Towards Narrative Understanding

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computational Linguistics
by Research

by

Ujwal Narayan N
20171170
ujwal.narayan@research.iiit.ac.in

International Institute of Information Technology
Hyderabad – 500 032, INDIA
March 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "*Towards Narrative Understanding*" by *Ujwal Narayan N*, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: *Prof. Manish Shrivastava*

To embracing the wibbly-wobbly, timey-wimey spirit of curiosity and adventure.

Acknowledgments

I would like to express my deepest gratitude to everyone who has supported and guided me throughout my thesis journey.

First and foremost, I would like to thank my family - my parents Udayashankar and Ishwari, and my sister Urja, for their unwavering love and constant support. Their encouragement has been invaluable to me in this endeavor.

I am immensely grateful to my advisor, Manish Shrivastava, for his passion, guidance, and encouragement. Our stimulating discussions have greatly contributed to my growth as a researcher.

I would also like to extend my thanks to IIIT, Hyderabad and its faculty members, especially Kamal Karlapalem, Vasudeva Varma, Dipti Misra Sharma, Radhika Mamidi, and Lini Thomas, for their mentorship and guidance throughout this journey.

My heartfelt gratitude goes to my roommate, Rohan Bhandari, for his patience and understanding during our time together before the COVID pandemic separated us.

I would like to thank my coauthors - Priyank, Sriharsh, Suhan, Sumukh, Alok, and many others - for their invaluable contributions and tireless efforts in bringing this work to completion.

Special thanks to Saujas, who, at this point, deserves to be an honorary author, for his meticulous proofreading and insightful suggestions that have significantly improved our work.

I am grateful to my friends - Deepti, Shelly, Sravani, Shantanu, Souvik, Sayar, Zubair, Aashna, Abighyan, Abhinav, Aaron, Ajay, Aryan, Athreya, Varun, Sriven, Suryansh, Debo, and many others whose names elude me as I type this at 3 AM - for their camaraderie and for making these years truly memorable.

I also want to express my gratitude to Manchester United and Royal Challengers Bangalore for teaching me the importance of perseverance and faith in the face of adversity and failure, year after year.

I would also like to extend my gratitude to the support staff at my institution for their invaluable assistance in various aspects of my research. Their dedication and hard work have been instrumental in the success of my thesis.

Thanks to Rammstein as well for providing the perfect soundtrack for my epic, thesis-writing montages.

Lastly, I would like to extend my appreciation to the anonymous reviewers of our papers, whose constructive feedback and insightful suggestions have greatly enhanced the quality of our work.

Thank you all for being an integral part of my journey.

Abstract

Research in NLP has seen increasing attention toward narrative understanding over the past decade. Narratives have a broad area of applications in various domains such as economics, political science and literature. Understanding narratives is critical to perform well in discourse-level tasks such as summarization, question answering, and multi-hop reasoning.

In this thesis, we explore a framework for understanding narratives. Narratives are broken down into two fundamental parts: events and characters. The task of understanding narratives is then posed as the task of understanding the interplay and relations between these two constituents. We focus on two major relations. How are characters related to other characters, i.e. the character-character relations, and how are events related to other events, i.e. event-event relations.

We utilize the concept of character arcs, a popular literary device that shows how the character changes with time to model character-character relations. We build MARCUS, an automated pipeline to generate and visualize these character arcs and character relations given a novel. We take two famous literary works, “Harry Potter” and “The Lord of the Rings” and analyze the character relations generated by MARCUS. We evaluate the quality of these arcs and relations through both quantitative and qualitative methods and show the effectiveness of the arcs created through MARCUS.

For event-event relations, we focus on the task of identifying temporal relations between an event pair in the narrative. Narratives, by their very nature, are discourse-level phenomenons. Yet, most of the current work on identifying event temporal relations focuses on local event pairs, i.e. event pairs found close together typically within adjacent sentences. We thus, build DELTA, a discourse-level event temporal relation dataset to facilitate document-level event timeline generation. In DELTA, we introduce the concept of multiple timelines, where we distinguish between the real timeline where the events have actually occurred, and hypothetical timelines with events that may not have actually happened. We also develop a new user-friendly annotation tool that not only streamlines and makes the timeline annotation efforts more efficient but also helps visualise and understand the timeline. We train strong baseline models based on RoBERTa to predict discourse-level event temporal relations. In addition, we qualitatively analyze the timelines generated by our dataset, and evaluate these timelines against the timelines generated by existing datasets.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The meaning of a sentence, while belonging to the meaning of the words present in it, can not be reduced to the sum of the meaning of the words. Similarly, in a discourse setting, the meaning of the discourse cannot be reduced to the sum of the sentences in it. How can we analyze and understand what the discourse is trying to convey? In this thesis, we look at a particular but prominent type of discourse: narrative discourse and build a framework to understand it.

Longacre defines narrative discourse as a type of discourse that is an account of events, usually in the past. Narrative discourse usually employs verbs of speech, motion, and action to describe a series of events that are contingent on one another and typically focuses on one or more performers of actions [Longacre (1990), Longacre (1996), Larson (1984)]. With the advent of mass media, it has become almost impossible to not encounter narrative discourses in our daily lives. Narratives can be found almost everywhere, and examples of narratives include newspaper reports, and many forms of literature such as novels and stories. Telling your friends what happened yesterday is also an example of narrative discourse.

Thus, to build language technologies that help us better our lives, it becomes important for machines to understand narratives. In this thesis, we look into narrative understanding, and we build a new framework for narrative understanding. To do so, we break down narratives into their two fundamental concepts:

1. Characters

2. Events

To understand narratives, it becomes important to understand the interplay between these two components, as shown in Fig 1.1, namely:

1. Character-Character relations

2. Event-Event relations

3. Character-Event relations

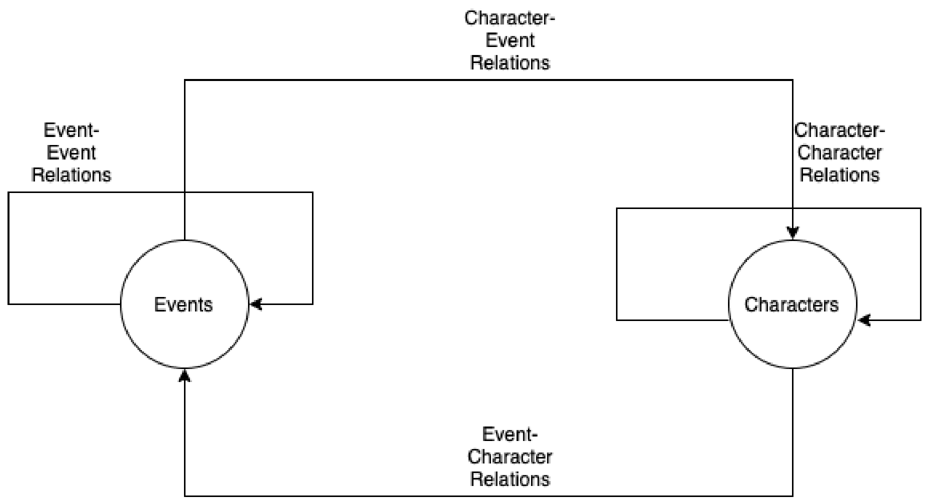


Figure 1.1: Interplay between characters and events

4. Event-Character relations

We explore the first two, i.e. character-character relations and event-event relations, in greater detail.

1.1 What is a narrative

In recent years, there has been an increased interest in the study of narratives and it has found applications in diverse areas. Nobel laureate R.J Shiller explains changes in human economic behaviour through changes in popular narratives and their interpretation when passed from person to person in a theory known as "narrative economics" [Shiller (2020)]. Simon Bushell, the founder of Sympower [1] utilizes the concept of narratives to address the "action gap" present between what scientists recommend for climate change and what the goverments, industries, and the public are doing [Bushell et al. (2017)]. The advent of social media has also increased the amount of polarization, be it political or moral, in society. Studies have shown that personal narratives bridge these divides better than just stating facts [Kubin et al. (2021)]. Even in fields like mental health, narratives have proven to be an invaluable tool, and Bruner's theory of narrative identity [Bruner (1991)] plays an important part in predicting well-being [Adler et al. (2016)]. Mental activity at a neural level, according to Dennet, can be explained as the continuous emergence and decay of narrative drifts [Dennett (2014)].

Because of its usage across multiple domains, it's hard to decide on an exact definition i.e what a narrative is [Ryan et al. (2007)]. The word narrative is often misused due to its recent popularity in place of words like theory, message, or explanation. At the basest level, Gennete [Genette et al. (1982)] defines narratives as follows,

> "a narrative can be thought of as the representation of an event or a sequence of events."

[1] `https://sympower.net/`

However, there can and often are multiple narrators and multiple narratees, and thus Prince expands the definition of a narrative to be

> "the representation of one or more real or fictive events communicated by one, two or several narrators to one, two or several narratees"

[Prince (2003)]. Narratives are not just any sequence. They are an ordered sequence, and we see temporality playing an essential role in ensuring that these events form a narrative. Multiple definitions of narratives focus on this aspect, with Ricouer [Ricoeur (1980)] stating

> "I take temporality to be that structure of existence that reaches language in narrativity, and narrativity to be the language structure that has temporality as its ultimate reference."

But is temporally ordered structures of events alone enough to constitute a narrative? A list of appointments is also a set of temporally ordered events. But that does not make it a narrative. A lot of work has gone into a deeper or a fuller definition of a narrative. Prince [Gerald (1982)] attempts to add logical relations to narratives by defining narratives to be

> "the representation of at least two real or fictive events in a time sequence, neither of which presupposes or entails the other."

Onega [Onega and Landa (2014)] emphasises on the importance of causality, as narratives have events which cause other events to occur, and defines narratives as

> "The semiotic representation of a sequence of events, meaningfully connected in a temporal and causal way."

Ball [Bal (1997)] expands on this with the addition of an experiencing subject and the concept of change, defining narratives to be

> "the transition from one state to another state, caused or experienced by actors."

. This is the definition of a narrative that we use in this thesis.

1.2 Fundamental Concepts of Narratives

As we saw in the earlier discussions on what a narrative is, one of the consistent themes that emerge is narrative as a "change of state". Aristotle is considered by many to be the father of Narratology. His work, "Poetics" is the first to establish the foundational work on narrative theory. Aristotle establishes the concepts of change through the use of the following notions.

- Beginning
- Middle

- End

Ricoeur [Ricoeur (1980)], and Bruner [Bruner (1991)], in particular, as shown in the earlier section, expand on these. Schärfe expands on these to establish the three grand principles of narratology. [Schärfe (2003)]

- The principle of succession
- The principle of transformation
- The principle of mediation

We'll expand on each of these principles below:

1. **The principle of succession**

   As Kant notes, "all things are in time, and in the substratum of time only coexistence or succession can occur" [Kant (1908)]. But in co-existence, things simply are, i.e. they exist without change, as if there is change then by definition, there is succession, and thus they are outside temporal constraints. Therefore things in co-existence are typically not part of narratives. So for narratives, incidents and objects must be arranged in sequence.

2. **The principle of transformation**

   This principle extends the Aristotelian principle of change. The principle of succession is closely tied to the principle of transformation, as succession lends itself to transformation. Succession generates a coherent narrative structure. The elements of this structure are then transformed from one situation to another.

3. **The principle of mediation**

   The meditation principle distinguishes narratives from other text forms like recipes and accounts. Mediation deals with the rationale, intention and relevance. This principle deals with the epistemological condition under which narrative structures are employed and the pragmatic condition intrinsic to storytelling.

If we look at these principles, they can be reduced to the interactions between characters and events. The principle of succession boils down to event-event temporal ordering. The principle of transformation is the interplay between characters and events. The actions of characters cause events. These events, in turn, shape the development of characters, which in turn controls the actions that they will take in the future, thus shaping future events. The changes in situations are fairly evident, as the events themselves indicate them. The changes in the character, however, can be seen through literary devices like character arcs. The principle of mediation involves the rationale behind the transformations and the intention behind the characters' actions. Thus, as shown above, are a natural consequence of the interplay between the events and the characters.

1.3 The interplay between Characters and Events

Broadly speaking, we can categorize the interplay between characters and events into two major categories:

- Homogeneous relations
- Heterogeneous relations

Homogeneous relations consist of relations between the same type of objects. Thus, event-event relations and character-character relations belong to this categroy. In this thesis, we focus on homogeneous relations, leaving heterogeneous relations consisting of the relations between events and characters as future work. Heterogeneous relations look into how characters are shaped by events and how the characters shape events themselves. This is beyond the scope of the thesis.

There are many different types of event-event relations. Events can be causally related to other events. For example, consider the following sentence. "Harry flicked the switch, and the stereo blasted some music". The event `flicked` causes the event `blasted`. Thus these events are causally related. Another type of event-event relation is the temporal relation. If an event A occurs before event B, then event A is temporally related to B with the relation "before". In this thesis, we focus only on the temporal relations between events.

We utilize the concept of character arcs to understand how relations between different characters change as the plot develops. We look at two popular literary works, "Lord of the Rings" and "Harry Potter" and analyze the evolving dynamics between the various characters.

1.4 Contributions of this thesis

The major contributions of the thesis are as follows.

- An event-centric framework for analyzing and understanding narratives based on the interplay between characters and events
- An automated way to generate and visualize character arcs for character pairs in a novel based on sentiment and emotions. Aggregating character arcs across all characters also help us view the narrative arc of the given novel.

- A new dataset and framework for identifying discourse level temporal relations between event pairs from narratives.

1.5 Thesis Outline

The thesis is organized into five chapters:

- **Chapter 2** explores the related work in the field. I explore prior work on computational narrative understanding. Existing literature relevant to the particular tasks this thesis covers in character-character relations and event-event relations is also explored in greater detail.
- **Chapter 3** deals with the work on character-character relations. In this, I utilize the theory of character arcs to show the character's development as the plot progresses. This graph is then used to analyze the relationship of the major characters in two popular literary works, Harry Potter and Lord of the Rings.
- **Chapter 4** showcases the work on event-event relations. Amongst the many event-event relations that exist, I focus on temporal relations. we showcase our efforts in building *DELTA*: a dataset for discourse-level timeline generation for a given narrative. I also built a strong baseline based on RoBERTa [Liu et al. (2019)] to validate our dataset.
- **Chapter 5** concludes the thesis and discusses the third type of relation, i.e. character-event relations. we explore further work, the challenges present and the shortcomings of the approaches explored in this thesis.

