Modeling Event Implications via Multi-faceted Entity Representations

Evangelia Spiliopoulou

CMU-LTI-22-017

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:

Eduard Hovy
Yonatan Bisk
Lorraine Levin
Alan Ritter

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Language and Information Technologies

Keywords: events, entity representations, event implications, limited-data, commonsense reasoning, large language models.

Abstract

Representing knowledge is a foundational aspect of *Natural Language Understanding*, ranging from meticulously designed notational vocabularies to high-dimensional automatically-created numerical distributions. Good representations must have high coverage with respect to the meanings inherent in a task and an unambiguous interpretable structure that is machine-readable. Once the representation is defined, constructing instances of such representations for given examples (either manually or automatically) faces two main sub-problems: (1) extracting the information conveyed by a word or sentence and (2) structuring the extracted information appropriately and filtering out what is unimportant, based on context.

Recent advances focus on representation learning by deep neural networks, where numerical distributions representing words or sentences are learned using a machine learning model by processing huge volumes of data. However, these representations do not have an internal schematic representation and are hence not interpretable by humans. This leads to a currently central question in AI of whether these high-performing models are able to reason over events and their implications in the real world, or whether they simply memorize all the training examples and perform a small amount of generalization.

In this thesis, we address the problem of representing and reasoning about events and their implications in the physical world. We propose methods to create more explainable representations of knowledge that retain only the parts of the encoded information that are relevant to a task at hand. Our approach results in models that learn underlying reasoning mechanisms and apply them to unseen situations (i.e., generalization). We study representations at different levels of semantics (lexical/conceptual, sentence, and discourse); representations for words, sentences, and event chains. Our methods address the following questions:

1. Can we separate different aspects of meaning in our representations and identify the aspects relevant for a task at hand, either via a fixed structure or through learning?
2. How do task formulation and representation structure affect performance in limited-data scenarios?
3. May infusing representations with human knowledge replace the need for huge volumes of training data? We study how definitions, ontologies or task explanations can be combined with a machine learning model.
4. Can a model trained only on language learn physical event implications and reasoning mechanisms that generalize across domains?

The answers to these questions enable us to create multi-faceted representations of entities that guide a deep neural network to learn reasoning mechanisms and avoid shortcut learning, which is a major impediment in limited-data or domain-transfer scenarios.

Contents

Chapter 1

Introduction

Designing algorithms that extract human knowledge from natural languages like English and transform it into a structured, machine readable representation is a fundamental subject for Natural Language Processing. This is a particularly hard problem, as human languages are highly complex and semantics are multi-faceted. The meaning of an utterance (word, sentence or phrase) is not one-dimensional, but can instead be separated into distinct facets. Depending on the task or context, we might be interested in a different aspect of a term's or sentence's meaning that contains the most relevant information.

Although humans have developed extraordinary skills on extracting and reasoning about the meaning of utterances, automatic approaches are still struggling to achieve this level of intelligence. The first step to achieve this is to design robust and generalizable structures that represent the meaning of a concept and can be used for reasoning tasks. Ontology-based methods constitute one of the oldest approaches to organize and represent knowledge, and are still widely used in NLP tasks. They can be in the form of lexical resources like WordNet [Miller, 1995] and FrameNet [Baker et al., 1998a] or domain-dependent ontologies carefully designed for particular problems/domains. Ontologies are particularly useful since they contain accurate and semantically interpretable information that can be easily accessed and filtered by humans according to the task of interest. However, this information is typically constructed manually, which is a very time-consuming and difficult process. This results in representations that are not easily extensible, so they cannot be modified or fine-tuned in the presence of new information.

Recent advances focus on learning representations by training a language model on extremely large corpora. This is a more data-driven approach than the meticulous construction of an ontology, since constructing distributed representations is fully automated and can be fine-tuned for any new task. Most recent representations that take context into account to encode the most relevant meaning of concepts are context-aware embeddings based on models like BERT [Devlin et al., 2019] or GPT-3 [Brown et al., 2020], which have been shown to achieve significant improvements in various downstream NLP tasks. However, a major problem with these representations is that they are not interpretable by humans, since they are a byproduct of deep transformer networks. Thus, we currently have no way to understand how information is encoded and whether semantics are also somehow encoded in the representation. This impedes us from choosing which part of the representation is relevant to a specific task, forcing us instead to rely solely on the availability of huge amounts of data for fine-tuning. The resulting representations are based on models with billions of parameters, which leads us to the following critical question; can such representations capture generalizable reasoning patterns about complex tasks, such as event implications, or can they only solve problems based on previously extracted surface correlation patterns? For example, given the sentence *The glass fell on the floor*, how do we learn a representation that can answer where is the glass or if it is broken? What if the *glass* was a *pen* instead? Such tasks, which are simple implications of events, despite being trivial for humans, are almost impossible to solve by current NLP methods. This is partly due to the way the models learn representations of words or sentences, where meaning is defined solely based on explicitly stated context (i.e., other words in the sentence). As a result, other facets of semantics, such as the implications of an event to the entities, are frequently ignored, unless explicitly stated in the context.

In this thesis, we explore how to design multi-faceted representations of semantics that capture and reason over implications of events in the real-world. To this end, we propose methods to learn more explainable representations that separate different aspects of meaning based on specific tasks. These representations have higher generalization power and show better performance, particularly in scenarios when only limited or no in-domain data is available.

We design representations for various levels of semantic units: entities, sentences and events. We propose a set of projects for each sub-area of semantics, all of which aim to construct more explainable representations, where meaning is split into independent dimensions, and show how we can practically use these representations to improve performance in NLP tasks. The thesis initially studies entity representations, which have fixed facets and structure, and incrementally explores representations of event relations, which are significantly more complex as they depend on the participating entities.

Chapter Overview

Chapter 3: Lexical Representations This chapter discusses Definition Frames, a multi-part representation matrix in which each facet (each row) corresponds to a particular relation, and the content of each facet is the embedding of the terms related to the represented word. The relations are based on the Qualia structure proposed in Boguraev and Pustejovsky [1990], which represent the modes of explanation of the given entity. Their values are automatically extracted from WordNet [Miller, 1995] definitions using a domain-adaptation approach.

As shown in Spiliopoulou et al. [2020b], a word representation that separates meaning into different facets is both interpretable and achieves better performance in limited-data scenarios. By disentangling the Qualia structure relations, Definition Frames can capture different types of similarity (relatedness and similarity) and achieve improved performance on word similarity tasks. Finally, we demonstrate the explainability of Definition Frames via a human study showing that they provide valid insights on how terms are related.

Chapter 4: Sentence Representations This chapter focuses on learning sentence representations so only information that is relevant to a specific topic or context is encoded. Given the context (task and topic), the goal of this work is to disentangle the meaning of a sentence and filter information irrelevant to that context. This is particularly important in a covariate shift scenario, where there are systematic differences between training and test data due to an underlying cause. In such scenarios the model deals with shortcut learning; it learns simple associations that explain the training data, but do not generalize in the test set.

Our work on this area focuses on a real-time application of classifying tweets with respect to their importance for crisis response. This is a covariate shift scenario, since we have no labeled data for an ongoing crisis when our training set consists of tweets from previous similar in-nature catastrophes (e.g., earthquake). As we can imagine, each tweet contains information about its importance in the context of the type of the disaster (e.g., earthquake) or of the specific disaster (e.g., Nepal 2015 earthquake). Given that there is data for only a handful of these crisis events, each event serves as the underlying cause of a distribution shift across the data.

As shown in Spiliopoulou et al. [2020a], our approach teaches the model to disassociate tweet importance from the specific crisis event, which is the underlying cause of the covariate shift. By disentangling the aspect of meaning that is useful for our task, the model learn representations in the context of the event type and achieves better performance in the test set. Since the main task is to classify the importance of the information contained in a tweet (criticality), we use an adversarial classifier that intends to learn which specific event the tweet refers to, hence remove the event specific bias through a reversal gradient. Our experiments represent a real-life crisis management scenario, where the model is evaluated on a new incoming event through a leave-one-out experimental setup, and show substantial improvement over baseline classification methods.

Chapter 5: Event Representations Events are complex semantic units due to their temporal dimension and their ability to modify the world state and its entity characteristics. Furthermore, an event may consist of other sub-events (*eating* is a sub-event of *dining in a restaurant*) or cause other events (*turning off the light* implies that *I cannot see*). In the NLP community, events are represented by a subset of fixed thematic relations (i.e., agent, patient, etc.), an approach that splits the semantics of an event to distinct, independent dimensions. As shown in Spiliopoulou et al. [2017], we can use such predefined, fixed structures to extract information from text, by combining rich semantic knowledge with deep learning methods.

Chapter 5 discusses events and their representation schemes, as well as their interactions with other events. While event detection within the NLP community focuses on event representations on sentence level, our interest is to study complex, large-scale events that are described across multiple documents. Such events do not necessarily correspond to a particular snippet of text and, instead, consist of several smaller sub-events that are mentioned in text.

The goal of this chapter is to model such events in terms of their sub-events: events of smaller duration that might influence the outcome of the large event. In day-to-day scenarios, we see several such large-scale events (e.g., financial crises, elections, covid-19), where the severity and outcome of the event is actually driven by the smaller sub-events that occur during the same period. However, due to the complexity of such large-scale events and their apparent dependence on their sub-events, we cannot represent them using a fixed schematic representation.

As shown in Spiliopoulou et al. [2021], we can represent large-scale events using a dynamic series of their sub-events. In this work, we develop a framework to extract influential sub-events that occur during a large-scale event disaster event and could potentially impact the evolution of the crisis (e.g., power outage, road blocks, etc.), based on natural language input. Our framework processes information acquired both in the sentence-level (sub-events per tweet) and across tweets in the context of the large-scale event, via the use of a dynamic graph neural network.

Chapter 6: Event Implications on Entities In Chapters 3–5 we addressed representations of entities, sentences, and events, using the assumption that their semantics are multi-faceted. Entities are the least complex semantic unit, since they can be defined based on a set of physical attributes and ontological relations. Events are more complex, since their semantics depend on both ontological relations (such as inheritance from higher-level events, like *move* → *run*) and their participants, which are typically entities.

Chapter 6 studies the impact of an event on the real world: specifically, how an event modifies the state of the entities impacted by the event. Given an event, humans are able to infer a great amount of information about how it affects other entities and events without this being explicitly stated. Although inferring such commonsense information is a trivial reasoning task for humans, it is particularly hard for neural models or any other algorithmic methods. This chapter aims to focus on a part of a core AI problem known as the Frame Problem. First introduced by McCarthy and Hayes [1981], the Frame Problem was formulated as the problem of updating all the beliefs about the state of the real world that were modified as the result of actions.

Although the Frame Problem is extremely hard, this chapter focuses on only one particular aspect: how an event's participant entities are modified by the event occurrence. Although this studies only the direct implications of the event on an entity's state, it still requires a deep understanding of the semantics of the single event. In comparison, more complex relations, such as causality of events, require further reasoning and understanding of how multiple events interact. For example, consider the event *I broke my phone*. A direct implication in an entity's state would be that the phone's *composition* changed, which only requires knowledge about the semantics of the event *break* and how it influences the entity. However, the event *I need a new phone* is a result of the first event and must be deduced via reasoning over the space of several possible events.

As part of this thesis, we study event implications as entity change-of-state with respect to physical attributes. Unlike Definition Frames, where the essential, definitional, facets of an entity are represented via its relation to other entities, here the different aspects of meaning are driven by the attributes of the entity. The reason behind this difference is the goal of the representations and task: given some textual description of an event, we want to model the properties of the entity that could change, instead of the fundamental properties that do not change.

In this work, we show that, by facetizing the meaning of an entity via its attributes, both the in-domain performance and the generalization abilities of our models significantly improve. We achieve this by verbalizing the different aspects of meaning in the form of attributes and feeding it as input to a large Language Model (LLM) via prompting. The attributes work as a bottleneck that retains only the information from the sentence that is relevant to them, guiding the model towards learning reasoning patterns for the task.

Thesis Summary

The underlying theory of this work is that semantics are multi-faceted and it is possible, with appropriate care, to disentagle and represent separately the major facets and use them individually as needed. In this thesis we show how meaning decomposition via proper task formulation results to better model performance and generalization abilities. This is possible even for representation notations that are opaque to humans, like large language models, where the separation functions as a bottleneck for the models to connect relevant information and learn reasoning patterns.

Supporting Publications

In order to support this thesis, we use our findings of previous and current research for each chapter. More specifically, the following papers support each chapter:

Chapter 3 Spiliopoulou, Evangelia, Artidoro Pagnoni, and Eduard Hovy. "Definition Frames: Using Definitions for Hybrid Concept Representations." In Proceedings of the 28th International Conference on Computational Linguistics, pp. 3060-3068. 2020.

Chapter 4 Spiliopoulou, Evangelia, Salvador Medina Maza, Eduard Hovy, and Alexander G. Hauptmann. "Event-Related Bias Removal for Real-time Disaster Events." In Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 3858-3868. 2020.

Chapter 5 Spiliopoulou, Evangelia, Eduard Hovy, and Teruko Mitamura. "Event detection using frame-semantic parser." In Proceedings of the Events and Stories in the News Workshop, pp. 15-20. 2017.

Spiliopoulou, Evangelia, Tanay Kumar Saha, Joel Tetreault, and Alejandro Jaimes. "A Novel Framework for Detecting Important Subevents from Crisis Events via Dynamic Semantic Graphs." In Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021), pp. 249-259. 2021.

Chapter 6 Spiliopoulou, Evangelia, Artidoro Pagnoni, Yonatan Bisk, and Eduard Hovy. EvEntS ReaLM: Event Reasoning of Entity States via Language Models. Under-review.

Chapter 2

Background & Related Work

This chapter briefly discusses prior work in NLP problems related to the thesis. This serves as the background to understand the contributions of each chapter to specific NLP problems and how they differ from previous approaches.

We follow a structure that facilitates the introduction of the main thesis chapters; work is categorized based on the representation type (i.e., lexical, sentence or event) or application area. The purpose of this chapter is to inform the reader and facilitate their understanding of our work, providing context of the thesis and current NLU challenges.

2.1 Lexical Representations

This section discusses prior work on lexical representations, with a particular focus on entities. The goal is to provide the background necessary to motivate Chapter 3 and settle the foundations for the entire thesis.

2.1.1 Lexicons & Ontologies

One of the earliest ways to organize and represent knowledge is via ontologies and hand-crafted lexicons. Ontologies aim to categorize concepts based on their meaning in a hierarchical tree-structure. Carefully built by experts, they contain semantically meaningful information in the form of relations between concepts. Throughout this thesis, we refer to several widely used ontologies, such as WordNet [Miller, 1995] and ConceptNet [Speer and Havasi, 2012], which contain concepts, their definitions and essential relations to other concepts.

Ontologies can represent detailed semantic relations of concepts and are still used as knowledge representations in several NLU tasks. Since they are meticulously designed by experts, they contain precise information and are easy to interpret by humans. However, designing or adding new nodes to an ontology requires manual effort from domain experts, which means that they are not scalable and easy to extend. This is a significant barrier when we want to represent words and extract semantic relations in a new domain.

2.1.2 Distributional Semantics

Recent work on lexical representations focuses on distributional semantics, where each word is represented by a large numerical vector of fixed dimension, learnt by a machine learning model. These vectors called *embeddings* and the overall goal is to minimize the distance between vectors that correspond to semantically similar words.

Early approaches in distributional semantics aimed on building a language model based on a large corpus of text. Firstly introduced by Baroni and Lenci [2010] as the Distributional Memory framework, such models are used to generate a vector for each word, based on all the context this word was previously seen. The vector for each word is unique and it does not change after the language model is trained. Other prominent work on this area includes GloVe [Pennington et al., 2014], word2vec [Mikolov et al., 2013a], and fastText [Bojanowski et al., 2017].

A common assumption among these approaches is that a word can be represented by a unique vector. However, the meaning of a word frequently depends on context and, thus, a fixed representation might not be sufficient to represent the semantics. This was the main motivation behind new methods that learn contextualized embeddings. Such methods produce embeddings as a function of the entire sentence (i.e., context), which means that they produce different embeddings for the same word, if it occurs in a different sentence. Most work in this area uses masking to learn a language model. ELMo produces embeddings based on the internal states of a BiLSTM model [Peters et al., 2018], while BERT [Devlin et al., 2018] and GPT [Radford et al., 2018] use transformer models. These approaches are shown to generalize well and outperform previous models in a large variety of NLP tasks.

Despite their exceptional performance, these models have a few shortcomings. Firstly, training them requires a large number of resources; specifically compute power and data. Many of these models rely on fine-tuning, which means that even pre-trained models need a large amount of data of the same domain and task in order to achieve significantly better performance than previous representations.

The second short-coming of distributed representations is their lack of interpretability. Unlike lexicons and ontologies, word embeddings are not interpretable by humans, since it is hard to understand the semantics behind a vector. Despite research efforts to interpret word embeddings [Mikolov et al., 2013b], there is no clear evidence of how their semantic relation. Recent work on interpretability focuses more on the effect that each word representation on the final model decision or how changes in context are reflected in the representations [Ethayarajh, 2019].

2.1.3 Representations from Definitions

Dictionary definitions constitute an excellent source of human knowledge, as they contain essential relations to identify the meaning of a concept. Despite written in natural language, definitions follow a very specific structure. Most definitions of a concept contain the type to which it belongs (*Genus*) and the properties that differentiate it from other concepts of the same class (*Differentia*). This distinction of definitions and their importance in scientific knowledge dates as far as Aristotle [Barnes, 1994] and sets the foundations of modern logic [Granger, 1984, Parry and Hacker, 1991].

In addition to their structure, definitions contain generic information that is sufficient to uniquely identify a concept that represents a *universal* term, whereas most natural language text (i.e., news articles, books, online forums) typically contain information about specific instances (i.e., individual) of a concept that portray only one aspect of meaning, based on context. These interesting properties of definitions motivate a series of work that uses them as sources to extract knowledge.

Earlier work in computer science literature uses definitions to extract the type of a concept (*Genus*) and the relations distinguishing it from other members of the same type (*Differentia*) via syntax and string matching heuristics [Binot and Jensen, 1993, Calzolari, 1984, Chodorow et al., 1985]. Recent approaches directly encoded definitions to distributed representations. Tissier et al. [2017] obtained embeddings via a skip-gram model trained on definitions, while Bosc and Vincent [2018] used an auto-encoder. Other work includes definition generation [Noraset et al., 2017], binary classification of sentences on whether they are definitional [Anke and Schockaert, 2018], reverse dictionary look-up [Hill et al., 2016, Zock and Bilac, 2004], and extraction of hypernymy relations from definitions using syntactic patterns [Boella and Di Caro, 2013].

Another related line of work focuses on structuring and extracting relations that define a concept, without explicitly using definitions. Prior research on lexical semantics has established a set of relations that is ideally sufficient to define a concept. Part of this work includes the Qualia structure [Boguraev and Pustejovsky, 1990] and the generative lexicon theory [Pustejovsky, 1991], which set the theoretical foundation for Chapter 3. The theory behind this approach is that a set of relations contain information about different aspects of meaning about a concept, creating a representation based on multiple, fixed facets. Other approaches include fine-grained definition-based frames like Semagrams [Moerdijk et al., 2008].

2.2 Semantics in a Sentence

In this section we discuss different approaches on capturing and representing the semantics of a sentence. In this area, there is a large amount of work that focuses on sentence meaning and representations, which are applied to numerous NLU applications. However, for the scope of this thesis, we narrow our literature review on the two problems that we subsequently discuss throughout the thesis; bias removal from sentence embeddings and event detection.

2.2.1 Learning Unbiased Representations

In this section we discuss sentence representations and recent techniques on detecting and removing biases, in an effort to obtain more generalizable models that transfer across domains. Bias removal is a problem parallel to multi-faceted sentence representations, as we care only about a specific aspect of the representation, based on the context that the representation is learnt (i.e., the bias). Although most recent work on bias removal assumes that the biases are known and take a binary value, this is not always true. Biases can be complex and latent, based on the data used to learn the representation. In Chapter 4 we discuss such an application of event-specific biases and how we can obtain more generalizable representations by removing the bias.

Most recent work on bias removal [Elazar and Goldberg, 2018] focuses on using adversarial learning to remove demographic bias from representations. Examples include adversarial generative networks that create fair representations [Madras et al., 2018], metrics to quantify unintended biases [Borkan et al., 2019] and applications that show substantial improvements on traditional NLP tasks like NLI [Lu et al., 2018], Coreference Resolution [Belinkov et al., 2019] and text classification [Zhang et al., 2018] by using unbiased representations. Our approach is inspired by the work of Elazar and Goldberg [2018] on bias removal through an adversarial attack. The authors use an adversarial setting to remove demographic information from text and construct cleaner representations. In our case, the adversarial classifier attempts to predict the event to which the tweet belongs. Another difference with our work is the imbalanced data used for training the classifier of the main task. Other related work includes domain adaptation based on a gradient-reversal layer [Ganin et al., 2016], text classification based on adversarial multi-task learning [Liu et al., 2017], and multi-adversarial domain adaptation across multi-modal data [Pei et al., 2018].

2.2.2 Events in a Sentence

The second area of work that we discuss involves events and how they can be represented. Events are a complex semantic unit that is analyzed in several chapters of this thesis. Chapter 5 discusses different ways to represent events; via their thematic roles and via their sub-events. In this subsection we discuss prior work on event detection from sentences; a well-studied field of NLP.

Although events may span across more than one sentences, event detection in text-level helps us formalize what an event is. The first part of Chapter 5 discusses into greater detail the task of event detection, while the second part of the same chapter uses techniques and representations of event detection to build a model that extracts sub-events in a large-scale event. Finally, Chapter 6 studies implications of events; thus, a deep understanding of events and prior work is necessary.

Early approaches on event detection were based on extracting features that map the potential event triggers to an ontology. Such approaches are still widely used and efficient when there are limited in-domain training data. The ontology used might be simple (i.e., a list of words in tree-structure) or more complex, designed to represent the semantics of events. Related work follows the *frame semantics* theory, and includes prominent examples like FrameNet [Baker et al., 1998a], VerbNet [Schuler, 2005] and Abstract Meaning Representations [Banarescu et al., 2013].

Research in event detection can be divided into two categories; domain-specific and open-domain. Domain-specific means that we are interested only on a subset of events that are related to a specific domain, which are given based on a pre-specified ontology. Most recent work on this area uses BERT models combined with other techniques. Tong et al. [2020] uses BERT and a knowledge distillation techniques with external sources, while Cao et al. [2020] focuses on incremental learning. Other techniques used include adversarial learning [Wang et al., 2019], graph neural networks [Nguyen and Grishman, 2018] and data generation based on BERT [Yang et al., 2019]. Prior to contextualized embeddings, Chen et al. [2015] proposed a dynamic multi-pooling convolutional neural network (DMCNN), which automatically induces lexical-level and sentence-level features from text, achieving state-of-the-art results. Nguyen and Grishman [2015]'s work focuses on CNNs using word embeddings in order to achieve a more generalizable event detection system. Other approaches include Ghaeini et al. [2016]'s FBRNN, which is a modification of RNNs using word and branch embeddings, and Liu et al. [2016]'s ANN & Random ANN, which exploits the direct relationship between the FrameNet and the ACE Ontology in order to construct an out-domain ANN model. Peng et al. [2016] showed that it is feasible to achieve state-of-the-art results with minimal supervision. In their approach, they use only a few examples and the SRL of a candidate event in order to construct a structured vector representation, which maps the event to an ontology.

Open-domain event detection or, in other words, open information extraction, is the problem of extracting events from a sentence in any domain. All sentences contain some events and we are called to extract information about all these events. Recent work on open information extraction includes Open IE [Stanovsky et al., 2018b], which uses a deep BiLSTM sequence prediction model and systems that combine BERT embeddings with other neural models, such as a BiLSTM decoder [Kolluru et al., 2020].

2.3 Crisis NLP & Information in Social Media

In this section we discuss work in Crisis NLP and relevant modeling techniques, which provide the necessary background for Chapters 4 and 5. Related modeling techniques focus on two main directions: (1) information extraction or classification in the tweet / sentence level, and (2) information aggregation across documents or social media posts.

2.3.1 Information Extraction & Classification in Tweets

Given the large volume of noisy data from social media, most tasks focus on sentence classification problems, where the goal is to filter only the most important posts that might be helpful for first responders. As discussed by Imran et al. [2015], Tapia et al. [2011b], there are several types of sentence classification for disaster response, such as determining if a message is related to a specific crisis event [Caragea et al., 2016, Kruspe, 2019, Nguyen et al., 2016, Neubig et al., 2011], if it is actionable [Leavitt and Robinson, 2017, Munro, 2011] or critical [Mccreadie et al., 2019, Spiliopoulou et al., 2020a]. Other work classifies tweets with respect to the type of information they contain, a problem that is formulated as a multi-class tweet classification (typically five major information types) [Miyazaki et al., 2019, Burel et al., 2017, Nguyen et al., 2017, Imran et al., 2016b, Padhee et al., 2020].

Related work outside of Crisis NLP can also be used to extract information from tweets, in the form of events. Chen et al. [2018] use an encoder-decoder framework to extract sub-events from each tweet, while Rudra et al. [2018] use noun-verb pairs to represent sub-events, where each pair is ranked based on their overlap score in tweets. Some approaches outside the crisis domain that focus on extracting textual sub-events from tweets or documents, in a sequence classification setup [Bekoulis et al., 2019]. Other related work includes Open IE methods (open information extraction), which extract tuples of expressions from text that represent the events of the sentence. Such work includes Open IE by AllenNLP [Stanovsky et al., 2018a], which uses a deep BiLSTM sequence prediction model and systems that combine BERT embeddings with other neural models, such as a BiLSTM encoder [Kolluru et al., 2020].

2.3.2 Aggregating Information Across Documents or Tweets

Processing information without the context of a crisis event is a bottleneck for big data crisis analytics, as discussed by Qadir et al. [2016]. Although most work in Crisis NLP focuses in tweet-level information and classification, most emergencies require to process information across documents or social media posts. Towards that goal, recent work aims at extracting sub-events that are important in the context of the larger crisis event. The notion of sub-events varies within this area; a sub-event could correspond to an entire cluster of words / tweets or to a textual span from a single tweet. Earlier work in sub-event extraction forms clusters of tweets during a crisis event based on a set of shallow features, such as tf-idf and metadata Abhik and Toshniwal [2013], Pohl et al. [2012]. Other approaches use topic clustering to form sets of words (topics) that represent sub-events Srijith et al. [2017], Xing et al. [2016]. Most recent work forms clusters based on verb-noun pairs from individual tweets Jiang et al. [2019], which are then ranked based on an ontology grounding score Arachie et al. [2020]. In all these methods each cluster is considered to correspond to a different sub-event. However, the elements within each cluster are not necessarily related via temporal or other relations, which raises questions with respect to the interpretability of the cluster/sub-event.

In a different direction, other methods use a group of temporally ordered messages to detect large-scale events. For example, Sakaki et al. [2010] use statistical and keyword features in a spatio-temporal model to detect crisis events based on Twitter streams. More recently, Meladianos et al. [2015, 2018] use a graph representation of tweets to extract important sub-events by detecting weight changes, a problem formulated as a summarization task. Early approaches that use text from social media to represent context for social events rely on linear classifiers using topic-related features Wang et al. [2012], graph features Keneshloo et al. [2014] or combination of heterogeneous data sources Korkmaz et al. [2015] and dynamic query expansion models with a static vocabulary Zhao et al. [2015] or fused with logistic regression Ramakrishnan et al. [2014]. Ning et al. [2018], Zhao et al. [2015] use multi-task models with shared parameters across different locations and events to model spatio-temporal correlations. Most recent work that inspired our approach uses dynamic graphs to represent information from social media, which model temporal constraints from precursor events Deng et al. [2020, 2019], Ning et al. [2018]. A common theme of this work, as further discussed by Ning et al. [2019] underlines the importance of explainability, since it is helpful for experts to analyze which factors led to the development of a large-scale event and potential ways to prevent or mitigate it.

2.4 Commonsense Reasoning in Events

Work in commonsense reasoning about events follows two directions: (1) predict event implications as entity changes, and (2) use commonsense knowledge about events and their implications as necessary intermediate steps in reasoning. Furthermore, work that studies event implications as entity change-of-state is distinguished into two broad categories based on the type of events studied: physical or social events.

2.4.1 Modeling Social Interactions

Research that directly studies event implications mostly explores causality between social events and emotional states, based on social norm expectations [Rashkin et al., 2018, Sap et al., 2019b, Forbes et al., 2020, Emelin et al., 2020, Hwang et al., 2020]. Jiang et al. [2021] study specific linguistic phenomena such as contradiction and negation, while Sap et al. [2019a] study the role of social biases and predicting implications of social events. Although this line of research highlights the difficulty of predicting cause-effect relations, social scenarios are typically ambiguous and require knowledge of event chains. For example, in order to answer whether *X gives a gift to Y* implies that *X hugs Y*, we must be aware of the relation between X and Y, their personalities, and the social context. On the other hand, event implications as physical changes of state of entities are, mostly, objective and depend on simple relations that a model could know a priori (e.g., the material of a mug), allowing us to isolate and study the reasoning abilities of a model.

2.4.2 Physical Event Implications

Closer to the subject of this thesis, is the prediction of physical implications of events. This problem often takes the form of entity changes in procedural text, such as in cooking recipes [Bosselut et al., 2017] or WikiHow articles [Tandon et al., 2020]. However, most datasets primarily focus on changes in location compared to other attributes, such as ProPara [Mishra et al., 2018] and bAbI [Weston et al., 2015]. Modeling approaches in both areas of commonsense explore the generation of explanations in a multi-task setting [Dalvi et al., 2019], the use of external knowledge graph [Tandon et al., 2018], and automatic knowledge base construction to keep a representation of the state of the world and generate novel knowledge [Bosselut et al., 2019, Henaff et al., 2016, Hwang et al., 2020].

