Extracting Event Mentions from Text and Anchoring their Occurrences to a Timeline

Ved Kurien Mathai

Masters of Technology Thesis

June 2017

International Institute of Information Technology, Bangalore

Extracting Event Mentions from Text and Anchoring their Occurrences to a Timeline

Submitted to

International Institute of Information Technology, Bangalore

in Partial Fulfillment of

the Requirements for the Award of

Masters of Technology

by

Ved Kurien Mathai

MT2015125

International Institute of Information Technology, Bangalore

June 2017

Thesis Certificate

This is to certify that the thesis titled **Extracting Event Mentions from Text and Anchoring their Occurrences to a Timeline** submitted to the International Institute of Information Technology, Bangalore, for the award of the degree of Doctor of Philosophy is a bona fide record of the research work done by **Ved Kurien Mathai**, **MT2015125**, under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Bengaluru,
15^{th} of June, 2017

G. Srinivasaraghavan

ABSTRACT

For tasks like information extraction, question answering, summarizing and, deductive reasoning, machines need to have a semantic understanding of the text and the facts it contains. Facts extraction includes converting information into forms that can be saved in a machine readable database, like DBpedia. Temporal facts form a major portion of these. In day to day, life events are dealt with in speech commonly and often are not fully specified, with listeners being able to glean enough information from context, implicitely placing the event on a timeline and ordering its occurrence with respect to another event.

Listeners also aggregate information about the same occurrence or fact through the use of discourse structures in order to form a summative understanding of the occurrence or fact, by resolving parallel mentions of the same event or fact and linking the two. They also aggreate information on event happening not from direct statements of their time of occurence but from a narration of the events as they have taken place or imminent to take place, with discourse words providing the necessary ordering information between events.

If machines are to develop understanding of events in order to deduce, derive, and glean meaningful information from spoken or written natural language statements, the way humans do, then, splitting the problem into three, they have to recognize and decipher event mentions in the text, resolve event referencing, and resolve relationships between the events so that ordering may be possible.

This thesis discusses each part of the problem in order to provide a solution for end to end system which takes free written discourse and give events on a timeline system. We have implemented one such system based on rules, while noting places for improvements as a matter of further research.

ACKNOWLEDGEMENTS

I would like to thank Prof. G. Srinivasaraghavan for the opportunity to do my thesis under him and for all his inputs. I would like to thank IIIT-B for providing the right environment prime for research.

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

In this chapter we discuss informally the meaning of tense and discourse and the linguistic appoach to these topics, while giving an overview of the method we followed in this dissertation for the extraction of events and their ordering, and noting down the flow of the rest of the thesis and its main contributions.

1.1 Time in language

Events are an action or happening that takes place at a given time or over time, may have an agent or person or thing who is doing or making the event happen, a location at which it is happening, and may have a recepient of the consequences of the action, which directly correspond to questions one may want to ask about event, to fully have a handle over them (*When; where; who did; and to whom?*) Each of these may have a direct answer (*December; in Bangalore; Ram; Sita*) or may have a reference to another event or descriptor. (*In the same month of Christmas; his home town; he; his wife.*) In general knowledge can be chained to form strings of coherent information to describe an agent or thing, but when the domain is limited to time, the general necessity of all entities to exist in a partial ordering and our ability to coerce them to one (given enough information), makes the problem interesting.

Text can be classified as spoken or narrative, reported or, encyclopedic. In each of these the way temporal information is represented is different. In narrative there would be few anchors around the text with other events represented in reference to these and in an a narrative fashion using discourse words like *then, and, after*, but the flow of events are usually in the same order of their occurences. Event mentions in news and other reports are mostly anchored to the date of publication, while order of their mentions within a report may not follow the chronological order since articles are written in the order of importance. Encyclopedic text, in particular *Wikipedia*, in general do not follow a narrative style with sentences being statements of facts. Anchors to event mentions can be found in the same or previous sentence as opposed to in narrative text which may have anchors multiple sentences before. Each of these styles warrant different ways in which temporal information can be extracted, which is not to say that a single piece of text will not have elements from all three styles.

Given the different styles of writing, the applications of such an extractor can be boundless: news reports, medical reports, discovery of evidence within emails, text etc for lawyers. Inferencing of information becomes much faster and deeper.

Either due to text interpretor inefficiency or the ambiguities in the text itself, when extracting temporal informations large amounts of inaccuracies may creep in. But given a large amount of text and a large number of mentions of the same event, logical coherence and closures between the events can be used to iron out the inaccuracies.

We know events can either be *before, after, during* other events, or *start, end* or be *simultaneous* with other events, more on this later in the next chapter. This extends to events and times too. Knowing

- *event1* happened before *event2* and *event2* happened before *event3* will tell us that *event 1* happened before *event3*
- *event1* happened after *event2* and *event2* happened before *event3* and if a time of occurence of *event1* and *event3* is determinable then *event 2* can be bounded between *event3* and *event1*.

The larger the network created the more specific the probabilities will become.