Chapter 2

Related Work

2.1 Temporal Relations between Events

The ACL Workshop on Temporal and Spatial Reasoning, 2001 and the LREC workshop Annotation Standards for Temporal Information in Natural Language, 2002 marked a renewed interest in the importance of temporal aspects in tasks like information retrieval and extraction. Works following these differed from the earlier work with the emphasis they placed on linguistic structures that encode and convey the temporal information present in the text. The development of TimeML specifications [Pustejovsky et al. (2005)], and the TimeBank Corpus [Pustejovsky et al. (2003)] paved the way for the task of temporal relation ordering.

TimeBank annotates temporal relations between any two events or temporal expressions with a fine-grained temporal relation scheme consisting of 14 labels. These temporal relations are called "T-Links". These T-Links can occur not just between two events, but also between events and temporal expressions. TimeML defines these temporal expressions as follows.

> "A temporal expression in a text is a sequence of tokens (words, numbers and characters) that denote time, i.e. they express a point in time, or a duration or a frequency."

Thus, for example, the date "3rd January 2000", "sixty minutes", or "fortnightly" are all examples of temporal expressions. Non-grounded expressions such as "today", "last year", etc are also considered temporal expressions. The fourteen T-Links are listed and briefly explained below.

1. **Before**:
   Two entity instances are marked as `before` if entity A occurs strictly before entity B. Consider the following sentence. "Vaujas Saduguru gave the test a while back. He was told he passed today". The event "gave" and "told" have a `before` relation.

2. **After**:
   This is the inverse of the above relation, `before`. If an entity A occurs before B, then the entity B occurs after entity A.

3. **Includes**:
   Two entity instances are marked as `includes` if entity A includes the entity B. Consider the following sentence, "Vaujas arrived in Hyderabad last week". The temporal expression "last week" includes the event "arrived".

4. **Is Included**:
   This is the inverse of the above relation, `includes`. If an entity A includes B, then the entity B is included by entity A.

5. **Holds During**:
   Two entity instances are marked as `holds during` if entity A persists throughout the duration of entity B. Consider the following sentence, "Vaujas taught for three years". Here "taught" is an event that persists for "three years" and as the event occurs during the above temporal expression, thus they are tagged as `holds during`

6. **Held During**:
   This is the inverse of the above relation, `holds during`. If an entity A holds during B, then the entity B is held during entity A.

7. **Simultaneous**:
   Two entity instances are marked as `simultaneous` if they occur at the same time. The simultaneous relation can also be marked for two entities that occur close enough that further distinguishing the difference does not change the temporal understanding of the text. Consider the following sentence, "Vaujas was smiling and dancing". Here, the events "dancing" and "smiling" can be marked as simultaneous. Even though in realty, Vaujas might have danced before smiling or started smiling before dancing, as it makes no difference to the temporal understanding of the text, we mark the relation as simultaneous.

8. **Identity**:
   Two entity instances are marked as `identity` when both entity instances refer to the same entity. Consider the following sentence, "Vaujas drove to Hyderabad. During his drive, he ate a banana". Here, *drive* and *drove* both refer to the same event, and thus have the relation identity.

9. **Immediately before**:
   Two entity instances are marked as `immediately before` if event A occurs immediately before event B. Consider the following sentence, "Vaujas drank the spoilt milk and felt queasy." Here the action of drinking the spoilt milk immediately causes discomfort. Hence these two events are tagged as `immediately before`.

10. **Immediately after**:
    This is the inverse of the above relation, `immediately before`. If an entity A occurs immediately before entity B, then the entity B occurs immediately after entity A.

11. **Begins**:
    Two entity instances are marked as `begins` if entity A indicates the start or occurrence of entity B. Consider the following sentence. "Vaujas was studying in his room from 8 AM to 10 AM". Here, the event studying occurs at the beginning of the temporal expression, "8 AM". Thus these entities have the `begins` relation.

12. **Begun by**:
    This is the inverse of the above relation, `begins`. If an entity A begins with entity B, then the entity B is begun by entity A.

13. **Ends**:
    Two entity instances are marked as `ends` if entity A indicates the end or the termination of entity B. Consider the following sentence. "Vaujas was studying in his room from 8 AM to 10 AM". Here, the event studying stops at the end of the temporal expression, "10 AM". Thus these entities have the `ends` relation.

14. **Ended by**:
    This is the inverse of the above relation, `ends`. If an entity A ends with entity B, then the entity B is ended by entity A.

As there can be a large number of such T-Links, TimeBank annotators were asked to mark only the relations critical to understand the document and thus only annotate a subset of the total T-Links resulting in 6418 relations from 183 documents. This sparse annotation leads to many false negatives. The labelling schema also leads to confusion amongst the annotators. The distinction between labels such as `before` and `immediately before`; `begun by`, `during`, and `includes` etc are not well defined, and thus lead to poor inter-annotator agreement.

To solve the problem with sparse annotation, later works focused on densely annotating the events. One of the first works to do this type of dense annotation was Temporal Directed Acyclic Graphs (TDAG) [Bramsen et al. (2006)]. TDAG considers text as a linear ordering of temporal segments. A temporal segment here is defined to be a fragment of text that maintains temporal coherence. Thus, while within the temporal sequence, temporal ordering is retained but can vary between different temporal sequences. In other works such as in TempEval-1 [Verhagen et al. (2007)] and TempEval-2 [Verhagen et al. (2010)], only relations between event pairs in specific syntactic pairs were annotated. In Joint Event Timeline [Do et al. (2012)], they extend the annotations on the ACE 2005 corpus [Walker et al. (2006)]. Annotators were not required to mark the relations between all the event pairs in the given context but were instructed to mark as many as possible. TimeBank-Dense (TB-Dense) [Cassidy et al. (2014)] was one of the first datasets to mark all the relations between entities in the given context. As it is expensive to annotate event relations between all the event pairs, TB-Dense annotates densely for local graphs. I.e., TB-Dense considers a context window consisting of neighbouring sentences and annotates temporal relations densely within all the entity pairs present in this context window. TB-Dense considers the following entities:

1. Events
2. Temporal Expressions
3. Document Creation Time (DCT)

TB-Dense also chooses not to opt for the fine-grain relations present in TimeBank and opts for a relation schema consisting of six relations, namely:

1. Before
2. After
3. Includes
4. Is Included
5. Simultaneous
6. Vague

Relations like *before*, *immediately before* present in TimeBank are all represented as *before* in TB-Dense. Most of TB-Dense's relations are based on the relations found in TimeBank, with the only difference being the relation label vague. TB-Dense adopts the relation vague from TempEval [Verhagen et al. (2007)]. Since TB-Dense considers all event pairs, there is greater ambiguity between the correct relation pairs. TB-Dense allows annotators to utilize the relation vague to indicate that no particular relation could be established. The vague relation is also used in another interesting way. Annotators are asked to mark a relationship if they are over 80% (including 80%) sure that it is the right relation, else they mark the relation as vague. If conflicts with the annotation labels cannot be resolved by taking the simple majority, the annotations are discarded, and the relationship is marked as vague. Consider, for example, a document with three annotators. If a relationship has no majority agreement, the three annotators chose three different labels. Thus, the relation is vague, as there is no certainty between the annotators as indicated by the three annotators marking three different labels. While annotating, TB-Dense also runs transitivity inference checks to ensure that the marked relations are consistent and there are no contradictions. TB-Dense randomly samples 36 documents from the TimeBank corpus and annotates these documents to produce 12715 relations. As expected, the relation vague dominates the statistics, with 5910 out of these 12715 relations being vague. Out of these 12,715 relations, 6,088 relations are event-event relations.

Densely annotating events can turn out to be an expensive exercise. But choosing a small context window like adjacent sentences can reduce the precision of when the event occurs by as much as 58% [Reimers et al. (2016)]. EventTime [Reimers et al. (2016)] attempts to reduce the annotation expenses by automatically inferring event-event relations based purely on the temporal expressions associated with these events. For example, consider the following sentences, “He was sent into space on May 26,

1980. He returned to Earth on the 1st of June." EventTime here detects the temporal expression "May 26, 1980" and associates it with the event *sent*. Similarly, it associates the temporal expression "1st of June" with the event *returned*. Now it compares the two events times, and as June 1st occurs after May 26th the event *sent* is classified as occurring `before` the event *returned*. EventTime distinguishes between two types of events. Punctual events occur at a point in time. Thus there is only a single temporal value associated with this event. However, events that have a duration are characterized by a start time and an end time. These are termed as SingleDay events and Multi-Day events based on the duration of these events. EventTime opts to keep days as the lowest level of granularity as none of the documents contained any information on the hour, minute, or second of occurrence for the events. EventTime finds that Multi-Day events are not handled well by TB-Dense, and these events account for roughly 41% of all the events in the corpus. EventTime, with its method of considering temporal expression, expands the context window from adjacent sentences to the whole document. But, these relations are not incomplete as not all the event-event relations are associated with time expressions. Most of the temporal relations between events can be easily identified relative to each other,i.e. event E_1 occurs before event E_2. Still, since they are not grounded with time expressions, they cannot be automatically inferred by EventTime. EventTime annotates the same set of documents as TB-Dense (36 documents) and has a total of $12,715$ T-LINKS.

Manually annotating events at a discourse level is expensive and time-consuming. So TDDiscourse (TDD) [Naik et al. (2019)] utilizes existing annotation work and augments them for long-distance temporal relations. TDD starts with all the relations present in TB-Dense save for the vague relation. TDD thus has the following set of relation labels:

1. After
2. Before
3. Simultaneous
4. Includes
5. Is included

TDD, then, proceeds to augment TB-Dense by using a heuristic algorithm to automatically infer relations between event pairs based on the data from EventTime. Automatically inferring relation labels from the EventTime corpus is a non-trivial task as not all the assigned dates and intervals are exact. TDD handles this by splitting the EventTime annotations into three cases.

1. SingleDay-SingleDay relations (SD-SD)
2. SingleDay-MultiDay relations (SD-MD)
3. MultiDay-MultiDay events (MD-MD)

For SD-SD relations, there are four possible ways to express this, namely:

1. `DATE`
2. Before `DATE`
3. After `DATE`
4. Between two dates such as after `DATE1` and before `DATE2`

When inferring TDD chooses to maintain precision. Thus there is no vague label if a relation cannot be determined. Because of this the *vague* label is discarded. For SD-MD relations, the SD-SD inference is applied between the start and end dates of the MultiDay events to determine the label. Similarly, for MD-MD relations, the SD-SD inference is applied for their respective start and end dates which are then compared to generate the relation label. The dataset generated by automatically inferring these labels is termed as TDD-Auto. TDD augments TDD-Auto by randomly sampling and annotating unlabeled event pairs. This subset of manually sourced relations is termed as TDD-Man. TDD-Man, however, is still incomplete as not all the event pairs are labelled with the correct relation label. In many narratives, it is impossible to accurately determine the correct relation label between an event pair due to a lack of information. By dropping the *vague* label, annotators are forced to choose incorrectly since they are not allowed to leave an event pair unmarked. TDD, in total, has $43,000$ relations with the manually annotated TDD-Man comprising $6,000$ relations.

Serial Number	Example
1	Serbian police tried to eliminate the pro-independence Kosovo Liberation Army and (e_1:`restore`) order. At least 51 people were (e_2: `killed`) in clashes between the Serbian police and ethnic Albanians in the troubled region
2	Service industries (e_3: `showed`) solid job gains, as did manufacturers, two areas expected to be hardest (e_4:`hit`) when the effects of the Asian crisis hit the American economy.
3	We will act again if we have evidence he is (e_5: `rebuilding`) his weapons of mass destruction capabilities, senior officials say. In a bit of television diplomacy, Iraq's deputy foreign minister (e_6: `responded`) from Baghdad in less than one hour, saying that ...

Table 2.1: Examples that can be ambiguous when generating event relations in a single axis

Datasets like TB-Dense suffer from poor inter-annotator agreement. An ideal annotator can remain clear-headed and resolve ambiguities accurately. However, in reality, it is hard for annotators (even harder if said dataset is crowd-sourced) to remain clear-headed for hours and annotate all the documents accurately. Consider the examples shown in Table: 2.1 [1]. In example 1 from the table, the Serbian police attempted to restore order but clashed with the pro-independence faction. We can think of it as the attempt to restore order, i.e. the event e_1 occurred before the conflict where the people were killed,

[1]These examples are taken from the MATRES [Ning et al. (2018)] paper

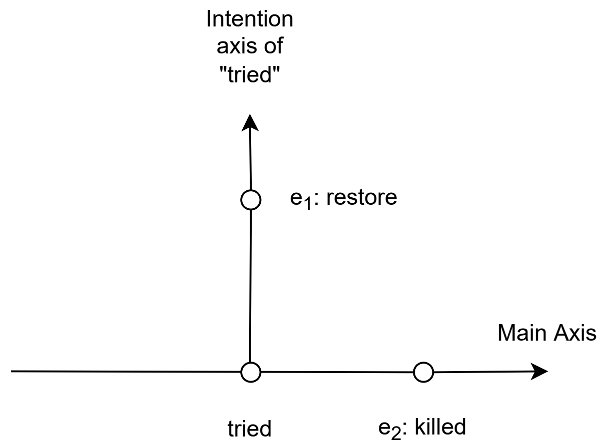


Figure 2.1: A multi axis view of Example 1 in Table 2.1

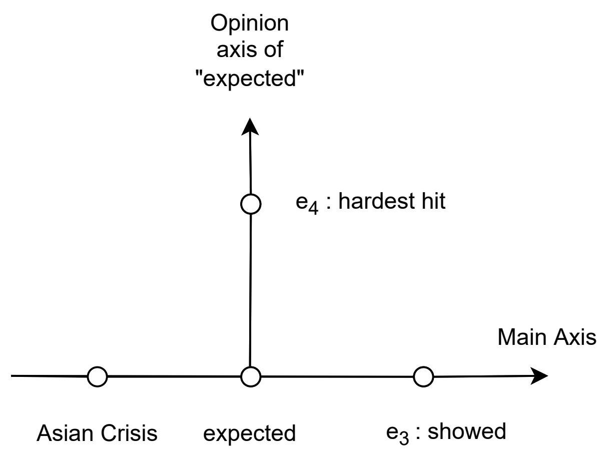


Figure 2.2: A multi axis view of Example 2 in Table 2.1

i.e. event e_2. However, we can also view the statement as the people were killed (e_2), but the order has not been restored (e_1) yet. In the former, e_1 occurs before e_2, whereas it occurs the other way around in the latter with e_2 occurring before e_2. In the second example, similarly, service industries and manufacturers were expected to be hit the hardest but actually ended up showing gains, so hit (e_3) occurs before showed (e_4). We can also view it as the two areas are showing gains because they are not yet hit, and thus e_4 occurs before e_3. In the last example, i.e. example 3, the event rebuilding (e_5) is a hypothetical event, and the reader is not sure if the event has occurred or not. Depending on the interpretation, namely "he is already building weapons, but we have no evidence" and "he will be building weapons in the future", one can assign different labels to the relation for e_5 and e_6. This ambiguity primarily arises due to the constraints of one axis. If we are allowed to annotate events on multiple axes, and label event relations only for the events within the same axes, we can greatly

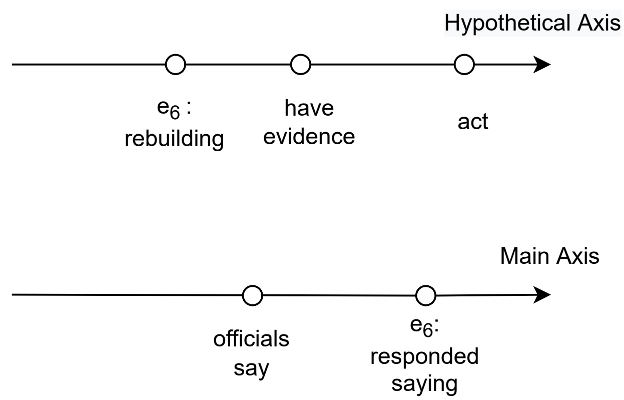


Figure 2.3: A multi axis view of Example 3 in Table 2.1

reduce the ambiguity and assign the correct relations. Figures 2.1, 2.2 and 2.3 show the multi-axis representation for the above sentences respectively.