2.4.3 Reasoning about Event Chains

While the work discussed earlier focuses on single events implications, open-ended commonsense reasoning tasks may involve multiple events and their causal relations. Reasoning about event chains and causality is one of the most difficult problems, which requires a deep understanding of event-to-event dependencies.

This category typically includes question answering tasks that assume knowledge of commonsense relations and their implications given a sentence and/or knowledge base as context. This line of work includes short questions, such as OpenBookQA [Mihaylov et al., 2018b], CommonSenseQA [Talmor et al., 2019], SWAG [Zellers et al., 2018] and COPA [Roemmele et al., 2011], or questions based on a provided document [Huang et al., 2019] or knowledge base [Clark et al., 2018b].

Chapter 3

Definition Frames: Entity Representations from Definitions

3.1 Introduction

Ontologies have been widely used in lexical semantics to organize and represent knowledge. Carefully built by experts, they contain semantically meaningful information in the form of relations between concepts. However, being manually constructed, they struggle to assimilate new information.

Compared to ontologies, distributed representations are fully automated and can be fine-tuned for new tasks. Despite their exceptional performance, most distributional methods do not have an explicit semantic interpretation. The resulting representations encode a tremendous amount of information, but afford no way to interpret what this information is and how it relates to the concept. Thus, one cannot choose which type of information is useful for a specific task, unless one has a lot of data and resources to fine-tune. Although a few approaches have tried to bridge the gap between semantics and distributed representations [Faruqui et al., 2015, Mrkšić et al., 2017], (1) they only encode information from ontologies, which are not extensible, and (2) the final representations are still not semantically meaningful.

Motivated by these problems, we introduce a novel hybrid representation called **Definition Frames** (DF), which encodes semantic information extracted from definitions. Definition Frames are matrix representations, where each row corresponds to a particular relation based on the Qualia structure proposed in Boguraev and Pustejovsky [1990]. For example, given the word *sun*, the Qualia structure relations would be:

Formal (hypernymy): star
Constitutive (meronymy): solar system
Telic (used for): light, heat
Provinence / Agentive (created via): solar nebula

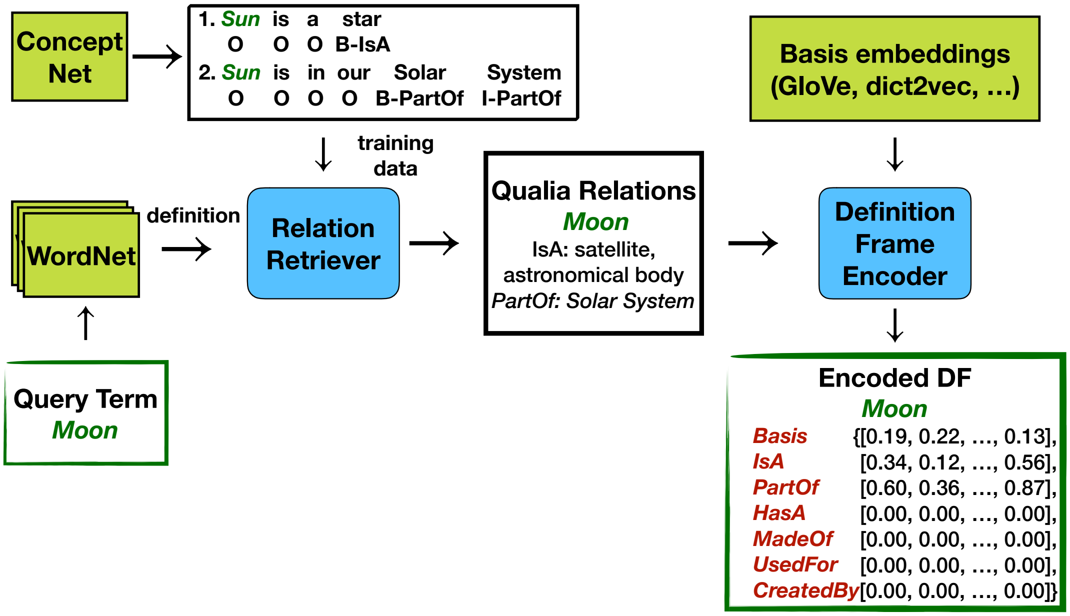

Concept Net
1. Sun is a star
O O O B-IsA
2. Sun is in our Solar System
O O O O B-PartOf I-PartOf
training data
Basis embeddings (GloVe, dict2vec, ...)
WordNet
definition
Relation Retriever
Qualia Relations
Moon
IsA: satellite, astronomical body
PartOf: Solar System
Definition Frame Encoder
Query Term
Moon
Encoded DF
Moon
Basis {[0.19, 0.22, ..., 0.13],
IsA [0.34, 0.12, ..., 0.56],
PartOf [0.60, 0.36, ..., 0.87],
HasA [0.00, 0.00, ..., 0.00],
MadeOf [0.00, 0.00, ..., 0.00],
UsedFor [0.00, 0.00, ..., 0.00],
CreatedBy[0.00, 0.00, ..., 0.00]}

Figure 3.1: Architecture diagram.

Definition Frames are lexical representations of entities and their goal is to improve performance and interpretability by disentagling different aspects of meaning. Towards this end, we use information found in definitions to "split" the meaning with respect to a certain set of independent dimensions, corresponding to relations. The set of the relations used is based on the Qualia structure and they are extracted automatically from the definitions via a domain-adaptation approach. Coming back to our example, if the given definition is *The Sun is a star in our solar system that transmits heat and light*, the set of our extracted relations would include the Formal, Constitutive and Telic dimensions, but not the Provinence, since that we are not given any relevant information. To the best of our knowledge, Definition Frames is the first hybrid representation, combining an explicit structure through semantically meaningful rows, while still being decomposed into distributional vectors.

3.2 Approach

Our framework consists of two parts: the Relation Retriever and the Definition Frame (DF) Encoder. The WordNet definition for any given term is used by the Relation Retriever model to extract the Qualia structure relations. The set of extracted terms pertaining to these relations form the Definition Frame. The DF Encoder encodes this output to a distributed matrix representation, which can be used in downstream NLP tasks.

Qualia Structure The Qualia structure (formal, constitutive, telic, and origin) is defined as the complete modes of explanation associated with an entity [Boguraev and Pustejovsky, 1990, Pustejovsky, 1991]. These relations are inspired by Moravcsik [1975]'s interpretation of the *aitia* of Aristotle, which can be used to define a concept. In fact, several Relation Extraction tasks [Hendrickx et al., 2009, Gábor et al., 2018] contain relations similar to Qualia describing the type (*isA*), structure (*partOf, hasA*) and material (*madeOf*), function (*usedFor*), or provenance (*createdBy*) of a concept.

Qualia	**Relation**	**# Wikipedia Def.**	**# WordNet Def.**	**WordNet Overlap**
Formal	IsA	235	146	59% (87/146)
Constitutive / Structure	PartOf	82	57	2% (1/57)
	HasA	39	33	6% (2/33)
	MadeOf	27	19	5% (1/19)
Telic / Function	UsedFor	59	54	0% (0/54)
Origin / Provenance	CreatedBy	26	17	0% (0/17)

Table 3.1: Annotated Relations for 300 Wikipedia and 150 WordNet definitions. *WordNet Overlap* indicates the number of relations expressed in the definition that were present in the WordNet ontology.

To automatically extract the Qualia structure of a term, we use dictionary definitions, as they uniquely describe a term. We confirm the prevalence of those relations in definitions by annotating 300 Wikipedia and 150 WordNet definitions, chosen at random from nominal terms in WordNet (Table 3.1). We empirically find that WordNet definitions express more relations than the hypernymy (*isA*) and meronymy (*madeOf, partOf, hasA*) relations directly encoded in the WordNet ontology (usedFor and createdBy relations are not part of WordNet ontology). Furthermore, as shown in Table 3.1, we observe that meronymy relations are more prevalent in WordNet definitions compared to the ontology.

Training Data Because there are no definitions annotated with Qualia structure and Relation Extraction datasets [Hendrickx et al., 2009, Gábor et al., 2018] are very domain specific without encoding general knowledge, we deploy a domain adaptation technique. We use ConceptNet to pre-train the Relation Retriever model (Section 3.2) and then fine-tune it on and apply it to WordNet definitions. We fine-tune on a set of 150 manual annotations, since WordNet definitions tend to have more complex sentences than the ones in ConceptNet.

ConceptNet [Speer and Havasi, 2012] is a general purpose ontology that contains relations between pairs of concepts, accompanied by a small source-sentence. Figure 3.1 shows that the Concept-query *Sun* is linked to two sentences (*Sun is a star* and *Sun is in our solar system*) from ConceptNet with the corresponding relations *isA* and *partOf*. The training data for the Relation

Retriever is composed of all ConceptNet source-sentences that contain one of the Qualia structure relations.

Extracting Definition Frames The Relation Retriever uses the WordNet definition of a term to extract words that are related to that term via a Qualia-type relation. The set of extracted relations with their corresponding related words form the **Definition Frame** (DF). More specifically, we define a Definition Frame for a term t as $F_t = \{r_1 : S_1, r_2 : S_2,.., r_k : S_k\}$, where $r_i \in \{$ *isA, usedFor, partOf, hasA, madeOf, createdBy* $\}$ and S_i is the set of words related to t via the relation r_i. For example, to extract the DF for *moon* (Figure 3.1), we use the WordNet definition of *moon* as input. The Relation Retriever extracts the terms that are related to *moon* via a Qualia-structure relation (i.e. *satellite*, *astronomical body* and *solar system*). These terms with their corresponding relations constitute the Definition Frame F_{moon}. More examples of Definition Frames are shown in Table 3.2.

The Relation Retriever uses a BiLSTM model to extract the relations from each sentence. The task is formulated as a sequence tagging problem where we identify both the relation type and the related entities, and optimizes the cross-entropy loss. For model selection, we perform experiments with strong baseline architectures for RE tasks (BiLSTM, BERT-BiLSTM, BiLSTM-CNN). In Table 3.3 we show the performance of the pre-trained Relation Retriever model on ConceptNet test data, which is evaluated on a held-out test set. We observe that the performance is very high, which is our main motivation to fine-tune on the Qualia annotations of WordNet definitions.

The Definition Frame is encoded via the DF Encoder into a matrix where each row w_i corresponds to one of the Qualia relations. The DF Encoder uses an embedding space ($Basis$) to construct each row vector w_i. Note that $Basis$ can be any distributional embedding model. Given a DF F_t, we define w_i as the average of word embeddings from the set of related terms S_i through relation r_i:

$$w_i = \frac{1}{|S_i|} \sum_{s \in S_i} Basis(s)$$

where $Basis(s)$ is the embedding for word s. We include an additional row for the $Basis$ vector of the term itself. This encoding of DF maintains a semantically meaningful structure as each row always corresponds to the same relation. If no terms are extracted for a relation, we use the zero vector of appropriate size. An example of the encoded DF$_{moon}$ is shown in Figure 3.1, where each dimension corresponds to a unique relation like *isA* and *partOf*.

3.3 Experiments

Word-Similarity Task We perform experiments on benchmark word-similarity datasets provided by Faruqui and Dyer [2014]: SimLex999 [Hill et al., 2015], MC30 [Miller and Charles, 1991], RG65 [Rubenstein and Goodenough, 1965], WS353 [Finkelstein et al., 2002] and MEN [Bruni et al., 2012]. Following Agirre et al. [2009], we split them into word-similarity (WS-Sim,

Word 1	Definition Frame, word 1	Word 2	Definition Frame word 2	Relatedness
shore	IsA: land, edge PartOf: body, water	sea	IsA: body PartOf: ocean, salt, water CreatedBy: land	0.86
wool	IsA: fabric MadeOf: hair, sheep	fabric	IsA: artifact MadeOf: weaving HasA: fibers CreatedBy: felting, knitting	0.86
restaurant	IsA: building, people UsedFor: eat	dinner	IsA: main, meal PartOf: day, evening, midday	0.86
day	IsA: time UsedFor: earth, make, complete, rotation	dusk	IsA: time PartOf: day, following, sunset	0.76
dress	IsA: one-piece, garment UsedFor: woman HasA: skirt, bodice	bride	IsA: woman CreatedBy: married	0.76
feather	IsA: light, horny, waterproof, structure PartOf: external, covering	hawk	IsA: diurnal, bird HasA: short, rounded, wings	0.82
orange	IsA: round, yellow, orange, fruit PartOf: citrus, trees	fruit	IsA: ripened, reproductive, body PartOf: seed, plant	0.82
harbour	IsA: sheltered, port, ships UsedFor: discharge, cargo	boat	IsA: small, vessel UsedFor: travel, water	0.76

Table 3.2: Extracted Definition Frames (before encoding) for pairs with high Relatedness score (MEN dataset). The Relatedness score, is the ground truth score, as noted in the original dataset. We observe that the two terms share characteristics of their Definition Frame, like being part of each other's frame or having common related terms.

Model	Pr	Re	F1
BiLSTM	97.6	97.7	97.6
BERT BiLSTM	95.1	95.0	95.1
Stacked-BiLSTM	97.6	97.6	97.6
BiLSTM-CNN	97.4	97.6	97.4

Table 3.3: Relation Retriever on ConceptNet data (held-out test set).

SimLex999, MC30, RG65) and word-relatedness (WS-Rel, MEN) datasets, as they evaluate different semantic affinities. We only consider nominal terms that exist in WordNet and report Spearman's correlation ρ. We perform experiments with three types of embeddings used as $Basis$: GloVe [Pennington et al., 2014], dict2vec trained on Wikipedia [Tissier et al., 2017], and retrofit embeddings [Faruqui et al., 2015] based on GloVe. Since the task comprises of pairs of words without any context, we do not compare against context-based representations.

Ablation Study We perform an ablation study by varying the set of relations used in DF. In this study, both $Basis$ and DF are encoded with dict2vec, as it achieves the best performance (Table 3.4). The goal of this study is to measure how each extracted relation affects the performance of DF in word similarity tasks. The results in Figure 3.2 show that, for similarity tasks, pruning relations sometimes improves performance over both the original DF (with all relations) and the $Basis$ embeddings. However, we observe that DFs consistently have worse performance than $Basis$ in relatedness tasks, particularly in the MEN dataset. As we further discuss in detail in Section 3.3, although DFs capture relatedness, this is not reflected when using the cosine similarity metric directly, since it cannot compare information across different dimensions. For example, consider the pair (*car*, *wheel*). If we compare row-vectors of DF_{wheel} and DF_{car} for each relation separately, the representations are very different. Each Qualia structure relation defining *car* and *wheel* is different for the two terms. However, the Structure dimension of DF_{car} would contain the information that *wheel* is part (meronym) of *car*, thus it should be compared to the $Basis$ dimension of DF_{wheel}.

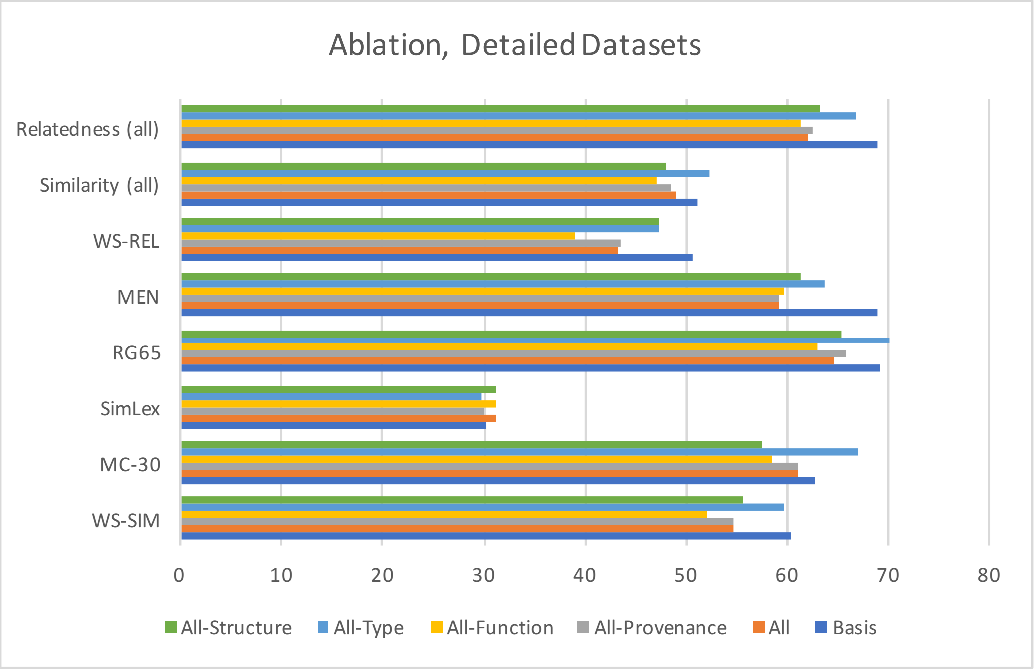


Figure 3.2: Ablation study for merged datasets.

Results To account for the cross-dimension problem described in the previous section, we design a slightly modified version of the previous experiments. We apply a linear transformation with the weights varying according to which type of word similarity (relatedness or similarity) we are measuring.

Datasets	**GloVe**				**Dict2vec**				**Retrofit**			
	Basis	Basis*	**DF**	**DF***	Basis	Basis*	**DF**	**DF***	Basis	Basis*	**DF**	**DF***
Similarity CV	0.39	0.50	0.35	**0.53**	0.53	0.52	0.45	**0.56**	0.44	**0.59**	0.35	0.56
Relatedness CV	0.68	0.77	0.38	**0.80**	0.71	0.76	0.61	**0.79**	0.67	0.78	0.51	**0.80**
MEN-test	0.70	0.79	0.56	**0.81**	0.73	0.74	0.62	**0.79**	0.71	0.79	0.53	**0.80**

Table 3.4: Spearman's correlation for embeddings before and after the linear transform. All cross-validation (10-fold) experiments have p-value $p < 0.01$.

This allows us to: (1) give more weight to more important relations and (2) compare the representations across different Qualia structure relations.

Using a linear transformation allows us to recover the initial DF representation from its transformed counterpart, which is important in order to maintain the semantic interpretability of DF (i.e. which words are related to t and how). Thus, given DF_t for a term t, we get $DF^*_t = W \times DF_t + b$, which we use in our experiments. The parameters W, b are learnt separately for similarity and relatedness tasks, since different relations and cross-relation comparisons have varying importance for the two tasks. The training objective for the linear transformation is the minimization of the mean squared error between the cosine similarity of the transformed representations and the normalized ground truth similarity score. For fair comparison, we also apply a linear transformation to the baseline $Basis$ by learning parameters W_{basis}, b_{basis} as described above for DF. For our experiments on similarity and relatedness datasets we use 10-Fold cross-validation and report the average performance, while on MEN we use the provided split into training and test data (it is the only dataset with a train/test split).

Our results show that Definition Frames achieve the best performance, compared to any of the baselines. In Table 3.4 we compare the performance of the Basis embeddings before and after the linear transformation ($Basis$ and $Basis^*$), with the Definition Frames (DF and DF^*). DF^* benefits much more of the dimension weighting and achieves better results compared to $Basis^*$, particularly with GloVe embeddings. Furthermore, we observe that Relatedness datasets (including MEN) gain the greatest advantage from the linear weighting. This lines up with our previous hypothesis, since the relatedness task requires more cross-relation comparisons (DF_{car} vs DF_{wheel}).

Qualitative Analysis One of the distinguishing features of DFs is that they are semantically interpretable. Beyond determining whether two terms are related, we find that DFs can be used to infer *how* they are related. We perform a qualitative analysis on 100 randomly selected terms from the MEN dataset that have high relatedness score (higher than 35 out of 50). The goal of this study is to assess whether we can use the explicit structure of DFs to predict the type of the relation between two terms.

We conduct a Mechanical Turk study, where we present (1) the pair of related words, (2) their corresponding definitions and (3) a Qualia structure relation, in the form of question. We phrase the annotation task as a binary question such as "*Is an aquarium created by a fish?*". We include all possible Qualia structure relations for each of the 100 pairs of related words. We ask three annotators to annotate each sample (1200 questions, each annotated three times, for a total of 3600 annotations).

To identify the most probable relation between two terms t_1 and t_2 using the encoded DF, we conduct a set of row-to-row comparisons. We measure the cosine similarity of each row of DF_{t_1} with $Basis(t_2)$ and vice-versa DF_{t_2} with $Basis(t_1)$. The relation corresponding to the row with highest cosine similarity is taken to be the most probable relation. We test if the relation predicted by the DFs is correct according to humans. By taking the majority vote of the annotations, we find that 77% of the extracted relations are considered valid by the workers. Furthermore, 54% of the relations were considered accurate by all three annotators and the inter annotator percent agreement is 60% over the 1200 relations. In Table 3.5, we show the accuracy per relation of the Definition Frames extracted relations, when all three MTurk participants agree.

Qualia	**Relation**	**Agreement %**
Formal	IsA	0.43
Constitutive / Structure	PartOf, HasA, MadeOf	0.79
Telic / Function	UsedFor	0.50
Origin / Provenance	CreatedBy	0.25

Table 3.5: Accuracy per relation.

3.4 Conclusion & Future Work

In this chapter we present a hybrid representation of entities whose dimensions, grounded in lexical semantics, can also be employed to form a structured distributed representation. Our goal is to create Definition Frames that disentangle the different aspects of meaning of a word into independent, interpretable dimensions. In Definition Frames we use a set of dimensions corresponding to the Qualia structure relations, which are shown to be sufficient to represent the meaning of concrete entities. One could, however, use any other relations if they are sufficiently clearly defined. This is necessary since Qualia structure is not able to represent words that denote abstract ideas (e.g., democracy) or eventualities (events, states, processes), since it offers no fixed dimensions that capture the main aspects of their meaning.

As we show in this chapter, Definition Frames capture different types of semantic overlap (both relatedness and similarity) and improve performance on word similarity tasks. We further examined how each dimension affects the performance due to data biases, which drove our approach of using a linear re-weighting of each relation. Finally, we demonstrated the explainability of Definition Frames via a human study showing that they provide valid insights on how terms are related.

There are several directions where future work could achieve further improvements via the use of Definition Frames. Firstly, in our experiments we only used static embeddings as the basis. However, given that Definition Frames work with any distributed representation, contextual word embeddings may be a better basis, since they are able to express more information. A second direction is to modify Definition Frames so they can disambiguate between different senses of the same word. In the current version of Definition Frames, we are given a single definition of the word that we want to represent. However, we could access all possible definitions of the word senses by using WordNet synsets. Then, instead of creating a matrix representation, one can create a tensor similar to Definition Frames where the third axis corresponds to the different senses of the word.

As shown by Scarlini et al. [2022], we can learn definitions of concepts from both visual and language input. Although their work involves only definition generation, it establishes that it is possible to learn about physical objects only from images. Given that images contain information about attributes complementary to language, an interesting future work direction is to use visual input to enhance Definition Frames with such information.

Chapter 4

Sentence Representations: Balancing Distribution Shifts in Data

4.1 Introduction

Sentence representations are an integral part of many NLP applications, as they encode the information contained in a sentence. Initial approaches on sentence representations used various combinations of lexical representations, such as the average embedding of the words in a sentence. However, such methods ignore syntax and typically lose a lot of important information. Instead, recent work focuses on learning an encoding of the entire sentence to represent meaning via unsupervised (i.e., contextualized embeddings) or supervised methods.

Similar to the lexical representations studied in Chapter 3, the meaning of a sentence has several aspects and, based on a specific context and task, we might be interested only on a certain part of it. The aspect of meaning that is most relevant to the data or task is what typically is encoded in a sentence representation. However, the fact that representations heavily depend on the training data and task means that they might not generalize well in out-of-domain scenarios, since a different aspect of meaning might be needed. While this might not be an issue when we have sufficient data to fine-tune the representations and learn to extract that different aspect of meaning, in many cases only limited or no in-domain data is available.

In this chapter, we discuss how to encode only information relevant to a task, when there is **a covariate shift** across our data. By covariate shift we mean a *systematic difference between training and test data from an underlying cause*. Our approach to this problem is to recognize the cause of the shift and teach our representations to remove information relevant to that cause. We apply our method on a sentence classification task in Crisis NLP, where we must classify tweets from a disaster event with respect to their importance. This is a real-time setting, where we have labeled data from previous events but no data for the ongoing disaster event, leading to a systematic difference across the data caused by the event each tweet refers to. Since the label of each tweet also correlates with local features from particular events (i.e., location, named entities), a model is prone to short-cut learning, where it learns these local features that do not generalize across events.

4.2 Background

Effective management of crisis situations like natural disasters (e.g., earthquakes, floods) or attacks (e.g., bombings, shootings) is an extremely sensitive and complex phenomenon that requires efficient coordination of people from multiple disciplines along with proper allocation of time and resources [Tapia et al., 2011a, Maitland et al., 2009]. Given that we live in the era of information and social media, filtering important nuggets of information from real-time data and using them into decision-making constitutes a crucial research direction [Tapia et al., 2011b].

Critical information from social media is found only in small amounts. Hence it is difficult to extract and analyze the data stream, since it is impossible to manually process the amount of information shared in social media in real-time. Therefore, it is important to detect data that contains useful information for decision-making and automatically extract it [Sutton et al., 2008, Palen et al., 2010]. Even though sentence classification is a well-studied NLP problem, common approaches do not bring the expected results [Reuter et al., 2018].

The main reason why common approaches fail is the lack of in-domain data [Mccreadie et al., 2019, Hiltz et al., 2014]. Most emerging crisis are unexpected and data analysis must be done real-time, within a small time-frame [Plotnick et al., 2015]. Even if we might have high quality annotated data from previous similar crisis situations, we will not have data from the emerging event that we want to classify. For example, let us assume an earthquake in Seattle happens right now. Although we may have annotated data from a previous earthquake in Los Angeles, most of the parameters would be entirely different (e.g., location names, damages, times, etc) since the cities and populations differ. Furthermore, because some of those parameters might indeed play an important role in the classification of a tweet from the specific event (e.g., the epicenter of the Seattle earthquake), a traditional model would learn them as important features. This creates a model that learns shortcuts and does not generalize on future events, since we cannot fine-tune properly on-the-fly. On the other hand, some other features are actually important in the general setting (e.g., severity of the earthquake, casualties). The problem we tackle in this work is how to construct a model that learns generalizable representations, instead of relying on the local features seen in training data.

We explore a technique that helps a neural model to distinguish and discard information that is related only to specific events, resulting in a more generalizable model with improved performance on unseen events without any fine-tuning. Since the main task is to classify the importance of the information contained in a tweet (criticality), we use an adversarial classifier that intends to learn which specific event the tweet refers to, hence remove the event specific bias through a reversal gradient. Our experiments represent a real-life crisis management scenario, where the model is evaluated on a new incoming event through a leave-one-out experimental setup, and show substantial improvement over baseline classification methods.

4.3 Approach

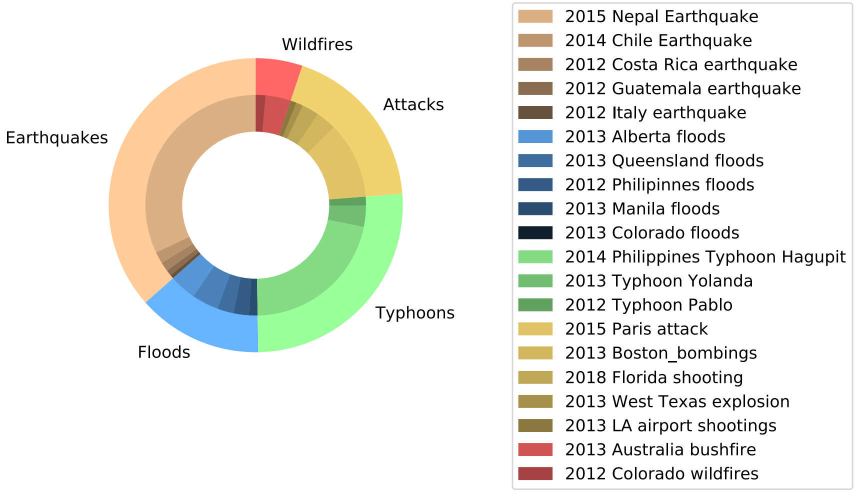


Figure 4.1: Crisis NLP Dataset Distribution. Outer circle: Color defines each of the event categories. Inner circle: The shade of colors describe the different events within a category.

In this work we use data from the TREC 2018 Incident Streams challenge [tre], which contains labels on criticality and information types [Mccreadie et al., 2019]. They define criticality as a score to identify posts that need to be shown to an officer immediately as an alert. The raw data and information about the specific event each tweet belongs to is extracted from the Crisis NLP [Imran et al., 2016a] dataset, which contains tweets in English from disaster events that occurred during 2012-2018. The crisis events in our dataset can be split into five main groups: earthquakes, floods, typhoons, wildfires and attacks. In Figure 4.1, we show that the data mainly consists of multiple earthquake, flood, and typhoon events, only two wildfire events, and five diverse attacks originated by humans.