The notion of temporal representation in language has been a popular topic of study in philosophy, linguistics and computer science. Early work [1] qualitatively classified verbs according to their aspect, tense and modality even if it were presented more as the puzzle of tenses in language than in a procedural method. [2] however gave a more structured form to tenses by describing all combinations of tense and modality as combinations of *reference, speech and event times.* These gave a starting point to other theoritical work which aimed to give a method to the madness that was time and its representation in text. Work like [3, 4] organized [1] ideas on aspect, tense and modality to attempt a deterministic structre that would fit these different permutations of tenses.

Modern work in computational linguistics took off in earnest with the Tempevals which are covered more in chapter 7

1.2 Central Thesis

The central thesis of this dissertation is that given many mentions of events and their relationships with other events or time in a large piece of text, creating probabilistic temporal graphs will tighten the bounds for the occurence of events and ground previously unanchored events to particular instances or durations on the time line.

Event extraction has a large portion of literature dedicated to it. [5, 6] aim to extract a temporal information that is globally correct over the corpus of text. After text extraction these use optimization techniques to find the most cohesive set of temporal statements. [7, 8] in particular explicitely creates a temporal graph over the events before optimizing it given constraints that aim to improve overall correctedness and [9] uses markov logic to form the largest cohesive set of cohesive extracted temporal facts.

factoid Many of these methods, however, do not account for durative events (events that occur over a long stretch of time nor do they have a provision to account for events that start and stop other events, where the duration of the event is not during the entire event but only the beginning and end. And they aren't also not end to end. They do not start from text to events and corresponding times. They instead work directly off of already annotated text or on specific domain like healthcare, in the case of [7]. [5] serializes other individual efforts to create an end to end system, which extracts text, recognizes them and then uses a markov logic network to create the closure.

In contrast the system described in this thesis recognizes events and resolves event coreferences based on heuristic rules and builds a probabilitic graph which models probabilities directly as a function of time. And propagates the probabilities through the graph iteratively until convergence. Events that are anchored to points on the time line propogate their effect that is based on the relationship between them and the neighbouring node, onto the neighbouring node. All events for which a relation between them and atleast one node with an anchoring to the timeline, will be guaranteed to have an anchoring to the timeline. Initial probabilities are based on count based statistics of the factoid mention, which assumes that the more number of times a factoid is mentioned the truer it would be.

Suppose the following sentences are uttered at different places in the discourse, but in the given order.

1. Sarah gave birth to her son on 14^{th} February.
2. Sarah was in London before the birth of her child, Christen.
3. Sarah had her first checkup 32 weeks before she gave birth to her child.
4. Sarah was in London just after her first checkup.

After (2) is uttered we would know that the event 2 <*Sarah, returned_from, London*> happened before <*Sarah's child, born,*>. But, 2 anchored the birth event to a point in time. Thus a distribution is assigned to event 2 such that probability of it happening after event 1 is reduced. But the occurence of event 2 itself is a distribution over the possible times it could have occured at. Thus the update will be weighted by this distribution. Following similar steps for (3) and (4) we find a distribution over time for the when Sarah was in London. And will be able answer a question such as *When was Sarah is London?,* with a distribution over the time that she was. Admittedly, this example is a little contrived since the events lexically match each other thus coreference is relatively easily than if the sentence (4) were *Sarah went to London just after her first checkup,* where the state of *Sarah in London* will have to be matched to the consequence of the event *Sarah went to London.* [1, 10].

The main contribution of this thesis is a method that assigns a probability distribution for the occurence of an event over time, and uses the known distributions over the relations between events to update the distributions for all events everytime a distribution is updated with new evidence from text, with the premise that strongly anchored events can propogate their strong anchoring over a event-event relationship with a weakly anchored event inorder to strength their anchoring in time

1.3 Challenge Posed by Temporal Information Extraction

It is characteristic of human language to not specify entities in their entirety in speech, instead using references and anaphora. Pronounial anaphora like *he, him, her, she* are used to replace nouns when references to the entity are repeated through the discourse. In many cases authors refer to the same entity using different words, for example *Narendra Modi*, *the Prime Minister* and *former Chief Minister*, all refer to the same entity, however it is left to the listeners to use previous knowledge to resolve these references. Similarly, when mentioning events, speakers tend to refer to event entities using different names, and sometimes using **shell nouns** such as *that, this* to refer to events mentioned previously in the same text. Compared to pronounial anaphora, event coreference resolution is a much harder task, which follows since pronouns form only a part of event definitions. Two references to events may be pointing at the same event yet have different terms for the subject, verb or the object. For example, *after the prime minister's swearing in*, and, *after Narendra Modi took office.* Most of the work in event recognition use a trained classifier to recognize events however do not recognize event coreference resolution as a separate problem which is hard to solve on its own.

The main marker for events does not necessarily have to be a verb. They can even be nouns and adjectives. For example, *birthday, Christmas, May Day* are all events that are nouns. Even noun verbs like *attack, fight etc* are events which are to be recognized.

Another challenge is the vagueness with which the times of occurence of some events are refered to. Writers of news articles sometimes vary the specificity of the time of occurrence according to the importance of the event they are refering to. The same event of Narendra Modi's swearing in can be refered to as *yesterday's* swearing in, *last week's* swearing in and *last year's* swearing in ceremony, for time of speech, a day after, a week after and a year after the actual event respectively. This makes it hard to find exact times for events if global knowledge is not readily available. This case is archetypical of why event coreference resolution is important.