MATRES [Ning et al. (2018)] is one of the first datasets to allow for such multiple axes. In addition to this concept of multiple axes, MATRES compares events based only on event start times to improve the inter-annotator agreement. They found that if endpoints are also considered, only one out of ten annotators passed the quality control check. But because of the lack of endpoints, information on when two events overlap is lost. MATRES annotates the same set of 36 documents as TB-Dense and annotates locally with a context window of adjacent sentences. As MATRES only considered start time, we have the following relation labels for MATRES:

1. Before
2. After
3. Equal
4. Vague

MATRES simplifies the annotation procedure compared to other datasets like TB-Dense. Let T_1 be the starting time of event E_1 and T_2 be the starting time of an event E_2. When annotating, annotators ask two questions:

- **Q1**: Is it possible that T_1 occurs before T_2?
- **Q2**: Is it possible that T_2 occurs before T_1?

Let A_1 and A_2 be the answers to question Q_1 and Q_2 respectively. Then based on the answers, we have the following four possibilities:

1. If A_1 and A_2 are both "yes", the relation is marked vague, as it shows a contradiction

2. If A_1 and A_2 are both "no", then the relation is marked as "equal", as both cannot occur before the other.

3. If A_1 is "yes" and A_2 is "no", then the relation between E_1 and E_2 is "before"

4. If A_1 is "no" and A_2 is "yes", then the relation between E_1 and E_2 is "after"

MATRES in total has around 1000 relations, out of which 800 relations belong to the main axis, with the rest belonging to the orthogonal axes.

2.2 Character-Character Relations

There is a rich plethora of literature analyzing character-character relations [Kim (2020)]. One of the major themes present when analyzing character-character relations is the use of sentiment and emotions. Elsner (2012a),Elsner (2015) present a kernel based on the frequency of occurrence and the sentiment of the characters to model the plot. The kernel can be used to generate a representation for the plot of different novels for higher-level processing.

Reagan et al. (2016) develop a character-arc-based analysis of popular works based on Kurt Vonnegut's thesis. Vonnegut claims that in a story, the protagonist or the main character has ups and downs. These ups and downs can be graphed to reveal the story's shape. A story can now be represented as a graph, and broadly Vonnegut breaks stories down into six major categories

1. **Rags to Riches**:

   In these stories, the character starts from nothing and achieves something extraordinary. At the beginning of the story, the character in poor circumstances, and as the story develops, they improve their circumstances and work their way to the top. A visualisation of this category explained with an example can be found in Fig: 2.4. Some examples of stories in this category are listed below:

   - Pride and Prejudice by Jane Austen
   - A Winter's Tale by William Shakespeare
   - Matilda by Roald Dahl

2. **Tragedy**:
   Also known as "Riches to Rags", In these stories, the character starts off in a good place but experiences a fall from grace. Oftentimes, their downfall is a result of their own hubris or fatal flaw. We can see a clear fall in the character's circumstances as the story progresses. These stories usually end in the death or destruction of the main character. A visualisation of this category explained with an example can be found in Fig: 2.5. Some examples of stories in this category are listed below:

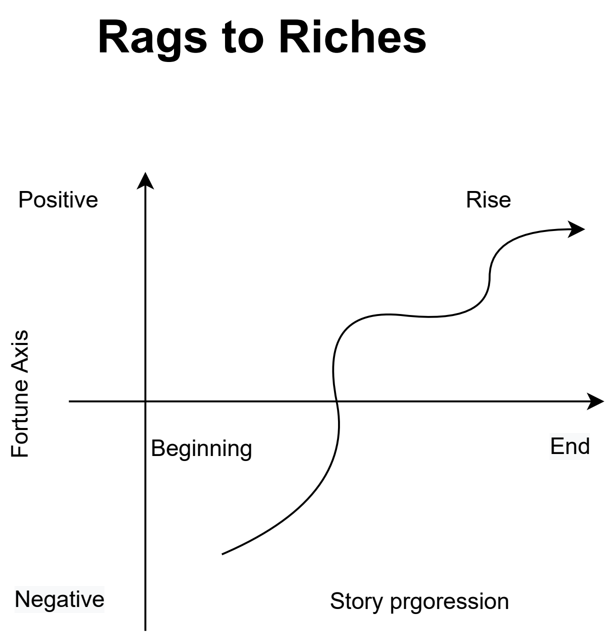


Figure 2.4: The shape of stories that are in the "Rags to Riches" category is visualized in the above graph. Consider the story of A Winter's Tale by William Shakespeare. Perdita is the daughter of King Leontes and his wife Hermoine. Leontes suspects his wife of infidelity and exiles Perdita. Shepherds then raise Perdita for sixteen years. Thus Perdita, at the beginning of the story, is in poor circumstances. As the story progresses, Perdita falls in love with Florizel, the son of one of Leontes' friends. Initially, Leontes thinks Perdita is not a suitable match for Florizel but upon discovering that Perdita is his long-lost daughter he becomes amenable to their betrothal. In the end, a statue of Hermoine that was made comes to life, and everyone lives happily ever after. To summarise, Perdita, who starts off in poor circumstances has experienced a continuous rise in her fortunes and ended up much better than where she started.

- Catch-22 by Joseph Heller
- Animal Farm by George Orwell
- Catcher in the Rye by J.D. Salinger

3. **Icarus**:

   These stories follow a character who has a moment of success or glory, but ultimately crashes back down to earth. The circumstance of the character first experiences a rise, and the story develops favourably to the main character. However, as the story develops, the protagonist experience challenges and difficulties, and their circumstances worsen, often leaving them worse than they started. A visualisation of this category explained with an example can be found in Fig: 2.6. Some examples of stories in this category are listed below:

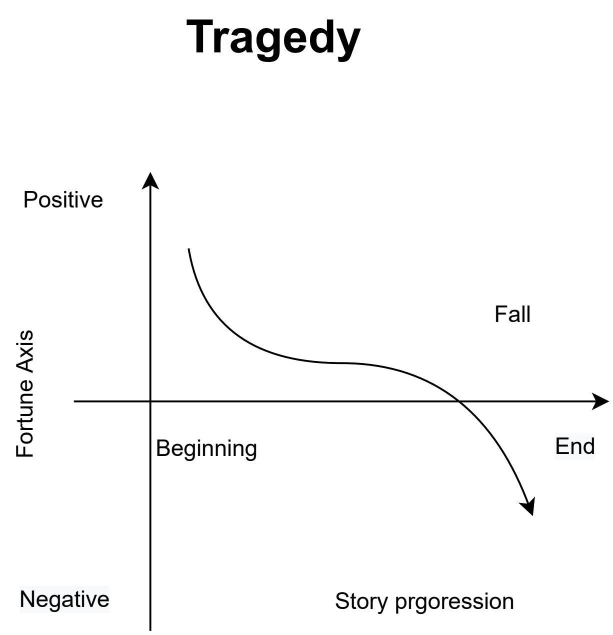


Figure 2.5: The shape of stories that are in the "Tragedy" category is visualized in the above graph. Animal Farm, a story by George Orwell, tells a story of farm animals. The farm animals rebel against their human masters and hope to create a society where animals can be free, happy and treated equally. The pigs, due to their intelligence, are put in charge of running the farm. Napoleon, one of the pigs, grows power hungry and chases or kills all animals that do not agree with him. By manipulating, the other animals, he becomes a dictator and starts violating the principles they stood for when the animals rebelled against the human masters. At the end of the story, the pigs act exactly as the humans did, and the situation for the other animals does not improve. One can even argue that it was worse, as many of the animals present at the beginning are no longer amongst the living. Here we see the farm animals continuously experience a decline in their fortunes, ending up much worse than where they started.

- Macbeth by William Shakespeare
- The Fault in Our Stars by John Green
- The Great Gatsby by F. Scott Fitzgerald

4. **Man in Hole**:

   In these stories, the character starts with favourable circumstances but very quickly experiences a fall. As the story progresses, the character works their way through adversities and continuously improve their circumstances. A visualisation of this category explained with an example can be found in Fig: 2.7. Some examples of stories in this category are listed below:

   - Alice in Wonderland by Lewis Carroll
   - The Hobbit by J.R.R. Tolkien

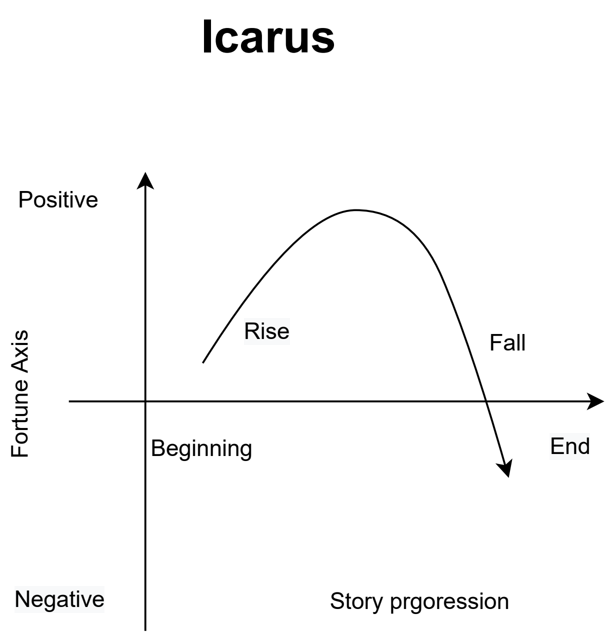


Figure 2.6: The shape of stories that are in the "Icarus" category is visualized in the above graph. The Great Gatsby, a novel by F. Scott Fitzgerald, is an example of this category. The story follows the rise and fall of Jay Gatsby. Gatsby is a wealthy man who throws lavish parties in an attempt to win over the love of his life, Daisy Buchanan. However, Daisy is already married to Tom Buchanan, and Gatsby's attempts to win her over are fruitless. As the story progresses, Jay and Daisy rediscover their love. However, Tom discovers the affair and confronts them about it. Eventually, Gatsby's business dealings and criminal past are revealed, and in the end, he is shot and killed by Daisy's husband.

- It is important to note that while stories are the most encountered narratives, narratives are not confined to stories. Donald Trump's 2016 presidential campaign, "Make America Great Again" is an example of this category.

5. **Cinderella**:
   Like in the story of Cinderella, these types of stories follow a character who experiences a rise, a fall, and then a rise again. The character starts from humble beginnings, experiences a period of hardship, and then ultimately has their situation improve. A visualisation of this category explained with an example can be found in Fig: 2.8. Some examples of stories in this category are listed below:

   - Jane Eyre by Emily Bronte
   - New Testament
   - A variety of popular Disney movies such as Frozen, Up

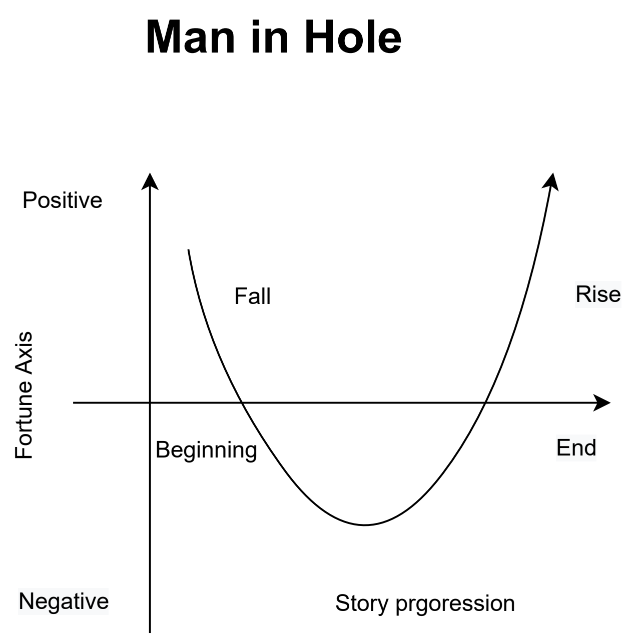


Figure 2.7: The shape of stories that are in the “Man in Hole” category is visualized in the above graph. Alice in Wonderland a novel by Lewis Carroll is an excellent example as the protagonist Alice literally falls in a hole at the beginning of the book. After falling into the hole, Alice discovers that she is in a strange world and nothing makes sense. She meets all manners of peculiar creatures as she faces challenges and overcomes obstacles.By the end of the story, Alice has learned valuable lessons and has grown as a person. As a result, her fortunes have changed for the better.