4.3.1 Data Description

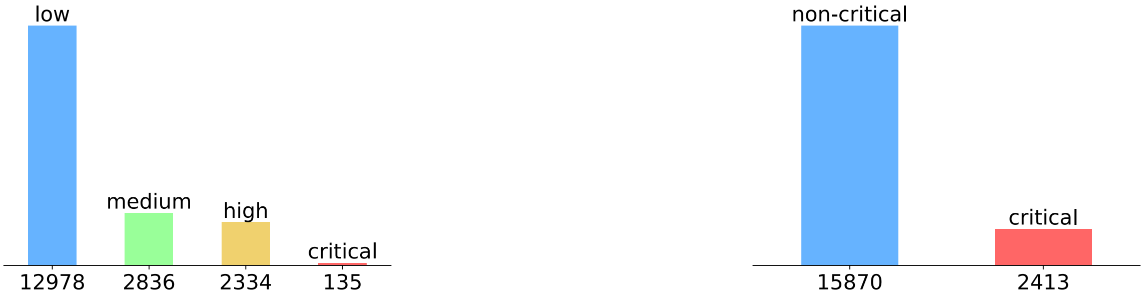
In our experiments we use a labeled subset of the data formed by 18,283 tweets which are labeled into four categories according to their level of importance for the authorities: low, medium, high, and critical. The distribution of the labels is highly skewed towards the low and medium labels as shown in Figure 4.2a. These types of tweets do not provide important information for decision-making during a disaster event. Since we are aiming to sieve the actionable tweets, we grouped together the low and medium labels as *non-critical*, and the high and critical as *critical*. The new distribution of the data after relabeling is shown in Figure 4.2b. As we see in the examples shown in Table 4.1, the latter have actionable information for the authorities, first responders, and population on distress.

Label	Event	Tweet
non-critical	2014 Philippines Typhoon	Good morning! keep safe everyone!
critical	2013 Colorado Floods	RT: Seek higher ground immediately wall of water coming down Boulder Canyon move away from Boulder Creek
non-critical	2013 Boston Bombings	I am honestly sick who could be so disgusting to do this to someone we will get answers and find you #prayforboston
critical	2015 Nepal Earthquake	RT: News at epicenter of Nepal tragedy local church mission offers help!

Table 4.1: Examples of *critical* and *non-critical* Tweets

4.3.2 Data Pre-processing

Our target dataset comes from Twitter. Therefore, we performed a series of pre-processing steps for data-cleaning. First, we removed links, hashtags and mentions, since most of them are event specific. We also removed non-English words to reduce the noise. Next, we removed all non-English characters and emojis. Finally, we observed that many times white spaces were omitted between words, which resulted in multiple words being clustered as a single token. To solve that, we stripped the text from punctuation marks and, subsequently, used a heuristic for word segmentation, where we split the token into the least number of possible English words via greedy search.



(a) Original labels

(b) New labels

Figure 4.2: Dataset label distribution. (a) Label distribution of original dataset, (b) Distribution of the labels after grouping {low, medium} as non-critical and {high, critical} as critical

4.3.3 Models

Our experimental setup consists of a dataset $\mathcal{D}$ composed of tweets $t_1, ..., t_n$ and two sets of labels; $y_{e_1}, ..., y_{e_n}$ representing the event that the tweet belongs to, and $y_{r_1}, ..., y_{r_n}$ representing the

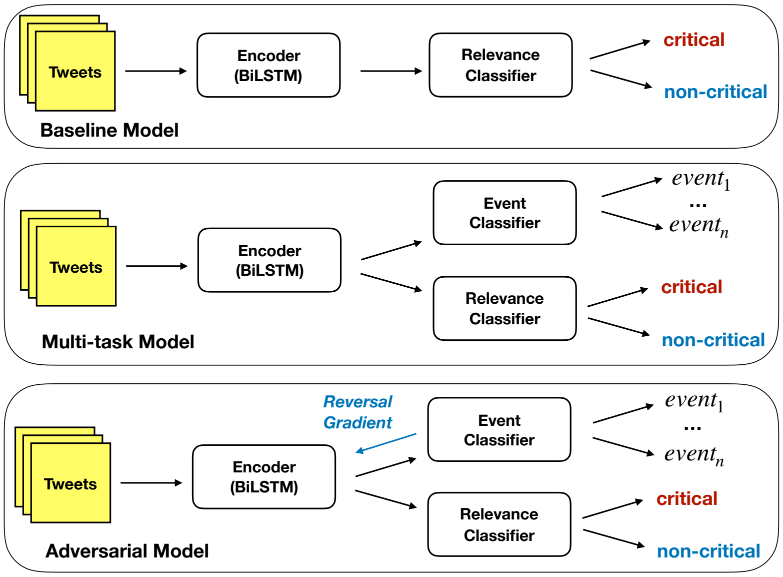


Figure 4.3: Evaluated Model Architectures. The adversarial model was compared against the baseline and multitask models to show the removal of event specific information.

importance of the tweet, where $y_{r_i} \in \{non\text{-}critical, critical\}$. For this task we want to find the optimal classifier f for predicting labels y_{r_i}. In this work we compared three models to measure if an adversarial training contributes to the detection of *critical* tweets on unseen events.

Our main hypothesis states that an adversarially trained model removes event-specific information, while focusing on features that determine how important the tweet is. For our experiments we compare the adversarially trained model against a binary classifier and a multi-task model. The comparison between the multitask and the adversarial models helps us evaluate whether the explicit removal of event-specific information benefits the relevance classifier or if using a model that jointly learns both tasks suffices.

Baseline Model

In our baseline model setup, a tweet t_i is a sequence of word embeddings $w_1, ..., w_{m_i}$ which are encoded through an LSTM Graves et al. [2013] encoder h. Then the generated embedding $h(t_i)$ is fed to a binary classifier c_r that learns to predict if the tweet is *critical* or *non-critical*. The architecture of this model is shown in Figure 4.3.

The training loss $\mathcal{L}$ used across all the models and experiments is cross-entropy. The optimization of the baseline model is described in Equation 4.1.

$$\operatorname*{argmin}_{h,c_r} \mathcal{L}(c_r(h(t_i)), y_{ri}) \tag{4.1}$$

Multi-task Model

The multitask learning setup described by Caruana [1997] aims to improve the performance of a model by learning multiple tasks at the same time. Since the dataset is divided per disaster event, we take advantage of this information given by the structure of the dataset, and define event detection as the second learning task along with the criticality classification. Hence, the multitask model adds an event classifier c_e on the encoding of the incoming tweet $h(t_i)$ which trains simultaneously with the classifier c_r, as seen in Figure 4.3.

The optimization procedure for this model is described in Equation 4.2.

$$\underset{h,c_r,c_e}{\operatorname{argmin}} \mathcal{L}(c_r(h(t_i)), y_{ri}) + \mathcal{L}(c_e(h(t_i)), y_{ei}) \tag{4.2}$$

Adversarial Model

The adversarial model used in this work follows the adversarial training setup proposed by Goodfellow et al. [2014], Ganin et al. [2016], and Xie et al. [2017]. In essence, the adversarial model is similar to the multitask model except for the addition of a gradient-reversal layer g_λ [Ganin et al., 2016] between the encoder h and the event classifier c_e. The gradient-reversal layer during a forward step works as the identity function $\mathcal{I}$, but during the back-propagation step the gradient from c_e is reversed and scaled by a value λ. In our work, we intend to achieve domain adaptation from previous events to a new incoming event by minimizing the information related to previously seen events provided by c_e, while maximizing the information gain obtained from classifier c_r, as described in Equation 4.3.

$$\underset{h,c_r,c_e}{\operatorname{argmin}} \mathcal{L}(c_r(h(t_i)), y_{ri}) + \mathcal{L}(c_e(g_\lambda(h(t_i))), y_{ei}) \tag{4.3}$$

4.4 Experiments

For our experiments we used two of the main popular word embeddings to represent the tokens of the tweets in the target dataset: GloVe [Pennington et al., 2014] embeddings, and BERT [Devlin et al., 2019] embeddings.

We used the 100-dimensional GloVe embeddings pre-trained on Wikipedia and Gigaword, which were made publicly available by the authors[31]. For extracting BERT embeddings we used the Python package *bert-embeddings*[42] as we built the networks for our experiments in PyTorch. This package offers a pre-trained 768-dimensional hidden state transformer model with 12-layers and 12-headed attention. In our experiments, the BERT model was frozen with no fine-tuning during training.

[13] https://nlp.stanford.edu/projects/glove/

[24] https://github.com/imgarylai/bert-embedding

Event Type	**Embedding**	**Model**	**Macro F1**	**Non-Critical F1**	**Critical F1**
Earthquakes	GloVe	Baseline	0.6432	0.9082	0.3782
		Multitask	0.5890	0.8960	0.2819
		Adversarial	**0.6602**	0.9170	**0.4034**
	BERT	Baseline	0.6138	0.9062	0.3213
		Multitask	0.5844	0.8863	0.2826
		Adversarial	**0.6154**	0.8888	**0.3420**
Floods	GloVe	Baseline	0.6010	0.8674	0.3346
		Multitask	0.6130	0.8679	0.3581
		Adversarial	**0.6326**	0.8454	**0.4198**
	BERT	Baseline	0.6145	0.8834	0.3455
		Mulitask	0.6062	0.8793	0.3331
		Adversarial	**0.6403**	0.8642	**0.4164**
Typhoons	GloVe	Baseline	0.5714	0.8965	0.2462
		Multitask	0.5832	0.8961	0.2702
		Adversarial	**0.5887**	0.8916	**0.2858**
	BERT	Baseline	0.6249	0.9189	0.3310
		Mulitask	0.6291	0.9091	0.3491
		Adversarial	**0.6302**	0.9086	**0.3517**
Attacks	GloVe	Baseline	0.6049	0.9047	0.3052
		Multitask	0.5994	0.8917	0.3071
		Adversarial	**0.6056**	0.8975	**0.3137**
	BERT	Baseline	0.5744	0.8840	0.2649
		Multitask	**0.6165**	0.9009	**0.3322**
		Adversarial	0.5492	0.8511	0.2472

Table 4.2: Event based zero-shot test results. The best model per disaster type is highlighted with the color assigned to the disaster type. The best model per embedding type is highlighted in bold.

Through all of our experiments the tweet encoder h is an LSTM with two layers. Each of the LSTMs have a hidden dimension of 100, which results in a tweet embedding of size 200. Both classifiers c_r and c_e are linear layers with output size 2 and the number of events per experiment, respectively. During our initial experimentation, we set the gradient-reversal layer scaling value lambda to different values within the range $[0.1 - 10]$. The most stable result throughout the experiments was obtained with $\lambda = 1$.

The models were trained using the Adam optimizer [Kingma and Ba, 2014], with an initial learning rate 0.01, batch size 16 and trained for 40 epochs. We employed dynamic batching by padding each batch to the sequence length of the longest sample in the batch.

To test the performance of the model at every epoch we calculated the micro F1 on the *critical* class from c_r and considered as the best model the one which showed the highest critical-F1 score, since for disasters it is important to recall as many *critical* tweets with the highest possible precision.

4.4.1 Model Evaluation

Since we intend to evaluate the models for a real-life scenario, we use data from each disaster type separately (e.g., model trained and tested only on flood events), to perform an analysis in a disaster-based zero-shot learning scenario simulating an incoming unseen event. To achieve this, the training data consists of all the events of the same disaster type except one, as it is used for testing the model. We generated n splits for each event type, where n is the amount of events per event type. We evaluated the three models on each split obtaining the macro-F1 and the micro-F1 scores from the c_r predictions. Finally, we calculated the mean of these metrics, which we can see in Table 4.2. The best models for each event type are highlighted in the representative color of the event, as shown in Figure 4.1.

Since we follow a leave-one-out testing procedure, we could not include the wildfires event type since this category only has two instances. This makes it impossible to train the multitask and adversarial models on this type of event.

Our experiments show an improvement of the F1 score for all disaster events that use adversarial training except for the attacks group, where the improvement is not consistent with the rest of the events. The earthquake and flood events show a significantly better performance of the adversarial model when compared to both the baseline and the multitask model. For the typhoon events the multitask model improves slightly over the baseline, but the adversarial model is the best for both embedding types, while BERT has better results than GloVe by a large margin.

Most similar to our setting, Nguyen et al. [2016] performs an experiment in an online training scenario using the Nepal 2015 Earthquake as test set, while more than 10,000 tweets from the dataset are used for pre-training the model. Their work reports an AUC of 0.73 at the beginning of the event, which would be comparable to our zero-shot learning scenario. To compare our model to their work, we used the data split where the Nepal earthquake was left out for testing the model. On this data split, the adversarial model using BERT embeddings obtains an AUC of 0.62 for the critical class while training with only 815 tweets from all the other earthquake events.

4.4.2 Event Types Data Mix

In Figure 4.1, we observe that the attack events group consists of diverse types of events such as shootings, bombings, and explosions. Even though all of those events contain violence-related incidents, the adversarial model with BERT embeddings has lower performance than the baseline and the multitask learning model, as shown in the results in Table 4.2. Our hypothesis is that the adversarial model fails to remove the event-specific biases in the Attack group, because of the

Model	**Macro F1**	**Non-Critical F1**	**Critical F1**
Baseline - GloVe	**0.5376**	0.7602	0.**3150**
MultiTask - GloVe	0.5331	0.7529	0.3133
Adversarial - GloVe	0.5157	0.7428	0.2885
Baseline - BERT	0.5593	0.7602	0.3584
MultiTask - BERT	**0.5625**	0.7558	0.**3692**
Adversarial - BERT	0.5539	0.7500	0.3578

Table 4.3: Mixed flood and typhoon test results

mixture of different event types. A potential solution to this problem would be to include more events to facilitate the disentanglement of the Attacks group.

To test this hypothesis, we created a synthetic event type where we mix flood and typhoon events, since both are disasters that would result in flooded cities and towns. We repeated the same experimental procedure by leaving out one event for testing and obtained the mean scores across all splits, as reported in Table 4.3. The results from this experiment verify our hypothesis that the adversarial training of the classifier is sensitive to the entanglement of events in the training data. This supports our claim on why we have low performance on attacks and highlights the importance of not mixing different event types when training under an adversarial setup.

4.5 Qualitative Analysis

We took a deeper look into our experimental results by comparing which patterns are learnt by the adversarial model but not the baseline. For this analysis, we focused on flood and earthquake event types, as they show the greatest difference in F1 score between the baseline and the adversarial model.

4.5.1 Critical Detection Comparison

For the first part of the qualitative analysis, we examined tweets where the baseline and the adversarial models disagree upon. We looked at both critical and non-critical tweets in order to find common patterns where the models fail. In Table 4.4 we show some examples of tweets where the baseline model failed, but were correctly classified by the adversarial model. The examples used come from the Philippines flood (performance shown in Table 4.5).

A consistent pattern observed for the critical tweets is that they mostly contain information about a need for emergent help or a situation currently happening. Furthermore, we see a strong sentiment of despair, where we may assume that the users are directly affected by the event. On the other hand, if we look at the non-critical tweets that were incorrectly classified as critical by the baseline, they mostly contain location information and named entities. These examples validate

True Label	Tweet Text
Critical	rt flood in the ust hospital is now on the 2nd floor no food for the patients & staff pls help ...
Critical	rt please help rt rt those who are in u erm the flood is now goi ...
Critical	ust hospital and u erm in need of immediate help u sts morgue is flooded ue rms nursery is near being flooded please please
Critical	philippine flood fatalities hit 23
Non-Critical	metro manila flood updates nlex is now north luzon express river pls rt and spread
Non-Critical	ndr rmc nearly 50 of metro manila submerged in floodwater due to heavy monsoon rains
Non-Critical	rt lets all pray for those who lost their homes and now living in cold and starving ...
Non-Critical	rt pal passengers to/from manila who are unable to take their flights due to floods may rebook their tickets with rebooking c ...

Table 4.4: Examples captured by the adversarial model (true-positives), but not the baseline (false-negatives).

that our approach learns representations focusing on the aspect of meaning relevant to the task, by removing information related to specific events.

Model	Macro F1	Non-Critical F1	Critical F1
Baseline - BERT	0.5844	0.8413	0.3274
MultiTask - BERT	0.5875	0.8766	0.2985
Adversarial - BERT	**0.6535**	0.8832	**0.4238**

Table 4.5: Test results on 2012 Philippines Flood

4.5.2 Model Comparison via Saliency Maps

For the second part of our analysis, we use saliency maps to visualize the relevance of each word in a tweet for the models. We selected tweets that contain named entities (e.g., locations, names) or information that is generally important to classify a tweet, such as casualties. For this part, we only used GloVe embeddings, since BERT is context-based and each embedding may encode information from the rest of the tweet.

In order to construct the saliency map, we use back-propagation to estimate the first-order derivatives from each word, as a measure of their contribution to the model's decision. This strategy was adopted from the vision community [Erhan et al., 2009, Simonyan et al., 2013], and recently adapted in NLP research [Li et al., 2016].

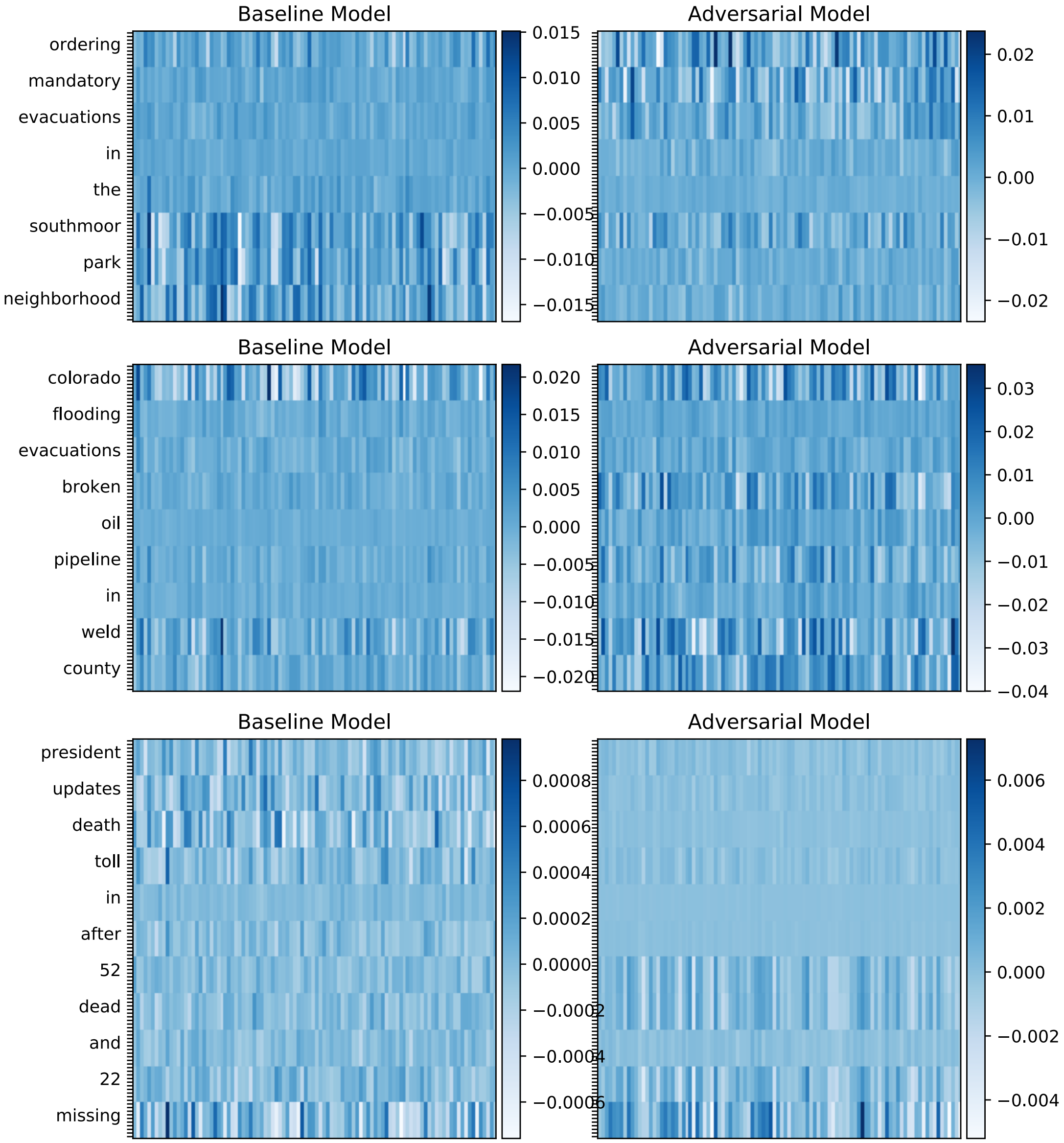


Figure 4.4: Saliency map visualization of tweets with strong event-bias.

In Figure 4.4 we visualize the saliency map of each word embedding for the baseline and adversarial models. The higher the absolute value of the first-order derivative (dark blue and white), the more important role it plays into the classifier's decision. We observe that, for the first and second sentences, the baseline puts more weight on the location, which is a strong event-bias since it includes information only for a particular event and not a disaster type (e.g., floods). On the other hand, the adversarial model focuses more on important sub-events, like *mandatory evacuations* and *broken pipeline*, which we desire to capture in a zero-shot scenario, and is generally ignored by the baseline model. We further observe a similar trend for the third sentence, where the baseline gives mostly uniform weight with a small focus on *president updates death*, while the adversarial model focuses more on generally informative text that describes casualties.

Through this analysis we observe that the two compared models encode different information even for identical sentences, as the baseline model is biased towards event-specific features and words, while the adversarial towards words that relate with an event-type or any disaster event. This shows that sentence meaning changes and depends on the data and task, since context drives which aspect of meaning we want to retain.

4.6 Conclusion

In this chapter we discuss the different aspects of meaning of a sentence and the dependency of the information we want to retain on context. We show how this manifests in a real-time application where there is unavoidable covariate shift across training and test data due to an underlying cause. In our scenario, the cause of this shift is the disaster event that each instance refers to, which leads to sentence representations focused on the specific event. Although such representations are not wrong, they do not generalize in the context of our task.

To address this problem, we propose an approach where we can use our knowledge about the task to drive sentence representations and improve performance by balancing the distribution shifts across training and test data. In our approach, we compare an adversarially trained model against a baseline classifier and a multitask learning model. The main task for all the models was to predict if a tweet is *critical* or *non-critical* over four types of disaster events: earthquakes, floods, typhoons, and mass attacks in public spaces. We presented a thorough analysis on how a simple classification model trained on crisis event data can be improved through adversarial training. Our results show how the addition of an adversarial network removes the bias from specific events, allowing the network to put more attention in disaster related information rather than specificities of a particular event.

Chapter 5

Representing Events via their Sub-events

5.1 Introduction

The purpose of Chapter 5 is to study event representations and the relations of an event to the participant entities and its sub-events. Although the chapter does not directly investigate novel event representations, it provides knowledge and techniques essential to understand the problem of event implications, functioning as a precursor to Chapter 6 and motivating future work towards this direction.

Most methods in NLP rely on the concept of frame semantics to represent and detect events in text. Semantic frames describe a schema to represent events based on their relations with entities from the same sentence, which is particularly important in order to decompose and analyze event implications. However, events are complex semantic units that are related not only to entities, but also to other events. For example, two events might be linked via a causal or temporal relation or also a sub-event relation. Such relations are essential for reasoning tasks and are extremely common in problems where we need to combine information across sentences and documents (e.g., event scripts).

Chapter 5 focuses on representations of large-scale, complex events via their sub-events. To obtain a deep understanding of the semantics behind an event, one has to take into account not only its relations to entities, but also how it interacts with other events. Such is the property of events to function as components of more complex events, called *sub-events*. Many events consist of smaller sub-events that describe a different aspect of their semantics. For example, the event *restaurant dining* consists of the sub-events *eating* and *paying the bill*. In such scenarios, the larger event and its sub-events depend on each other; the sub-event would not exist without the larger event and the larger event would not be the same if we alter any of its sub-events. Although not all sub-events have a causal connection to the larger event or goal, some of them are essential for the completion of the goal. This need for understanding sub-events and their relations to a larger event / goal is particularly prominent in reasoning tasks based on procedural text. To answer complex reasoning questions in such tasks, a model has to extract reasoning patterns by understanding event semantics and learning event dependencies.

Towards this end, we propose a framework that combines information across a group of textual sources related to a given large-scale event and extracts its important sub-events. Our framework is evaluated in crisis NLP scenarios, where we extract sub-events from a large volume of tweets related to a given large-scale disaster.

5.2 Events in a Sentence

5.2.1 Background

Event Detection Event Detection in NLP is defined as a sequence labeling problem, where we have to identify the snippets of text that correspond to an event mention and, sometimes, also classify the extracted event to a predefined event type. An event mention consists of the **event trigger**, a word or phrase that denotes an action, and the **event arguments**, a list of participant entities that are related to the event nugget via a specific relation. Given a sentence s, the goal is to identify all words or phrases that correspond to some event trigger and classify each trigger to an event type. Typically most NLP methods study event detection in the setting of a single sentence, where all information about the event mention can be found in the same sentence. That means that we can only extract events explicitly stated in text, without combining information across sentences or documents. As we study in this chapter, this simplified assumption is often not true, as events are complex units that depend on their participants and other events.

Frame Semantics Frame semantics is a linguistic theory developed by Fillmore and Baker [2001], relating lexical semantics to encyclopedic knowledge. The basic idea is that one cannot understand the meaning of a single word without knowledge of all the concepts that are related to that word. This knowledge takes the form of a **semantic frame**; a structure of relations linked to the word that we want to represent. For example, consider the word *run*. According to frame semantics, in order to fully understand the meaning of this word, one must know about the situation of *Motion*, which involves an *agent*, *time*, *source* and *destination*. Thus, a word activates, or evokes, a frame of semantic knowledge relating to the specific concept to which it refers.

Frame semantics assume the existence of an underlying ontology, where words belong to concept classes, based on hypernymy and synonymy relations. Based on this ontology, each concept class corresponds to a specific semantic frame, which is activated every time a word belonging to the same concept class is found in text. For example, consider the sentence *John runs to school*, where we want to represent the word *run*. This event trigger belongs to the higher-level concept of *Motion* and, thus, inherits the semantic frame from that concept. If this semantic frame includes the following relations f_{motion} = { *agent*, *source*, *destination*, *time*}, then we are trying to fill-in the arguments for each of these relations, if possible.

5.2.2 Event Detection Using a Frame Semantics Parser

Similarly to frame semantics, events also consist of a set of predefined relations for which we want to fill their slots based on information provided in text. Although some relations are common across all events (e.g., time), others depend on the semantics of the event trigger (e.g., source and destination). Despite their similarities, event detection and frame semantics are not exactly equivalent; frame semantics provide the theory of representations for any type of semantic units (i.e., events, entities and relations) following an approach close to the lexical aspect of language. Thus, in order to take advantage of frame semantics tools for event detection tasks, one needs to somehow ground the problem of event detection on the frame semantics theory.

As shown by Spiliopoulou et al. [2017], a frame-semantic parser can become a useful tool for event detection if used in combination with a filtering method. Their approach is to manually construct a mapping from the domain-specific ontology used in an event detection task to FrameNet [Baker et al., 1998a], a taxonomy of semantic frames. Then, they use this mapping to filter the output of Semafor [Das et al., 2014], a semantic-frame parser system that is grounded to FrameNet. Finally, by using a simple random forest classifier, they rank and further filter the initial low-precision, high-recall output of Semafor.

Their proposed methodology results in a system that, despite its simplicity, has competitive performance to other state-of-the-art models of its time. Unlike previous approaches that focus solely on elaborate deep learning techniques, this work shows that it is feasible to achieve good results by leveraging richer semantic representations of events and combining them with machine learning methods.

5.3 Events Across Documents

Social media are widely used for informing humanitarian aid efforts in crisis events [Nazer et al., 2017, Reuter et al., 2017]. During a large-scale crisis event, there is a large set of smaller events in duration and impact that are essential components of the larger event, the *sub-events*. Detecting important sub-events that occur during a crisis (e.g., road blocks, people trapped) can aid authorities to prevent and respond to urgent situations (e.g., rescue efforts) [Nazer et al., 2017]. However, this requires connecting information from multiple posts as they contain repetitive or complementary information which needs to be aggregated (e.g., the number of trapped people and their location) for disaster response.

Several approaches in crisis NLP aggregate information across multiple tweets in the form of clusters, where each cluster is considered a sub-event [Abhik and Toshniwal, 2013, Pohl et al., 2012, Arachie et al., 2020]. However, these methods have several shortcomings. First, the output clusters may not refer to a single sub-event, but to a list of sub-events that share similar information types. For example, consider the tweets *25 people killed in Everest base-camp* and *200 people killed in Gorkha*. They contain the same information type (i.e., number of people killed), but clearly refer to two different sub-events. This results in large, non-interpretable clusters that lack cohesion [Jiang et al., 2019]. Second, most work ignores or uses heuristics to model the temporal dependencies of sub-events, without explicitly modeling time-sensitive information that gets updated, such as the number of injured people. Third, tweet content is represented by shallow semantics, such as bag-of-words or verb-noun pairs. Such representations miss information that distinguishes between different sub-events of the same information type and are inadequate to model semantic dependencies across sub-events. To provide an example, consider the Nepal earthquake in April 2015. In Table 6.7, we show tweets referring to a deadly avalanche in mountain Everest which was triggered by the earthquake. We may extract *100 climbers were trapped in camp 1 and 2* from one tweet and *the route to camp 1 and camp 2 was completely destroyed* from another. Although these two tweets refer to different sub-events, they are related and an event extraction framework would benefit from modeling their dependencies.

Representing events in text is a complex problem. Most work on event detection relies on the **semantic frame** theory [Fillmore, 2008], as explained in Section 5.2. In this work, we use the same notion of event representations. Thus, we can distinguish different sub-events even if they have the same predicate and/or partially share entities, as in the example discussed earlier.