Without it, finding further details for events becomes harder.

1.4 Thesis Organization

This thesis goes through the three difficult problems of *event recognition, event coreference resolution* and *placing the events on a timeline with respect to each other.*

In chapter 2 the linguistic and theoritical results of the treatment of time and events in language are discussed. These give a basis upon which a thorough rule based temporal extractor can be attempted to be created.

In chapter 3 the rules for event extraction are discussed including methods for resolving noun verbs. We discuss the working of the external tools used and show the need for the uses. We also discuss alternate methods in the section.

We discuss event coreference resolution in chapter 4 and the closure mechanism we use in chapter 5. Along with alternate methods that are implemented in literature.

The experimentation and results are discussed in chapter 6 and in continuation to the related work that are discussed in the individual chapters where relevant, other related work is illustrated in chapter 7. Finally, the possible improvements conceptual and in implementation and, future work is discussed. 8

CHAPTER 2

TIME AND ITS ATTRIBUTES

2.1 Introduction

In general narration, be it written or spoken, events are seldom always uttered in the order they occur. Speakers tend to refer back to events (or even those that are going to happen) that may have happened that may or may not have an influence on the event currently being spoken about. The handles that listeners use to place these happenings in rough partial ordering are tense, aspect and aspectual classes.

Though this will be elaborated on further, *Tense* places the event relative to the time of either the current time, as past, present or future. An event can be in one of the many *Aspectual Categories*: culmination, culminated process, point, process, to use the terms as in [10], who in turn adapted them from [1]. Events are classified into one the above categories based on their *Aspects* like whether the event is that of progressive or perfect. Another aspectual modifier is the use of adverbs like *in* and *for*.

We first discuss the *Tenses* and the three points and their arrangements that Reichenbach [2] uses to define all nine forms of tenses: simple, perfect and continuous for each past, present and future.

We then discuss the aspectual classes and how they are defined in terms of their aspects and tenses and the various approaches taken by [10], [1] and [4]. These papers discuss many anomolous cases where sentences and discourse throw up what seems to be paradoxes. We address some of these important cases.

Table 2.1: Reichenbach Tense with Single Point Event

Tense	Examples	Order of Points
Past Perfect	I had seen John	E<R<S
Simple Past	I saw John	R,E<S
Present Perfect	I have seen John	E<S,R
Present	I see John	S,R,E
Simple Future	I shall(will) see John	R,S<E
Future Perfect	I shall(will) have seen John	S<E<R

After which we discuss the first part of Allen's seminal paper [11] about representing all events as periods.

2.2 Reichenbach's Theory of Tenses

Reichenbach defines three points, whose arrangements for a given sentence defines the tense of the sentence: the *point of speech, point of reference and point of the event.* As can be derived from the names: the *point of speech* is the point in time at which the sentence is uttered, the *point of the event* is the point at which the event occurs and *point of reference* is the point to which the spoken sentence is anchored to. Though the time period is refered to as a point by Reichenbach, he does allow for events to stretch over periods, thus allowing for a sense of progressive tenses.

We illustrate on Reichenbach's own example: *Peter had gone.* Reichenbach argues that there are actually two points being alluded to in the sentence. The time at which Peter had actually gone and a time between his going and the speaker uttering the sentence at which it is known that Peter had definately left before. But this is clearly different from the sentence *Peter went*, and *Peter has gone*, even though the act committed by Peter has happened before the speaker uttered the sentence. The point at which the speaker anchors the clause changes in each of the sentences. In *Peter had gone* the tense is past perfect, where the event is definitely over before the time alluded to which in turn is definitely before the speech time. In 'Peter has gone' the tense is present perfect, thus the event is definitely before the reference point which is the same as the point of speech. However, in *Peter went* the tense is simple past, where the point of event is the same as the point of reference which in turn is definitely before the speech time. Extending this the present and future we get forms as in table 2.1.

Table 2.2: Reichenbach Tense with Extended Events

Tense	Examples	Order of Points
Past Perfect Extended	I had been seeing John	$\mathrm{e}_1<e_2<R<S$
Simple Past Extended	I was seeing John	$\mathrm{e}_1<R<e_2<S$
Present Perfect Extended	I have been seeing John	$\mathrm{e}_1<e_2<S,R$
Present Extended	I shall(will) be seeing John	$\mathrm{e}_1<R,E<e_2$
Simple Future Extended	I shall(will) be seeing John	$\mathrm{R,S}<\mathrm{e}_1<R<e_2$
Future Perfect Extended	I shall(will) have been seeing John	$\mathrm{S}<\mathrm{e}_1<R<e_2<R$

As was mentioned Reichenbach allows for an extended tense, which according to him doesn't stand for the actual event occuring through the whole time rather that the event happens at points throughout the length of the period. So it stands for a sense of repitition, as in *I am seeing John* doesn't necessarily mean the viewer's (my) eyes are constantly trained on John as it can also mean that through out that period the viewer every now and then only did the act of seeing John in the literal sense.