6. **Oedipus**:

   These stories follow a character who experiences a fall, a rise, and then another fall. The character starts off in a good place but either through their own bad choices or through encountering harsh difficulties, they suffer, and their situation starts to worsen. They may have a moment of redemption or glory and thus experience an improvement in their circumstances, but ultimately end up worse than they started. A visualisation of this category explained with an example can be found in Fig: 2.9. Some examples of stories in this category are listed below:

   - Moby Dick by Herman Melville
   - Gone with the Wind by Margaret Mitchell
   - The Godfather by Mario Puzo

They generate emotional arcs by analysing sentiment over the text using a sliding window of 10,000 words. The sentiment is analysed using a hedenometer based on the labMT dataset [Hammond et al.]. The stories are decomposed into orthogonal emotional arcs using Singular Value Decomposition. Ward’s

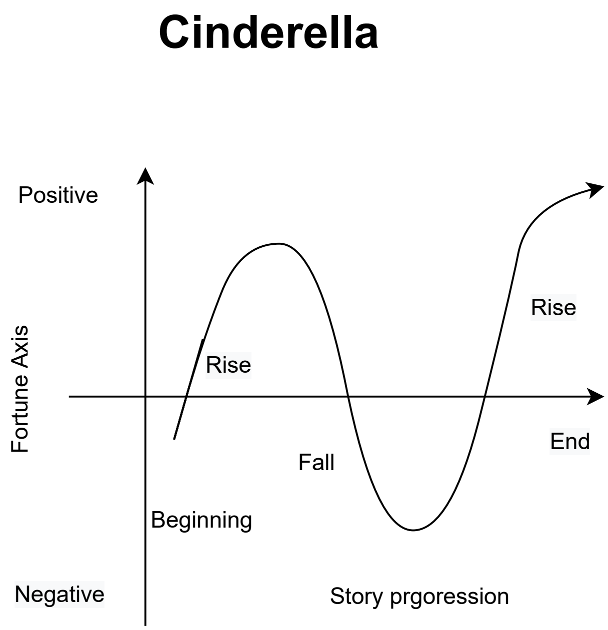


Figure 2.8: The shape of stories that are in the "Cinderella" category is visualized in the above graph. The novel Jane Eyre by Emily Bront tells the story of a young girl who is orphaned and sent to live with her aunt and cousins. She is treated poorly by her aunt and cousins and is made to do all the work around the house. When she is old enough, she goes to a boarding school where she is also treated poorly. However, she gets an education and eventually becomes a governess at Thornfield Hall. She falls in love with her employer, Mr Rochester, but discovers that he is already married to a woman who is locked in the attic. Mr Rochester's wife sets fire to the house and Mr Rochester is blinded. Jane leaves Thornfield and goes to live with some friends. At the end of the story, Mr Rochester comes to find her and they are married. They live happily ever after. Jane Eyre fits into the Cinderella category because she goes from being a poor, orphaned girl to being a governess at a wealthy estate. She falls in love with her employer, but it is not meant to be. However, in the end, she gets her happy ending when she marries Mr Rochester.

algorithm [Ward Jr (1963)] of hierarchical clustering is then used to minimize variance across different books. A Self Organizing Map [Kohonen (1990)] is then used on these clusters to find the core arcs. They experiment with 1,327 stories collected from Project Gutenberg and analyse the popularity of different novels based on their arcs.

Barth et al. [Barth et al. (2018)] develop a tool called rCat to analyze the character relations. rCat measures the closeness of two characters through a distance measure based on the number of tokens between the two character mentions in the text. rCat also allows to apply a filter to only consider words in the emotional domain to visualize the development of the plot.

Jhavar et al. [Jhavar and Mirza (2018)] develop a similar tool called EMOFIEL (EMotion mapping of FIctional rELationships). EMOFIEL can capture the emotion flow for a character pair, and organizes it on the story timeline. EMOFIEL also uses emotions and models emotions with two different models.

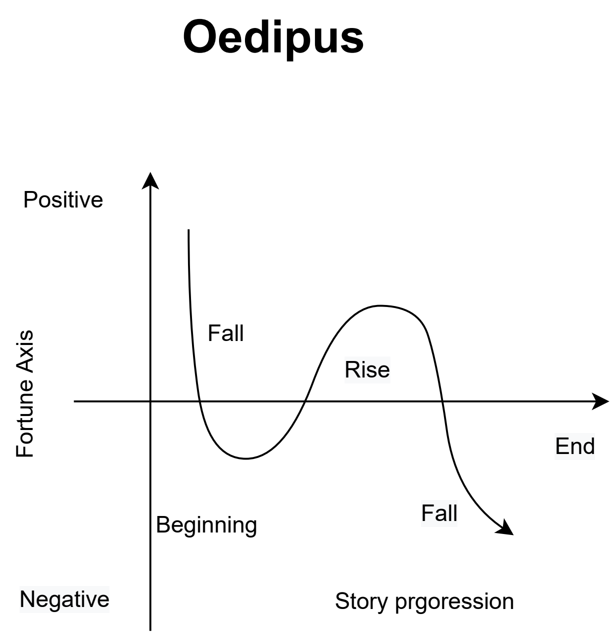


Figure 2.9: The shape of stories that are in the "Oedipus" category is visualized in the above graph. The Godfather by Mario Puzo tells the story of the Corleone family, a powerful crime family in New York City. The story follows the family's patriarch, Vito Corleone, as he ages and his son, Michael, takes over the family business. Michael quickly proves himself to be a ruthless leader, and the family prospers. However, as Michael expands the family's power, he attracts the attention of the government and the other crime families, which leads to a bloody war. In the end, the Corleones are victorious, but at a great cost. Michael is the last surviving member of his immediate family, and he is left with a broken heart and a dark soul.

The first model is a categorical model and is based on EmoLex [Mohammad and Turney (2010), Mohammad and Turney (2013)]. EmoLex is a list of words and their associations with the eight basic emotions[2]. The second model is a dimensional model and utilizes the Valence-Arousal-Dominance model (VAD). In the VAD model, emotional states are described relative to three fundamental emotional dimensions:

1. **Valence** which captures the degree of pleasure or displeasure of a particular emotion.
2. **Arousal** which captures the level of mental activity. Arousal can range from low engagement to high engagement or ecstasy.
3. **Dominance** which captures the extent of control faced in a given situation.

Character relations can also be modelled mathematically. Rinaldi et al. [Rinaldi et al. (2013)] develop a mathematical model to capture the relationship in the popular Disney movie, "Beauty and the Beast".

[2] The eight basic emotions are anger, fear, anticipation, trust, surprise, sadness, joy and disgust

They utilize ordinary differential equations to model the relations, and changes in the relationships are viewed as a disturbance in the equilibrium and are seen in the model due to the prescience of a saddle-node bifurcation. Jafari et al. [Jafari et al. (2016)] develop a dynamic model involving differential equations that describe the changes in behaviour between a couple. They utilize complex variables, utilizing the fact that complex variables have both phase and magnitude to capture to treat feelings as a two-dimensional vector rather than a scalar. This way, they are able to capture more complex emotions such as co-existing love and hate.

Chapter 3

Character-Character Relations

In this chapter, we explore character-character relations. We utilize the concept of character arcs to show the overall relationship of a character with all the other characters.

Characters in narratives attempt to influence their circumstances to resolve conflict, while circumstance itself shapes the characters with events that develop them [Weiland (2016)]. This character journey is integral to narratives [Vonnegut (1995)]; the transformative journeys of compelling characters drive good stories. This work addresses the challenge of quantifying these journeys using character arcs modelled around events and relations.

In computational literary studies, we seek to understand and represent narratives. Prior work has focused on plot units [Elsner (2012b); Lehnert (1981)], extracting social networks from narratives [Agarwal et al. (2013)], and character-centric approaches [Flekova and Gurevych (2015); Bamman et al. (2014)]. In this work, we focus on the character-centric approach. Motivated by research that examines narrative events [Sims et al. (2019)], we claim that character development is greatly influenced by the result of events involving said characters. Additionally, we leverage the recent developments in both emotion analysis [Demszky et al. (2020), Zad and Finlayson (2020)] and semantic role labelling (SRL) [Shi and Lin (2019)] to understand how events quantitatively affect both the agent and recipeient characters.

3.1 Key Concepts

In this section, we discuss key concepts that we use when generating a character arc.

3.1.1 Events

Similar to Sims et al. (2019) we focus solely on *events* with asserted *realis* (depicted as actually taking place, with specific participants at a specific time) instead of those with other epistemic modalities (hypotheticals, future events). These *events* are used as indicators of inter-character relationship states. For every event, we extract the latent predicate-argument structure to identify the participants of said

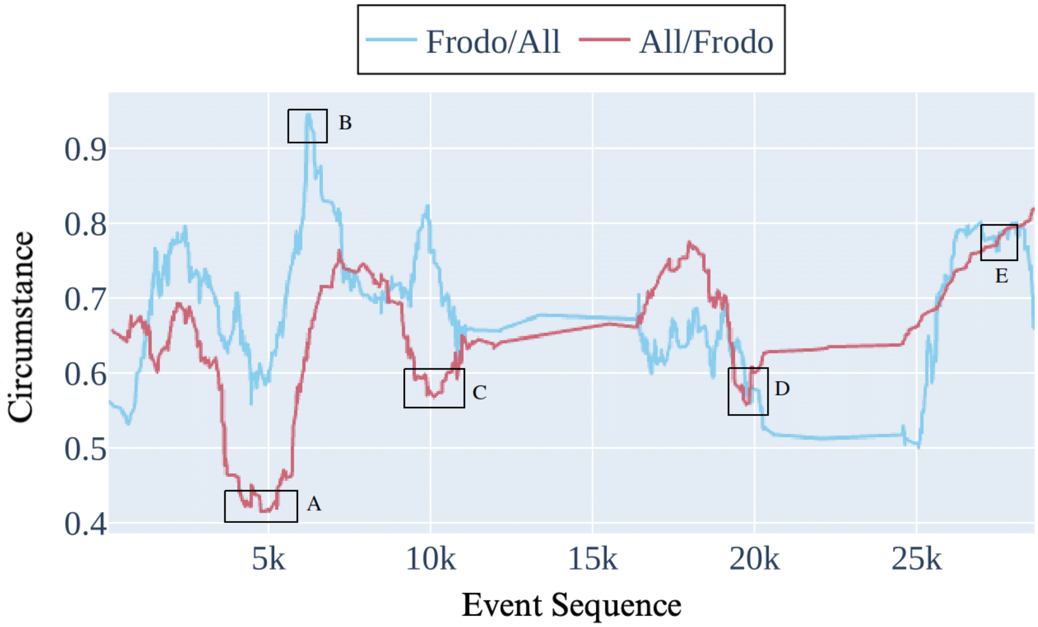


Figure 3.1: Relation-Based Character Arc for Frodo Baggins in The Lord of the Rings trilogy.Here the blue line represent the character arc for Frodo when Frodo is the *actor*. Similarly, the red line represents Frodo, when Frodo is the *experiencer*. **A**: Frodo is stabbed by the poisoned blade of the Nazgul at Weathertop (valley). **B**: Frodo reunites with loved ones at Rivendell (peak). **C**: Frodo is in grief after the wizard Gandalf falls to a Balrog (valley). **D**: Frodo is attacked by Shelob in her lair (valley). **E**: Frodo succeeds in his quest and returns home to The Shire (peak).

event. For example, consider the sentence “Sam carried Frodo”. Here “carried” is the *event* with “Sam” being the *actor* i.e. the person performing the event and “Frodo” the *experiencer* i.e. the person experiencing the event.

3.1.2 Circumstance

We take inspiration from Vonnegut (1995) and use implied sentiment and emotion to quantify *circumstance*, the state of fortune associated with a directed pair of characters participating in an *event*. Narratives and plots span a plethora of settings - each character ‘reacts to’ as well as ‘influences’ their unique *circumstance*; it would be beyond the scope of this paper to establish a universal scale. Instead, we focus on the relative *shift of circumstance* across the narrative.

3.1.3 Relation Arcs and Character Arcs

The relationship between two characters transforms as the plot develops. A *relation arc* is a plot that depicts how a specified pair of characters’ circumstances change during their involvement in a narrative’s events. While the *actor* of an event shapes the circumstances that have an impact on the *experiencer*, the latter's actions subsequently are captured in other relation arcs, allowing the effect of events to trickle through numerous connected arcs constituting the plot of a narrative. We posit that a character's journey at any point in the narrative can be represented by contextually assessing the amalgamated effect of their interactions with both themselves and other characters on their circumstance; a *character arc* is thus defined as a pair of quantitative aggregations of all the corresponding *relation arcs* of a character as both *actor* and *experiencer* respectively.

Fig 3.1 shows the character arc for the protagonist of the Lord of the Rings, Frodo Baggins, capturing the shift of circumstances he experiences as both an *actor* and *experiencer*. The events Frodo participates in change his circumstances - a drop to the valleys (minimas) represents a deterioration of circumstances, whereas a rise to the peaks (maximas) denotes an improvement. It is evident from **C** and the arc slice between **D** and **E** that actor and experiencer arcs of a character are not always aligned, allowing for the intuitive explanation of instances when characters oppose the circumstances they find themselves in, for better or for worse.

3.2 Data

We choose to focus on longer narratives. Narratives such as short stories typically have less than 500 events leading to a scarcity of data. Each character has only a fraction of these data points; thus, we fail to build meaningful character arcs.

To showcase our model's performance, we choose two of the most popular series praised for their rich character development and relations, namely the "Harry Potter" heptalogy by J.K Rowling [1] and the "Lord of the Rings" trilogy by J.R.R Tolkien [2].