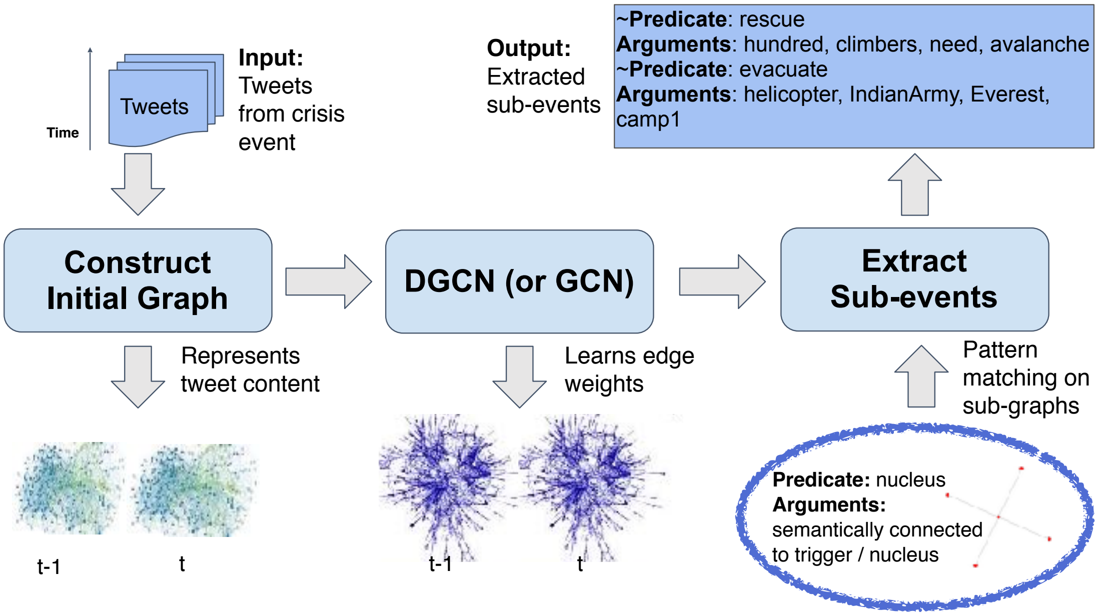
1. #NepalQuake avalanche kills 8 at Nepal's Everest base-camp
2. Obliterated Everest basecamp where at least 10 people were buried alive by avalanche after Nepal earthquake
3. Route to camp1 completely destroyed by avalanche.#NepalQuake
4. Avalanche sweeps Everest base-camp, killing 17: An avalanche triggered by Nepal's massive earthquake. . .
5. #Everest avalanche more than 100 climbers stuck in camp1 awaiting rescue.! #NepalQuake

Table 5.1: Example tweets from **April 2015 Nepal Earthquake** crisis event.

Recent work on social event understanding proposed a method to model event dependencies. They construct a sequence of graphs representing all the documents [Deng et al., 2019]. They preserve temporal dependencies of events by using a Dynamic Graph Convolutional Network (DGCN); a model that learns an expressive graph representation of nodes not only from their connections in a certain time-step, but also from the dynamic context of the previous time-step. In our framework we exploit the expressive power of a DGCN to aggregate tweet content and model large-scale crisis events, by learning graph edge weights. These weights let us identify important nodes and relations; a critical step for sub-event extraction [Meladianos et al., 2018].

We propose SD^{2}SG, (Sub-event Detection via Dynamic Semantic Graphs): a novel framework to extract important sub-events from a temporally-ordered group of tweets related to a crisis event. Our approach combines information across tweets into a set of temporally-ordered graphs, which are used to extract sub-events. Since we have limited data, we impose structural (entities can be connected only via a predicate) and semantic constraints (predicates are defined via the FrameNet ontology) in each graph. Thanks to these constraints, our model learns valid relations instead of coincidental co-occurrences of words via the use of a DGCN model.

5.4 SD^{2}SG : a Framework to Extract Sub-events



Time
Tweets
Input: Tweets from crisis event
Output: Extracted sub-events
~Predicate: rescue
Arguments: hundred, climbers, need, avalanche
~Predicate: evacuate
Arguments: helicopter, IndianArmy, Everest, camp1
Construct Initial Graph
DGCN (or GCN)
Extract Sub-events
Represents tweet content
Learns edge weights
Pattern matching on sub-graphs
t-1
t
t-1
t
Predicate: nucleus
Arguments: semantically connected to trigger / nucleus

Figure 5.1: Framework architecture diagram.

In this section, we discuss our approach on extracting sub-events from a stream of temporally-ordered tweets related to a crisis event. Our framework consists of the following steps, as shown in Figure 5.1: (*i*) construct the initial dynamic semantic graph, (*ii*) learn the graph's weights via a graph neural network, and (*iii*) extract sub-events from the learned graph via random walks that satisfy our semantic constraints. The pseudocode of our framework's steps is in algorithm 1.

Algorithm 1: Steps of SD^{2}SG

```
Input  : $C$ = A set of crisis events,
         $tweet_1, \dots, tweet_n$ in a temporal order,
         $n$ = total number of tweets in **a crisis event**
         $t$ = number of time-steps,
         $k$ = number of sub-events,
         pre-trained embeddings
Output: Extracted sub-events for each time-step $t$: $S_1, S_2, \dots, S_t$ from each crisis event.
1. for each crisis event, $C_i \in C$ do
     for tweets $\in \{relevant, irrelevant\}$ do
         Divide $tweet_1, tweet_2, \dots, tweet_n$ into groups of equal size, $D = D_1, D_2, ..., D_t$;
         for each $D_i \in D$ do
             Construct semantic graph $G_i$;
2. Run learning framework, DGCN on the output from Step 1 and extract indicator function $I$
   based on DGCN's parameters to extract graph weights;
3. for each crisis event, $C_i \in C$ do
     // Only run on graph constructed from relevant tweets
     for each $G_i \in G$ do
         $a$) Extract sub-graph $G_i^{'}$ based on $I$;
         $b$) Sample sub-events from random graph walks in $G_i^{'}$;
         $c$) Collect sub-events that meet semantic constraints (event structure);
         $d$) Rank extracted sub-events with tf-idf score. Choose top $k$;
```

5.4.1 Constructing Initial Dynamic Semantic Graphs

Given a large-scale crisis event, our first step is to represent the content of the related tweets in a graph structure by merging information across all the messages, which we call **initial graph**. An initial graph represents the tweets for a given time-step; it can be used dynamically in a sequence of initial graphs (i.e., one graph per time-step) or as a single graph (i.e., one time-step).

There are multiple ways to build the initial graph representation of tweets. In SD^{2}SG, we use a sequence of **initial semantic graphs**, where each graph is based on semantic relations from text. Given a set of tweets from a specific time-step, the initial semantic graph is a bipartite graph that connects predicates with their arguments, as they appear together in text. A group of tweets can be represented by a single initial graph, where the same predicate may be connected to different arguments from different sentences. An example of an initial semantic graph is shown in Figure 5.2, which is constructed based on tweets 3, 4 and 5 in Table 6.7.

Formally, given a tweet t_i, we use a dependency parser to extract the part-of-speech tags from tweets. Based on these tags we form two groups: (1) verbs and nominalized verbs (i.e., nouns that are derived from verbs, like *explosion*) $V_i = \{v_1, v_2, ...\}$, by matching the tokens to the Lexical Units provided in FrameNet [Baker et al., 1998b] and (2) nouns (excluding nominalized verbs) $N_i = \{n_1, n_2, ..\}$. The output graph has as nodes $\cup_i N_i \cup V_i$ for all tweets t_i. We form weighted

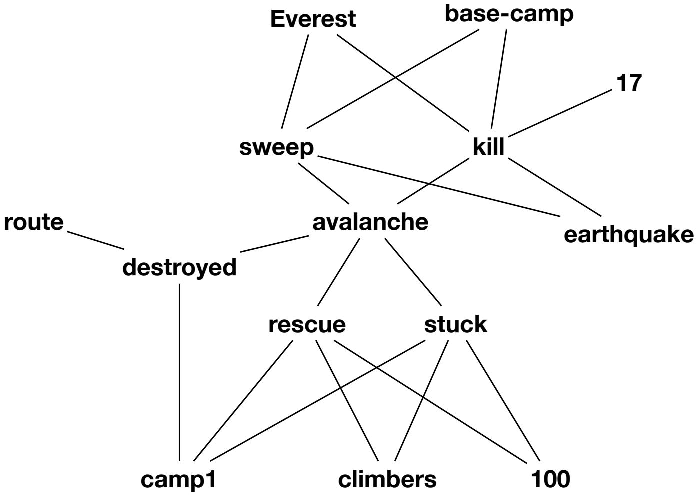


Figure 5.2: Initial semantic graph, based on subset of Nepal 2015 earthquake tweets (tweets 3,4,5)

edges only across the two groups (verbs V and nouns N), which are initialized based on the PMI of each pair [Church and Hanks, 1990]. More specifically, for each tweet t_i, we have $(v_i, n_j) \forall i, j$ but no (n_i, n_j) edges. This ensures that we link the sentence's predicate with its arguments, while avoiding to link arguments that appear together under different relations/predicates. As shown in Figure 5.2, this results in a graph that combines information across tweets in a more explainable way compared to previous approaches [Deng et al., 2019], while maintaining semantic relations from text.

5.4.2 Learning Edge Weights via a DGCN

The initial semantic graphs mainly capture information mentioned in sentence level, without taking context into account (i.e., information based on neighboring relations). This results in large graphs with noisy relations, where it is hard to extract important information. In order to get a smaller, less noisy graph, we formulate our problem as a classification task using dynamic graph convolutional networks (DGCNs) where the goal is to learn edge weights, a mechanism introduced by Deng et al. [2019] to detect social events for news articles. With this setup the model learns which neighbourhoods or sets of nodes in the graph are important and correspond to sub-events that occur during a crisis.

A DGCN model consists of a sequence of GCNs that are linked together by feeding information from the previous time-step to detect important factors in the context of social event understanding. Each initial graph is fed into a different GCN layer by time. For each GCN layer except the first one, the input features are processed by a temporal encoded module, involving the output of the last GCN layer and the current word embeddings, to capture temporal features. Finally, there is a masked nonlinear transformation layer to unify the final output vector from the final GCN layer.

The loss is calculated between the model output and ground truth label, which, in our case, is whether a group of tweets is related to a crisis event or not.

Formally, given a sequence of initial semantic graphs, we form their normalized adjacent matrices $A_1, A_2, ..A_t$ for each time-step t. We are also given a matrix of initial node features H_0, which in our case corresponds to the pre-trained word embeddings of the vocabulary. At each time-step, the convolutional layer of the DGCN is computed by: $H_{t+1} = g(A_t \bar{H}_t W^{(t)} + b^{(t)})$, where $W^{(t)}, b^{(t)}$ are model parameters and g is a non-linear activation function. Note that $\bar{H}_t$ does not correspond to the GCN output, but instead to the temporal encoding embeddings, calculated from the last TE layer. The temporal encoding is defined based on the following equations, where W_p, W_e, b_p, b_e are the parameters learned by the model:

$$H_p^{(t)} = H_t W_p^{(t)} + b_p^{(t)} \tag{5.1}$$

$$H_e^{(t)} = H_0 W_e^{(t)} + b_e^{(t)} \tag{5.2}$$

$$\bar{H}_t = tanh(|H_p^{(t)}||H_e^{(t)}|) \tag{5.3}$$

In order to classify the group of tweets as related or not to a crisis event, we set the output feature dimension of the last layer as 1. Due to dynamic graph encoding, the output feature vector of the last GCN layer is a combined representation of all graph nodes, which is different for each large-crisis event (i.e., different graph nodes). To guarantee the consistency of the model across instances, the DGCN uses a masked nonlinear transformation layer to map the final output vector to the prediction of the task. Finally, for each node i in the graph, we use the scalar value $h_{i,t}$ from the last GCN layer and $w_{i,m}$ from the masked nonlinear transformation layer to define an indicator function $I_i = h_{i,t} \times w_{i,m}$. This indicator function is used to select important nodes and their edges from the graph.

5.4.3 Extracting Sub-events

Given the sequence of learned graphs, the last part of our framework aims at extracting significantly smaller sub-graphs that represent sub-events (i.e. a typical sub-event contains 3-6 terms, while a graph might have 100-200 nodes). Although we use bipartite graphs to represent tweets, during learning, we treat them as homogeneous with zero edge-weight because of the nature of GCN/DGCN models (they operate on homogeneous graphs, where all nodes are treated equally). In order to generate valid sub-event candidates, we use a pattern matching method based on iterations of random walks on each graph [Bressan et al., 2018, Saha and Hasan, 2015]. The patterns used correspond to the typical structure of an event, where the predicate (usually a verb) is linked to a set of arguments (entities/nouns). Similarly to the semantic constraints in each initial semantic graph, we generate sub-events of star-like patterns of variable size (3–6 nodes), where the center node is an event predicate, as defined by the FrameNet lexicon. The size of these patterns is determined based on the maximum number of arguments in event detection tasks. More specifically, we looked at the annotation instructions of the ACE 2005 event detection dataset [LDC, 2005], according to which each event type has at most 6 arguments.

After extracting the candidate sub-events, we use a ranking method to remove duplicate or redundant information. To do that, we employ a tf-idf scoring scheme, where each sub-event is treated as a single document; the score of each sub-event equals to the average tf-idf score of its words. While other ranking or filtering methods can be applied, tf-idf is most appropriate as it retrieves important information (tf) and avoids similar, almost duplicate sub-events (idf).

5.5 Training & Evaluation of SD^{2}SG

Crisis Event Dataset. To train the model described in algorithm 1, we use a subset of the combined dataset described in Alam et al. [2020], which consists of Twitter data from 59 crisis events, including natural and man-made disasters. The tweets in this dataset were all manually annotated by Alam et al. [2020] as either being related or unrelated to their corresponding crisis event. The statistics of the dataset are shown in Table 5.2.

Crisis Event Type	# Crisis Events	# Related Tweets	# Unrelated Tweets
Hurricane/Typhoon	13	22,154	13,219
Crash/Explosion	11	7,689	9,718
Flood	11	12,366	9,747
Earthquake	10	12,164	6,911
Terrorist Attack	3	5,956	4,205
Tornado	3	6,242	5,034
Wildfire	3	2,842	348
MERS	1	1,113	69
Ebola	1	1,420	210
Volcano	1	104	191
Haze	1	476	136
Landslide	1	364	2,800
Total	59	73,070	52,588

Table 5.2: Dataset Statistics; number of tweets refers to tweets related to the large-scale crisis event.

Training Details. We execute Step 1 of algorithm 1 on these 59 crisis events. For the models based on dynamic graphs (SD^{2}SG and Simple Dynamic Graph) we use time-step $t = 3$.

We randomly split the dataset in Table 5.2 into train, development, and test sets, where an event belongs to only one of these sets. Related and unrelated sub-events are positive and negative examples, respectively. This set-up was used to train the DGCN model (Step 2 of algorithm 1). Out of 59 crisis events, we use 33, 10 and 16 as training, dev and test sets. For word embeddings we used 100d GloVE [Pennington et al., 2014] pre-trained on Twitter, and the DGCN was trained using the Adam optimizer with learning rate 5e-4, weight decay 5e-4, and dropout rate 0.2.

Once the DGCN is trained, we execute Step 3 in algorithm 1 and extract sub-graphs of interest for our evaluation. We evaluate our extracted sub-events with respect to two factors: (*i*) validity and (*ii*) importance in the context of a large-scale crisis event.

5.5.1 Baselines

To verify our assumption that information aggregation is important for our task, we chose baselines consisting of various methods that either aggregate information across tweets or not. We use Open IE as the baseline that does not aggregate information, while the remaining baselines are ablations of our proposed model. Our ablations study lets us verify the impact of every component of the proposed model. Unfortunately, since no prior work in crisis NLP extracts sub-events in the form of semantic relations, we cannot compare with these methods. Here is a brief overview of each baseline:

Simple Dynamic-Graph: uses a complete graph (i.e., edges across all pairs of nodes) without any constraints. The weight of each edge is based on the PMI of the two nodes. This model was proposed by Deng et al. [2019] to model social events.

Static Sem-Graph: (1 time step) constructs only one graph for all the tweets, without taking into account their temporal dimensions. The initial graph is constructed in a similar manner as the proposed model, but the weights are learned via a static GCN model.

Init Sem-Graph: uses the sequence of initial semantic graphs as-is (no learning of graph weights).

Open IE: uses the output of an Open IE system for each individual tweet to directly produce sub-event candidates. For this baseline, we use the OpenIE system developed by Stanovsky et al. [2018a]. Each sub-event is formed by using Open IE's *predicate* as the sub-event predicate and the head nouns of each *argument phrase* as the sub-event arguments. Since the output is already a set of sub-events instead of a graph, we directly rank them based on tf-idf features, similarly to the last step of the proposed model.

5.5.2 Validity of Sub-events

We perform a human evaluation of our extracted sub-events based on crowdsourced annotations via Amazon Mechanical Turk.

Data: We collect a total of 13,000 sub-events (details in Table 5.3) by selecting the top 100 sub-events for each baseline per time-step from 16 large-scale crisis events (after removing events with few instances). At this step we allow the sub-events to contain almost duplicates; sub-events that have a partial match with respect to their arguments. This helps us to evaluate each extracted sub-event and avoid inaccurate merging.

Large Scale Crisis Events	**Extracted Sub-events**
2014 India-Pakistan floods	1467
2012 Colorado wildfires	693
2013 Alberta floods	680
2013 Balochistan earthquakes	715
2013 Dhaka garment factory collapse	800
2013 Los Angeles International Airport shooting	687
2013 South Wales bushfires	683
2017 Puebla earthquake	710
2015 Nepal earthquake	939
2019 Covid pandemic	818
Cyclone Oswald	850
2013 Spuyten Duyvil derailment	771
Hurricane Harvey	688
MERS epidemic	894
2014 Typhoon Hagupit	690
West Texas Fertilizer Company explosion	915
Total	13,000

Table 5.3: Large-scale crisis events and their number of extracted sub-events.

Annotation Guidelines: First, we want to assess whether the extracted sub-events are valid. We showed every candidate sub-event $s_i = (t, a_1, a_2, ..)$ (where t is the predicate and $a_1, a_2, ..$ are the event arguments) to three MTurk annotators and asked them the following questions:

1. Does the predicate represent a crisis incident (e.g., outage, collapse, injury) during a major crisis event? Possible answers: *yes* or *no*.
2. How many of the argument words describe a crisis scenario with or without the predicate? Possible answers: *all*, *some*, or *none*.

We estimate the inter-annotator agreement of these judgments via Fleiss' Kappa; the predicate accuracy has $k = 0.5$, while sub-event accuracy $k = 0.37$. Based on our analysis of the annotations, we conclude that the major challenge for human annotators is to determine whether the argument words could potentially describe a crisis sub-event. This problem stems from the fact that some sub-events without any useful information may still be part of some crisis scenario. To address this challenge, we conducted a second evaluation by domain-experts (subsection 5.5.3), which aims to evaluate each sub-event also based on its importance, filtering out such insignificant sub-events.

Metric & Evaluation Summary: In Table 5.4 we show the results of our human evaluation. To estimate the accuracy of the sub-events overall (predicate and event arguments) for each baseline, we merged the answers of three categories; Yes and All, Yes and Some, and No and All. Sub-events belonging to these three categories are all considered valid, while the sub-events belonging to any of the other three categories invalid.

We decided this merging for two reasons. First, some sub-events might have predicates that are not clearly related to a crisis, but in combination with proper arguments the entire structure is a valid, meaningful sub-event in a crisis scenario. For example, the sub-event (predicate: *fly*, arguments: *rescuers, climbers, Everest*) is a valid sub-event for the Nepal 2015 earthquake given that *rescuers flew to save climbers in Everest*. However, if we look strictly at the annotation guidelines, this sub-event will belong to the category No and All, given that *fly* is not a crisis predicate. Second, some sub-events may be partially valid; the event predicate and some (but not all) of the arguments are valid. Such instances belonging to the Yes and Some class still contain meaningful information for the crisis event and could be used to inform decisions. Given that the ultimate goal of our tool is to be used as an initial filter mechanism that retains potentially useful information for humanitarian aid, we decided to count instances with minor argument inconsistencies as valid sub-events.

Our results show that SD^{2}SG outperforms all baselines. This highlights that all the components of the model (the initial semantic graph, the temporal aspect and the learned weights) contribute to a better model overall. However, we observe that the initial semantic graph is the second best performing model, with only 3% difference. From that, we conclude that the semantic and structural constraints are a crucial component to extract valid sub-events.

Crisis Predicate ⟶	Yes			No				
Event Arguments ⟶	All	Some	None	All	Some	None		
Models ↓	Sub-event Validity Score						**Predicate Accuracy**	**Sub-event Accuracy**
Open IE	0.30%	0.50%	2.70%	1.00%	9.20%	86.30%	4.50%	1.80%
init sem-graph (no learning)	7.70%	14.60%	13.00%	7.00%	18.80%	42.00%	39.20%	29.30%
Simple dynamic	8.00%	10.00%	11.30%	7.00%	13.10%	56.80%	30.00%	18.70%
static sem-graph	5.99%	8.90%	9.85%	1.60%	17.50%	56.10%	26.20%	16.49%
SD^{2}SG (proposed)	9.70%	15.30%	13.70%	7.10%	17.70%	36.50%	**45.40%**	**32.10%**

Table 5.4: Percentage of valid sub-events. Sub-event Accuracy represents instances that fall under the *Yes and All/Some* and *No and All* categories.

5.5.3 Importance of Sub-events

Determining the importance of a sub-event is a complex task that requires expert annotators, as they need to consider the context of the crisis event. Even though a sub-event may be valid with respect to its structure, we still need to validate if it is important in the context of the large-scale crisis event.

Data: We used the sub-events from the top performing baselines that were previously classified to belong to one of the following categories (sub-event accuracy); *Yes and All*, *Yes and Some* and *No and All*. Out of a total of 1,756 valid sub-events, we randomly select a subset of 300 ($\sim$ 80 sub-events per baseline).

Annotation Guidelines: To evaluate the importance of the sub-events, we conducted another human evaluation, where we asked two expert annotators the following question, for each sub-event:

1. Go to the provided Wikipedia link. Is the proposed sub-event important for this crisis event? Suppose the proposed sub-event did not happen, would the consequences of the major crisis or the humanitarian aid response be different? Possible answers: *yes* or *no*.

We estimate the inter-annotator agreement by Cohen's Kappa, which was $k = 0.48$.

Metric & Evaluation Summary: To evaluate sub-event importance we estimate the percentage of all extracted sub-events that are important, per model (important sub-event accuracy). The goal of this metric is to reflect how good each system is in extracting important sub-events.

To estimate the important sub-event accuracy we use the results obtained from both human evaluations. The first evaluation tells us how many valid sub-events each system extracts, while

Models	**Important Sub-event Accuracy**
init sem-graph (no learning)	19%
simple dynamic	15%
static	
sem-graph	14%
SD^{2}SG	**25%**

Table 5.5: Percentage of important sub-events. The second column is an estimate of the important sub-events that each model extracts.

the second how many of these valid sub-events are important, per system. Each annotated sub-event was considered important if any of the two annotators labeled it as such. Thus, the important sub-event accuracy per system is estimated by:

$$\frac{valid\_sub}{extracted\_sub}\frac{important\_sub^*}{valid\_sub^*} = Accuracy_1\frac{important\_sub^*}{valid\_sub^*} \tag{5.4}$$

where $Accuracy_1$ is the sub-event accuracy for a given model (subsection 5.5.2, while $valid\_sub^*$ and $important\_sub^*$ correspond to the number of valid and important sub-events respectively in the annotated sample (i.e., $valid\_sub^*$ = 300).

The results of this evaluation are shown in Table 5.5. We observe that our proposed model performs substantially better than the baselines (6% higher than the second-best). Although the accuracy of all systems is relatively low, this is due to the low percentage of valid events, since a sub-event must be valid in order to be important.

5.5.4 Discussion

In the previous section we show that despite SD^{2}SG performed significantly better than our baselines, our numbers are overall low; 45.5% of our extracted sub-events are valid and only 25% important. In this section we identify and discuss a set of reasons why sub-event extraction of tweets is a challenging problem and how we can improve.

The first step of our analysis is to compare our extracted sub-events to an existing resource manually curated by experts in crisis NLP. We used the EMTerms (Emergency Management Terms) ontology [Temnikova et al., 2015]; a resource of 7,000 manually annotated terms that are used in Twitter to describe crisis events, classified into 23 information-specific categories. Based on this lexicon and our extracted sub-events, we estimate the percentage of predicates and arguments that exist in the EM terms by a partial string matching (many EMTerms are phrases of 2-3 words). In Table 5.6 we show the results of this grounding. We observe that, overall, a large percentage of the predicates can be grounded in the ontology, while the argument overlap is significantly lower. This

Models	**Predicate Overlap**	**Argument Overlap**
init sem-graph (no learning)	85.10%	65.90%
PMI dynamic	**92.85%**	58.20%
static sem-graph	88.75%	67.60%
SD^{2}SG (proposed)	90.40%	**68.00%**

Table 5.6: Overlap of extracted sub-events with terms from the EM ontology.

can be explained by our semantic constraints on the predicates of the extracted sub-events (must exist in FrameNet), while the event arguments had no such constraints. However, given the results of our human evaluation in the previous section, we conclude that, even though a word might be a crisis related keyword, a sub-event formed by such keywords is not necessarily valid. This is due to the fact that sub-events aim to represent relations between several terms, thus grounding to an ontology is not a sufficient metric of the quality of the extracted sub-events.

Crisis Event	**Predicate**	**Arguments**
1. Alberta floods	evacuation	flooding, zone, Canada
2. Puebla earthquake	follow	rescuer, victims
3. Typhoon Hagupit	keep	safety, flee
4. Puebla earthquake	school	kill, child, dead
5. MERS	cough	healthcare, surveillance
6. Duyvil derailment	fatality	derailment,helicopter, major, abc, amtrak
7. MERS	emergency	infection, Fukuda discover
8. Dhaka garment factory collapse	rescue	survivors, number, labor,factory

Table 5.7: Example output from SD^{2}SG

In Table 5.7, we show a few real-output examples that highlight the complexity of sub-events. These sub-events belong to any of the three accepted categories of valid sub-events (*Yes and All/-Some* or *No and All*). Although they were all considered valid by human annotators, we observe a few major challenges. Although some predicates are not crisis words, they could still form a valid crisis sub-event with the appropriate arguments. Such an example is the predicate *follow* in sub-event 2, where *rescuers followed the victims* is a valid sub-event during a crisis. However, for some other instances, the predicate might be an entity in the particular context and, thus, not a valid sub-event. Such an example is the sub-event 4, where although the sentence *children were killed dead at school* seems like a valid sub-event, the model has incorrectly identified *school* as the predicate. Several of the annotation inconsistencies in the first part of our evaluation were due to similar instances, something that highlights the importance of domain expert annotations.

5.6 Future Work

A major problem in our framework is that we don't know how the arguments are related to the predicate. Although semantic frames consist of specific relations (e.g., agent, patient, location), our framework provides only a list of the related entities without their relations. An open challenge is to use thematic roles both for tweet representation and for the extracted sub-events, as this will result in more meaningful sub-events that would be easier to evaluate. Given that SD^{2}SGis modularized, it can be modified to represent thematic roles by using heterogeneous graph neural networks (HGNNs) instead of DGCNs. HGNNs are a type of network that consists of multiple types of edges or nodes. However, extracting thematic roles from text (first step of SD^{2}SG) would still be a particularly hard task due to the nature of social media text, which does not always conform to proper syntax and grammar.

A second challenge is the removal of duplicates and merging of information. As we see in Table 5.3, for each event we extract hundreds of sub-events (100 sub-events per baseline). However, this set contains several almost duplicates: sub-events that share a predicate and some of their arguments. Determining whether two sub-event mentions that are similar actually refer to the same sub-event or their argument differences are substantial to consider them as different sub-events is an open challenge that needs a solid understanding of both the crisis context and the event semantics. Although our filtering and ranking methods at the end of SD^{2}SGaim to partially address this, we conclude that this is a challenging problem and an interesting direction for future work.

5.7 Conclusion

In this chapter we discussed two types of event representations: semantic frames and sub-events. A semantic frame separates the event meaning into a structure of several dimensions (i.e., the semantic roles), each of them describing a different aspect of meaning for the event. This is a parallel to multi-faceted representations that we studied in earlier chapters but, unlike previous representations for entities and sentences, the multi-faceted representation of an event is reflected in language. In other words, frame semantics provide a framework to represent what is already stated in text.

We further show that the meaning of an event can be decomposed to a list of smaller sub-events. Each sub-event represents a different aspect of meaning of the larger event, forming together a detailed explanation by breaking down the semantics of the event. Such an event representation is complementary to semantic frames, which instead focus on a schematic, high-level description of the event semantics.

Towards that direction, we propose an approach to extract important sub-events for a larger event by combining information across different messages, which can be applied in Crisis NLP. As our second part of our evaluation shows, the extracted sub-events are a critical piece in the representation of the larger crisis event, as their existence influences the outcome of the larger event and other sub-events.

Decomposing an event or goal into smaller components is frequently used to provide deeper explanations of the semantics of the event or goal. Such a decomposition of the meaning is prominent in tasks such as script learning and event schema induction [Li et al., 2020, Wang et al., 2022], where the focus is to understand and predict missing events during a narrative or event graph leading to the completion of a given goal. Despite recent progress, there are several open-ended questions that future work should try to address, like how to decide the level of abstraction of a sub-event or how to quantify sub-event importance to the larger event / goal. Most previous work deals with a given list of sub-events that assumes answers to these questions. Although such questions feel natural to humans, they are extremely difficult for NLP systems as we have shown in this chapter.