Moens and Steedman [10], as we will discuss later, make discourse a function that can coerce an utterance into one or the other meaning. They form a construct which allows the above sentence to mean either seeing in the periodic sense (once every day) or the sense of continuous seeing.

What others treat as states Reichenbach doesn't explicitely call a state, yet just an extended state which again is true for every point in the period. This includes adjectives like, *He is hungry.* Which is a state of being hungry and which will be true for every time point until the point the subject stops being hungry. According to Reichenbach adjectives are used mostly for a sense of permanent state of things, without clarifying whether the state can change.

Permanence of the reference point is the principle by which sentences or clauses that follow each other have to follow to be grammatically coherent with respect to tenses. The other implicit constraint being that the time of speech of the clauses should also remain same. Using his own first example we illustrate:

I had mailed the letter when John came and told me the news.

The first clause is of past perfect and has the form, E<R<S. While the second and third clauses

are of R,E<S. Here the time of R and S are the same. As a non felicitous case we take

I had mailed the letter when John came and has told me. *

Which transfers the reference time to the present and makes the tenses unacceptable.

Sometimes an explicit time position is mentioned in the utterance, such as *yesterday* and *now*, or even a date like *January 1, 2017*, which in this project is taken as a very important contributor to the beliefs of when an event occured on a timeline and correctly anchors the event to that point. This according to Reichenbach is actually the reference point to which they are alluding to. His example includes, *I met him yesterday*, which due to the fact the reference point and event point are matching, the event did indeed take place the previous day. However, he claims that *I had met him yesterday* actually places the reference point to the previous day and the event could in fact have happened any time before that. Even two days before.

Reichenbach shows a case where the permanence of the reference point seemingly is violated, *he was healthier when I saw him than he is now.* The first clause is simple past of the form, R,E<S, the second clause is also the same, while the last clause is of simple present. Remember, the time of speech is always the same, therefore $R_2 < R_3$. And this violates the principle of reference point. However, according to Reichenbach the use of 'now' is a usage of the *positional use of the reference point*, which is more general and has a priority over the permanence of reference point.

2.3 Aspectual Categories: States, Process, Accomplishments, Achievements

In addition to tenses, sentences can belong to one of many aspectual classes such as states, processes, accomplishments and achievements. This nomenclature is not standard but we start by using the ones Vendler [1] uses and we change it as we shift focus to the other's works such as Pustejovsky [3] and Moens and Steedman [10] and to stay consistent with literature.

According to Vendler, a process are actions such as running or writing which have sub-parts like lifting a leg and moving a pen on paper actively going on in order for them to be an ongoing action, however knowing doesn't have a sub part which we can claim to be going on at the current time.

Also, actions which have a certain end result like *running a mile* and *drawing*, have a subpart has to be completed in full to claim that they have occured. If they are stopped midway then they have not occured. These actions are labeled as **accomplishments**, as opposed to activities like pushing a cart which is does not have a culmination. This is why a question such as, *How long did he draw the circle?* can be begged of that action while, *For how long did he push the cart?*, can be begged of the other.

As a further refinement to accomplishments, Vendler adds one more class know as **achievements**, which are point events that occur at the end of a preperative period of another action which this event capped off. Like winning a race is the end of a period of participating in the race, and reaching a summit of a mountain comes at the end of a period of climbing. The need for two distinctive categories for activities that happen over a period has been debated over by [10], will discuss their arguments later.

However, by way of illustration provided by Vendler the differences between the 4 types of time schemata is subtely shown:

1. Activities: **A** was running at time t means that time instant t is on a time stretch throughout which A was running.
2. Accomplishments: **A** was drawing a circle at t means that t is on the entire time stretch in which **A** was in the process of drawing and completing a circle.
3. States: **A** loved somebody from t_1 to t_2 means that any instant between t_1 to t_2 it is true that **A** loved someone.

Vendler then goes on to describe how these 4 schemas are not complete and there are in fact many cases in which clauses don't fit into one of these schema. But we are not interested in trying to classifying individual sentences as much as having a broad idea as to what the possible categories clauses can be in. As [4] points out, some temporal adverbials are point events while some others are those that occur over a period. John arrived at 3 p.m is an example of the former. Recognizing the usage of the different adverbials for the different aspects are important.

Mary drank a beer

Mary drank beer

belong to two different aspectual classes seemingly because of the presence of a determinant.

2.4 Moens and Steedman's Verb Coercion

Moens and Steedman [10] extend Vendler's classification into that of a functional ontology where under certain conditions verbs can be 'coerced' into a particular aspectual class. Their first example is that of a punctual event *Harry was hiccupping*, which is a point event coerced into a process. Then the meaning is now interpreted as iteration or repetition of the basic event. His functions are of the following kind

progressive(process(iteration(point(Harry hiccup))))

Moens illustrates with examples the many ways verbs can be coerced to a certain aspectual types, using aspect, tense and adverbials. Even though these may actually just be a revisit of Vendler's classes with a new lens. We repeat selected examples here because a handle over this kind of coercion can help lay a starting point and both, enrich by enumerating the many ways verbs are coerced and, not overcomplicate by clubbing many rules in the same bracket and reducing redundancy in the process of writing rules for extracting temporal meaning from sentences.