We process all the books in each of the series and convert the novels into plain text. We further clean the text and removed all the irrelevant information such as page numbers, chapter titles, additional annotations, links etc. At the end of all the processing, we merge the different novels in the series and get a single document for each of the series that contains a clean version of the narrative. The details of each dataset are present in Table: 3.1

Table 3.1: Dataset Details

Data Source	Word Count	Event Count
Harry Potter	1,095,940	93,782
Lord of the Rings	478,329	28,670

[1] https://en.wikipedia.org/wiki/Harry_Potter

[2] https://en.wikipedia.org/wiki/The_Lord_of_the_Rings

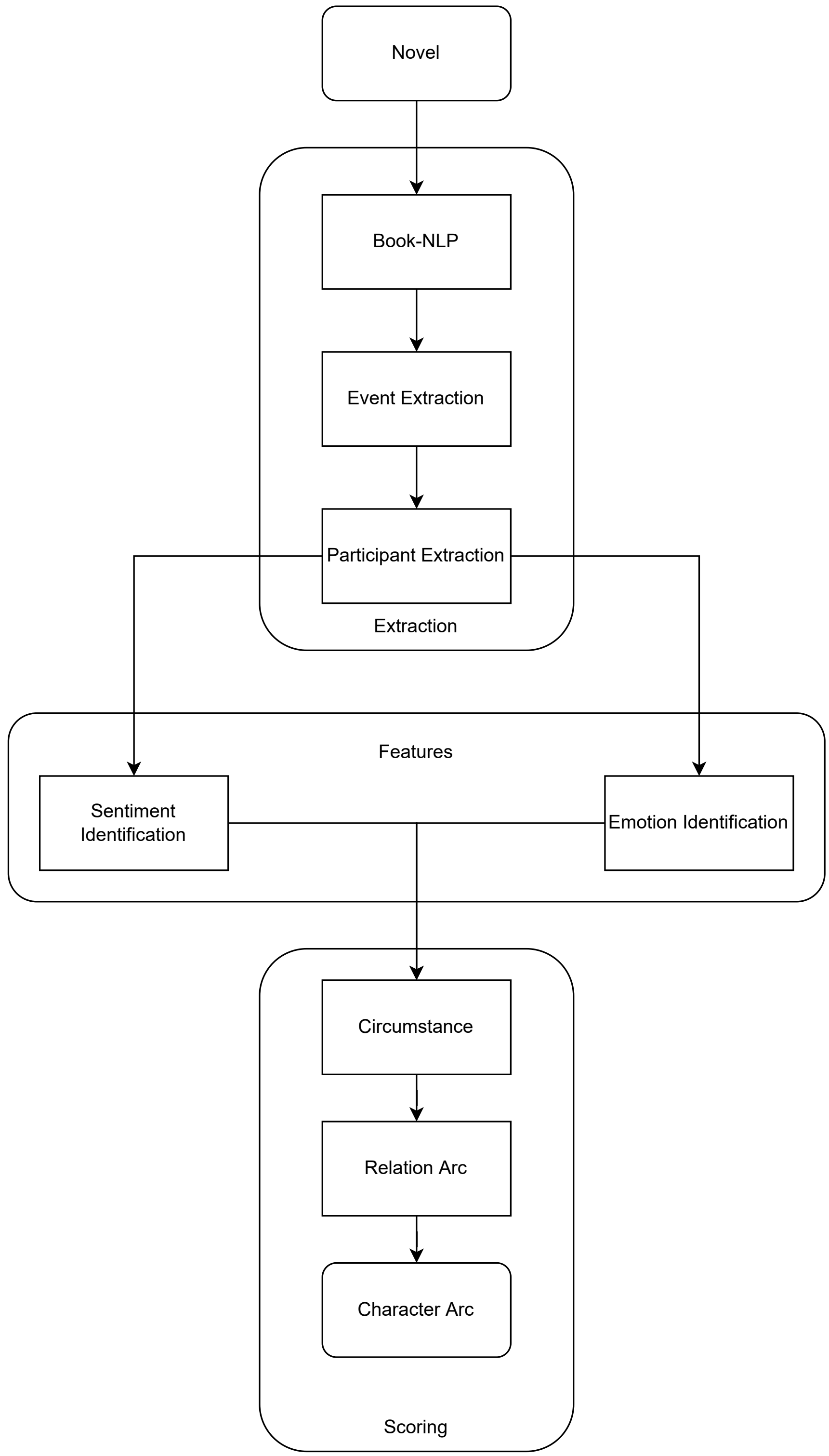


Figure 3.2: MARCUS (Modeling Arcs for Understanding Stories), an NLP pipeline that plots a character's arc as their quantitative interaction with circumstance as both actor and experiencer, represented by the proxy amalgamation of their event-centric relations across the narrative.

3.3 Components of MARCUS

MARCUS as a pipeline is made of multiple components as shown in Fig: 3.2. Broadly speaking MARCUS can be divided into three main components:

1. Extraction: This component deals with the information extraction from the text. This includes both event extraction as well as participant extraction.

2. Features: This component involves the various features such as sentiment and emotion identification

3. Scoring: This component involves combining the different features derived in the earlier section. This combination happens across features, events, and characters.

3.3.1 Extraction

3.3.1.1 BookNLP

Most of the information extraction components deal with relatively shorter text lengths. Since we are working with lengths of multiple novels, architectures like the Transformer cannot be directly used to the limitations of the context window. There are, however, some works that deal with information extraction at this scale, and the most prominent amongst them is BookNLP. BookNLP [Bamman et al. (2014)] is an NLP pipeline[3] that scales to books and long documents (in English). It performs most of the standard NLP tasks using existing tools at scale. For tasks such as part-of-speech tagging and named entity recognition, it utilizes StanfordCoreNLP, while for other tasks such as dependency parsing, it utilizes the Malt Parser. In addition, it handles essential tasks for literature analysis, such as character name clustering and co-reference resolution. Characters in a literary work can be referred to in a variety of ways. For example, consider "Harry James Potter" from the Potterverse. He is referred to as Harry Potter, Harry Potter, Mr. Potter, the boy who lived, undesirable number one, the chosen one etc. While BookNLP can handle trivial cases such as Harry Potter, Harry, Mr Potter, more oblique references are missed. MARCUS uses BookNLP to retrieve character occurrences and linguistic features needed for event and participant extraction.

3.3.1.2 Event Extraction

LitBank [Sims et al. (2019)] is an annotated dataset of a hundred works in English from Project Gutenberg. It contains annotations for entities, events, entity coreference, and quotation attribution from these works. The features extracted from the previous BookNLP step is fed into a BiLSTM model with BERT embeddings which is subsequently trained on the LitBank dataset to extract the events from the sentences in the text.

[3] `https://github.com/dbamman/book-nlp`

3.3.1.3 Participant Extraction

Semantic Role Labeling (SRL) recovers the latent predicate argument structure of a sentence, providing representations that answer basic questions about sentence meaning, including "who" did "what" to "whom," etc. We label the sentences associated to each event with an SRL model. The SRL model is a linear classifier on top of a BERT based model trained to extract participant characters [4]. If there are multiple events in the same sentence with the same actors and experiencers, we consider only the first event to avoid any sentence being processed more than once.

3.3.2 Circumstance

Every actor/experiencer pair corresponding to an event in the narrative is assigned a quantitative measure of circumstance. We argue that proxy indicators such as sentiment and implied emotion implicitly capture an event's circumstances and are specifically well suited to our focus on *shift of circumstance*. Absolute measures of circumstance can therefore be interpreted as a characteristic of the genre or trope of narrative, while their shifts are a characteristic of the character's journey. Since circumstances evolve over time (or event sequences), MARCUS considers the effect of previous circumstances between characters when defining *relation arcs*.

3.3.2.1 Sentiment Identification

We pose sentiment extraction as a regression task to capture the subtleties of relationships. We fine-tune a RoBERTa model on the Stanford Sentiment Treebank (SST) [Socher et al. (2013)] dataset to obtain a fine-grained sentiment score in the range of 0 to 1. The SST dataset provides $12k$ sentences and phrases with their associated sentiment scores lying between 0 to 26, which we normalize before training for ten epochs in a 60:20:20 split. The metrics for this model are reported in Table 3.2. MARCUS uses the model to assign sentiment scores to events extracted earlier in Section 3.3.1.2.

Metric	Score
Mean Squared Error	0.01620
Mean Squared Error	0.09693

Table 3.2: RoBERTa Sentiment Regression Model Metrics

3.3.2.2 Emotion Identification

Sentiment alone may not give us enough information about circumstance - we argue that in such cases, multi-faceted emotional states help capture shift in circumstance by leveraging the nuances of relationships. To identify emotions, we use a BERT model[5] trained on the GoEmotions [Demszky et al.

[4] https://docs.allennlp.org/models/main/models/structured_prediction/models/srl_bert/
[5] https://github.com/monologg/GoEmotions-pytorch

(2020)] dataset in a multi-label setting, as interactions can have more than one emotional undertone. The GoEmotions dataset consists of 58k Reddit comments manually annotated for 27 emotion categories: admiration, amusement, sorrow, fear, etc. MARCUS uses the confidence of the model's predicted labels as well as a manually assigned value for each label to contribute to the measure of circumstance. These labels, in the range of -2 to 2, are assigned based on intensity of emotion; higher intensity corresponds to higher absolute value.

3.3.3 Relation and Character Arcs

MARCUS generates relation arcs by plotting the measure of circumstance, t, for every event, e, belonging to an actor/experiencer pair of characters across the narrative.

$$t_{\mathrm{e}} = \alpha * s_{\mathrm{e}} + \sum_{i=1}^{L} \beta_i * c_{\mathrm{i_e}} \tag{3.1}$$

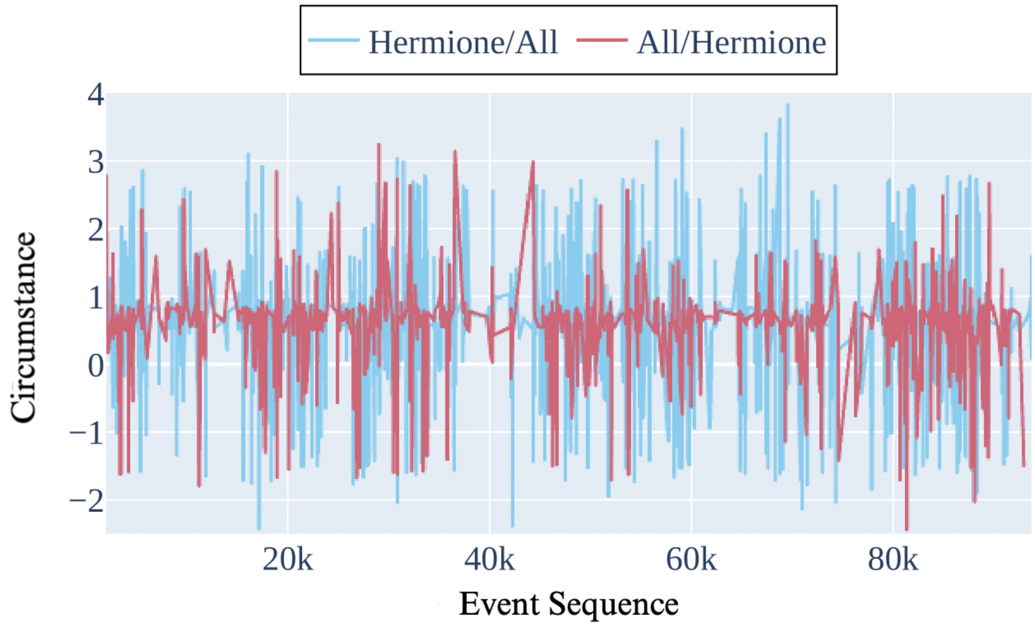
where t_{e} is the measure of circumstance for event e, $\alpha \in (0,1)$ is the sentiment co-efficient that controls how much influence the fine grained sentiment score should have over relation arcs, $s_{\mathrm{e}} \in (0,1)$ is the sentiment of that event, L is the total number of emotion labels for that event, $\beta_i \in [-2,2]$ is the fixed score for emotion label, and $c_{\mathrm{i_e}} \in (0,1)$ is the corresponding confidence score for each emotion label in the event. We run multiple experiments to choose optimal values of α and β: their final values are listed in the code.

We apply a window function $R(.)$ over $\mathbf{t_e}$, the set of all measures of circumstance corresponding to the event set **e**, to calculate the relation arc, **r**, given by

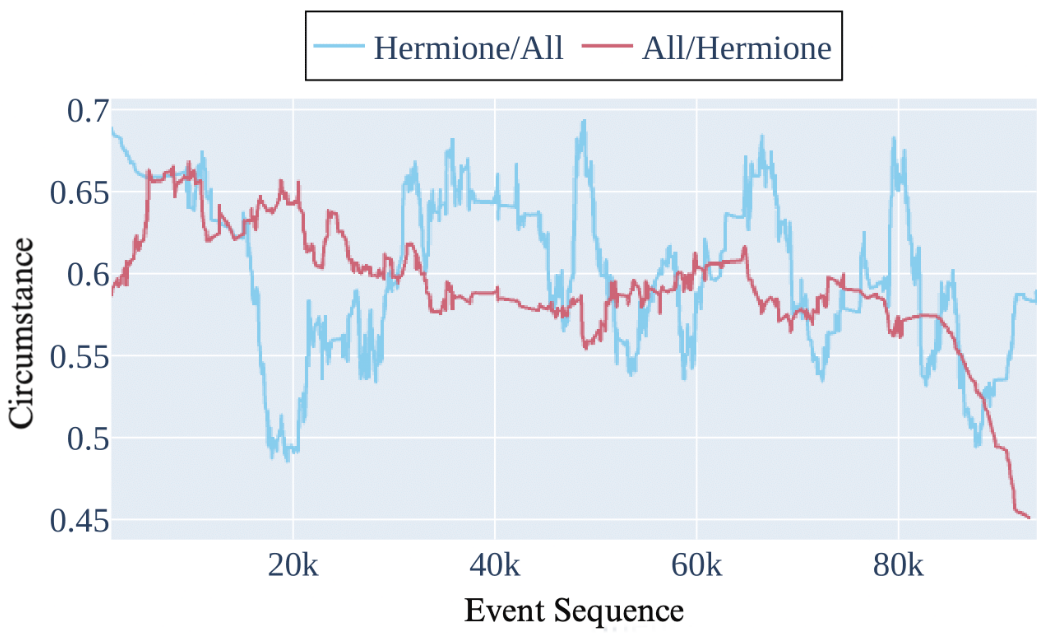
$$\mathbf{r} = R(\mathbf{t}_{\mathrm{e}}, n, p) \tag{3.2}$$

where $R(.)$ is the window function that helps smoothen the arcs while retaining previous state information, n is the window size, and p is an optional parameter for specifying order for polynomial fitting.

As shown in Fig 3.3a, the relation arcs are too noisy without smoothing or retention of previous circumstance information. We experiment with three standard window functions: Rolling Mean, Triangular Rolling Mean and Savitzky-Golay Filter [Savitzky and Golay (1964)]. We find that the Savitzky-Golay Filter represents the narrative most accurately as seen in Fig 3.3b, and use the same for all arcs represented in the paper. We generate the character arc, **c**, by adding up the corresponding relation arcs **r** of that character over all events **e** in the role of actor and experiencer respectively.



(a)



(b)

Figure 3.3: (a) Character Arc for Hermione, No Rolling Function Applied; (b) Character Arc for Hermione, with a savgol filter of window size 1/10th of her event sequence length, fitted with a third degree polynomial.

3.4 Evaluation

3.4.1 Survey

We ask a set of 16 human volunteers (avid fiction readers aged 20-31) to peruse both the Harry Potter and Lord of the Rings series, following which they evaluate our system by answering surveys on two tasks: **idchar**, where the volunteers are given a list of relation arcs and character pairs and asked to match the arcs to their corresponding pairs, and **idplot**, where the volunteers are given character arcs, pertinent plot events and asked to identify the points in the arc that they think represent the corresponding events.