Chapter 6

Event Implications & Entity's State Changes

6.1 Introduction

In Chapters 3–5 we studied representations for entities, sentences and events, using the assumption that their semantics are multi-faceted. Given a particular task and context, only some aspect of meaning is relevant to the task and generalizes across data distribution shifts. We showed that by decomposing meaning into smaller units according to the task, we obtain semantically rich representations that achieve performance and interpretability improvements in limited-data scenarios. Via the use of multi-faceted representations, machine learning models are able to: (1) identify which information is relevant, and (2) learn reasoning patterns by connecting the correct pieces of information.

While previous chapters focus on the design of meaning decomposition schemes to retain the aspect of meaning relevant to the task in question, Chapter 6 addresses the deeper challenge of learning how to combine pieces of relevant information together to solve reasoning problems. In order to do this, a machine learning model needs to learn patterns that connect information and generalize across different scenarios, imitating the underlying reasoning rules required to solve a reasoning challenge. Unlike early work that relied on meticulous handcrafted reasoning rules, heuristics and formal logic, such as the Logic Theorist system [Shaw et al., 1958] and the General Problem Solver [Newell et al., 1959], the patterns in a machine learning model are automatically learned from few examples. This poses the challenges of enhancing the generalization abilities of a machine learning model and guiding it towards the underlying reasoning rules, while avoiding the intense manual effort of creating and/or annotating precise reasoning rules.

Similarly to the problems described in Chapters 3–5, the main challenge in reasoning tasks is that they often occur in limited-data scenarios. By the term *limited-data scenarios*, we refer to situations where the training data is not sufficiently large to represent all possible configurations of the problem and, thus, inducing the correct decision rule is a significant challenge. Since reasoning problems require to connect multiple pieces of information, some of which may be ’commonsense knowledge’ and hence outside the domain and given context, a machine learning model using only surface level features such as lexical semantics and syntax may need to see all possible combinations of the available information to learn the underlying reasoning rules.

Our approach to address this problem is based on two basic principles: decomposition of meaning into facets and diversification of the facets. First, as we showed throughout this thesis, decomposing meaning into facets guide a machine learning model to learn which aspect of meaning is relevant and connect it to the task. However, in the case of reasoning problems, our "task" is not the downstream task that the model is evaluated on, but instead learning and applying the underlying reasoning rules. Thus, a model also needs guidance to learn patterns of connecting the facets of multiple entities to events. These patterns are necessary in order to generalize to unseen configurations, which is a crucial step in reasoning. This leads us to the second principle, where instead of using a fixed facetization scheme of each entity as seen in Definition Frames of Chapter 3, a model is trained on different meaning decompositions of an entity for the same context. As we show in this chapter, our mechanism helps to learn reasoning patterns that generalize to new, unseen facets and, thus, results in a model with better generalization abilities. These two principles form together the main theory of this thesis, as described below.

Thesis Theory: *By decomposing entity meaning into facets, the model learns to identify and focus on the aspects of meaning relevant to the underlying reasoning mechanism of the question. Furthermore, by varying the facets across instances, the model learns patterns that imitate the reasoning mechanisms involved for each event. This leads to good generalization for unseen attributes, which is a crucial step towards reasoning.*

6.2 Forms of Reasoning in NLU

While most tasks in Natural Language Understanding involve some degree of reasoning, recent work has identified a certain class of problems as more challenging than others, referring to them as **commonsense reasoning** tasks. The main difference that distinguishes them from other NLU tasks is that they require information that is not stated in the context. Identifying the commonsense knowledge necessary for a reasoning task is extremely challenging for a machine learning model, as it requires an understanding of the underlying reasoning mechanism that connects different pieces of information (both stated and unstated) together. On the other hand, since non-reasoning problems rely on simpler reasoning mechanisms, a machine learning model is able to solve them by using only lexical and syntactic information. Thus, while for non-reasoning tasks a model learns to link each context word to the prediction, for commonsense reasoning it needs to learn the underlying reasoning mechanisms that link relevant information to the prediction and use them to identify which is the relevant information, both in-context and unstated.

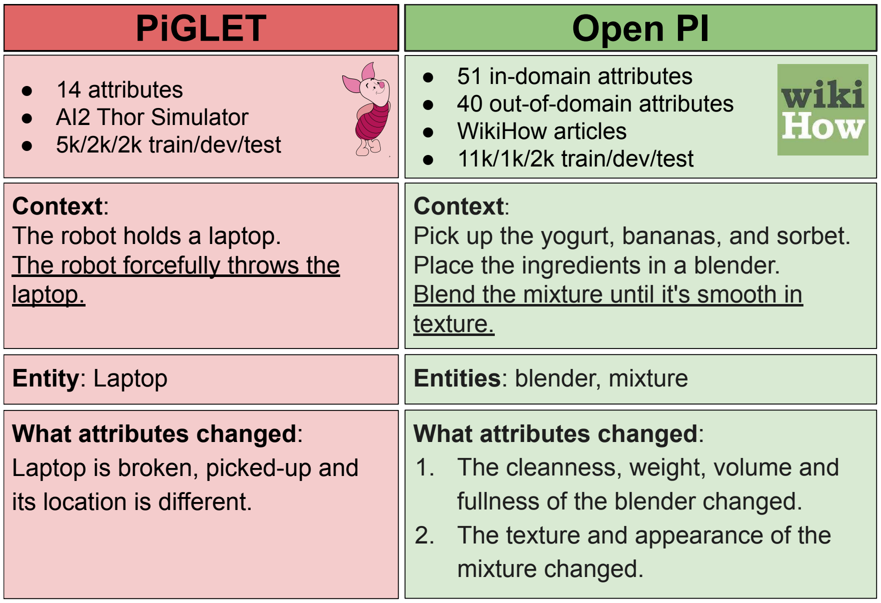
The difficulty of the underlying reasoning mechanism may vary even within the class of reasoning problems. For example, it is easier to model the effect of a single event compared to a series of events or when multiple entities are involved. While automated methods become increasingly more successful in solving reasoning tasks, a remaining challenge is how to measure and evaluate their reasoning abilities. The reasoning abilities of most models are evaluated indirectly, based on the number of correct predictions in a given task, without verifying the way the reached to this conclusion. The direct evaluation of reasoning is an extremely difficult problem for two reasons: first, it requires a manual annotation of the reasoning mechanisms and fixed criteria on how to compare mechanisms, and, second, most high-performing machine learning models are not interpretable.

These difficulties lead research in NLU to design increasingly challenging tasks under the assumption that they contain fewer dataset biases and, thus, the model should develop some degree of reasoning in order to reach a valid conclusion [Zellers et al., 2018, Mihaylov et al., 2018a, Clark et al., 2018a, Tandon et al., 2020]. Such challenging tasks often involve the use of information that is not explicitly stated or connecting information across different sentences and documents in order to make a prediction (multi-hop reasoning). However, even for these challenging tasks a model may still learn to directly connect the prediction to certain aspects of the input instead of developing an internal reasoning mechanism, falling into the trap of short-cut learning. To ensure that this is avoided, many datasets include an out-of-domain or “challenging” portion in their test set, where the model is evaluated with respect to its generalization abilities. Out-of-domain data typically require reasoning rules similar to the training data, but vary certain aspects of the data such as using unseen entities, events or topic. The underlying research assumption is that, in order to make correct predictions in the out-of-domain data, the model learns some research patterns that generalize across different contexts, and, thus, go beyond a surface correlation of the entities and the prediction.

6.3 The Problem of Event Implications

Modeling the effect of actions on entities (*event implications*) is a fundamental problem in AI spanning computer vision, cognitive science and natural language understanding. Most commonly referred to as the Frame Problem [McCarthy and Hayes, 1981], early solutions relied on a set of handcrafted rules and logical statements to model event implications. However, such methods require substantial manual effort and fail to generalize. More recently, modeling event implications has reemerged under the guise of common sense reasoning within NLP [Sap et al., 2019b, Bisk et al., 2020b, Talmor et al., 2019] and action anticipation in Computer Vision [Damen et al., 2018, Bakhtin et al., 2019].

Predicting event implications is a particularly difficult problem due to the complex nature of language and implicit knowledge required to answer such questions. For example, if we are given the sentence *the mug fell on the floor* and we want to determine whether *the mug* is *whole* and *functional*, we need to know of several facts such as the material of the mug, the fragility of ceramics, the hardness of the floor, etc. and also how to combine these facts together to **reason** whether the mug will break or not. Furthermore, while for some events we need all this information to predict the potential changes in entities, for other events such as *I broke the mug*, these changes are intrin-



PiGLET	Open PI
• 14 attributes • AI2 Thor Simulator • 5k/2k/2k train/dev/test	• 51 in-domain attributes • 40 out-of-domain attributes • WikiHow articles • 11k/1k/2k train/dev/test
Context: The robot holds a laptop. The robot forcefully throws the laptop.	**Context**: Pick up the yogurt, bananas, and sorbet. Place the ingredients in a blender. Blend the mixture until it's smooth in texture.
Entity: Laptop	**Entities**: blender, mixture
What attributes changed: Laptop is broken, picked-up and its location is different.	**What attributes changed**: 1. The cleanness, weight, volume and fullness of the blender changed. 2. The texture and appearance of the mixture changed.

Figure 6.1: We use the PiGLET and OpenPI datasets to probe if LLMs contain the necessary grounded and world knowledge to reason about event implications.

sic (i.e., they are always true). None of this knowledge is explicitly stated, instead being classified as *common sense knowledge*, and is traditionally acquired from observations or interactions with objects and the environment.

Core to this line of work is the assumption that events can be learned via language, without depending on other forms of perception. While physical interactions may be intuitively necessary [Bisk et al., 2020a], the limitations of language-only models are not apparent in existing reasoning datasets. To explore the utility of other modalities and interaction, Zellers et al. [2021] train a language model to predict physical changes in a virtual environment. According to their findings, such a model can substantially benefit from physical interactions compared to its language-only equivalent. However, as we show in this chapter, the purported limitations of the language-only models are not always well-founded and, instead, may be the result of an incorrect use of the baseline model. More specifically, we show that the language-only baselines in PiGLET use an encoding of the input that is inappropriate for a LLM. In their approach, the input is given in the form of a dictionary of (*attribute*, *value*) pairs, instead of natural language text. However, LLMs are trained as language generators by learning to predict a group of masked words (Masked Language Modeling) in a text, or continue a given text by producing a new sequence of words. In both cases, LLMs are trained to generate and, thus, expect a natural language text that maintains syntactic coherence. This implies that PiGLET has an unfair comparison of the language-only baselines, which, as we show in our results, achieve significantly better performance. Thus, we conclude that key to the success (or failure) of proving the importance of physical interactions are (1) How we use the language models to ensure a fair model comparison, and (2) The difficulty of the task domain and dataset.

We find that the difficulty of the task and dataset often indicates whether reasoning is required. Others have also noted that despite the tremendous gains in NLU made possible by Large Language models (LLM), they still stumble when reasoning is required [Brown et al., 2020]. Although LLM may not have inherent reasoning abilities, in this work we aim to investigate whether they are able to indirectly learn patters that imitate the underlying reasoning rules. To do this, we are present with two new challenges: (1) Can we evaluate reasoning via the generalization abilities of a model, and (2) Can a model learn reasoning patters more effectively if we share which information may be relevant to these patterns?

The nascent field of “prompting" [Liu et al., 2021, Wei et al., 2021, Ouyang et al., 2022] hints at a possible approach for humans to indirectly share information about reasoning patterns with models. Most recent work uses prompting mainly as a template to reformulate a new task, differentiate across tasks in multi-task learning or provide examples in few-shot learning scenarios. As we discuss in this chapter, prompting can also be used to share information about the relevant aspects of meaning and guide a model to extract reasoning patters by varying the information conveyed in each prompt. However, the best structure and the amount of information to convey via a prompt for a given task still remain as an open question.

In this chapter we discuss the problem of event implications as entity change-of-state with respect to physical attributes and identify the major challenges for LLMs. Our work tackles two major issues: (1) are language-only models able to predict physical event implications, and (2) can the meaning decomposition of an entity into facets function as an intermediate step for the extraction of reasoning patterns. As we discussed in Chapter 5, event semantics are challenging to represent, since events can form complex relations such as causality, temporal relations or sub-event relations. Furthermore, due to their dependence on multiple entities in the form of *event arguments*, event semantics and, consequently, event implications may significantly differ even across events with the same predicate. For example, if we consider the events *I dropped the glass on the hardwood floor* and *I dropped the fork on the hardwood floor*, the event implications will be different (the *glass* could break, unlike the *fork*).

Due to the complex nature of events, language models struggle to determine which information is relevant to the task and to extract reasoning patterns that generalize to unseen instances. To address this challenge, we use prompting as a means to share information about the desired facets of the entity and guide the model to focus on the aspect of meaning relevant to the event and context.

While Chapter 3 discussed multi-faceted entity representations in the form of Definition Frames, such a decomposition is not unique. Instead, meaning can be split to small units and aggregated, as it is fit by the information needs of the task. Definition Frames provide a meaning decomposition for **concepts**, which refers to relations that generalize across different contexts. This is the reason why we used definitions to extract the Qualia relations, as they describe fundamental properties that are always true. On the other hand, the problem of event implications requires a meaning decomposition for specific instances of an entity. Attributes that could change, such as the location of an object, determine the outcome of a physical event and, thus, they are a potentially relevant aspect of meaning we want to retain. Furthermore, different event types and context may rely on different facets, showing that there is no unique meaning decomposition scheme. An open question is whether meaning decomposition helps language models to learn event implications and, if so, how to choose the facets in order to ensure generalizability.

Our work aims to answer the first question by leveraging prompting; a modern technique used to share information with language models. Unlike a fixed structure of meaning decomposition, prompting is a more versatile technique that allows us to communicate with a model via natural language. This implies that we are able to: (1) change the decomposition scheme as we see fit according to the task, and (2) reuse the same model in different domains, without the need of further fine-tuning.

To answer the second question, we use a simple yet effective meaning decomposition that helps us explore how the choice of facets affects in-domain and out-of-domain performance. Given that our task provides a list of fixed attributes that can change due to an event, these attributes can be used as facets of the entities involved. By verbalizing a list of attributes to a model, we provide a meaning decomposition scheme which pushes the model to retain the aspect of meaning that corresponds to the queried attributes. Our attribute verbalization is an effective mechanism to communicate with a LLM via natural language text, which aligns with the LLM objective of generating words that, in combination to the input, are a syntactically coherent text. Finally, due to versatility of prompting, we can query different subsets of attributes across instances and study the effects on learning. As we show via our experiments, a model substantially benefits both with respect to performance and generalization by varying the attributes queried per instance (k-attribute prompt). This proves our assumption that attributes function as a bottle-neck to retain the aspect of meaning relevant to the underlying reasoning mechanisms and, thus, help the models to learn these mechanisms.

6.4 Task Formulation and Dataset Overview

The problem of predicting event implications can be formulated in several ways, with varying levels of difficulty. For example, Tandon et al. [2020] generate triplets of *entity, attribute, post-state* given some context, while Zellers et al. [2021] are given the entity, attribute, pre-state to only predict the *post-state* of the entity.

Our experiments imitate a realistic setting in naturally occurring language with a minimal set of assumptions. The context used in our setting is a small paragraph followed by a sentence describing an action, for which we want to predict the event implications on the participant entities. Our task is formulated similar to Zellers et al. [2021], where we are given an entity of interest and a predefined list of attributes, for which we need to determine whether a change-of-state occurred with respect to the list of attributes. However, unlike Zellers et al. [2021], our setup does not use the *pre-state* of the entity (the value of the attributes before the action), since this information cannot be extracted from text without the aid of in-domain human annotations.

As we see in Figure 6.1, in the OpenPI example we are given a paragraph, a list of entities (*blender, mixture*) and a fixed list of 51 attributes. Based on the action-sentence *Blend the mixture until it's smooth in texture*, we need to predict a binary value for each attribute denoting whether it changed or not for each of the given entities. Such a question can be quite challenging, since it requires to track entities and their states based on context in order to predict how they change due to the new action.

6.4.1 PiGLET

The PiGLET dataset [Zellers et al., 2021] consists of encodings of the pre-state and post-state of entities, as a result of an action. Each instance is accompanied by the **context**, a natural language description of the pre-state of the entities, followed by a description of the action. PiGLET is a small dataset that contains only 5k examples in the training set, while the development and test set have 2k examples each. Furthermore, PiGLET studies entity change-of-state with respect to only 14 attributes: *temperature, is_cooked, is_dirty, sliceable, is_sliced, mass, is_open, is_filled, is_picked_up, is_toggled, distance, is_broken, breakability* and *size*.

PiGLET is a semi-artificial dataset, where the *entity, pre-state, post-state, action* tuple was generated by exploring the virtual environment AI2 Thor [Kolve et al., 2017]. In addition to this encoding of context as a dictionary, the authors of PiGLET asked human annotators to construct natural language sentences describing the *pre-state* and *action* for every instance. These sentences were annotated by Amazon Mechanical Turkers, who were given the *entity, pre-state, post-state, action* tuple that was generated by the virtual environment. This results in simpler concise statements compared to the ambiguous language that humans naturally use to communicate. While the models studied by previous work used the tuple generated by the virtual environment as their input, our approach and baselines use only the natural language descriptions. The reason is that we supposed that this is a more appropriate way to communicate with a LLM: a hypothesis proven correct via our experiments.

The domain of the dataset is constrained by the set of possible interactions and entities in AI2 Thor, which corresponds to only 8 distinct events and 120 distinct entities. Due to the fact that each event cannot be applied to all entities present in the environment, PiGLET has a relatively small number of possible configurations. As we prove through our experiments, LLMs can achieve very good performance in PiGLET, invalidating previous claims that a model with physical interaction abilities is required and that, instead, PiGLET is not a challenging enough dataset to prove such claims.

6.4.2 OpenPI

Open PI [Tandon et al., 2020] also studies the change-of-state of entities with respect to physical attributes. However, unlike PiGLET, Open PI is based on articles from WikiHow, containing realistic natural language descriptions of physical changes. The context in this dataset is the entire WikiHow article preceding the action sentence from the article.

Open PI is a substantially larger dataset, containing an initial set of 51 pre-defined attributes from WordNet [Fellbaum, 2010]. This list of attributes was augmented by human annotators, resulting in two sets of attributes for each instance: **in-domain** (initial 51 attributes) and **out-of-domain** (introduced by Amazon Mechanical Turkers). The purpose of the out-of-domain attributes was to show that a manually curated list of attributes may not be sufficient in a real set-up to describe all possible event implications and additional attributes would be needed, highlighting the challenging aspects of the problem of physical event implications. However, this also leads us to the core of the **Frame Problem**, according to which we want to predict changes *without having to model all the attributes that do not change*. This means that we must introduce constraints on which attributes we want to predict changes for, since the list of all possible attributes may be infinitely large.

Semantic Type	In-domain Attributes	Out-of-domain Attributes
Spatial	location, volume, shape, size, orientation, length, distance, organization	angle, direction, area, height, width, pose, posture, spacial relation
Material	texture, electric conductivity, thickness, hardness, strength, pressure	tenseness, tension, tightness, softness, material, flexibility, thermal conductivity, density, granularity
Entity-Specific	cleanness, wetness, fullness, ownership, openness, cost, composition, coverage, focus	contents, wholeness, capacity, hydration, consumption, documentation, emotional state, pain, usage
Behavioral	knowledge, speed, motion, stability, complexity, skill	activity, balance, consistency, safety, familiarity, exposure, viability, resistance
Quantifier	amount	intensity, quantity, magnitude
Temporal	availability	age, life, existence, time
Sensory Perception	visibility	color, taste, temperature, smell, sound, appearance, weight, brightness

Table 6.1: Semantic types of attributes, both in-domain and out-of-domain.

In-domain attributes: The in-domain attributes were chosen by the authors of Open PI, based on a list of attributes relevant to physical interactions present in WordNet. Given their expertise on the subject, this is a meticulous list of attributes that contains only a few synonyms. This predefined list was given to all human annotators, so all attributes in the list were considered for every instance while being annotated. This means that we rarely have false negatives in the annotations; attributes that changed but the annotators failed to note this.

Out-of-domain attributes: The out-of-domain attributes were introduced by the human annotators as attributes that were not present in the pre-defined in-domain list. The annotators were asked to come up with such attributes themselves for each instance, which means that across different annotators we may find different attributes, even though all of them are correct. This results in many synonymous attributes, such as *angle* and *direction*, and false negatives in the annotations. For example, for $instance_1$, $annotator_1$ identifies that attribute *width* changed, and for $instance_2$, $annotator_2$ identifies that *height* changed. However, *width* may also change for $instance_2$, but the annotator failed to consider this attribute. As we show in our error analysis (Section 6.9), such false negatives in the annotations are common due to the large number of possible attribute changes.

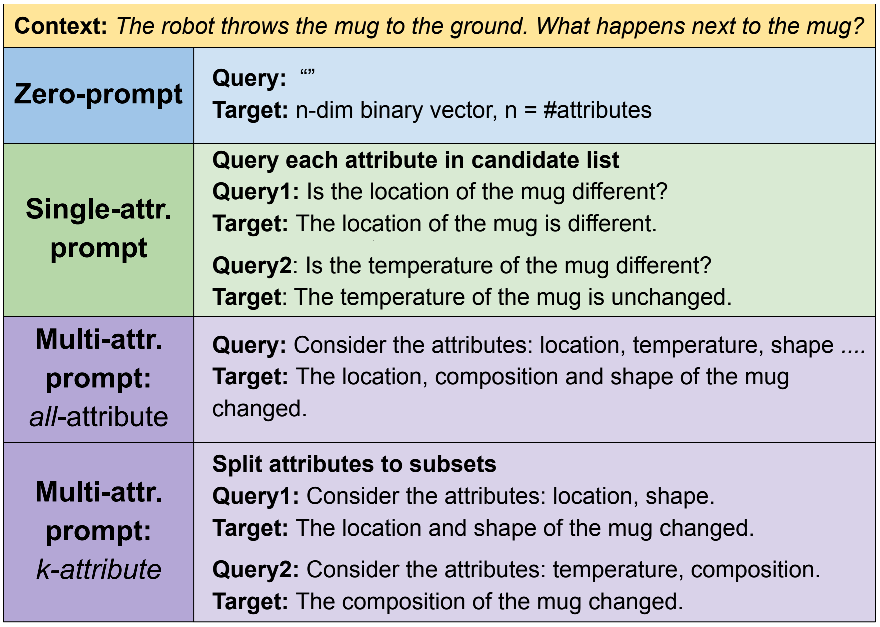
Although the total number of attributes (in-domain and out-of-domain) reaches ∼800 unique attributes, the initial 51 attributes cover more than 80% of instances. Furthermore, the vast majority of the newly introduced attributes appear only once and many of them contain typos or abbreviations. For these reasons, we filtered the attributes provided by the human annotators and construct a curated, less noisy list of out-of-domain attributes. Our filtering removes attributes that occur less than three times and words that correspond to the entity or event of the sentence instead of an attribute. This list, to which we will refer during our experiments as the out-of-domain attributes, consists of 49 attributes, which are shown in Table 6.1. In our experiments, all models are trained on the in-domain attributes.

As we see in Table 6.1, we divide both in-domain and out-of-domain attributes according to their semantic type. Via the resulting ontology, we gain useful insides about the type of attributes and study the performance per semantic cluster (subsection 6.8.4). We observe that although there is a large semantic overlap of the in-domain and out-of-domain attributes, almost 50% of the out-of-domain attributes could not be matched to a synonym. These attributes consist the most difficult out-of-domain group, showing the greatest challenges for our models.

6.5 Methodology

Next, we introduce our prompting techniques, which vary with respect to per-instance information content. Each technique is tested with different LLMs and fine-tuning methods. The goal of each prompting mechanism is to show how model performance and generalization vary based on the information conveyed in our queries. Our study focuses on four prompting methods depicted in Figure 6.2: zero-prompt, single-attribute, multi-attribute, and a variant of the latter, the k-attribute prompt.

Our approach builds on literature demonstrating benefits in using prompting to distinguish different tasks when a model is trained in a multi-task setting [Raffel et al., 2020, Wei et al., 2021]. In our study, however, we explore how to use prompts as a medium to convey the task-specific information that a model must know in order to solve the task, similar to how one would ask a human. To the best of our knowledge, we are the first ones to demonstrate advantages and disadvantages of different ways to codify intermediate steps required for reasoning via prompting and use them to study LLMs' understanding of event implications.



Context: *The robot throws the mug to the ground. What happens next to the mug?*	
Zero-prompt	**Query:** "" **Target:** n-dim binary vector, n = #attributes
Single-attr. prompt	**Query each attribute in candidate list** **Query1:** Is the location of the mug different? **Target:** The location of the mug is different. **Query2**: Is the temperature of the mug different? **Target**: The temperature of the mug is unchanged.
Multi-attr. prompt: *all*-attribute	**Query:** Consider the attributes: location, temperature, shape .... **Target:** The location, composition and shape of the mug changed.
Multi-attr. prompt: *k-attribute*	**Split attributes to subsets** **Query1:** Consider the attributes: location, shape. **Target:** The location and shape of the mug changed. **Query2:** Consider the attributes: temperature, composition. **Target:** The composition of the mug changed.

Figure 6.2: Prompting techniques used in our models. **The k-attribute prompt has better learning abilities by: (1) verbalizing the decomposition facets (attributes), and (2) diversifying the facet decomposition across instances.**

6.5.1 Large Language Models

We explore three transformer-based language models: an autoregressive, an autoencoder, and a seq-to-seq model. We include models with different architectures to investigate the effect of our prompting strategies across model families. Our goal is to use each model in combination with prompts that enhance their individual strengths, based on their pretraining schemes.

RoBERTa [Liu et al., 2019]: is an autoencoder model widely used in classification tasks. RoBERTa is a very robust and efficient model that achieves good performance in a large variety of tasks. This makes it a good candidate to study new tasks and evaluate new methodologies.

T5 [Raffel et al., 2019]: is a seq-to-seq model that has shown excellent performance in multi-tasking by using the task name as a prompt. Although T5 is pre-trained as a generator model, it can be used both for text classification and generation. Its excellent performance in multi-tasking shows that T5 is able to use prompt information to identify the task, from a list of predefined tasks (shown in pre-training). Since we also convey task-specific information via our attribute verbalization, T5 is an appropriate model to evaluate our methodology.

GPT-3 [Brown et al., 2020]: is an autoregressive model and is primarily used in zero and few-shot settings due to its substantially larger size. GPT-3 is used in language generation and classification, and has shown excellent performance in few-shot settings when queried with appropriate prompts. Due to its large size and pre-training data, GPT-3 may have already seen many of the common-sense knowledge required to solve reasoning tasks and developed good generalization abilities, as its high-performance in few-shot scenarios hints.

These backbone models are used with one of the three prompting techniques, as described in the following paragraphs and shown in Figure 6.2.

6.5.2 Multi-label Classifier: Zero-prompt

Our baseline model is a multi-label classifier with no explicit information about the nature of the task or the attributes themselves. The model takes the context and the prompt *Now what happens next to the [entity]?* as inputs, and predicts a binary vector, where entries correspond to changes in specific attributes. We test this mechanism with RoBERTa, as it performs well in classification tasks.

With this model we test the traditional "finetuning assumption" that, given enough data, the model can learn the correspondence between attributes and dimensions in the output vector and correctly predict their changes. This model serves as a baseline of how a LLM performs when fine-tuned to a specific task. Crucially, it does not have the ability to generalize to new attributes as the output vector is of fixed size.

6.5.3 LM as Classifier: Single-attribute Prompt

Our second prompting technique provides information about individual attributes. Via this technique we evaluate whether a model benefits from the verbalization of each attribute, as a means to retain useful information from the context. Unlike the zero-prompt model, this model can be used out-of-domain, with unseen attributes.

In this setup, we query the model about each individual attribute separately, for every *context-entity* pair, as shown in Figure 6.2. This mechanism was tested with all three models: RoBERTa (fine-tuned and zero-shot), T5 (fine-tuned) and GPT-3 (few-shot).

By querying each attribute individually, the model is able to focus only on information related to that specific attribute. This can both benefit and hurt performance, as we show in subsection 6.5.4. On one hand, the model pays more attention to the sentence semantics related to the queried attribute. By using the attribute as a bottleneck, the model learns which aspect of meaning is important in that instance. This is particularly beneficial in limited-data scenarios where generalization is necessary. On the other hand, by querying only a single attribute per instance, the model does not learn correlations across attributes. This weakness becomes more apparent in scenarios with many correlated attributes.

6.5.4 LM as Generator: Multi-attribute Prompt

Our final prompting technique focuses on retrieving information about a set of attributes, by querying multiple attributes together. This technique combines strengths of the zero-prompt and the single-attribute prompt models, as it is able to both verbalize the attributes and capture correlations across them. Unlike other mechanisms, this method allows us to control the information content per instance, by varying the set of queried attributes. As we show in subsection 6.5.4 and subsection 6.8.2, varying the attribute queries across training instances is crucial to achieve generalization.

For this technique, the prompt lists the attributes that the model should consider. This list is dataset specific and can vary between training and testing (i.e., out-of-domain) or even across training instances. The model is trained to generate the attributes that changed, as shown in Figure 6.2. This technique works with text generation models and was tested on both T5 (fine-tuned) and GPT-3 (few-shot).