The basic structure of Moen and Steedman's event is something they call a *nucleus*. It consists of a *preperatory* process, *culmination* and *consequent state*. If reaching the top of a mountain is the culmination then the climb was the preperatory part, and the *fact* that you have reached the top being true is the consequent state. Each of these parts may be a aggregate of other events. So the culmination itself may have it's own preperatory process, culmination and consequent state.

2.4.1 Tense

We already saw an example of coercing a point event into a process by interpreting it as a repetitive action.

Another coercion to a progressive is that of a culminated process. In this, the sense of the culmination is removed and only the preparatory process is present. An example is, *Roger was running a mile.*

Taking the two coercions above, one can combine them and make the entire culminated process to a point and then iterate it. As an example: *Roger is baking cakes nowadays. He has baked almost twenty-five cakes till now.*

By way of removing a sense of the culmination, nothing is known about the truth of the culmination or whether it really happened. Thus by making it a process, the sense of the culmination is lost.

2.4.2 Perfect

The Perfect Tense requires there to be a culminating state, as in *Harry reached the top*, tends to mean he is still at the top. However, if the consequent is not true then the perfect seems to be wrong. For example, *I have spilled my coffee,* means that the state of the consequences of spilling coffee are in effect, however if the consequences are later mitigated then the perfect present seems unsuitable.

If the input does not have a culmination then the perfect will coerce it to a culmination, as long as previous world knowledge allows it to be. Like the unusual sounding, *the star has twinkled* can only be acceptable if the act of twinkling itself is more special than otherwise.

Has John worked in the garden? is coerced to a culmination if it is treated as a single unit and as the starting point for another event: as part of a prearranged plan.

Culminating process are mostly independent of a time anchors, however if the time at which the event takes place is important and consequential then it may be mentioned. For example, *They have married on Friday the 13th!*

The difference in a pair of sentences like *Einstein has visited Princeton,* and *Princeton has been visited by Einstein,* is the fact that the in the first it is imperitive that the subject has a continuing consciousness, but it does not, while the second, *Princeton* does in fact have a enduring conciousness.

2.4.3 Adverbials

As pointed out in Vendler, *for* and *in* adverbials are used for process and culminating actions respectively. However, something like *Sue played the sonata for about eight hours*, though can take the 'for'. For a process that went on for eight hours, our *knowledge about the world* and about how long sonatas take to play rules out any attempt to make it so and instead makes it an iterative process by coercing playing a sonata to a point.

Coercing a verb to a progressive requires a -ing form of verb to be present while coercing to a culminated verb requires a perfect. There are no markers in English to explicitely coerce a verb to an iterative form. The ambiguity given by the following two sentences:

Nikki Lauda won the Monaco Grand Prix for several years

Nikki Lauda stopped racing for a few years

displays that.

If a culminating event is attached with an *in-adverbial* then the preperatory period is what is alluded as the process that got over in that time.

To convert a process variable to culminated process with the in-adverbial is to attach a culmination point to the process. So to make *John ran in 10 minutes* acceptable world knowledge would have to add the fact that *John ran a mile* in those 10 minutes to make a culminating process like before.

Another way to look at it is, as seen before, we coerce the event into a point and then interpret it as an iteration.

The power of the functional form of dealing with tenses and adverbials is that they can be clubbed together to form loops. Like in

It took me two days to play the "Minute Waltz" in less than sixty seconds for more than an hour

The verb phrase *play the Minute Waltz* is coerced by the in-adverbial into a cuminated process, this in turn is coverted to a process with the for-adverbial by making the culminated process into an iteration. Finally the *it took me* is like an in-adverbial, thus it takes a culminated process and attaches a duration to it.

2.4.4 Temporal Focus

This is an extention to Reichenbach's notion of reference time where tenses are treated as having an anaphoric character, as in they reference a time elsewhere mentioned in the speech. In clauses that follow each other in a narrative the reference time seems to move on from clause to clause, as in:

At exactly five o'clock, Harry walked in, sat down, and took off his boots.

Pragmatically the events have to follow the previous ones else unless context doesn't allow for it the clause may sound absurd. The structure of a event having a preperatory phase, a culmination and a consequent can be used to explain if the temporal focus should advance, if the next even falls within the consequent state of the event in the current focus point then the focus will move on. If it does not then it will start a new event structure entirely.

2.4.5 When-clauses

When clauses do not require a previously established temporal focus but brings a completely new event into focus. Mostly ones that can be refered to by the listeners understanding of the world.

An example where three different end clauses to a *when* sentences:

When they built the 39th Street bridge...

...a local architect drew up the plans.

...they used the best materials.

...they solved most of their traffic problems.

Either the core event is decomposed into a nucleus (event structure) and a transition is made to one of the compenents, as in to the preperatory activity of the event of the consequent state of having done the event. Or else the entire event is made the culmination event and then a preperatory or consequent state is taken from context.

In the first clause above the entire building of the bridge becomes a culmination point and the preperatory phase is the drawing up of the plans. In the second clause the building of the bridge is decomposed into three parts and the using of the materials fall into the preperatory part of the decomposed event. The last is the same as the first except now the clause lies in the consequent of the event.