For both these tasks mentioned above, we calculate accuracy. Task **idchar** achieved an accuracy of 71.5% and task **idplot** had an accuracy of 72.9%. We also evaluate with the Fleiss Kappa metric where category 1 indicates a complete match, and 0 indicates otherwise. For both the tasks, we have a score of 0.675 and 0.528 indicating strong and moderate inter-annotator agreement, respectively. Thus, most of the volunteers consistently identified both the character pairs and the relevant points in the graph given the plot sequence.

3.4.2 Gold Labels

We have a volunteer extremely familiar with the story annotate the first 300 events of Lord of the Rings trilogy involving Frodo Baggins as a participant character. The annotator marks each event with three labels denoting a *positive*, *neutral* or *negative* shift in circumstance. Tables 3.3 and 3.4 illustrate the positive and negative shifts as tagged by our system and the corresponding gold labels for the same events provided by our annotator. Our system tends to assign positive labels to neutral events and has higher accuracy for negative shifts of circumstance.

Table 3.3: Positive Shifts

Gold Label	Percentage
Positive	0.36
Neutral	0.30
Negative	0.33

Table 3.4: Negative Shifts

Gold Label	Percentage
Positive	0.12
Neutral	0.15
Negative	0.73

3.5 Challenges and Future Work

We identify five notable challenges in our approach that can be addressed in future work. Firstly, since MARCUS is a sequential pipeline, it is challenging to determine the effect of errors cascading through the system quantitatively. Secondly, our rolling window makes the arcs dependent on the availability of data; event paucity in short stories or characters with low interactions hinders accurate arc generation. Thirdly, we observe in our arcs that our fine-grained events do not represent an abstract view of the discourse - a more contextual representation of events is needed. Fourth, in our understanding of a character's circumstance, localized interactions with other event-specific characters heavily influence shifts; we need an effective means of capturing the latent relative importance of character-specific interactions. And lastly, our approach does not aim to handle non-linear narratives where events are not sequentially presented.

3.6 Applications

Providing tangibility to the theoretical concept of character arcs, MARCUS can be employed in a variety of applications. Character arcs can be used in a more nuanced approach for detecting similarity between narratives by focusing on character journeys, leading to a possible improvement in book recommendations and movie recommendations based on stories and scripts. Character arcs can also help with digital enrichment in e-readers, adding to the rich metadata provided by devices like Kindle. Character arcs can also function as guidance for natural language generation tasks in the field of fiction. And lastly, they can help narrative studies by identifying character tropes and for identification of personality traits.

3.7 Conclusion

We propose MARCUS (Modeling Arcs for Understanding Stories), an NLP pipeline that addresses the novel task of generating character arcs from narratives. We explain key concepts like *events* and *circumstance* and delve into the details of our event-centric approach which leverages proxy markers like sentiment and emotion. We then evaluate our pipeline, discuss challenges, elucidate future work and outline potential applications.

Chapter 4

Event-Event Relations

4.1 Introduction

Most of the available datasets for event relations extraction and timeline generation focus on short-distance temporal relations between events. These event-event relations are either intra-sentential i.e. identifying event relations for events occurring in the same sentence or between events occurring in adjacent sentences. This greatly limits the applications of such systems as the majority of the temporal event relations are between events that occur in sentences spread across the discourse i.e. long-distance event temporal relations. In this paper, we focus on the task of identifying temporal relations between events at a discourse level beyond adjacent sentences.

There have been numerous efforts on the task of identifying temporal event relations. However, the existing annotation schemas are ill-suited to extending these relations at a discourse level. We define a timeline to be a temporal ordering of events relative to each other. Creating a document-level timeline is an inherently challenging task due to the extreme long-distance event relations. An event in the beginning of the document could very well occur simultaneously with an event at the end of a document. Another key challenge is that not all events are anchorable to the same timeline as there can exist multiple branching timelines (with branched timelines capable of branching further).

To address this, we introduce a new annotation schema for annotating event-event temporal relations. Similar to the different axes present in MATRES [Ning et al. (2018)], our schema places emphasis on different timelines. Broadly we define a "real timeline " which includes all the events that have actually happened, and other hypothetical timelines of events which includes events whose occurrence is not certain. These events can include events present in the discourse due to subjunctive mood such "imagine", events occurring in reported speech, events in the future tense etc.

In order to facilitate a fair comparison, we annotate the same selection of documents as existing literature such as TDD [Naik et al. (2019)] and MATRES [Ning et al. (2018)]. These documents are a set of English news articles. We annotate temporal relations for all the event pairs in the documents, and create a dataset with nearly 45,000 temporal relations.

Because we are considering context beyond adjacent sentences, annotating event timelines with existing annotation tools is cumbersome, and often leads to difficulties in keeping track of document context across long distances. Thus, in order to make annotation efforts efficient we also developed a new annotation tool to enable annotators to mark and visualize event timelines. We open-source both the dataset the annotation tool. We build a baseline based on RoBERTa context encoder based classifier and achieve an F1 score of 0.492

To summarize, our major contributions are as follows:

- We extend the task of identifying temporal relations between event pairs to the discourse level and formulate the task as an event timeline generation
- We introduce a new annotation schema for event timeline generation. Our schema differs from existing schemas as we branch event timelines, paying special attention to the real timeline which consists of events that have actually occurred
- We build and release a novel annotation tool which allows annotators to easily mark long distance event-event relations

4.2 Relations

We have a total of six relations namely:

1. Before
2. Hypothetical Event Timeline (HET)
3. During
4. Simultaneous
5. Indeterminate
6. Vague

Out of these six, the first three relations namely `before`, `during` and `HET` are directed relations. The last three relations namely `simultaneous`, `indeterminate` and `vague` are undirected relations.

We define each of the relations and illustrate these relations with examples.

4.2.1 Before

An event pair has the temporal relation `before`, if event A occurs strictly before event B.

Consider the following example, ”Manchester United fans were celebrating with fireworks after their team won the Premier League this season”. Here the event *won* occurs before the event *celebrated*. Thus the relation between *won* and *celebrate* will be marked with the temporal relation `before`. Most cause-effect scenarios bear the `before` temporal relation.

4.2.2 Simultaneous

We opt for the same definition of Simultaneous relations as that of the TimeBank [Pustejovsky et al. (2003)]. That is, an event has the temporal relation `simultaneous` if event A happens at the same time as event B. The events are also considered simultaneous if they are indistinguishable in context i.e. occur close enough to the same time that further distinguishing their times makes no difference to the temporal interpretation of the text.

Consider the example, "Jack ate a burger and drank a glass of coke". Here the events, *ate* and *drank* are marked as simultaneous. It could be the case that Jack first drank the coke, and then ate his burger or vice-versa but as it makes no difference to the structure of the timeline we mark these events as `simultaneous`.

4.2.3 During

An event pair has the temporal relation *During* when an event A occurs completely within another event B.

Consider the following example, "I flew to Norway and ate a burger on the plane". The event *ate* occurs during another event *flew*. Thus the event *ate* has a during relation between event *flew*

4.2.4 Indeterminate

An event pair has the temporal relation *Indeterminate* when both the events belong to the same timeline, but there is not enough information given in the document to ascertain the sequence of events.

Consider the following example, "I ate cake for Harry's birthday last week. I also sold my bike after I crashed it last week." Both the events *ate* and *sold* happened last week, but since the ordering of events between *ate* and *sold* is not known, the temporal relation for this event pair will be marked as *indeterminate*.

4.2.5 HET

Events in a document can be expressed with varying degrees of certainty. Thus, when constructing an event timeline it becomes important to consider the modality of an event [Mitamura et al. (2015)]. Broadly speaking, an event can be considered as either a "real" event i.e an that event that has actually occurred or a hypothetical event i.e an event with an uncertain status of occurrence [1]. This task

[1] There have been other terminologies as well that describes the same underlying idea such as "factive" or "non active" [Feldman et al. (1986)] or "actual" and "potential". [Ekdahl and Grimes (1964).]

of determining the certainty of occurrence is referred to as event factuality prediction. [Saurí and Pustejovsky (2009); Lee et al. (2015)]

We define the “real timeline” to be the timeline which consists of only real events, and conversely the hypothetical timeline is a timeline which only consists of “hypothetical” events. As there is only one real world and all the “real” events have actually occurred there exists only one “real” timeline and thus all of these events will be anchored to one timeline. There can be multiple hypothetical timelines as they do not have the same constraints.

These hypothetical events are indicated with irrealis moods [ELLIOTT (2000)]. The irrealis modal suffix indicates that the activity expressed by the verb is unreal.

Some of the most common cases of these irrealis moods that we observe are, “subjunctive moods” and “dubitative moods”. Events with subjunctive moods are used to express desire or imagination. Common subjunctive verbs are: dreamed, imagined, want etc. Events with dubitative moods are used to express if a particular statement is uncertain. For example, ”We believe Rooney scored the winner” or ”As far as I know, Rooney scored the winner”. Another common type of hypothetical events that we see are verbs in future tense. The last major instance of hypothetical events that we observe arises from reported speech.

These irrealis verbs are termed as anchors as they anchor the hypothetical timelines to the real timelines. We call the relation between the anchors and the closest verb (in terms of lexical distance from the anchoring event) in the hypothetical timeline as the HET relation.

Consider the following example. “Jack said, “Mary kicked the ball and the goalkeeper saved it””. Here there are three events, namely *said*,*kicked* and *saved*. The event *said* is an anchor as it is indicating reported speech and thus creating a new hypothetical timeline. Since, between the two hypothetical events *kicked* and *saved*, *kicked* is the event lexically closest to the anchor, the relation between *saved* and *kicked* is marked as HET.

4.2.6 Vague

We use the relation vague to indicate that the events are in different timelines. Since the events are in two different timelines, they cannot be directly compared and thus the event relation between such events is marked as vague. It is important to note however that while vague implies that the events are in different timelines, the converse is not true. As shown earlier, an event pair E_1 and E_2 in different timelines can also have the relation HET if E_1 is an irrealis verb, and E_2 is the lexically closest hypothetical event to E_1.

Consider the same example as before, “Jack said, “Mary kicked the ball and the goalkeeper saved it””. Here the relation between *saved* and *said* will be marked as vague since *saved* occurs in.a hypothetical timeline, and as it is not the lexically closest event to the irrealis verb *said* which occurs in the real timeline the relation between *saved* and *saved* will be marked as vague.

4.3 Annotation Schema

Similar to TDD and MATRES, we annotate TB-Dense data consisting 36 English news articles. Also similar to MATRES we restrict ourselves to verbal events only and do not annotate relations between nominal events. Nominal events often need to be grounded with temporal expressions for annotators to ascertain their location in the timeline with high confidence. Since these temporal expressions are usually unavailable, and timeline generation captures the relative ordering between events and not their absolute position with reference to a global time, we choose to ignore these types of events.

Another interesting set of events is the "not events", i.e events that are explicitly stated to not have occurred. The non-occurrence of an event can also cause another event or vice versa. For example: "John *failed* because he did not *submit* his assignment." Note that, the event *submit* never happens but is the cause for failure. Due to this, for the annotation of temporal relations, we do not distinguish between this case and the regular case of events that have actually occurred. We treat these events as real events i.e. as if they had occurred, and mark the temporal relations accordingly. The way we handle such "not events" is similar to [Chambers et al. (2014)], however we differ from them in the way we handle hypothetical and conditional events. We treat these hypothetical and conditional events differently as there is a degree of uncertainty associated with these events, but in the case of "not events" there is certainty that the event has not occurred, and thus we annotate the "not events" on the real timeline.

Similar to TimeBank-Dense, annotators were asked to mark a relation only if they were very sure of the relation. In cases where there is confusion between multiple relation pairs, we ask the annotators to mark that relation as `indeterminate`. Inter annotator disagreements are also resolved by marking the disagreed relation as `indeterminate`.

Inference based Dense Annotation

We choose a dense annotation schema i.e for every event pair in the document, there must be a temporal relation. As we operate in a document-level context, it becomes challenging to ensure that every relationship has been marked and that no relationship is missed.

While this schema ensures completeness, it increases the effort required to annotate the documents. In order to ease the load on the annotators, we automatically infer new relations from existing relations based on a set of simple logical rules. These inferred relations are then added to the graph. Some of the rules are listed below.

1. Transitivity: If there are three events, E_1, E_2, E_3 and the relations E_1 `before` E_2, and E_2 `before` E_3 exist, then the relation between E_1 and E_3 can automatically be inferred as `before`.

2. Temporal Equivalence: If there exists two events:E_1 , E_2, and there exists a relation E_1 is `simultaneous` with E_2 , then E_1 and E_2 will share the same relation with all other events. That is, if there exists another event E_3, such that the relation between E_1 and E_3 is `r`, then the relation between E_2 and E_3 is also `r`.

3. Timeline Parallelism: If there exists two different timelines, T_1 and T_2, then the relations between all events on T_1 and T_2 will be `vague` except for the event that splits the hypothetical event timeline from the real event timeline which will be the relation `HET`.

We showcase our annotation schema on a simple example. Consider the following paragraph. John has *loved* apples since a young age. He can distinguish between nice and bad apples just by looking at them. The other thing he is passionate about is football. He has *followed* Liverpool for a long time now. When the game was being broadcast last night, the CNN reporter *said*, "It looks unlikely that the Liverpool defence will have any answers to Manchester United's formidable attack". At this, John *laughed* and *exclaimed*, "This reporter *likes* being wrong".

The above paragraph's timeline has been depicted in Fig.4.1

There are two narratives occurring here in this paragraph. One dealing with food, the other dealing with football. These are depicted as different branches of the main timeline. As there is no concrete temporal expression to ground the two events of the timeline i.e *loved* and *followed*, it is not possible to assign an exact relation between them. However, since both of them have actually occurred they must occur on the same timeline and are thus given the relation `indeterminate`. But if you consider the next event *said*, since we know *said* occurred today, and the former two events occurred in the distant past we can mark the relation as `before`.

As shown above, with a document-level context, there are often multiple narratives occurring simultaneously. These narratives even if they are not causally related can often be temporally related due to the presence of temporal expressions which let you know when these events occurred. Thus we introduce the concept of multiple timelines, where annotators can mark events as they occur with multiple timelines occurring simultaneously. Then at the end of the document, they can review the different timelines, and add the relations to connect across timelines be it an additional relation such as `before, during, simultaneous, HET` or classifying the relation between the two timelines as `indeterminate or vague`.