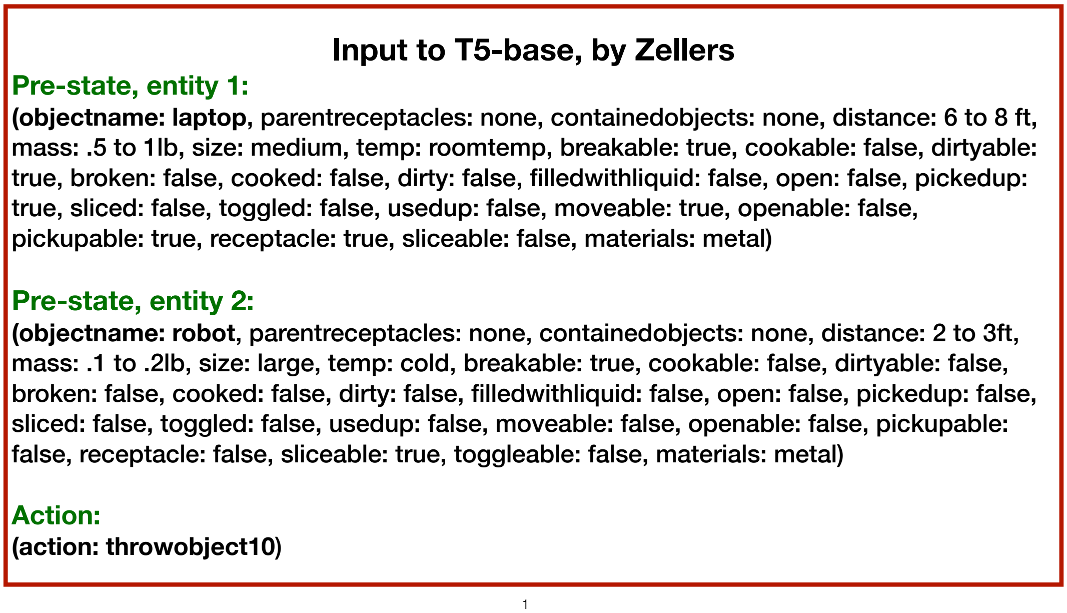
The first version of this model, the *all-attribute prompt*, queries all attributes that could change in the same instance. However, the risk with this approach is that, because the prompt is fixed, the model learns to pay little attention to the specific attributes that appear in it. We therefore propose a variant of this method, the k*-attribute prompt*, aiming to achieve high performance in both in-domain and out-of-domain scenarios. The objective is to learn about attribute dependencies but also force the model to pay attention to the specific attributes being prompted. To achieve this, we prompt the model with k random attributes and train it to predict changes *only* among these k attributes. More specifically, for each training example, we partition the 51 attributes into q random groups where q is a random integer between 1 and 5. k refers to the number of attributes in each partition. This method ensures that the model is queried with k random attributes and that all 51 attributes are always queried for each example.

6.6 PiGLET

6.6.1 Baselines

In this part we briefly describe the previous models evaluated on PiGLET dataset and how they relate to our proposed models and baselines. One of the main claims of Zellers et al. [2021] is the necessity of physical interactions, by showing that a physical interaction model performs substantially better than models that rely solely on language. However, via our experiments we invalidate their claim by showing that the same language-only models achieve comparable performance to the physical interaction model, if used with suitable input.

Physical Interaction model: This model is our strongest baseline and it was proposed by the authors of PiGLET as the state-of-the-art model. According to their findings, this model achieves 81.1% hard accuracy, substantially improving upon all language-only baselines. This model consists of two components: (1) a physical interaction model trained in the virtual environment AI2 Thor, and (2) a language model based on GPT-2 Radford et al. [2019]. Similarly to their language-only baselines, this model has access to the pre-state of each entity and the action.



Input to T5-base, by Zellers

Pre-state, entity 1:
(**objectname: laptop**, parentreceptacles: none, containedobjects: none, distance: 6 to 8 ft, mass: .5 to 1lb, size: medium, temp: roomtemp, breakable: true, cookable: false, dirtyable: true, broken: false, cooked: false, dirty: false, filledwithliquid: false, open: false, pickedup: true, sliced: false, toggled: false, usedup: false, moveable: true, openable: false, pickupable: true, receptacle: true, sliceable: false, materials: metal)

Pre-state, entity 2:
(**objectname: robot**, parentreceptacles: none, containedobjects: none, distance: 2 to 3ft, mass: .1 to .2lb, size: large, temp: cold, breakable: true, cookable: false, dirtyable: false, broken: false, cooked: false, dirty: false, filledwithliquid: false, open: false, pickedup: false, sliced: false, toggled: false, usedup: false, moveable: false, openable: false, pickupable: false, receptacle: false, sliceable: true, toggleable: false, materials: metal)

Action:
(action: throwobject10)

Figure 6.3: The input format for the language-only baseline model that was used by Zellers et al. [2021]. **The same core model (T5-base) used with suitable input performs significantly better.**

Language-only models: In our experiments we use three models based only on natural language (no physical interaction component): an n-gram Logistic Regression, a RoBERTa zero-prompt model and a T5 all-attribute prompt model. Via these experiments we aim to show that: (1) a physical interaction model is not needed to achieve good performance in PiGLET, and (2) the performance of an LLM may significantly vary based on how we use it. The input in all our experiments with language-only models takes the following form: a sentence describing the pre-state, followed by the action and a specific prompt based on the evaluated model. This input has the form of a naturally occurring text, similar to what a large language model is pre-trained on. For example, for the instance in Figure 6.1, the input would be *[prompt] The robot holds a laptop. The robot forcefully throws the laptop. Which attributes changed for the laptop?*, where *[prompt]* corresponds to the verbalization of the names of all possible attributes for the T5 all-attribute prompt model and to the empty string for the Logistic Regression and the zero-prompt models.

This input formulation is very different from the one used by Zellers et al. [2021] for the language-only baselines. In their approach, the input is a dictionary encoding of the entity pre-state and action, instead of a natural language description, as we see in Figure 6.3. As we show via our experiments, this difference is essential, as it leads to incorrect results about the abilities of the evaluated models. Namely, Zellers et al. [2021] use a T5-base model with this input and achieve only 53.9% in hard accuracy, compared to 81.1% of the Physical Interaction model. However, when we use the same model (T5-base) with appropriately formatted input, it achieves very similar results to the Physical Interaction model.

	All attributes			Per-attribute F1				
Model	Pr	Re	F1	Dist	Size	Mass	Temp	isBroken
Physical Interaction, (PiGLET)	97.4	91.6	**94.4**	**93.6**	79.2	98.3	**99.6**	92.8
n-gram LogReg (baseline)	87.8	88.0	87.9	78.8	74.7	97.8	94.0	79.4
RoBERTa-base, zero-prompt	95.2	92.6	93.9	90.6	82.7	**100.0**	95.3	**94.7**
T5-base, all-attribute prompt	93.0	95.4	94.1	91.7	**83.5**	**100.0**	95.8	90.3

Table 6.2: Micro-Precision, Recall and F1 scores across all 14 attributes in PiGLET. Per-attribute F1 scores for challenging attributes, as in Zellers et al. [2021]. **Language-only models perform competitively with PiGLET.**

6.6.2 Evaluation

As shown in Table 6.2, all models perform relatively well on the PiGLET dataset. The extremely small margin in performance between Physical Interaction and the proposed models (RoBERTa zero-prompt and T5 all-attribute) indicates that language models can learn about physical attributes even without the need of physical interactions with the environment. However, we highlight that this conclusion holds for datasets similar to PiGLET and the importance of physical interactions remains an open question that must be tested in more realistic and challenging datasets.

Despite the high performance of our proposed models, previously reported baselines on PiGLET show significantly lower performance than the Physical Interaction model. Notably, their baseline using T5-base achieves only 53.9% in hard accuracy, compared to 81.1% of the Physical Interaction model [Zellers et al., 2021]. Unfortunately we cannot directly compare these results to our proposed models due to their choice of metric (hard accuracy) and different problem formulation, where the input and output is the encoding of the pre-state and post-state of the entity. Despite the use of different metrics, we observe a minimal performance difference between language-only models and PiGLET. This highlights the importance of using proper prompting techniques and task formulation to take full advantage of LLMs, otherwise we risk of misrepresented results and invalid conclusions due to unfair comparison of the baselines.

Our final observation is that there is a larger gap between the n-gram LogReg model and the rest of the models. This shows that, although language is very useful to predict physical event implications, pre-trained language models still have a significant advantage due to the information they have previously seen. This raises the question of how can we better exploit the relations that pre-trained language models already know, which we explore via the next set of experiments.

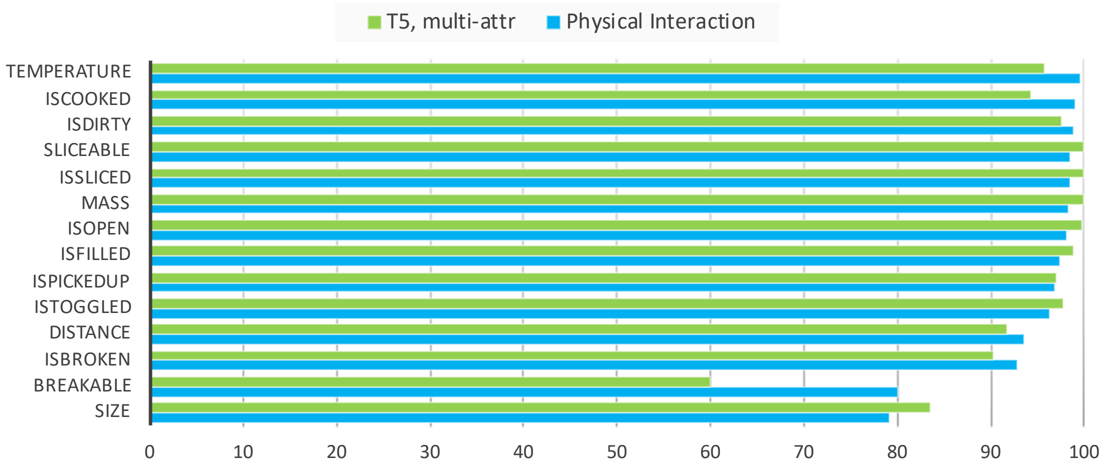


Figure 6.4: F1 score per attribute for Physical Interaction and T5 all-attribute prompt models, in PiGLET. **The high performance of both models highlights the limitations of PiGLET and the need of more challenging datasets to evaluate the effect of physical interactions and compare them to language-only models.**

In Figure 6.4 we also show the per-attribute performance in PiGLET for the two best-performing models: the Physical Interaction and the T5 all-attribute prompt models. We observe that overall for the vast majority of attributes the performance of both models is extremely high (F1 >90). Only for the attribute *breakability* the T5 all-attribute prompt model has relatively low performance (F1 = 60) compared to the Physical Interaction model (F1 = 80). Our hypothesis of why this occurs is that the Physical Interaction model has access to the values of all the attributes before the action (pre-state), while the T5 all-attribute prompt model only sees a short sentence description of what happened before. Thus, attributes like *breakability* are ambiguous without the pre-state information: an object may be considered breakable after it is broken (i.e., can break further) or not since it cannot be broken again.

6.7 OpenPI

Due to our finding that PiGLET is not a challenging dataset and all the LLM-based models perform extremely well, we use Open PI as the main dataset to compare the proposed prompting techniques. With the exception of the GPT-3 models, all models have relatively similar sizes, ranging from 123M (RoBERTa-base) to 354M (RoBERTa-large) parameters.

		In-domain			Out-domain		
Training	Model	Pr	Re	F1	Pr	Re	F1
Zero-shot	RoBERTa-large, single-attribute prompt	3.1	63.3	5.9	2.4	68.8	4.6
Few-shot	GPT-3-Babbage, single-attribute prompt	3.7	82.4	7.1	-	-	-
	GPT-3-DaVinci, all-attribute prompt	37.6	24.5	29.7	28.3	12.9	17.7
Fine-tuned	GPT-2 (baseline in Open PI)	49.8	11.8	19.1	-	-	-
	RoBERTa-large, zero prompt	65.1	40.1	49.6	-	-	-
	RoBERTa-base, single-attribute prompt	40.3	55.1	46.6	21.3	26.2	**23.5**
	T5-base, single-attribute prompt	34.6	53.3	42.0	15.9	21.5	18.2
	T5-base, all-attribute prompt	47.5	56.0	**51.4**	25.0	1.2	2.2
	T5-base, k-attribute prompt	52.8	50.0	**51.4**	16.8	22.7	19.3

Table 6.3: Micro-Precision, Recall and F1 scores for Open PI. In-domain attributes refers to the 51 originally curated attributes, while out-domain to the 41 attributes introduced by human annotators.**The k-attribute prompt combines best in-domain performance with generalization abilities. Verifies our hypothesis about: (1) decomposition of meaning into facets, and (2) diversification of the decomposition across instances.**

Few-shot: The GPT-3-based models were the only models used via few-shot learning, due to lack of resources for fine-tuning models of their size. For each instance in the test set, we pick 10 examples from the training set to be included in the prompt - there are marginal improvements beyond four [Min et al., 2022]. Performance in complex tasks like QA is sensitive to prompt selection [Liu et al., 2022]. Following previous work, we pick the relevant examples based on semantic similarity [Reimers and Gurevych, 2019]. In the single-attribute prompt setting, we include examples querying the same attribute, and balance both positives and negatives.

In-domain vs out-domain: All our models are trained on the initial 51 attributes (subsection 6.4.2). For in-domain experiments, the models are tested on the same set of attributes, while for out-of-domain on the new attributes introduced by human annotators, as explained in Section 6.4.2. After removal of rare attributes that occur in less than 3 instances, the out-of-domain set consists of 49 unique attributes.

GPT-2 baseline: This model is the strongest baseline proposed by Tandon et al. [2020] along the Open PI dataset. It is based on a GPT-2 model trained to generate sentences describing entity change-of-states. Each instance has as input a WikiHow article, followed by the prompt *What happens next?*. The model generates sentences that follow a specific format, *The [attribute] of the [entity] was [pre-state] before and [post-state] after*, where the fields between the square brackets are used to evaluate the predictions. Given that our models are evaluated on their ability to predict entity change-of-state with respect to an attribute, we post-process their output by removing generations where the *entity* is wrong or the *attribute* is not part of the given attribute list, so we have a fair comparison of the models.

6.7.1 In-domain Evaluation

As shown in Table 6.3, our models (F1 = 51.4) perform significantly better than the GPT-2 baseline (F1 = 19.1) provided by the Open PI authors [Tandon et al., 2020]. This highlights one of the core challenges of the Frame Problem, to predict event implications *without having to model everything that does not change*. Thus, a modeling approach of generating all attributes that change without any constraints is extremely difficult, as it has to consider also everything that does not change. Instead, by verbalizing and querying specific attributes from a given list, we help the model to narrow the domain of possible changes and then learn to generalize to unseen attributes, when queried about them.

Our second observation from Table 6.3 is that the best performing models in-domain are the all-attribute prompt and the k-attribute prompt, followed by the zero-prompt model. This shows that the verbalization of attributes has a positive impact in performance, given that both multi-attribute models beat the zero-prompt baseline. This difference is even more striking while studying the per-attribute performance, as shown in Figure 6.5. We observe that when attributes are sorted with respect to their frequency, RoBERTa zero-prompt completely ignores the lowest 50%, predicting always that there is no change for these attributes. This is an extremely undesirable behavior, as the model only focuses on learning frequent attributes to attain a good performance, jeopardizing its robustness to different domains. On the other hand, the T5 multi-attribute prompt model has decent performance for several less frequent attributes.

Based on Figure 6.5 and Figure 6.6, we observe that the RoBERTa zero-prompt model has higher performance than the T5 multi-attribute prompt for only five attributes: *location, cleanness, temperature, size* and *power*. Three out of these attributes belong to the most frequent attributes (more than 1000 instances). Furthermore, for each of these attributes the absolute difference in F1 for the two models is actually very small. However, due to the frequency of these attributes this difference counts much more for the overall micro-F1, which was reported in Table 6.3.

Another observation based on Figure 6.5 is that both models have F1 = 0 for certain attributes. Although this is more prominent for the RoBERTa zero-prompt model (24/41 attributes), the T5 multi-attribute prompt model also fails to learn some attributes (9/41 attributes). Furthermore, we notice that 8/9 of the attributes where the T5 multi-attribute prompt fails belong to the lowest-frequency class of attributes (attributes with less than 100 instances). This highlights that, although the T5 multi-attribute prompt model is substantially better than RoBERTa zero-prompt on learning rare attributes, some attributes are inherently more difficult than others and, thus, require more data to learn them. To explore the question of how performance relates to the semantics of an attribute, in subsection 6.8.4 we provide a detailed analysis of attribute performance according to their semantic type.

Our final observation from Table 6.3 is that both the T5 single-attribute prompt and the RoBERTa single-attribute prompt models have lower performance compared to the T5 multi-attribute prompt models. This observation shows that just verbalizing the attributes as part of the prompt is not

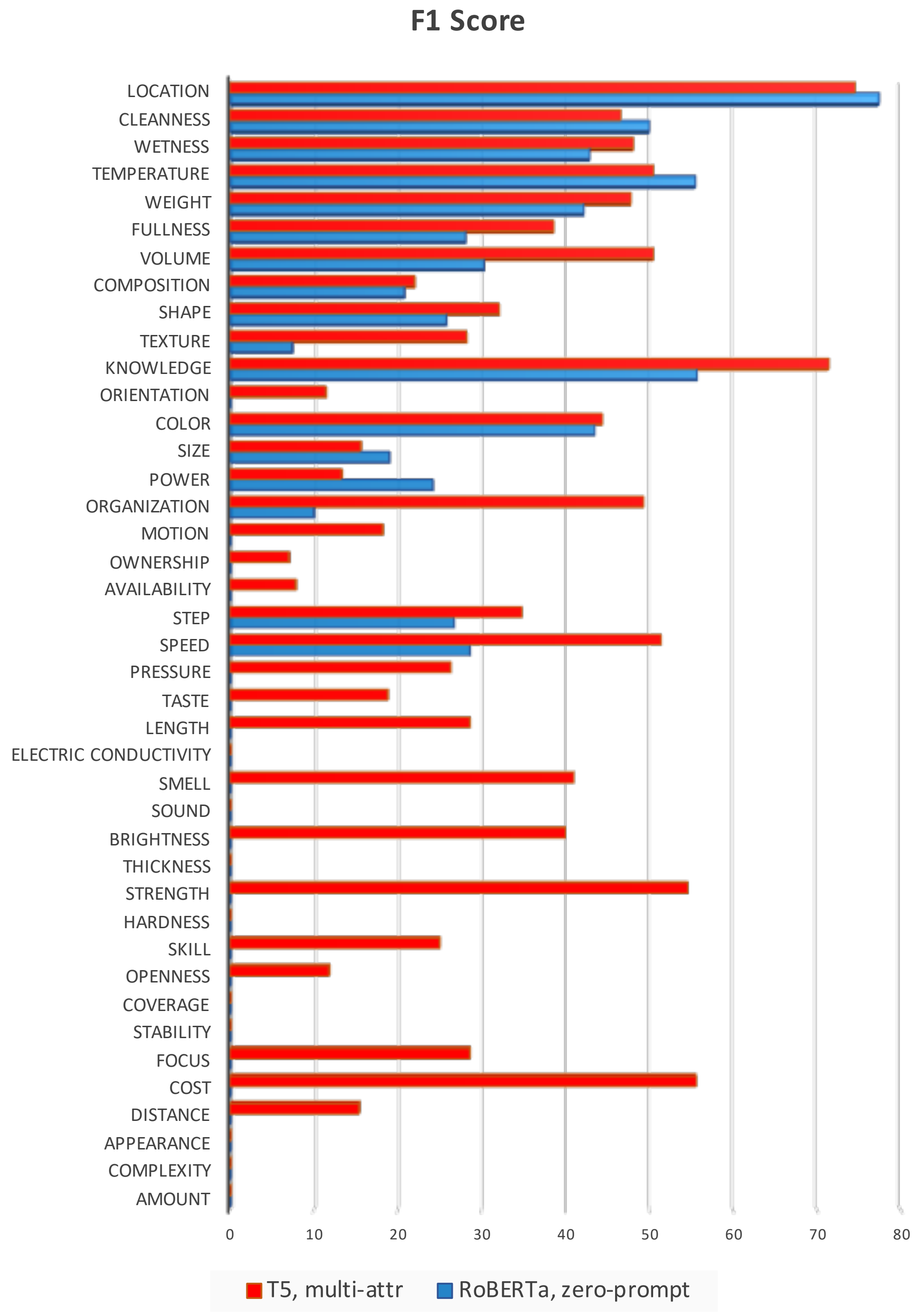


Figure 6.5: F1 score per attribute for RoBERTa zero-prompt and T5 multi-attribute prompt models in Open PI. The attributes are sorted according to their frequency (decreasing). **RoBERTa zero-prompt completely ignores all attributes with less than 150 instances.**

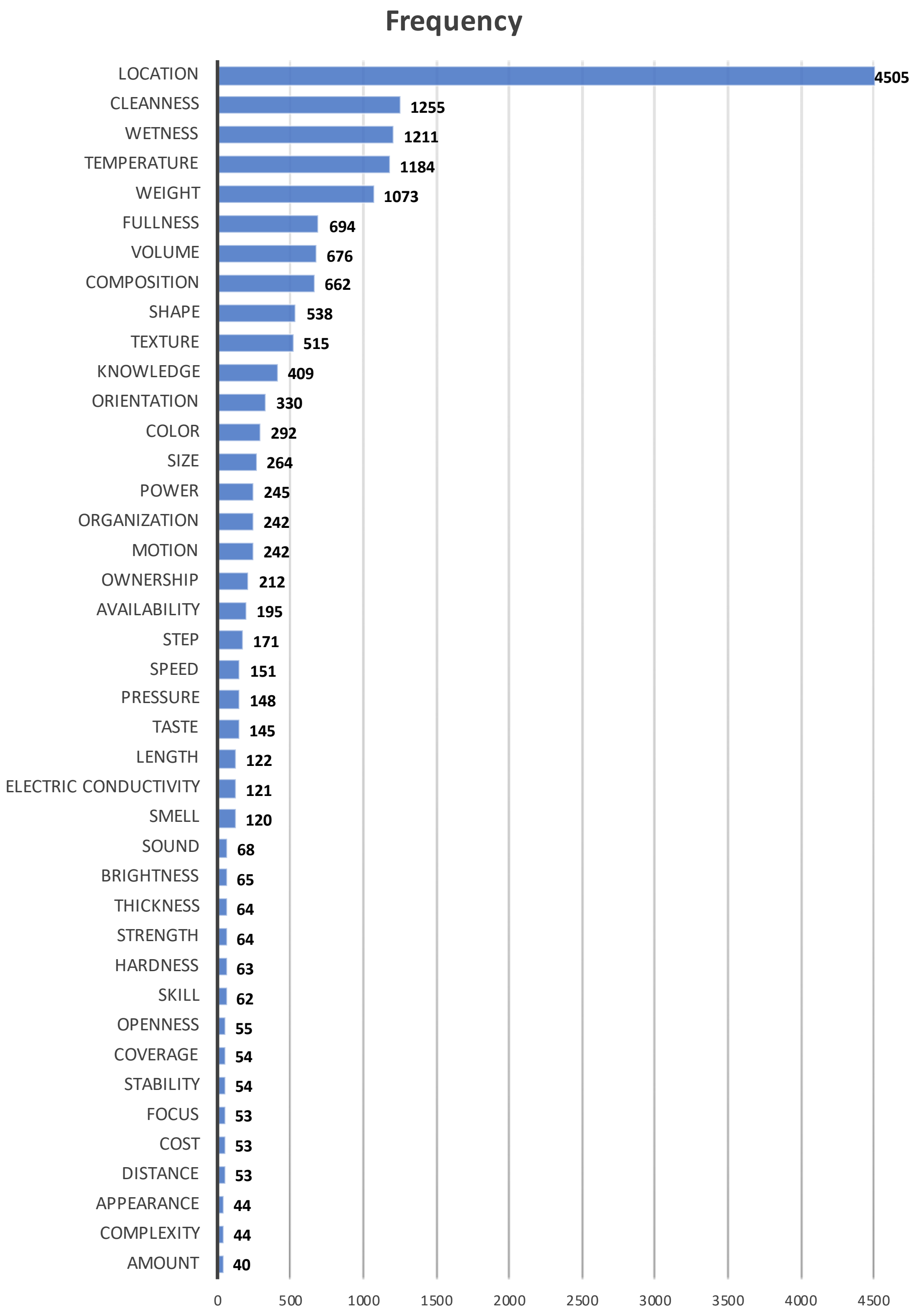


Figure 6.6: Frequency per attribute for in-domain attributes in Open PI. The attributes are sorted according to their frequency (decreasing).

sufficient and, instead, we need to consider how we diversify our queried attributes per instance. The diversification of the facet decomposition, which takes the form of attributes in our task, has to consider two dimensions: (1) modeling facet dependencies within a single query, and (2) querying different facets per instance. The first dimension is particularly important when our facets (attributes) are not independent and there is not sufficient data to learn their dependencies, which explains the lower performance of the single-attribute models. As we discuss in subsection 6.7.2, the second dimension is essential for the generalization properties of a model.

6.7.2 Out-of-domain Evaluation

As we observe in the Table 6.3, despite the poor out-of-domain performance of the T5-base all-attribute prompt model (F1 = 2.2), the other two variants of the same prompting technique have performance competitive to the best model. Namely, the T5 k-attribute prompt model reaches F1 = 19.3, while the GPT-3 all-attribute has F1 = 17.7. This confirms our hypothesis that fine-tuning a model with a fixed query hurts its generalization properties, as it does not learn to pay attention to the queried attributes. Instead, as discussed in Section 6.1, the diversification of the facet decomposition across instances guides the model to learn how to extract a different aspect of meaning of the entity, based on the queried attributes. This is an essential step for learning, since the facets function as a bottle-neck to retain the meaning that is relevant to the underlying reasoning mechanism. For example, given a sentence s where we want to predict the change-of-state of the entity e, we may create two instances for s, e with different sets of queried attributes, q_1 and q_2. If a model is trained on both instances, it will learn a different aspect of meaning for each instance, despite having the same context.

Although the GPT-3-DaVinci all-attribute prompt model does not use the diversification of facets, we observe that its generalization abilities are not negatively affected. This is due to few-shot learning, where the model is forced to learn to generalize based on very few examples. However, this performance may also be attributed due to the choice of language model, as GPT-3-DaVinci is considered the best version of GPT-3 models to predict complex intent and causality, with a total of 175B parameters. For comparison, T5-base and RoBERTa-base have 220M and 123M parameters, respectively. The further investigation of the extend of facet diversification on GPT-3 is left as future work, since experiments with diverse query sets require an extremely large number of resources (e.g., running just one configuration of GPT-3-DaVinci single-attribute prompt would cost 2,000$) and our T5 k-attribute prompt experiments indicate that the number of queried attributes must also be treated as a hyperparameter.

Our final observation is that, despite the low performance for in-domain experiments, the RoBERTa single-attribute prompt model performs the best in out-of-domain. Although the facet diversification explains the performance difference with T5 all-attribute prompt, it doesn't explain why RoBERTa single-attribute prompt is slightly better than the T5 k-attribute prompt. To further investigate this, we perform a manual error analysis of the output of both models and construct a typology of their errors, which is discussed in Section 6.9.

6.8 Model Analysis & Discussion

In this section we further analyze the models' behavior with respect to the type of attributes they see and their generalization properties. This study serves to uncover advantages and disadvantages of each technique and suggest promising methods for future work to enhance both model performance and robustness. For all our experiments in this section we use Open PI, as it is a more challenging dataset than PiGLET with a large number of attributes.

6.8.1 Reasoning with Rare Attributes

Since some attributes are significantly more frequent than others, fine-tuned models have been exposed to more data about them, which positively influences performance for these attributes. For example, performance across all fine-tuned models for the most frequent attribute *location* is substantially higher (F1 = 0.65–0.75) compared to other attributes such as *taste*, as shown in Figure 6.5. Although all models perform well on such high frequency attributes, our analysis shows that there are significant differences in performance for less frequent attributes.

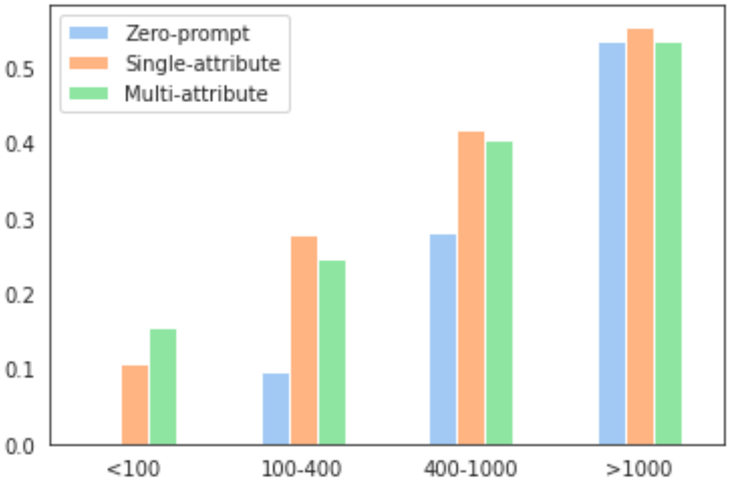


Figure 6.7: Performance per attribute frequency in training data. Each bar shows the weighted-F1 score across all attributes in the same frequency category. **Unlike our proposed models, RoBERTa zero-prompt's performance is much lower when in lower frequency attributes and reaching F1 = 0 for rare attributes (less than 100 instances).**

We study per-attribute model performance based on each attribute's frequency in training data for the three best performing models: RoBERTa zero-prompt, RoBERTa single-attribute, and T5 k-attribute. After grouping each attribute with respect to its frequency, we observe four distinct groups: low (<100 instances), medium-low (100-400 instances), medium-high (400-1000 instances) and high (>1000 instances) frequency. In Figure 6.7 we plot the weighted-F1 score per group for each of the three models.