If the second clause to the when was a state, then there is not causal or contigency relation between the two clauses rather just a predicate of the state at the time of the event culmination. If however the event is an inceptive event that begins the state of something then that takes the form of the stative predicating the state after the culmination.

2.4.6 Refering to Future Events

A sentence like

Robert was working on the speech project until he got a job offer from Sussex.

can be both the **imperfective progressive**, by answering questions about the past activities, or it can be **past futurate progressive**, by answering a question about the what the plans for the future were in the past. The imperfective progressive seems to decompose the event into its constituent parts and it places the focus on the preperatory process, indicating that it is in progress at the point of reference. The futurate progressive uses an adverbial to signal an event time after the reference, and the whole event is a structure of its own. The progressive part means that the event is still ongoing at the time of reference, and the futurate progressive is similar to the perfect in the sense that it speaks something about the reference point which is different from the event time.

The nonmodal future needs an adverbial to anchor the event. For example, one wouldn't say, *I leave* to describe what is currently happening but rather *I leave today*, or *I leave next Monday.* However, progressives do not necessarily require a temporal anchor, like *I am leaving.*

2.5 Dowty's Temporal Discourse Interpretation Principle

This principle deals with how the reference point changes (or doesn't) in succesive sentences in discourse. The principle is, given a sequence of sentences S_1 to S_n the reference of each sentence will be consistent with the definite time adverbial in S_i otherwise the t

2.5.1 Applied to Accomplishments/Achievements

Taking a pair of sentences such as, *John entered the president's office. The president walked over to him.* The second event's reference time will be placed immediately after the reference point of the first, however the understanding of immediately after is dependent on the listeners knowledge of the world and prior knowledge about how long the given events normally take to happen. In the next three sentences *John moved to Boston in July. He took a job in a steel mill. His boss became a close friend of his.* The time elapsed between the events can be in terms of days, weeks or any such time period. The important point is that the next clause includes the next event that warrants interest from the listener.

To deal with a past perfect second sentence. The reference time of the second event is actually placed after the reference point of the first sentence however the event can have taken place at an earlier time. Gricean principles (if a more specific placement of the event on the timeline is known to the speaker, then that specific definition will be uttered rather than the less specific one) rules out the fact that the two events were infact simultaneous.

A lot of information about when events overlap with each other are gained not only from the order of utterance but also from assumptions about when these events are to take place in the real world.

Another reason for why achievements and accomplishments do not overlap is that, for an accomplishment to be true for an interval I, it is not true for any subinterval I′, it cannot be true for any superinterval I′′ either because if it is then it is true for a subinterval of I′ of I′′ thus violating the original condition. Thus two successive periods during which the accomplishement is true will be non overlapping.

2.5.2 Applied to Statives

Unlike for activities, if a stative is asserted to be true at an interval I, then there is nothing that stops it from being true for a superinterval I′.

Compare the statements: *Yes, he was asleep between 1:00 p.m. and 2:00 p.m. in fact, he fell asleep at noon and did not wake up until 3:00.* and *Yes, Mary built that house between July 1 and December 1: in fact, she began building it in June and did not finish it until January.*

In the second sentence, the event is an accomplishment which is what makes it unacceptable. In all probability a hearer of a stative will assume that the state began well in advance of the point of utterance, like *Mary entered the president's office. There was a bound copy of the president's budget on his desk.*

Dowty argues that even if there are successive statives in a discourse, even though the states may be parallel, the order in which the narrator is observing the states are one after the other and the reference time is shifting ever so slightly with the time of the observation. Like in the sentence: *Mary entered the president's office. A copy of the budget was on the president's desk. The president's financial advisor stood beside it. The president sat regarding both admiringly. The advisor spoke*

However, the use of an adverbial like 'suddenly' can cancel the inference that the state was true from an earlier point in time and thus this point becomes an inceptive(start) point for the state. Dowty points out that some verbs like *sit, stand,* and *lie* are very frequently inceptives and thus can be ambiguous between stative and inceptive readings. It should be pointed out that without an adverbial too the second event may be an inception for a new state. If the second event is that of an inception then it is required that the presupposition of that change be true before the state changed. That is a sentence like: *The book was on the table at* t_0 ...

Mary put the book on the table at t_n.

2.5.3 Applied to Activities

Just like statives, **activity** is asserted to have taken place at interval I and could have begun before and continued beyond I. The surrounding discourse determines whether the time of activity is the first or the last part of the time it took place in. In an extreme case the sentence: *John entered the president's office. The clock ticked loudly*, the activity seems to have been both going on before the reference time and will continue to go on after that.

If the first clause is the cause of the second sentence then the causality will prevent overlap of the two events. However, some events are the consequences of another and they will be simultaneous to that of the first action.

2.5.4 Applied to Progressives

Progressives like statives allow their events to overlap the ones in the surrounding discourse. Unlike stative, however, the larger interval over which the event takes place is defined even if it overlaps the other events in discourse. Progressives unlike statives do not have the concept of inception. It may be possible that the reference time is infact the start of the progressive, but it will seldom be specified through the progressive and rather the event of starting or ending will be specified satisfying the Maxim of Quantity, that is the more explicit expression will be used.