Annotators also find it difficult to mark temporal relationships for events that happen in the future or for events that occur in reported speech. We resolve this with the creation of hypothetical timelines. Hypothetical timelines also ensure separation between events that may not have occurred and events that have occurred, with the real timeline only having events that are known to actually have occurred.

4.4 Annotation Tool

We found annotating timelines with the existing annotation tools such as Prodigy[2] or GraphAnno[3] involved frequent context switches, and this made annotating any text longer than a paragraph extremely cumbersome. The timeline data represented as a triple (event, event and label) is also hard to comprehend

[1] In the figure we show a minimal timeline only, as showing all the relations will cause the figure to become cluttered and hard to read.

[2] `https://prodi.gy`

[3] `https://github.com/LBierkandt/graph-anno`

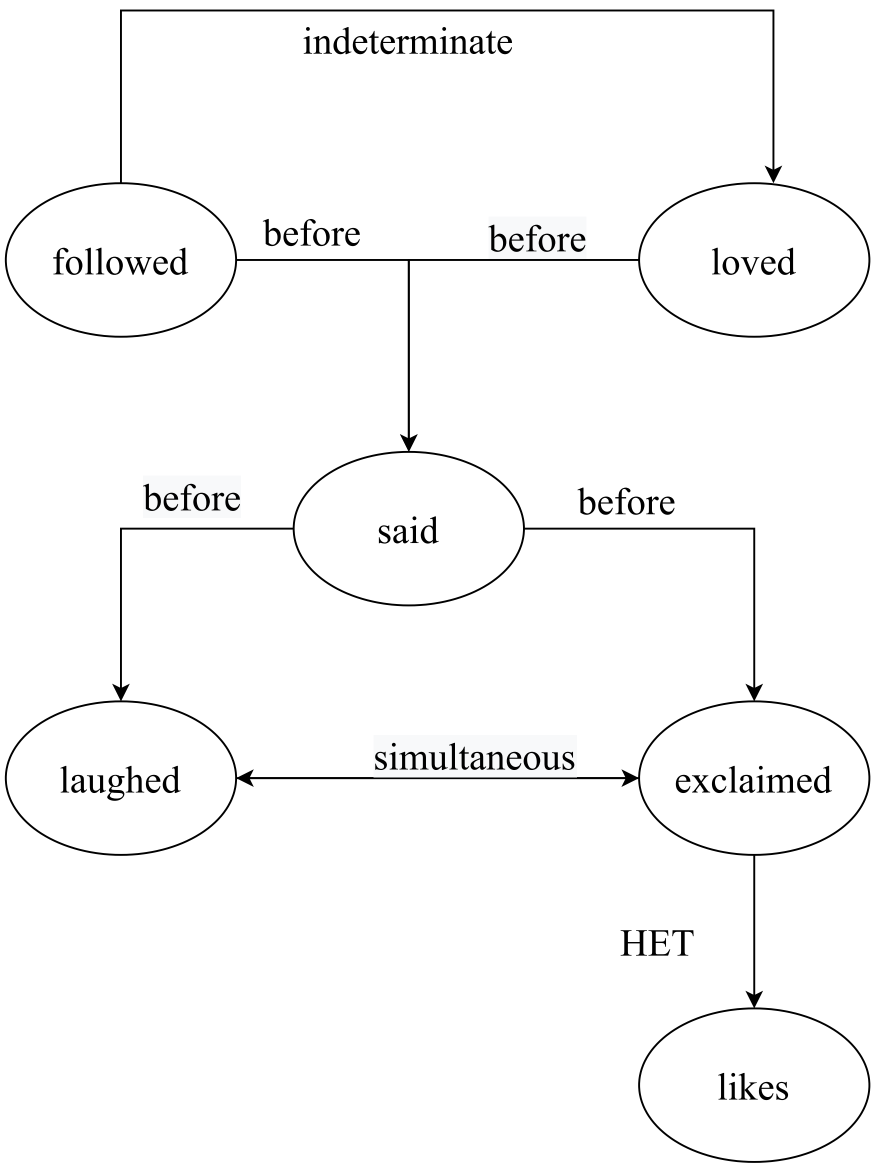


Figure 4.1: An example timeline

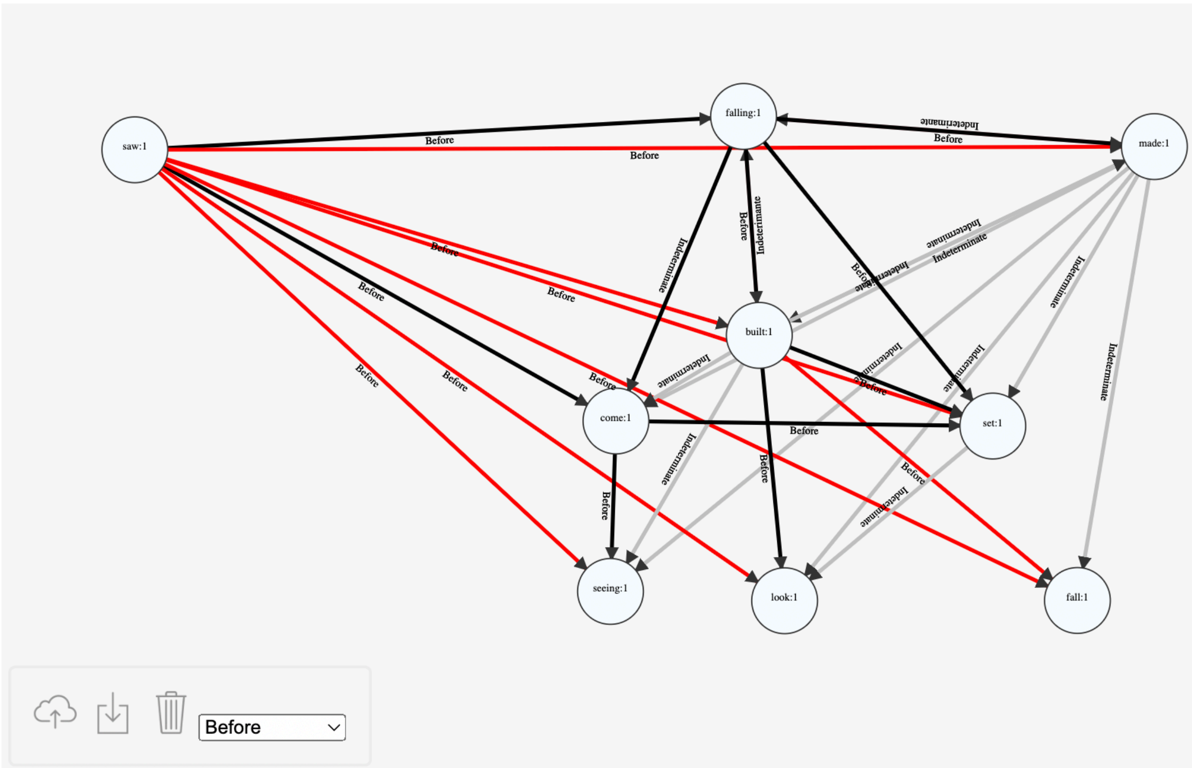


Figure 4.2: A screenshot of the annotation tool with a sample document

without visualising their respective relations as a graph. Thus, we developed an annotation tool to help both ease the annotation efforts and once annotated, aid in visualizing and thus understanding the various timelines. A screengrab of the tool is shown in Fig: 4.2

With the tool, the annotation is made as simple as drawing an edge between two event nodes. We choose to only visualize events from the document so to allow for easy annotation of cross-document relations. Annotators can also choose to infer new relations from the existing ones, and based on the type of relations as well as the rule used for inference, the edges are coloured differently, allowing annotators to identify what relation was inferred and verify if the inferred relation was indeed accurate. In cases where inferring is ambiguous and might lead to two relations, annotators were prompted to input the right relation. For example, consider the following case shown in Fig 4.3. Here event E_1 occurs before both events E_2 and E_3. But the relationship between E_2 and E_3 is ambiguous as it can potentially have the relation `before` or it can have the relation `indeterminate`.

4.5 Dataset Statistics

We annotate TimeBank-Dense, a collection of 36 news documents in English. The number of relations in each document vary from as low as 32 relations to as high as $\sim$ 17600 relations, with the average number of event-event temporal relations per file being $1,200$. We use existing TimeBank

Dataset	# of relations	Labels
TDD-Man	6150	a, b, s, i, ii
TDD-Auto	38302	a, b, s, i, ii
TB-Dense	6088	a, b, s, i, ii, v
MATRES	1800	e,a,b,v
DELTA (ours)	45,271	b,h,d,s,i,v

Table 4.1: Number of relations across datasets.
a: after, b: before, s: simultaneous, i: includes, ii: is included, e: equal, v: vague

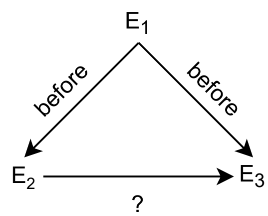


Figure 4.3: Ambiguous relations

annotations to identify the events in the corpus and remove all nominal events. A class-wise distribution of the relations can be found in Table 4.2 . Our annotations lead to a $\sim 150X$ gain over MATRES in terms of number of relations, and we are able to capture more long-distance relations. Table 4.3 shows a distribution of event relations based on sentence distance. Inferring new relations based on existing ones also proved to be extremely helpful to the annotators with the number of relations increasing by nearly 185 times. Due to their very nature, the relations `indeterminate` and `vague` don't add as much value to the timeline and are orders of magnitude more frequent than the other four relations, thus we group these relations together and call this group *frequent relations* leaving the other four relations namely `before`, `during`, `simultaneous` and `HET` as *infrequent relations*. The automated relation inference based on the manually annotated increases the total number of relations by ∼185 times. While most of the inferred relations belong to the *frequent relations* set, the number of *infrequent relations* also rises substantially with inference, leading to $\sim 2.5X$ gain in the number of relations. We report a Kappa score of **0.714** showing substantial agreement between the annotators.

Label	# of Relations
BEFORE	2038
HET	202
DURING	67
SIMULTANEOUS	807
INDETERMINATE	17095
VAGUE	25061

Table 4.2: Class wise distribution of event relations in DELTA.

Dataset	¡5	¡10	¡15
MATRES	1.00	0	0
TDD-Man	0.40	0.40	0.19
TDD-Auto	0.50	0.32	0.17
DELTA	0.58	0.34	0.08

Table 4.3: Distribution of sentence wise distance between events for all the event-event relations across various datasets.

4.6 Baseline Model

We adapt the RoBERTa [Liu et al. (2019)] based context encoder by Zhao et al. (2020) and expand context spans to allow global relation prediction.

Let a document $doc = [s_1, s_2 \dots s_{|doc|}]$, be represented as a sequence of $|doc|$ sentences. Let $p_i = (e_{i1}, e_{i2})$ represent the i^{th} event pair. Let P_doc be the list of all event pairs in doc. The task is to predict temporal relation, $y_{(i1,i2)} \in \{$before,after,simultaneous,hypothetical, indeterminate$\}$, $\forall(e_{i1}, e_{i2}) \in P_doc$. Let e_{i1} and e_{i2} belong to sentence s_{ia} and s_{ib} respectively. Without loss of generality, let $ia \leq ib$.

Further, let nbr_l and nbr_r be the number of sentences before s_{ia} and after s_{ib} (respectively) used as additional context for generating local context aware features for predicting $y_{(i1,i2)}$. Since [Zhao et al. (2020)] only present results on the MATRES dataset, where events can only occur in the same or adjacent sentences, nbr_l and nbr_r are both set to 0. However, in our case, we increase the window size to pass as much context as possible to the LM. Let seq_i be the concatenation of consecutive sentences used for generating features for $y_{(i1,i2)}$. Accordingly, $seq_i = [s_{ia-nbr_l} \dots s_{ib+nbr_r}]$, and when both event mentions are in the same sentence, then $seq_i = [s_{ia}] = [s_{ib}]$. A language model, LM, takes seq_i as input and generates contextualized embedding for each token in seq_i. Let h_{i1} and h_{i2} be the embedding generated by $LM(seq_i)$ for event mentions e_{i1} and e_{i2} respectively. Note that each event mention may correspond to multiple tokens, and its embedding is obtained by pooling the embeddings of all the tokens in it. Let $h_{(i1,i2)}$ represent the feature vector for predicting $y_{(i1,i2)}$. $h_{(i1,i2)}$ is obtained by concatenating 4 vectors: $h_{(i1,i2)} = [h_{i1}; h_{i2}; |h_{i1} - hi2|; h_{i1} \circ h_{i2}]$, where $|.|$ is the elementwise absolute value operator, and $\circ$ is hadamard product of two vectors.

The prediction in the baseline model is done by passing $h_{(i1,i2)}$ through a classifier C, i.e. $\hat{y}_{(i1,i2)} = C(h_{(i1,i2)})$.

We run our experiment on all 36 documents, in an 80-20 split. The overall baseline system has a total of $\sim$ 123 million parameters with RoBERTa base having $\sim$ 123 million parameters and the classifier having $18,438$ parameters. We train the model on a 2080Ti, and each epoch takes approximately thirty minutes to train. We run all experiments for a total of 20 epochs with a batch size of 32 and a learning rate of $2e^{-5}$.

We report the results of our experiments in Tab.4.4. We observe a 31% increase in the F1 score when compared with the same model on TDDiscourse. However, just like TDDiscourse, the results from our baseline suggest that existing models find it difficult to capture long-term dependencies. Despite this, unlike TDDiscourse, we see increased performance while predicting hypothetical events, because of the distinction we provide in our annotation scheme. This further strengthens the argument for the need of multiple timelines

Label	P	R	F1
BEFORE	28.2	18.8	22.6
HET	99	80	88.9
DURING	0	0	0
SIMULTANEOUS	68.3	32.2	43.8
INDETERMINATE	49.9	90.8	64.4
Overall	49.3	49.2	49.3

Table 4.4: Evaluation metrics for the model on DELTA. We report the metrics for non-vague event relations to avoid any bias caused by the number of *vague* relations

4.7 Timeline Evaluation

In order to evaluate the quality of our timelines, we choose a set of 10 documents from the corpus and construct timelines from two other datasets namely MATRES and TDD. We construct timelines for these datasets by adapting our inference rules for their corresponding labels, and automatically inferring relations wherever possible. We then proceed to make all the three timelines a minimal timeline. We define a minimal timeline to be a timeline which has the lowest number of relations while still ensuring that all possible relations can be inferred from these minimal set of relations. We ask a set of 6 human volunteers to analyze these timelines, and which timeline they feel best represents the document. We ask them to evaluate the timelines on two criteria: coverage and accuracy.