Our first observation is that all models perform well for high-frequency attributes and differences in performance are very small across the models (F1 = 55–60). This confirms our hypothesis that LLMs can learn physical interactions and achieve strong performance, if there is a sufficiently large amount of labeled data to fine-tune on. This conclusion also agrees with our findings from PiGLET, where LLMs achieve very good performance (F1 = 94), if trained with proper input. Although the training set of PiGLET (5k instances) is smaller than the one of Open PI (11k instances), PiGLET has a smaller number of attributes and entities. This results in more positive instances per attribute, which is sufficient to generalize to most configurations possible within the virtual environment.

	RoBERTa, zero-prompt	RoBERTa, single-attribute	T5, k-attribute
Spearman correlation	$\rho = 0.82$	$\rho = 0.80$	$\rho = 0.51$

Table 6.4: Spearman correlation between attribute frequency and F1 score. High correlation means the model learns primarily high-frequency attributes. All results have p-value < 0.001.

Our second observation is that, although performance in high-frequency attributes is similar across all models, it significantly drops for RoBERTa zero-prompt when attribute frequency decreases. This shows that the model struggles to learn with fewer examples. This difference is most striking in the low-frequency cluster, where the model completely ignores the attributes (F1 = 0.0). On the other hand, the T5 k-attribute prompt model (followed by RoBERTa single-attribute prompt) has relatively high performance even in low-frequency attributes. This supports one of our main hypothesis in this chapter that,

by verbalizing and querying specific attributes, models pay attention to each attribute and learn to imitate reasoning patterns, a crucial step in limited-data scenarios.

Finally, to quantify the effect of attribute frequency on performance, we estimate the per-attribute Spearman correlation between performance and frequency for all three models. As we see in Table 6.4, both the RoBERTa zero-prompt and the RoBERTa single-attribute prompt models have a very strong correlation of performance to frequence, while the T5 k-attribute prompt has only moderate correlation. Thus, the T5 k-attribute prompt model learns to predict attributes without overly relying on their frequency. This results in a model with better generalization abilities to limited-data scenarios and unseen attributes, as we also show in Section 6.9.

6.8.2 Diversifying the Facets via the k-attribute Prompt Model

To verify the generalization abilities of the models, we test them in out-of-domain scenarios with unseen attributes. Through manual inspection we find that the all-attribute models have an inherent

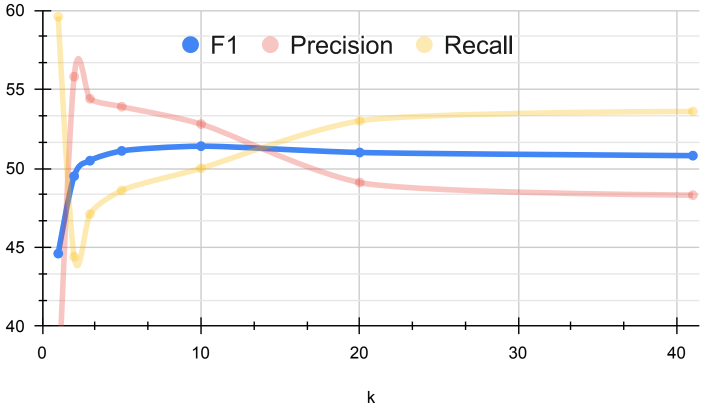


Figure 6.8: F1, Precision, and Recall scores as a function of the number of attributes used in the prompt during in-domain evaluation for the k-attribute model.

bias towards generating attributes that appeared in the training data, even when prompted with new ones. Their performance is in fact poor in the out-of-domain setting (2.2 F1, Table 6.3). Now the question is whether this is a limitation of the reasoning abilities of the multi-attribute models or a bias introduced by its training scheme.

We propose the k-attribute model to alleviate training biases by randomizing the queried attributes. Notably, this model still maintains the core assumptions behind the multi-attribute prompt model of querying multiple attributes at once. We observe that this simple technique results in the same in-domain F1 score as the all-attribute prompt model, while significantly improving its out-of-domain performance. This shows that the observed limitations with the all-attribute prompt model are due to training biases that prevent the model from generalizing to unseen attributes.

Once trained, the k-attribute prompt model can be queried with varying number of attributes. In Figure 6.8, we plot the performance of the model as a function of the number of attributes used in the query during in-domain evaluation. We observe a drop in performance when the model is queried with a single attribute (similar to the single-attribute prompt models). The performance is highest around 10 attributes and drops slightly beyond that. We also observe that by varying k, we can modulate precision and recall, suggesting that there are both lower and upper bounds on the optimal number of attributes that LLMs can consider at once.

We also experimented by grouping attributes in a prompt based on their semantic similarity, but this did not yield any significant changes in performance. We leave it to future work to investigate further how to optimally choose the groups to use in a prompt during training and inference.

6.8.3 Degree of Generalization Abilities

A major obstacle for NLP models is to apply the reasoning patterns they have learned to unseen attributes. Although the overall performance is lower in out-of-domain (best F1 = 23.5) compared to in-domain experiments (best F1 = 51.4), we observe that it varies significantly across different attributes. In this part of our analysis, we investigate the models' generalization abilities to out-of-domain attributes, based on their relation to in-domain attributes.

Essentially we identify two types of out-of-domain attributes: (1) these that are semantically similar to some in-domain attribute(s), and (2) these that have no similarity to any in-domain attribute. These two groups of attributes also evaluate the degree of the model's generalization abilities, as it is easier to generalize to different verbalizations of a previously seen attribute than to a completely new concept. For this part of the analysis we use the RoBERTa single-attribute prompt model, as it has the best out-of-domain performance.

Out-of-domain attribute	**In-domain synonym/antonym**
activity	motion
angle	orientation
area	shape
balance	weight
capacity	amount
consistency	stability
contents	composition
direction	orientation
flexibility	stability
granularity	composition
height	length
hydration	wetness
intensity	brightness
quantity	amount
safety	speed
softness	hardness
tenseness	pressure
tension	pressure
thermal conductivity	electric conductivity
tightness	pressure
width	length

Table 6.5: The most semantically similar in-domain attribute for each out-of-domain attribute (Group Matched).

To identify related attributes, we firstly use cosine similarity distance on top of an encoder trained for semantic similarity [Reimers and Gurevych, 2019]. After manual curation, we identify 21 out-of-domain attributes that are closely related to in-domain attributes (Group Matched), as we see in Table 6.5. The 20 remaining out-of-domain attributes are more dissimilar and do not have matching in-domain attributes (Group Dissimilar).

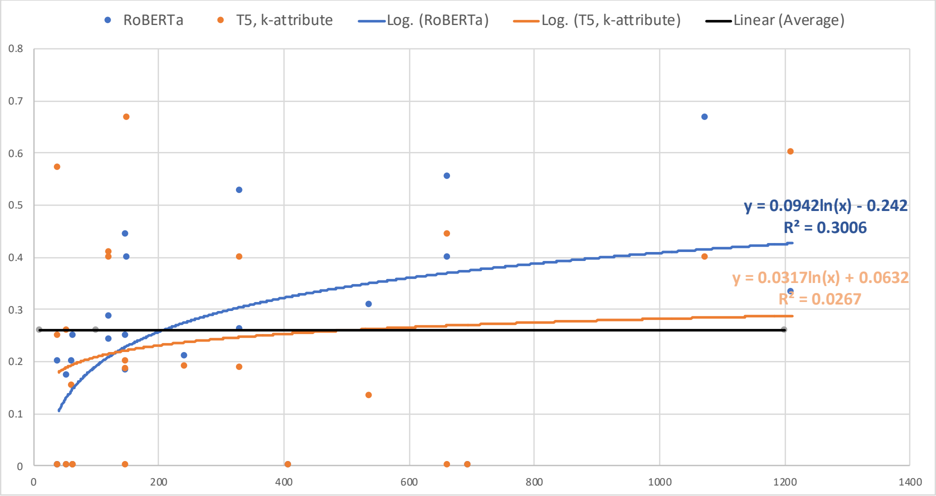


Figure 6.9: Out-of-domain performance in Matched group as a function of synonym's frequency in training data. **T5 k-attribute prompt shows no correlation between performance and synonym frequency, unlike RoBERTa single-attribute prompt (medium correlation).**

For each of the two groups (Group Matched and Group Dissimilar), we estimate the weighted-F1 score. We observe that Group Matched reaches **F1 = 29.4**, while Group Dissimilar **F1 = 13.6**. For Group Matched, we also verify that the model's performance on closely related attributes is similar by measuring their Pearson correlation, which is $r = 0.67$ (p-value < 0.05). Both results indicate that

the model understands the semantics of the attributes despite different verbalizations, however, it struggles with more complex reasoning mechanisms, such as applying the acquired patterns to entirely new attributes.

In the second part of our analysis, we further investigate the out-of-domain per-attribute performance in the Matched Group. More specifically, we study how the out-of-domain performance of an attribute relates to its in-domain synonym's frequency. For this part, we used both the T5 k-attribute prompt and the RoBERTa single-attribute prompt models. As we observe in Figure 6.9, the per-attribute performance of the T5 k-attribute prompt model is independent of the frequency of the attribute's synonym. However, the RoBERTa single-attribute prompt model has moderate correlation between performance and synonym's frequency. Combining this observation with our results in subsection 6.8.1, we conclude that the T5 k-attribute prompt model has overall better generalization abilities that do not depend on the frequency of an attribute or its verbalization, as long as it has previously seen some change-of-state of a similar attribute.

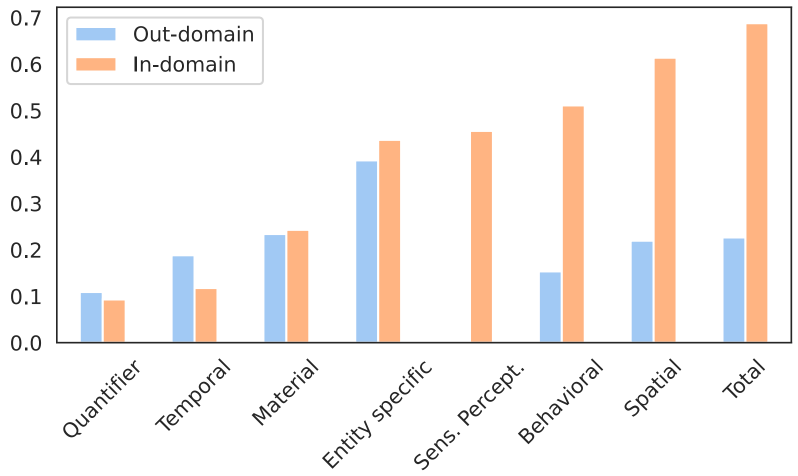


Figure 6.10: F1 scores per semantic type of attributes. **Model struggles to generalize to unseen attribute of *Spatial* and *Behavioral* types, highlighting that these semantic types are particularly challenging.**

6.8.4 Challenging semantic types

In the final part of our analysis, we explore why some classes of physical attributes appear to be inherently more difficult for LLMs. More specifically, we manually design an ontology of attributes into seven major semantic types and then grouped each in-domain and out-of-domain attribute according to the information it encodes, as seen in Table 6.6. Then, we study the performance for each semantic type, both in-domain and out-of-domain. Via this analysis we aim to identify evaluate each semantic type with respect to: (1) in-domain performance, and (2) generalization to unseen attributes. For this analysis we use RoBERTa single-attribute prompt, as it has the best out-of-domain performance.

As shown in Figure 6.10, with respect to in-domain performance, the model struggles to capture *Quantifiers* and *Temporal* semantic types, which are known to be challenging for current LLMs [Ravichander et al., 2019]. On the other hand, the semantic types *Spatial, Behavioral, Sensory Perception* and *Entity Specific* have high weighted-F1 score in in-domain experiments.

Our second set of observations are related to the differences between in-domain and out-of-domain performance. More specifically, we first observe that the *Entity-specific* and *Material* semantic types are equally challenging for both in-domain and out-of-domain attributes. These semantic types tend to describe inherent properties of an entity, such as *fullness*, that can only change due to very specific events, such as *put X into Y*.

On the other hand, the semantic types *Spatial* and *Behavioral* have a large discrepancy between in-domain and out-of-domain performance. This is surprising given that these semantic types contain high-frequency attributes, like *location*, and, thus, more training instances. This observation highlights

Semantic Type	In-domain Attributes	Out-of-domain Attributes
Spatial	location, volume, shape, size, orientation, length, distance, organization	angle, direction, area, height, width, pose, posture, spacial relation
Material	texture, electric conductivity, thickness, hardness, strength, pressure	tenseness, tension, tightness, softness, material, flexibility, thermal conductivity, density, granularity
Entity-Specific	cleanness, wetness, fullness, ownership, openness, cost, composition, coverage, focus	contents, wholeness, capacity, hydration, consumption, documentation, emotional state, pain, usage
Behavioral	knowledge, speed, motion, stability, complexity, skill	activity, balance, consistency, safety, familiarity, exposure, viability, resistance
Quantifier	amount	intensity, quantity, magnitude
Temporal	availability	age, life, existence, time
Sensory Perception	visibility	color, taste, temperature, smell, sound, appearance, weight, brightness

Table 6.6: Semantic types of attributes, both in-domain and out-of-domain. This ontology is constructed manually to study performance and generalization abilities per semantic type.

the limitations of current models to predict physical changes outside controlled environments and that future work on physical event implications should evaluate models with respect to their generalization abilities, besides in-domain performance.

6.9 Error Analysis & Challenges for LLMs

As we observed in subsection 6.4.2, performance drops significantly across all models when evaluated in out-of-domain attributes. Through our analysis section we investigated two groups of attributes based on the existence of an in-domain synonym and the frequency of such a synonym. Furthermore, we created a taxonomy of attributes to their semantic types and analyzed performance per semantic type.

Both these analyses and our results assume that the annotations are exhaustive, thus, no additional attributes from the given attribute list change. However, since the out-of-domain attributes were introduced by Amazon Turkers, there are attributes with significant overlap such as *angle* and *direction*. These attributes may frequently change together, but given that each human annotator had to introduce their own attributes, sparsity of annotations in such attributes becomes unavoidable.

Due to these reasons we hypothesize that our models generate correct attribute changes that are not present in the annotations and, thus, the real performance is higher than what is reflected by the F1 score in our main results section. To confirm this hypothesis and identify the major challenges of the models, we performed a manual error analysis. During this analysis, we inspect the output of our best-performing models: the T5 k-attribute prompt model and the RoBERTa single-attribute prompt. Given the expected attribute correlations, our error analysis is per-instance and not per-attribute, where an instance is the pair of context and entity, and each instance may contain several types of errors.

In Table 6.7 we show some real instances that we used in our error analysis. Although for each instance all the out-of-domain attributes were queried, for brevity we only show attributes that were identified as changed by either model or by the annotations. We observe that in many of these examples the models predict attribute changes that are correct, despite not being captured by the annotations. Such cases are Example 2, Example 4 and Example 5, where the T5 k-attribute prompt correctly predicts attributes that were not identified by the annotators. These attributes are not necessarily related to the annotated attribute, such as *width* and *resistance* in Example 2, or *hydration* and *softness* in Example 4. However, some other instances may have predicted attributes that are closely related to the annotated attribute, as we see in Example 1, where *posture* and *angle* oftentimes change together.

Our final observation from Table 6.7 is that the models are able to correctly predict attributes that require some common sense knowledge, which was not part of the provided context. For example, T5 k-attribute prompt predicts in Example 4 that *soaking beans* implies that *softness* changes, something that is not as an obvious conclusion as the change of *hydration*. Even more, in Example 5 we observe that the model is able to understand the intent of the paragraph, which is to change the *softness of lips*. These examples show that the T5 k-attribute prompt model is able to perform some degree of reasoning, even for predictions that were considered wrong due to missing annotations.

Based on our analysis of the full out-of-domain dataset, we identify four major types of errors according to the degree of understanding of context and entities involved. The first category (*Additional attributes*) includes errors where the model predicts attribute changes that are correct, but are missing from the annotations. This error type shows that the model is performing as expected and, instead, it was marked as an error due to the inadequacy of the annotations. We found that almost 53% (T5 k-attribute prompt) and 44% (RoBERTa single-attribute) of the instances in this category refer to attributes that are synonyms to the annotated attributes.

The second error category (*Wrong context*) includes attribute changes that are applicable to the entity, but are incorrect given the context. This error type shows that the model captures some relations about the entity involved and how they are typically affected, but struggles to understand the consequence of the participant events.

The third category (*Wrong context & entity*) consists of attribute changes that are entirely wrong

Example 1	
Context: Begin by standing in Mountain Pose. Bend your right leg back and hold on to the inside of your foot behind you with your right hand.	**Entity:** person **Annotated Attributes:** balance
T5 k-attribute prompts: Consider the following attributes: flexibility, angle, hydration, consumption. Which attribute changed for the person?	**T5 k-attribute output:** posture, flexibility, angle, pose
RoBERTa single-attribute prompts: Is the flexibility of the person different? Is the viability of the person different?	**RoBERTa single-attribute output:** No Yes
Example 2	
Context: Cut off a corner of a yeast packet.	**Entity:** packet **Annotated Attributes:** resistance
T5 k-attribute prompts: Consider the following attributes: contents, angle, width, resistance, softness. Which attribute changed for the packet?	**T5 k-attribute output:** contents, width
RoBERTa single-attribute prompts: Is the width of the packet different? Is the resistance of the packet different?	**RoBERTa single-attribute output:** Yes No
Example 3	
Context: Drink a glass of hot milk.	**Entity:** body **Annotated Attributes:** thermal conductivity
T5 k-attribute prompts: Consider the following attributes: contents, hydration, thermal conductivity. Which attribute changed for the body?	**T5 k-attribute output:** thermal conductivity
RoBERTa single-attribute prompts: Is the thermal conductivity of the body different? Is the hydration of the body different?	**RoBERTa single-attribute output:** No Yes
Example 4	
Context: Soak the dried beans and lentils overnight in a large bowl.	**Entity:** beans **Annotated Attributes:** hydration
T5 k-attribute prompts: Consider the following attributes: softness, contents, granularity, hydration. Which attribute changed for the beans?	**T5 k-attribute output:** softness
RoBERTa single-attribute prompts: Is the hydration of the beans different? Is the softness of the beans different?	**RoBERTa single-attribute output:** No No
Example 5	
Context: Take the honey and mix it with the sugar, then add in a little bit of Vaseline or petroleum jelly. When the mixture is all gritty, apply it on to your lips as you would with lip balm. Leave on the mixture for about one minute.	**Entity:** lips **Annotated Attributes:** granularity
T5 k-attribute prompts: Consider the following attributes: softness, pain, granularity. Which attribute changed for the lips?	**T5 k-attribute output:** softness, pain
RoBERTa single-attribute prompts: Is the softness of the lips different? Is the granularity of the lips different?	**RoBERTa single-attribute output:** No No

Table 6.7: Examples from out-of-domain and model predictions for the T5 k-attribute prompt and the RoBERTa single-attribute prompt models. **In many instances the predicted attribute is correct, but the annotations fail to reflect this.**

for both the given entity and the context. This is the most severe error since it shows that the model is not able to correctly predict neither the attributes that apply to the entity nor the potential implications of the event.

The fourth category (*No prediction*) consists of instances with null predictions. Here the model decides that there was no attribute change from the given list of attributes. Although such behavior is preferred to a wrong attribute prediction and is expected in out-of-domain scenarios, it results in significant drop in recall for both models.

Error Type	**T5, k-attribute**	**RoBERTa, single-attribute**
Additional attributes	41.5%	25.4%
Wrong context	7.6 %	6.5%
Wrong entity & context	2.7%	20.7%
No prediction	48.2%	47.4 %

Table 6.8: Error categories and prevalence of each category as a percentage of the number of instances. Based on out-of-domain attributes. **Many of the errors are due to missing annotations, which shows that the real out-of-domain performance (and, thus, generalization abilities) of T5 k-attribute prompt is higher than reported in subsection 6.7.2.**

As we show in Table 6.8, the most prominent error type for both models is *No prediction*, accounting for almost half of the errors. This highlights that both models struggle to identify which out-of-domain attributes are relevant to a particular context and entity, showing weaknesses in the generalization abilities. On the other hand, it is positive that they are self-aware of their limitations and the semantic differences across in-domain and out-of-domain attributes, which leads them to avoid wrong predictions in the presence of uncertainty. This observation shows that we can create high-precision systems with a certain degree of generalization abilities to unseen attributes.

The second most prominent error is *Additional attributes*, where the models make correct predictions about attribute changes that the annotations failed to include. This category does not reflect a failure of the models, but instead failures of the evaluation and the dataset. Unlike in-domain attributes, which is a predefined list of attributes given to human annotators, the out-of-domain attributes were introduced by the human annotators themselves. This results in significant noise and inconsistencies across this set of attributes and the annotations that mention them. As we mention in Section 6.4.2, we spent a considerable amount of effort to curate the attributes and remove noise and duplicates. However, since there was no given list of attributes, each annotator may introduce attributes that were not considered by others while annotating different instances. This is particularly prominent between almost synonymous concepts, such as *width* and *size*, which oftentimes change together. As we see in Table 6.8, this category is responsible for 41.5% of errors for T5 k-attribute prompt and 25.4% for RoBERTa single-attribute. These results highlight that our systems' real performance is significantly higher than what was reported in our previous section, since this error wrongfully hurts our reported precision and recall. Although performance for in-domain attributes is still better than out-of-domain, this gap is in fact more narrow than what we thought, showing that the models are able to generalize to some degree to unseen attributes.

This error is divided into two subcategories according to the type of additional attributes introduced by the model. In the first category, these attributes are synonyms of the annotated attributes and could replace them in the particular instance. In the second category, the additional attributes are significantly different and complement the annotated attributes, such as *flexibility* and *size*. We found that the first category is responsible for 53% (T5 k-attribute prompt) and 44% (RoBERTa single-attribute) of the instances with *Additional attributes* errors. This observation shows that the k-attribute prompt model is able to generate more detailed predictions of attribute changes.

The third category of error is *Wrong context*, which accounts only for 7.6% and 6.5% of instances for the k-attribute and single-attribute prompt models, respectively. This reflects a real challenge of the problem of event implications, since correctly answering these queries requires a deep understanding of context instead of focusing only on the entity. Having fewer errors from this category shows that the models are able to reason about event implications and how they affect entities.

The final error category is *Wrong entity and context*, which includes attribute changes with no obvious link to the entities or context. While this error is very rare for the T5 k-attribute model (only 2.7%), it is more frequent for the RoBERTa single-attribute model (20.7%). Given that this is the most severe error category, the large difference shows that the T5 k-attribute prompt model is significantly better than the single-attribute prompt to generate correct attribute changes.

6.10 Conclusion

In Chapter 6 we discussed the problem of predicting event implications as entity change-of-state with respect to physical attributes. In our work, we decompose the entity semantics into facets that guide the model to retain the aspect of meaning relevant to the underlying reasoning rule we aim to imitate. Compared to the multi-faceted representations of Chapter 3 where the facets were given in the form of a relation, Chapter 6 assumes that a LLM only requires a verbalization of the facets without a schematic representation of their values. By using prompting, a prominent modern technique, as a means of communication with a LLM, we convey which facets are most useful to complete the task. Our approach relies on two dimensions important for the facetization of meaning: (1) verbalization of the facets, and (2) diversification of the facet decomposition per instance, which boosts the generalization abilities of a model.

Predicting physical changes due to events is a challenging problem for current models, especially in out-of-domain or limited-data scenarios. We show that, by using proper task formulation, LLMs can learn physical event implications even without physical interactions. Future work should explore the question of whether physical interactions are necessary in more complex and realistic settings, by (1) providing more challenging datasets that test the model limitations, and (2) ensure a fair comparison of the language-only baselines.

Furthermore, we show that the performance of a LLM may significantly vary based on how we use it, and, overall, LLMs can benefit from: (1) verbalizing the attributes, (2) varying the prompt information content across instances, and (3) querying multiple attributes in the same instance. By following these guidelines, we show significant improvements in unseen attributes and attributes of low-frequency. Last, our error analysis and discussion sections provide useful insights for future work, with respect to prompt content and shortcomings of the current datasets that study physical event implications.

Chapter 7

Conclusion

Through this thesis we show that meaning has several aspects and, based on the context or task, we might be interested to retain only one of these aspects in our representations. We discussed how meaning can be decomposed to facets in representations for different semantic units: entities (Chapter 3), sentences (Chapter 4) and events (Chapter 5). We show that by decomposing meaning we achieve significant improvements with respect to both performance and generalization abilities of a model. These improvements are particularly important in limited-data scenarios, where a model is susceptible to shortcut learning.

In the final part of the thesis (Chapter 6), we explore meaning decomposition as a means to share relevant information with the model and as a guide to learn reasoning patterns. Even high-performing models such as LLMs are oftentimes unable to deduce which aspect of the entity or sentence is relevant to answer complex questions, such as in commonsense reasoning tasks. While in previous chapters we studied tasks that only require the extraction of the relevant aspect of meaning, in this final chapter we study complex reasoning challenges, where the model must both extract the relevant aspect of meaning and combine different pieces of information together in order to learn reasoning patterns. By studying the task of event implications as entity change-of-state with respect to a set of physical attributes, we evaluate the reasoning abilities of our models and conclude that meaning decomposition functions as a catalyst in learning, which guides models to learn reasoning patterns that generalize to new attributes and events.

7.1 Summary of Key Contributions

This thesis has the following contributions:

- Chapter 3 proposes an entity representation that decomposes meaning into a fixed set of dimensions based on relations found in definitions. This representation exploits the fact that definitions contain essential knowledge that generalizes across all instances of a concept. We show that decomposing meaning leads to higher performance and interpretability in limited-data scenarios.

- Chapter 4 discusses the problem of covariate shift, as a set of systematic differences across training and test data due to an underlying cause. It explores how such differences may negatively influence performance by encoding the aspect of meaning that is not appropriate for the task, while ignoring other important information (shortcut learning). We propose a novel approach to tackle this problem by removing information from sentence representations that correlates with the cause of the covariate shift.

- Chapter 5 discusses event representations in the form of semantic frames and sub-events. Both representations aim to capture and understand the semantics of an event by decomposing it into different aspects. While semantic frames provide surface information about the event extracted from text, a representation via its sub-events provides explanations of the semantics and how the event fits in the world.

- Chapter 6 discusses event implications as entity change-of-state with respect to physical attributes. The chapter explores two components: (1) how meaning decomposition influences performance and generalization of the models, and (2) the effect of language in learning compared to physical interactions. To examine the first component we design training mechanisms that vary with respect to their meaning decomposition. Through this work, we show that the generalization abilities of a model heavily rely on the meaning decomposition of entities via their attributes, which function as filters to retain the relevant aspects of meaning. Furthermore, we show that, despite the importance of physical interactions, we can still learn reasoning patterns about physical event implications solely based on language.

7.2 Future Directions

In this section we describe possible future directions that are motivated by this thesis. Although this list is not exhaustive, it is based on the main challenges and avenues for potential solutions that we have identified from our research conclusions.

7.2.1 Using Definitions as Source of Knowledge

Definitions were successfully used in Chapter 3 to form entity representations. Definitions are often used in language generation, where the main goal is to verify that a representation or model contains sufficient information and is able to abstract over this information in order to generate a correct definition.

Future directions in this area could focus on using definitions as a source of information instead of an attainable goal. Due to manual curation, definitions capture only important aspects of meaning that generalize across all instances of a class. This makes them a very useful tool to combine with LLMs, particularly in limited-data scenarios or rare words, where the model cannot see enough instances in order to be able to abstract over their meaning.

7.2.2 Learning Meaning from Vision & Language

Recent work in the area of multi-modal deep learning has shown that we can achieve significant improvements in various NLP tasks by combining information across multiple modalities, such as vision and speech. Research in multi-modal learning focuses primarily on emotion recognition [Busso et al., 2008, Zadeh et al., 2016] or sentiment analysis tasks [Zadeh et al., 2016], and, more recently, on visual question-answering problems. Visual QA involves simple reasoning tasks where, given an image and a natural language question about the image, the model has to generate or predict the correct answer [Hudson and Manning, 2019, Singh et al., 2019, Gurari et al., 2018, Goyal et al., 2017].

Despite the extensive use of multi-modal techniques for simple reasoning questions, these tasks require only one modality to learn (typically vision) and the other modality (typically language) to query and evaluate the model. Such an approach equates the problem to the task of object recognition, without learning any significant knowledge from language. However, as we show in Chapter 6, a model can learn from language even for physical reasoning challenges and, potentially benefit if both modalities are combined. While language provides the right level of details needed to answer reasoning questions, vision requires fewer data to learn physical properties of objects. Thus, future directions should focus on developing models learning from both modalities, where the model can both identify the relevant aspect of meaning and ground it in the physical world.

7.2.3 Determining Importance of Sub-events

In Chapter 5 we discussed our approach to extract sub-events during a large-scale crisis event. Through our work we observe that a major challenge is to determine which sub-events are important and could potentially change the outcome of the large event. Future research directions in that area may focus on quantifying the essentiality or importance of sub-events based on their influence on other sub-events and the large event as a whole.

7.2.4 Information Content in Query Formulation

In Chapter 6 we proposed different mechanisms to decompose an entity's meaning based on a set of attributes. An important observation from our experiments, is that the generalization properties of the model depend on how we decompose meaning per instance, even if the same amount of information remains through the entire training set. More specifically, we noticed that if we vary the attributes that we query the model about, it learns to better generalize to unseen attributes.

We believe that future work should explore how a model can maximize its learning capacity based on different decompositions of meaning. In our approach we randomly choose a subset of the attributes per query, but, potentially, a model benefits more by selecting more similar (or different) attributes per query and, thus, controlling the level of abstraction of the knowledge we want to infuse our models with.