However, there may be a case where a quasi-inceptive sense of progressive may exist. This Dowty argues is due to the lack of awareness on the part of the observer rather than the fact the event actually started at that point. For example, *John was felt his way up the staircase, there was a cracking sound and suddenly he was falling through.* Is helped by the adverbial 'suddenly.'

2.6 Allen's theory of Time

Allen introduces an interval base logic where the time of occurence of an event can be described as the interval over which it occured. The reason he gives for the use of intervals rather than time points is the following: If one were to describe the event of *opening the door*. One can always divide the action into subactions and the following subactions further. But are instantaneous points necessary after that? They may create problems according to him, suppose there is a time that proposition P is true and a time immediately after where P is not true (P), then there has to be a time when both the time periods meet and overlap. Then either both propositions are true at a point or else neither are true. We can have it such that only the lower end is included in the event and not the upper end, but that is artificial. Thus, we can settle to have events definied by their intervals and instantaneous events to be at points at which the event starts and stops.

Between any two temporal intervals there exists only 13 possible relations between them.

- During(t1, t2): the time interval of t1 is fully contained in time2.
- Starts(t1, t2): t1 shares the same beginning as t2 yet ends before t2 ends.
- Finishes(t1, t2): t1 shares the same end as t2 yet begins after t2 begins
- Before(t1, t2): t1 occurs before t2, yet they do not overlap.
- Overlap(t1, t2): t1 starts before t2 and they overlap.
- Meets(t1, t2): t1 is before t2 and there is not interval between them. t2 starts where t1 ends.
- Equal(t1, t2): t1 and t2 are the same interval;

In the next chapter we discuss the first step of the three part process for end to end temporal information extraction.

CHAPTER 3

EVENT EXTRACTION

A definition of what constitutes an **event** can be a main root word and its corresponding attributes. The root word could be a verb as in *go, brought, fought, shouted etc.* or a noun-verb like *death, birth, marriage, divorce etc.* Attributes of events consist of

- Subject: The doer or main agent of the action. *The pope retired today* has the subject as the *pope*. However, when noun verbs are used the subject can be the noun verb itself as in, *The pope's retirement brought thousands to the Vatican*, where the pope's retirement is the subject to the the verb *brought*.
- Object: The reciever of consequences of the action the agent does. In passive tense the object becomes the subject of the sentence, even though it is the reciever of the consequences. *The pope's retirement recieved backlash from some.* Direct objects have a change of state after the action due to being directly acted upon while indirect objects are acted upon due to circumstance.
- Prepositional Phrase: This can include an object that is attached to the event by a preposition. It more importantly is not the direct reciever of the consequence of the action yet provides a setting for the main action. It can include a location as in, *I am going to Rome for a vacation* where both *Rome* and *vaction* are prepostional with *Rome* giving a setting and *vaction* giving it a purpose. Parallel to the giving a geographical location prepositions also give temporal settings to the action as in, *The pope retired in 2013.*

The main idea of event extraction is to find the root word of the action and its attributes. Exctracting events from text and normalizing means to correctly search for verbs or nouns or adjectives that could denote an event and find the corresponding time to which it refered if mentioned. In text inasmuch events linked to a time of occurence by a prepositions they can also be linked to another event entity by a prepositional phare. Extracting the former gives us an event-time relation for the event and the second gives us an event-event relation, which follow relations as noted in section 2.6.

Early work in the field led to the creation of a de facto standard TimeML in text [12], which marks and normalizes the time mention in the text, yet does not markup the semantics of the action, such as the subject or object. In our work we use a tool SUTime, part of the Stanford CoreNLP toolkit to extract the information we need from the times recognized by the tool along with a dependency parser to extract other information such as the subject, object, propositions. In the next section we discuss the TimeML annotation language, after which we discuss the dependency parses that we use to extract the the verb attributes and the event-event relations and time-time relations.

3.1 TimeML

Simultaneous to the growth of TimeML as a de facto standard for time annotation was the growth of the TimeBank Corpus which is the gold standard for the TimeML language. The TimeML language itself was created to address four basic problems in events identification in text:

- Time-stamping an event by identifying it and anchoring it to a location in time.
- Ordering of events with respect to each other.
- Resolving underspecified temporal expressions.
- Dealing with the durative events.

TimeML had gone through three iterations and the current standard is TIMEX3. The latest version has the following imporvements:

- It can deal with intensionally specified expressions like *last week, next month, etc.*

- It can deal with both temporal prepositions (*for, during, in*) and temporal connectives (*before, after, while*)
- It has a provision to classify verbs as tensed verbs(build), stative adjectives(*built*, or event nouns (*World War*)
- It creates relations between events and times and differentiates between anchorings, orderings and embeddings. [12]

TimeML's definition of an *event* is a situtation that happens or occurs. They can be punctual or durative or even states in which some factor is said to be true. However, not all stative predicates are marked up as only those that are directly related to some temporal expression or those that noticeably change over the course of the document are. This does does not suit the case of interdocument annotations however since events states may change over the course of a corpus and not marking up states will lead to information loss.