We see that in the overwhelming majority of cases, people prefer *DELTA* over the other two datasets. In 68.3% of the cases the evaluators chose *DELTA* for having the best coverage, and in 58.3% of the cases, the evaluators preferred *DELTA* for accuracy. Overall 59.1% of the volunteers preferred *DELTA* over the other two datasets.

4.8 Conclusion

In this paper, we design a new annotation schema for the identification of discourse-level temporal relations. Based on this schema we build and release a new dataset **DELTA**.**DELTA** is the first dataset that provides the complete set of event-event temporal relations for all verbal events in a given document. We also introduce the concept of real timelines and distinguish between real timelines and hypothetical timelines. As annotating at a discourse level is an expensive task, we build an annotation tool to make annotating long distance relations easier, and automatically populate inferrable relations to reduce the efforts of the annotators. We observe that human volunteer prefer the timelines provided by *DELTA* over the timelines generated from existing datasets. We also release a baseline system that generates discourse level timelines.

Due to the limitations on the length of the context, context encoder based methods can only capture a narrow context which is insufficient for long distance relations across a document. For future work, one possible avenue of approach is the utilization of Graph Neural Networks [Scarselli et al. (2009); Schlichtkrull et al. (2018)]. The problem of timeline generation lends itself well to a graph formulation, and in recent years graph based approaches have outperformed context-encoders in not just the discourse setting but also in the local setting [Mathur et al. (2021);Liu et al. (2021b)]. Since GNN's don't have constraint on the fixed context, they should be able to handle long term relations much better. Introducing logical constraints (such as ensuring that transitivity of `before` relations hold true) to ensure global consistency has also shown promising results when creating a discourse level timeline [Chambers et al. (2014) ;Ning et al. (2017)].

Another possible avenue of work is to expand the scope of the dataset. This can include expanding the set of events that are handled (i.e. nominal events), by introducing new documents in the same domain or from other domains such as "short fiction".

Limitations

We hope our approach to be a starting point for more work in the area of timeline generation. While we see some promising results, we observe the following limitations.

- Even with the annotation tool easing the annotation efforts and the automatic inference automatically populating new relations, it is still time-consuming and expensive to generate complete discourse level temporal relations between events.
- As we utilize TimeBank-Dense event annotations, we do not identify the events. Thus, a model trained on the dataset can only used on a document if it is accompanied with annotations for the events which needs to ordered temporally.
- We only consider verbal events. Thus, we lose out on the information given by the nominal events
- We only build a context based encoder to classify the temporal relations. Since the context window is small compared to the total size of the document, it cannot get all the context required for the prediction of long distance relations.
- Due to the difficulty of the task, as well as the small size of the dataset, even if the F1 scores are better than existing literature, the model cannot be used to accurately generate a complete discourse level timeline for the document.

Chapter 5

Conclusions

In this thesis, we have looked at how narratives can be understood through events and characters. We focus on two major areas: character-character relations and event-event relations. We looked at how to generate character arcs and narrative arcs given a novel. This thesis also introduces a new dataset for discourse-level identification of event-event relations. We also introduce a new annotation tool to streamline annotation efforts and reduce annotator fatigue. We create a strong baseline and lay the groundwork for document-level event timeline generation.

5.1 Future Work

There are many possible avenues for future work. In this section, we'll explore some key directions.

In this thesis, we focus on a single type of event-event relation, namely the temporal relation. However, there are many other types of event-event relations, such as causal and co-referential relations. Identifying and understanding these relations, in addition to temporal relations, is crucial to achieving narrative understanding. Existing methods typically model the task of extracting causal relations between events as a classification task. We can broadly classify the methods based on whether they utilize external i.e. non-document knowledge. There has been exciting progress along both methods [Liu et al. (2020)]. The methods involving internal context cues typically use syntactic features, lexical features, causal patterns etc to identify these causal relations [Hashimoto et al. (2014);Riaz and Girju (2010); Hidey and McKeown (2016)]. External knowledge methods usually involve a reasoning engine which can help mitigate the noisy nature of these relations in a document. Kadowaki et al. (2019) has seen success with using BERT Devlin et al. (2019) trained with causality candidate documents. Other methods such as using common sense reasoning engines along with annotated resources such as ConceptNet [Speer et al. (2017)] have also shown promising results [Rashkin et al. (2018); Mostafazadeh et al. (2020); Liu et al. (2021a)].

Understanding heterogeneous interactions such as the way characters influence events and events influence characters is also extremely important to accomplish the goal of narrative understanding.

Works such as automated character description generation [Brahman et al. (2021)], and analysis of character networks [Labatut and Bost (2019)] are promising efforts in this direction.

Related Publications

1. Suhan Prabhu, **Ujwal Narayan**, Alok Debnath, Sumukh S, and Manish Shrivastava. 2020. Detection and Annotation of Events in Kannada. In 16th Joint ACL - ISO Workshop on Interoperable Semantic Annotation, pages 88–93, Marseille. European Language Resources Association. Accepted.

2. Sriharsh Bhyravajjula, **Ujwal Narayan** and Manish Shrivastava. MARCUS: An Event-Centric NLP Pipeline that generates Character Arcs from Narratives. In Proceedings of the Fifth International Workshop on Narrative Extraction from Texts (Text2Story), 44th European Conference on Information Retrieval (ECIR), Stavanger, Norway, 2022. Accepted, Best paper in track.

3. **Ujwal Narayan**, Priyank Modi and Manish Shrivastava. DELTA: A dataset for discourse level event timeline generation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP). Submitted to EMNLP (Rejected); Submitting to ACL

Bibliography

Jonathan M Adler, Jennifer Lodi-Smith, Frederick L Philippe, and Iliane Houle. 2016. The incremental validity of narrative identity in predicting well-being: A review of the field and recommendations for the future. *Personality and Social Psychology Review*, 20(2):142–175.

Apoorv Agarwal, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. 2013. SINNET: Social interaction network extractor from text. IJCNLP (System Demonstrations).

Mieke Bal. 1997. Narratology: introduction to the theory of narrative. toronto, buffalo.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A Bayesian mixed effects model of literary character. ACL.

Florian Barth, Evgeny Kim, Sandra Murr, and Roman Klinger. 2018. A reporting tool for relational visualization and analysis of character mentions in literature. In *Book of Abstracts – Digital Humanities im deutschsprachigen Raum*, Cologne, Germany.

Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. 2021. ”let your characters tell their story”: A dataset for character-centric narrative understanding. *ArXiv*, abs/2109.05438.

Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006. Inducing temporal graphs. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 189–198, Sydney, Australia. Association for Computational Linguistics.

Jerome Bruner. 1991. The narrative construction of reality. *Critical inquiry*, 18(1):1–21.

Simon Bushell, Géraldine Satre Buisson, Mark Workman, and Thomas Colley. 2017. Strategic narratives in climate change: Towards a unifying narrative to address the action gap on climate change. *Energy Research & Social Science*, 28:39–49.

Taylor Cassidy, Bill McDowell, Nathanel Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. Technical report, Carnegie-Mellon Univ Pittsburgh PA.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense Event Ordering with a Multi-Pass Architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Dorottya Demszky, Dana Movshovitz-Attias, J. Ko, Alan S. Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. *ACL*.

Daniel C Dennett. 2014. The self as the center of narrative gravity. In *Self and consciousness*, pages 111–123. Psychology Press.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687.

Muriel Ekdahl and Joseph E Grimes. 1964. Terena verb inflection. *International Journal of American Linguistics*, 30(3):261–268.

JENNIFER R. ELLIOTT. 2000. Realis and irrealis: Forms and concepts of the grammaticalisation of reality. 4(1):55–90.

Micha Elsner. 2012a. Character-based kernels for novelistic plot structure. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 634–644, Avignon, France. Association for Computational Linguistics.

Micha Elsner. 2012b. Character-based kernels for novelistic plot structure. EACL.

Micha Elsner. 2015. Abstract representations of plot structure. In *Linguistic Issues in Language Technology, Volume 12, 2015 - Literature Lifts up Computational Linguistics*. CSLI Publications.

Harry Feldman et al. 1986. *A grammar of Awtuw*. Dept. of Linguistics, Research School of Pacific Studies, The Australian ....

Lucie Flekova and Iryna Gurevych. 2015. Personality profiling of fictional characters using sense-level links between lexical resources. EMNLP.

Gérard Genette, Alan Sheridan, and Marie-Rose Logan. 1982. *Figures of literary discourse*. Columbia University Press New York.

Prince Gerald. 1982. Narratology: The form and functioning of narrative.

Adam Hammond, Krishnapriya Vishnubhotla, Graeme Hirst, and Saif M Mohammad. Voices speaking to and about one another: Introducing the project dialogism novel corpus.

Chikara Hashimoto, Kentaro Torisawa, Julien Kloetzer, Motoki Sano, István Varga, Jong-Hoon Oh, and Yutaka Kidawara. 2014. Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 987–997.

Christopher Hidey and Kathleen McKeown. 2016. Identifying causal relations using parallel wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1424–1433.

Sajad Jafari, Julien C Sprott, and SMRH Golpayegani. 2016. Layla and majnun: a complex love story. *Nonlinear Dynamics*, 83(1):615–622.

Harshita Jhavar and Paramita Mirza. 2018. Emofiel: Mapping emotions of relationships in a story. In *Companion Proceedings of the The Web Conference 2018*, pages 243–246.

Kazuma Kadowaki, Ryu Iida, Kentaro Torisawa, Jong-Hoon Oh, and Julien Kloetzer. 2019. Event causality recognition exploiting multiple annotators' judgments and background knowledge. 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 5816–5822.

Immanuel Kant. 1908. Critique of pure reason. 1781. *Modern Classical Philosophers, Cambridge, MA: Houghton Mifflin*, pages 370–456.

Evgeny Kim. 2020. Emotions in literature: computational modelling in the context of genres and characters.

Teuvo Kohonen. 1990. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

Emily Kubin, Curtis Puryear, Chelsea Schein, and Kurt Gray. 2021. Personal experiences bridge moral and political divides better than facts. *Proceedings of the National Academy of Sciences*, 118(6):e2008389118.

Vincent Labatut and Xavier Bost. 2019. Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–40.

Mildred L Larson. 1984. *Meaning based translation*. University press of America Lanham, MD.

Kenton Lee, Yoav Artzi, Yejin Choi, and Luke Zettlemoyer. 2015. Event detection and factuality assessment with non-expert supervision. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1648.

W. Lehnert. 1981. Plot units and narrative summarization. *Cogn. Sci.*, 5:293–331.

Jian Liu, Yubo Chen, and Jun Zhao. 2021a. Knowledge enhanced event causality identification with mention masking generalizations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3608–3614.

Jian Liu, Jinan Xu, Yufeng Chen, and Yujie Zhang. 2021b. Discourse-level event temporal ordering with uncertainty-guided graph completion. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3871–3877. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Kang Liu, Yubo Chen, Jian Liu, Xinyu Zuo, and Jun Zhao. 2020. Extracting events and their relations from texts: A survey on recent research progress and challenges. *AI Open*, 1:22–39.

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

Robert E Longacre. 1990. Storyline concerns and word order typology in east and west africa. *Studies in African Linguistics*.

Robert E Longacre. 1996. *The grammar of discourse*. Springer Science & Business Media.

Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha. 2021. TIMERS: Document-level temporal relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 524–533, Online. Association for Computational Linguistics.

Teruko Mitamura, Yukari Yamakawa, Susan Holm, Zhiyi Song, Ann Bies, Seth Kulick, and Stephanie Strassel. 2015. Event nugget annotation: Processes and issues. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 66–76.

Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational intelligence*, 29(3):436–465.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. Glucose: Generalized and contextualized story explanations. *arXiv preprint arXiv:2009.07758*.

Aakanksha Naik, Luke Breitfeller, and Carolyn Rose. 2019. Tddiscourse: A dataset for discourse-level temporal ordering of events. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 239–249.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1027–1037, Copenhagen, Denmark. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. *arXiv preprint arXiv:1804.07828*.

Susana Onega and José Angel García Landa. 2014. *Narratology: an introduction*. Routledge.

Gerald Prince. 2003. *A dictionary of narratology*. U of Nebraska Press.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

James Pustejovsky, Robert Ingria, Roser Sauri, José M Castaño, Jessica Littman, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Inderjeet Mani. 2005. The specification language timeml.

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

Andrew J. Reagan, Lewis Mitchell, Dilan Kiley, Christopher M. Danforth, and Peter Sheridan Dodds. 2016. The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science*, 5(1):31.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal anchoring of events for the timebank corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2195–2204.

Mehwish Riaz and Roxana Girju. 2010. Another look at causality: Discovering scenario-specific contingency relationships with no supervision. In *2010 IEEE Fourth International Conference on Semantic Computing*, pages 361–368. IEEE.

Paul Ricoeur. 1980. Narrative time. *Critical Inquiry*, 7(1):169–190.

Sergio Rinaldi, Pietro Landi, and FABIO DELLA ROSSA. 2013. Small discoveries can have great consequences in love affairs: the case of beauty and the beast. *International Journal of Bifurcation and Chaos*, 23(11):1330038.

Marie-Laure Ryan et al. 2007. Toward a definition of narrative. *The Cambridge companion to narrative*, 22.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43(3):227–268.

Abraham Savitzky and Marcel JE Golay. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Henrik Schärfe. 2003. Grand principles of narratology: Paper præsenteret ved pala2003, istanbul, tyrkiet.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Peng Shi and Jimmy J. Lin. 2019. Simple bert models for relation extraction and semantic role labeling.

Robert J Shiller. 2020. *Narrative economics: How stories go viral and drive major economic events*. Princeton University Press.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. Literary event detection. ACL.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. EMNLP.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic. Association for Computational Linguistics.

Marc Verhagen, Roser Saurí, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden. Association for Computational Linguistics.

Kurt Vonnegut. 1995. *Kurt Vonnegut on the Shapes of Stories*. YouTube.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

K. M. Weiland. 2016. *Creating Character Arcs, The Masterful Author's Guide to Uniting Story Structure, Plot, and Character Development*. PenForASword Publishing.

Samira Zad and Mark Finlayson. 2020. Systematic evaluation of a framework for unsupervised emotion recognition for narrative text. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 26–37. ACL.

Xinyu Zhao, Shih-ting Lin, and Greg Durrett. 2020. Effective distant supervision for temporal relation extraction. *arXiv preprint arXiv:2010.12755*.