7.2.5 Event Implications for Non-physical Attributes

Another observation from Chapter 6 is that attributes related to skills and behavior are more difficult to learn than other attributes. Although we only studied attributes that are related to physical event implications, some of these events also affected the object's abilities, such as *balance*.

Interesting future directions may analyze event implications for these inherent abilities of objects. Such attributes are more difficult to study because one can only test the acquisition or loss of a certain ability during specific events (e.g., moving an object to test its balance). On the other hand, to study changes in strictly physical characteristics such as *length*, we only need to observe the object. This distinction makes these attributes particularly difficult to predict and shows the complex nature of physical event implications.

7.2.6 Modeling Event Chains & Causality

In Chapter 6 we studied event implications as entities change-of-state. However, events do not cause only changes in entities, but could also cause other events. Predicting causal relations is an extremely difficult problem due to the number of causes or effects that an event may have. Thus, in order to predict causality we do not only need to model the event implications of a single event but a combination or series of events, typically called event chains. Naturally, although this problem adds further complexity to current reasoning challenges, it is fundamental in order to create models that can reason about the world.

7.2.7 Complex Coreference Resolution

Another future direction motivated by multi-faceted representations involves coreference resolution. Entity coreference occurs when two or more expressions (mentions) refer to the same person or thing. The task of coreference resolution aims to identify phrases that refer to the same entity, since each phrase may help us to extract more information about the entity.

In some cases, the coreferent mentions participate in similar or compatible events. This certainly facilitates the task of coreference resolution, since a similar aspect of the entity semantics is used in each sentence meaning. However, there are certain cases of more complex coreference, where the entity participates in unexpected events and, thus, the relevant aspect of its meaning is completely different across sentences. For example, consider the entity *paving stones*. A typical use of the entity may refer to the material aspect of it, as in *the road was constructed using paving stones*. However, we may later read the sentence *during the protest, the paving stones from the road were used to break the car windows*. Although both mentions refer to the same object, the relevant aspect of meaning for the latter is the hardness and sharp edges of the paving stone, making it a weapon.

By representing different aspects of meaning and being able to choose what is relevant to context, future work can construct a theoretical framework to model such difficult cases of coreference.

Bibliography

URL http://dcs.gla.ac.uk/~richardm/TREC_IS/2020/oldindex.html.

Dhekar Abhik and Durga Toshniwal. Sub-event detection during natural hazards using features of social media data. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 783–788, 2013.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics, 2009.

Firoj Alam, Hassan Sajjad, Muhammad Imran, and Ferda Ofli. Standardizing and benchmarking crisis-related social media datasets for humanitarian information processing. *arXiv preprint arXiv:2004.06774*, 2020.

Luis Espinosa Anke and Steven Schockaert. Syntactically aware neural architectures for definition extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 378–385, 2018.

Chidubem Arachie, Manas Gaur, Sam Anzaroot, William Groves, Ke Zhang, and Alejandro Jaimes. Unsupervised detection of sub-events in large scale disasters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 354–361, 2020.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, 1998a.

Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998b.

Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. Phyre: A new benchmark for physical reasoning. 2019.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186, 2013.

Jonathan Barnes. Posterior analytics. 1994.

Marco Baroni and Alessandro Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, 2010.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. Sub-event detection from twitter streams as a sequence labeling problem. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 745–750, 2019.

Yonatan Belinkov, Adam Poliak, Stuart M Shieber, Benjamin Van Durme, and Alexander M Rush. On adversarial removal of hypothesis-only bias in natural language inference. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (* SEM 2019)*, pages 256–262, 2019.

Jean-Louis Binot and Karen Jensen. A semantic expert using an online standard dictionary. In *Natural Language Processing: The PLNLP Approach*, pages 135–147. Springer, 1993.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience Grounds Language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020a. URL https://arxiv.org/abs/2004.10151.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020b.

Guido Boella and Luigi Di Caro. Extracting definitions and hypernym relations relying on syntactic dependencies and support vector machines. In *51st Annual Meeting of the Association for Computational Linguistics, ACL 2013*, volume 2, pages 532–537. Association for Computational Linguistics (ACL), 2013.

Branimir Boguraev and James Pustejovsky. Lexical ambiguity and the role of knowledge representation in lexicon design. In *Proceedings of the 13th conference on Computational linguistics-Volume 2*, pages 36–41. Association for Computational Linguistics, 1990.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5: 135–146, 2017.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 491–500. ACM, 2019.

Tom Bosc and Pascal Vincent. Auto-encoding dictionary definitions into consistent word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1532, 2018.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. Simulating action dynamics with neural process networks. *arXiv preprint arXiv:1711.05313*, 2017.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*, 2019.

Marco Bressan, Flavio Chierichetti, Ravi Kumar, Stefano Leucci, and Alessandro Panconesi. Motif counting beyond five nodes. 12(4), 2018. ISSN 1556-4681. doi: 10.1145/3186586. URL https://doi.org/10.1145/3186586.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics, 2012.

Grégoire Burel, Hassan Saif, and Harith Alani. Semantic wide and deep learning for detecting crisis-information categories on social media. In *International semantic web conference*, pages 138–155. Springer, 2017.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359, 2008.

Nicoletta Calzolari. Detecting patterns in a lexical data base. In *10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, 1984.

Pengfei Cao, Yubo Chen, Jun Zhao, and Taifeng Wang. Incremental event detection via knowledge consolidation networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 707–717, 2020.

Cornelia Caragea, Adrian Silvescu, and Andrea H Tapia. Identifying informative messages in disaster events using convolutional neural networks. In *International Conference on Information Systems for Crisis Response and Management*, pages 137–147, 2016.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Guandan Chen, Nan Xu, and Weiji Mao. An encoder-memory-decoder framework for sub-event detection in social media. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1575–1578, 2018.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun 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)*, 2015.

Martin S Chodorow, Roy J Byrd, and George E Heidorn. Extracting semantic hierarchies from a large on-line dictionary. In *Proceedings of the 23rd annual meeting on Association for Computational Linguistics*, pages 299–304. Association for Computational Linguistics, 1985.

Kenneth Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018a.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018b.

Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. Everything happens for a reason: Discovering the purpose of actions in procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4496–4505, 2019.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. Frame-semantic parsing. *Computational Linguistics*, 2014.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. Learning dynamic context graphs for predicting social events. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1007–1016, 2019.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1585–1595, 2020.

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

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

Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, 2018.

Denis Emelin, Ronan Le Bras, Jena D Hwang, Maxwell Forbes, and Yejin Choi. Moral stories: Situated reasoning about norms, intents, actions, and their consequences. *arXiv preprint arXiv:2012.15738*, 2020.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. 2009.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, 2019.

Manaal Faruqui and Chris Dyer. Community evaluation and exchange of word vectors at wordvectors.org. In *Proceedings of ACL: System Demonstrations*, 2014.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, 2015.

Christiane Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer, 2010.

Charles J Fillmore. Frame semantics. In *Cognitive linguistics: Basic readings*, pages 373–400. De Gruyter Mouton, 2008.

Charles J Fillmore and Collin F Baker. Frame semantics for text understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*, volume 6, 2001.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131, 2002.

Maxwell Forbes, Jena D Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. Social chemistry 101: Learning to reason about social and moral norms. *arXiv preprint arXiv:2011.00620*, 2020.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haifa Zargayouna, and Thierry Charnois. Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688, 2018.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Reza Ghaeini, Xiaoli Z Fern, Liang Huang, and Prasad Tadepalli. Event nugget detection with forward-backward recurrent neural networks. In *The 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Edgar Herbert Granger. Aristotle on genus and differentia. *Journal of the History of Philosophy*, 22(1):1–23, 1984.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE, 2013.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969*, 2016.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics, 2009.

Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4: 17–30, 2016.

Starr Roxanne Hiltz, Jane A Kushma, and Linda Plotnick. Use of social media by us public sector emergency managers: Barriers and wish lists. In *ISCRAM*, 2014.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*, 2019.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. *arXiv preprint arXiv:2010.05953*, 2020.

Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Processing social media messages in mass emergency: A survey. *ACM Computing Surveys (CSUR)*, 47(4):1–38, 2015.

Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016a. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. *arXiv preprint arXiv:1605.05894*, 2016b.

Liwei Jiang, Antoine Bosselut, Chandra Bhagavatula, and Yejin Choi. " i'm not mad": Commonsense implications of negation and contradiction. *arXiv preprint arXiv:2104.06511*, 2021.

Shan Jiang, William Groves, Sam Anzaroot, and Alejandro Jaimes. Crisis sub-events on social media: A case study of wildfires. In *International Conference on Machine Learning AI for Social Good Workshop, Long Beach, United States, July*, volume 1, 2019.

Yaser Keneshloo, Jose Cadena, Gizem Korkmaz, and Naren Ramakrishnan. Detecting and forecasting domestic political crises: a graph-based approach. In *Proceedings of the 2014 ACM conference on Web science*, pages 192–196, 2014.

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

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Soumen Chakrabarti, et al. Imojie: Iterative memory-based joint open information extraction. *arXiv preprint arXiv:2005.08178*, 2020.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Gizem Korkmaz, Jose Cadena, Chris J Kuhlman, Achla Marathe, Anil Vullikanti, and Naren Ramakrishnan. Combining heterogeneous data sources for civil unrest forecasting. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 258–265, 2015.

Anna Kruspe. Few-shot tweet detection in emerging disaster events. *arXiv preprint arXiv:1910.02290*, 2019.

LDC. The ace 2005 evaluation plan. *http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/ac e05-evalplan.v3.pdf*, 2005.

Alex Leavitt and John J Robinson. The role of information visibility in network gatekeeping: Information aggregation on reddit during crisis events. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, pages 1246–1261, 2017.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in nlp. In *Proceedings of NAACL-HLT*, pages 681–691, 2016.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. Connecting the dots: Event graph schema induction with path language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 684–695, 2020.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning*

Architectures, pages 100–114, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.deelio-1.10.

Pengfei Liu, Xipeng Qiu, and Xuan-Jing Huang. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, 2017.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. Leveraging framenet to improve automatic event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

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

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. Gender bias in neural natural language processing. *arXiv preprint arXiv:1807.11714*, 2018.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393, 2018.

Carleen Maitland, L Ngamassi, and Andrea Tapia. Information management and technology issues addressed by humanitarian relief coordination bodies. In *Proceedings of the 6th International ISCRAM Conference*. Gothenburg, Sweden, 2009.

John McCarthy and Patrick J Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Readings in artificial intelligence*, pages 431–450. Elsevier, 1981.

Richard Mccreadie, Cody Buntain, and Ian Soboroff. Trec incident streams: Finding actionable information on social media. 2019.

Polykarpos Meladianos, Giannis Nikolentzos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. Degeneracy-based real-time sub-event detection in twitter stream. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9, 2015.

Polykarpos Meladianos, Christos Xypolopoulos, Giannis Nikolentzos, and Michalis Vazirgiannis. An optimization approach for sub-event detection and summarization in twitter. In *European Conference on Information Retrieval*, pages 481–493. Springer, 2018.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018a.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013a.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013b.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

George A Miller and Walter G Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint*, 2022.

Bhavana Dalvi Mishra, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. *arXiv preprint arXiv:1805.06975*, 2018.

Taro Miyazaki, Kiminobu Makino, Yuka Takei, Hiroki Okamoto, and Jun Goto. Label embedding using hierarchical structure of labels for twitter classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6318–6323, 2019.

Fons Moerdijk et al. Frames and semagrams. meaning description in the general dutch dictionary. In *Proceedings of the Thirteenth Euralex International Congress, EURALEX 2008*, 2008.

Julius M Moravcsik. Aitia as generative factor in aristotle's philosophy. *Dialogue: Canadian Philosophical Review/Revue canadienne de philosophie*, 14(4):622–638, 1975.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324, 2017.

Robert Munro. Subword and spatiotemporal models for identifying actionable information in haitian kreyol. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 68–77. Association for Computational Linguistics, 2011.

Tahora H Nazer, Guoliang Xue, Yusheng Ji, and Huan Liu. Intelligent disaster response via social media analysis a survey. *ACM SIGKDD Explorations Newsletter*, 19(1):46–59, 2017.

Graham Neubig, Yuichiroh Matsubayashi, Masato Hagiwara, and Koji Murakami. Safety information mining—what can nlp do in a disaster—. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 965–973, 2011.

Allen Newell, John C Shaw, and Herbert A Simon. Report on a general problem solving program. In *IFIP congress*, volume 256, page 64. Pittsburgh, PA, 1959.

Dat Nguyen, Kamela Ali Al Mannai, Shafiq Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. Robust classification of crisis-related data on social networks using convolutional neural networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 2017.

Dat Tien Nguyen, Shafiq Joty, Muhammad Imran, Hassan Sajjad, and Prasenjit Mitra. Applications of online deep learning for crisis response using social media information. *arXiv preprint arXiv:1610.01030*, 2016.

Thien Nguyen and Ralph Grishman. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Thien Huu Nguyen and Ralph Grishman. Event detection and domain adaptation with convolutional neural networks. In *ACL (2)*, 2015.

Yue Ning, Rongrong Tao, Chandan K Reddy, Huzefa Rangwala, James C Starz, and Naren Ramakrishnan. Staple: Spatio-temporal precursor learning for event forecasting. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 99–107. SIAM, 2018.

Yue Ning, Liang Zhao, Feng Chen, Chang-Tien Lu, and Huzefa Rangwala. Spatio-temporal event forecasting and precursor identification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3237–3238, 2019.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. Definition modeling: Learning to define word embeddings in natural language. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Swati Padhee, Tanay Kumar Saha, Joel Tetreault, and Alejandro Jaimes. Clustering of social media messages for humanitarian aid response during crisis, 2020.

Leysia Palen, Kenneth M Anderson, Gloria Mark, James Martin, Douglas Sicker, Martha Palmer, and Dirk Grunwald. A vision for technology-mediated support for public participation & assistance in mass emergencies & disasters. *ACM-BCS Visions of Computer Science 2010*, pages 1–12, 2010.

William Thomas Parry and Edward A Hacker. *Aristotelian logic*. Suny Press, 1991.

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Haoruo Peng, Yangqiu Song, and Dan Roth. Event detection and co-reference with minimal supervision. In *EMNLP*, 2016.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, 2018.

Linda Plotnick, Starr Roxanne Hiltz, Jane A Kushma, and Andrea H Tapia. Red tape: Attitudes and issues related to use of social media by us county-level emergency managers. In *ISCRAM*, 2015.

Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. Automatic sub-event detection in emergency management using social media. In *Proceedings of the 21st international conference on world wide web*, pages 683–686, 2012.

James Pustejovsky. The generative lexicon. *Computational linguistics*, 17(4):409–441, 1991.

Junaid Qadir, Anwaar Ali, Raihan ur Rasool, Andrej Zwitter, Arjuna Sathiaseelan, and Jon Crowcroft. Crisis analytics: big data-driven crisis response. *Journal of International Humanitarian Action*, 1(1):1–21, 2016.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

Naren Ramakrishnan, Patrick Butler, Sathappan Muthiah, Nathan Self, Rupinder Khandpur, Parang Saraf, Wei Wang, Jose Cadena, Anil Vullikanti, Gizem Korkmaz, et al. 'beating the news' with embers: forecasting civil unrest using open source indicators. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1799–1808, 2014.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A Smith, and Yejin Choi. Event2mind: Commonsense inference on events, intents, and reactions. *arXiv preprint arXiv:1805.06939*, 2018.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1033. URL `https://aclanthology.org/K19-1033`.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL `https://aclanthology.org/D19-1410`.

Christian Reuter, Marc-André Kaufhold, Thomas Spielhofer, and Anna Sophie Hahne. Social media in emergencies: a representative study on citizens' perception in germany. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):1–19, 2017.

Christian Reuter, Gerhard Backfried, Marc-André Kaufhold, and Fabian Spahr. Iscram turns 15: A trend analysis of social media papers 2004-2017. *Proceedings of ISCRAM*, 2018.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95, 2011.

Herbert Rubenstein and John B Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

Koustav Rudra, Pawan Goyal, Niloy Ganguly, Prasenjit Mitra, and Muhammad Imran. Identifying sub-events and summarizing disaster-related information from microblogs. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 265–274, 2018.

Tanay Kumar Saha and Mohammad Al Hasan. Finding network motifs using mcmc sampling. In Giuseppe Mangioni, Filippo Simini, Stephen Miles Uzzo, and Dashun Wang, editors, *Complex Networks VI*, pages 13–24, Cham, 2015. Springer International Publishing.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860, 2010.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. Social bias frames: Reasoning about social and power implications of language. *arXiv preprint arXiv:1911.03891*, 2019a.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035, 2019b.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. Visual definition modeling: Challenging vision & language models to define words and objects. 2022.

Karin Kipper Schuler. Verbnet: A broad-coverage, comprehensive verb lexicon. 2005.

J Cliff Shaw, Allen Newell, Herbert A Simon, and TO Ellis. A command structure for complex information processing. In *Proceedings of the May 6-8, 1958, western joint computer conference: contrasts in computers*, pages 119–128, 1958.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Robert Speer and Catherine Havasi. Representing general relational knowledge in conceptnet 5. In *LREC*, pages 3679–3686, 2012.

Evangelia Spiliopoulou, Eduard Hovy, and Teruko Mitamura. Event detection using frame-semantic parser. In *Proceedings of the Events and Stories in the News Workshop*, pages 15–20, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2703. URL https://www.aclweb.org/anthology/W17-2703.

Evangelia Spiliopoulou, Eduard Hovy, Alexander G Hauptmann, et al. Event-related bias removal for real-time disaster events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3858–3868, 2020a.

Evangelia Spiliopoulou, Artidoro Pagnoni, and Eduard Hovy. Definition frames: Using definitions for hybrid concept representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3060–3068, Barcelona, Spain (Online), December 2020b. International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.coling-main.273.

Evangelia Spiliopoulou, Tanay Kumar Saha, Joel Tetreault, and Alejandro Jaimes. A novel framework for detecting important subevents from crisis events via dynamic semantic graphs. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 249–259, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.wnut-1.28. URL https://aclanthology.org/2021.wnut-1.28.

PK Srijith, Mark Hepple, Kalina Bontcheva, and Daniel Preotiuc-Pietro. Sub-story detection in twitter with hierarchical dirichlet processes. *Information Processing & Management*, 53(4): 989–1003, 2017.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, 2018a.

Gabriel Stanovsky, Julian Michael, Luke S. Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *NAACL-HLT*, 2018b.

Jeannette N Sutton, Leysia Palen, and Irina Shklovski. Backchannels on the front lines: Emergency uses of social media in the 2007 southern california wildfires. 2008.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, 2019.

Niket Tandon, Bhavana Dalvi Mishra, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. Reasoning about actions and state changes by injecting commonsense knowledge. *arXiv preprint arXiv:1808.10012*, 2018.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, 2020.

Andrea Tapia, Louis-Marie Ngamassi Tchouakeu, Edgar Maldonado, Kang Zhao, Harold Robinson, and Carleen Maitland. Exploring barriers to coordination between humanitarian ngos: A comparative case study of two ngo's information technology coordination bodies. *International Journal of Information Systems and Social Change*, 2(2):1–25, 2011a.

Andrea H Tapia, Kartikeya Bajpai, Bernard J Jansen, John Yen, and Lee Giles. Seeking the trustworthy tweet: Can microblogged data fit the information needs of disaster response and humanitarian relief organizations. In *Proceedings of the 8th International ISCRAM Conference*, pages 1–10. ISCRAM Lisbon, Portugal, 2011b.

Irina P Temnikova, Carlos Castillo, and Sarah Vieweg. Emterms 1.0: A terminological resource for crisis tweets. In *ISCRAM*, 2015.

Julien Tissier, Christophe Gravier, and Amaury Habrard. Dict2vec: Learning word embeddings using lexical dictionaries. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 254–263, 2017.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5887–5897, 2020.

Hongwei Wang, Zixuan Zhang, Sha Li, Jiawei Han, Yizhou Sun, Hanghang Tong, Joseph P Olive, and Heng Ji. Schema-guided event graph completion. *arXiv preprint arXiv:2206.02921*, 2022.

Xiaofeng Wang, Matthew S Gerber, and Donald E Brown. Automatic crime prediction using events extracted from twitter posts. In *International conference on social computing, behavioral-cultural modeling, and prediction*, pages 231–238. Springer, 2012.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng 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)*, pages 998–1008, 2019.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. In *Advances in Neural Information Processing Systems*, pages 585–596, 2017.

Chen Xing, Yuan Wang, Jie Liu, Yalou Huang, and Wei-Ying Ma. Hashtag-based sub-event discovery using mutually generative lda in twitter. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, 2019.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*, 2016.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*, 2018.

Rowan Zellers, Ari Holtzman, Matthew E Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. Piglet: Language grounding through neuro-symbolic interaction in a 3d world. 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)*, pages 2040–2050, 2021.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340. ACM, 2018.

Liang Zhao, Qian Sun, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Multitask learning for spatio-temporal event forecasting. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1503–1512, 2015.

Michael Zock and Slaven Bilac. Word lookup on the basis of associations: From an idea to a roadmap. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*, ElectricDict '04, pages 29–35, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1610042.1610048`.

Chapter 8

Appendix

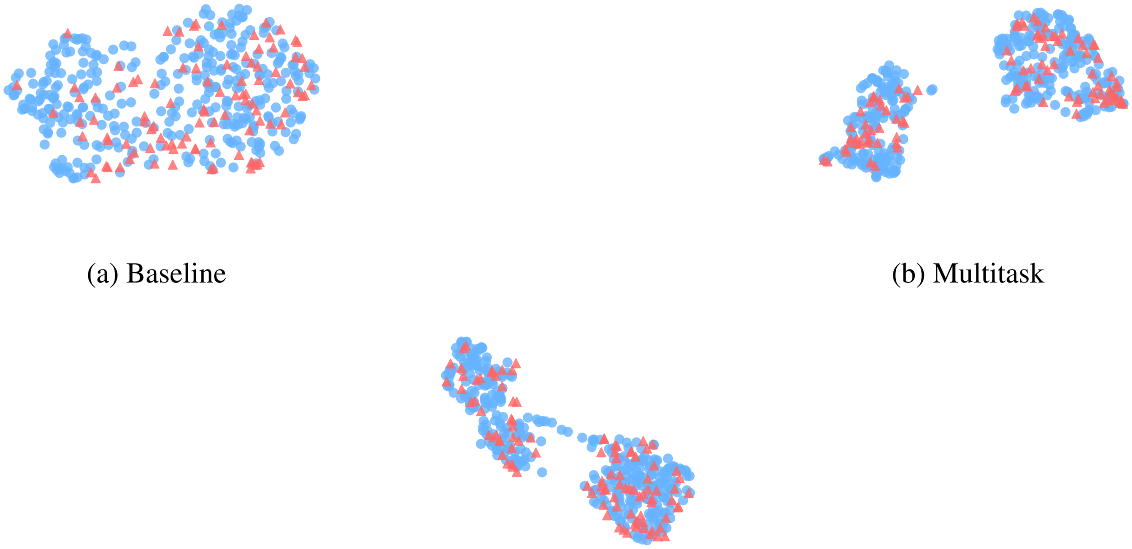
A Appendix

This appendix provides supplementary material for the work described in Chapter 4.

A.1 Embedding Visualization of Tweets

In Figure A.1 we visualize the 2-D projection of the 200-D tweet embeddings $h(t_i)$ for all the tweets from the 2012 Philippines flood data subset. We observe that the embedding distribution generated by each of the models is different. It is important to note that the multitask model achieved the lowest F1 score and the corresponding visualization shows two distinguishable clusters. The baseline embeddings are visualized as uniformly distributed, while the adversarial model's (best performing model) embeddings are mostly clustered in two groups with few points between the clusters.

The projection of the tweet embeddings from the adversarial model does not result into two separable clusters as seen in Figure A.1c, like the ones shown in Figure A.1b for the multitask embedding.



(a) Baseline

(b) Multitask

(c) Adversarial

Figure A.1: Tweet encoder embeddings visualization. Encoder embeddings dimensionality mapped to 2-D using UMAP. Blue circles represent non-critical tweets, red triangles represent critical tweets.

B Appendix

This appendix refers to details about work described in the second part of Chapter 5.

B.1 Crisis Event Types and Number of Tweets

In Table B.1 we show all the crisis events in our dataset and their corresponding event type. This data is publicly available by Alam et al. [2020].

Crisis Event Type	**Crisis Event Name**	**Crisis Event Name**
Crash/Explosion	2013 Glasgow helicopter crash 2013 Dhaka garment factory collapse 2012 Amuay Oil Refinery explosion in Venezuela 2013 Santiago de Compostela derailment in Spain 2014 Malaysia Airlines Flight 17	West Texas Fertilizer Company explosion (2) 2013 Lac-Mégantic rail disaster 2013 Kiss nightclub fire in Brazil 2013 Spuyten Duyvil derailment 2013 Chelyabinsk meteor
Earthquake	2013 Pakistan earthquake 2014 Iquique earthquake 2012 Costa Rica earthquake April 2015 Nepal earthquake 2013 Bohol earthquake	2017 Iran–Iraq earthquake 2014 South Napa earthquake 2017 Puebla earthquake 2012 Guatemala earthquake 2012 Northern Italy earthquakes
Flood	2012 Philipinnes Floods Srilanka Floods 2013 Manila Floods 2014 Pakistan Floods 2013 Queensland floods	2013 Alberta Floods (2) 2014 India Floods 2013 Italy Sardinia 2013 Queensland Floods-ontopic 2013 Colorado Floods
Hurricane/Typhoon/Cyclone	2015 Vanuatu Cyclone 2012 Philippines Typhoon-pablo 2012 US Sandy Hurricane 2014 Philippines Typhoon 2013 Phillipines Typhoon Yolanda 2014 Philippines Typhoon-Hagupit 2015 Vanuatu Cyclone-pam	Hurricane Harvey Hurricane Irma Hurricane Maria 2014 Mexico Hurricane Odile 2012 Sandy Hurricane-ontopic 2012 US Sandy Hurricane-a144267
Terrorist Attack	2013 Boston Bombings 2013 Boston Bombings-ontopic	2013 La Airport Shootings
Tornado	2011 Joplin Tornado-a121571 2011 Joplin Tornado-a131709	2013 Oklahoma Tornado-ontopic
Wildfire	2012 Colorado Wildfires (2)	2013 Australia Bushfire
Volcano	2014 Iceland Volcano	
Haze	2013 Singapore Haze	
Landslide	2014-2015 Worldwide Landslides	
Respiratory Syndrome	2014 Middle-east Respiratory-syndrome	2019 Covid pandemic
Ebola	2014 Worldwide Ebola	

Table B.1: All crisis events in the dataset.

C Appendix

This appendix refers to details about the work described in Chapter 6. Our experiments are built on top of the Huggingface library [Wolf et al., 2019]. The code for our experiments will be available as open source upon acceptance.

C.1 Metrics

Our task is a multi-label classification where, given some context and an entity of interest, we need to identify which attributes change. For most pairs *context, entity*, event implications affect only 1-2 attributes. This results in a few positive instances (i.e., attributes that change) and a large number of negative instances (i.e., attributes that do not change). Furthermore, we observe that the number of positive instances significantly varies across attributes: for example, in the training set of Open PI, *location* has 4505 positive instances, while *distance* only 53. Due to the significant label imbalance, in our experiments we report micro- Precision, Recall, and F1 for the positive instances, across labels. In addition to these metrics, we measure per-attribute Precision, Recall and F1 for both datasets.

C.2 Hyperparameters

We performed hyperparameter search in the following way. Based on the model size, we picked the largest batch size that could fit on our GPUs. Then we performed hyperparameter search on the dev set. We report in Table C.2 the hyperparameters we use in each case. We use Adam with betas (0.9,0.999) and ϵ =1e-08 for T5 experiments.

Data	Model	Epochs	Batch size	Learning Rate	Label Smoothing
PiGLET	RoBERTa, zero-prompt	30	20	4e-05	0.0
	T5 all-attr	50	32	3e-05	0.1
Open PI	RoBERTa, zero-prompt	20	32	1e-05	0.0
	RoBERTa, single-attr	6	16	1e-05	0.1
	T5 single-attr	8	16	5e-05	0.1
	T5 all-attr	8	16	5e-05	0.1
	T5 k-attr	10	16	5e-05	0.1

Table C.2: Hyperparameters.

To verify that model size differences do not impact our results, we also did experiments with RoBERTa-base zero-prompt, which shows very similar performance to RoBERTa-large zero-prompt.

C.3 In-domain Attributes and their Frequency

Attribute	Train	Dev	Test
location	4505	360	803
cleanness	1255	117	167
wetness	1211	80	215
temperature	1184	91	184
weight	1073	84	124
fullness	694	62	122
volume	676	56	174
composition	662	48	90
shape	538	55	65
texture	515	34	74
knowledge	409	27	119
orientation	330	15	45
color	292	13	33
size	264	26	50
power	245	11	18
organization	242	14	37
motion	242	15	33
ownership	212	6	19
availability	195	30	63
step	171	8	13
speed	151	3	18
pressure	148	4	14
taste	145	8	14
length	122	9	17
electric conductivity	121	9	18
smell	120	7	43
sound	68	6	6
brightness	65	0	7
thickness	64	4	16
strength	64	2	14
hardness	63	5	10
skill	62	3	4
openness	55	2	16
coverage	54	3	7
stability	54	6	14
focus	53	4	5
cost	53	6	9
distance	53	0	11
appearance	44	8	8
complexity	44	1	5
amount	40	3	16

Table C.3: Attribute occurrences in training, validation, and test sets.