contact.prevarication

Table A.1: Event Types in RAMS and WikiEvents. Common: List of Event Types Common to Both Datasets.