Timex's mechanism to deal with events that are repeated such as, *playing the piano three times last week*, is to create multiple MAKEINSTANCE tag each for each occurence of the event.

TimeML has a SIGNAL tag that denote function words that relate events together, which can be prepositional such as *before* in the sentence, *He met his friends after parking his car.*

TimeML separates the annotations of the text from annotations of the links that relate two events or an event and time. The links are

- TLINK: representing a temporal link between events or event and time. These include the cases of *simultaneous; identical; before; after; immediately before; immediately after; including; being included; during duration of other; beginning of the other; begun by other; being ending of other; ended by other* which are in essence an enumeration of Allen's 13 event-event relations described in section ref.
- SLINK: subordination links used for contexts introducing relations between two events, or an event and signal. These include *Modal* which introduce a possible world that would have taken place; *Factive* which assumes that the clause it refers to was in fact correct;

Counterfactive which assumes the clause it refers to was in fact wrong; *Evidential* the clause is brought on by reporting or perception; *Negative Evidential* the reporting or perception shows the negative polarity of the clause.

- ALINK: aspectual link representing relationship between aspectual event and it's arguments. These include *intiation; culmination; termination; continuation.* An example of initiation being *John started to read.*

The purpose of this section is to give an overview of the different features a state of the art annotation tool covers. Using this insight we can validate the features that our own annotion tool extracts. There are other features that TimeML annotates like *causality* through there occurrence tag. However, we skipped those since they are not imperitive to the end goal of this thesis. In the next section we overview dependency parsing, which extracts relations between words in a sentence. The rules based extractor use this parse to extract the attributes of the events.

3.2 Dependency Parse

Dependecy parsing is a syntactic parse which is used in natural language understanding, as it is rooted in the natural traditions of how sentences are syntactically formed. Dependency parses follow the concept of *lexical integrity* where individual words are the fundamental units of a sentence as opposed to phrases, since this make computation easier.

For further details of the design principles and evolution of the Stanford dependencies one is refered to [13, 14]. However, the stanford dependency parser has from the outset been designed to be easy on the uptake for new users. At its essence the dependency parse treats each word as a node in a tree rooted at the main verb with the attributes of the verb made children of the main verb. Other clauses are subtrees rooted at the main verb of the clause. The clauses are children of the main verb. Traversing the tree is intuitive and makes extracting syntax easier.

3.2.1 Definitions

In this section we will discuss the essential parser tags that we have used in the parses of English sentences.

Root

The root is normally the main verb of the first clause. However if the main verb is a form of the *be* family then the root is the head word of the noun pharse that is the argument of the *be* word.
The minister won the reelection, has *won* as the root. *He is the minister,* has as the root.

nsubj

This is the noun pharse that is the syntantic subject of the root of the clause.

nsubjpass

Is the subject of sentence that has passive voice.

dobj

The direct object of the verb phrase. The object that is directly acted upon by the agent and its action.

iobj

The dative object of the verb pharse. *She bought him cake. cake* is the direct object. *him* is the indirect object.

ccomp

Is the **clausal complement** which is clause that has its own subject and acts like an object of the root verb. *It was reported that the Macron had won the election.*
reported and *won* are related my a ccomp relation.

xcomp

Open clausal complement is a relation between the roots of the parent clause and subordinate clause, where the subordinate clause does not have a subject of its own. The subject may normally be the object of the next higher clause if available or the subject of the next higher clause. like to run. has a xcomp relation between *like* and *run.*

advcl

Is an adverbial clause modifying the verb. (temporal, consequence, conditional, purpose etc) *Birds migrate as it gets colder in winter.* where *migrate* and *gets* are related by advcl.

aux

Is the non-main verb of the clause. It is the modal auxiliary or a form of *be*, *do* or *have.*
The birds have flown away. Here, *have* is the auxiliary.

auxpass

The non-main verb of the clause that is the passive voice form of the auxiliary.
The birds were scared away.

cop

Is the relation between a copular verb and its complement. *The weather is good today.*

punct

Is used to link the main word and a punctuation in a clause.

conj

Conjuction is a relation between the head word of the two clauses connected by the conjunct. Which could be one of *and, or, etc.*

cc

Coordination is the relation between the head of the conjuct (the first phrase of the conjunct) and the coordinating word.

parataxis

It means to place side by side. This is a relation between the main verb and other sentential elements, that are separated by colon, semicolon or if they are just placed so without any coordination or subordination. *We might as well accept, we lost.* Where *accept* and *lost* are connected by a parataxis.

advmod

Adverbial phrase that is not a clause and modifies a verb or adjective. *Cryogenically preserved cells*

appos

Appositional modifier of an NP, is an NP to the right of the first NP that defines or modifies that NP. Paranthesized clauses and explainations of abbreviations are include as appos.

poss

The relation between the head of the NP and the complement of the genetive *'s*

num

The number part of a larger number phrase that can include currency or any other unit.
The clock tower is 4 kilometers away.

mark

A **marker** is the word that introduces a finite clause subordinate to another clause. *that, whether* for complement clause, and *while, although* for prepositional.
He reported that the president stepped down today. Here, *that* is related to *reported* through a mark tag.

compound

It links the compound words of a noun to the root noun which would be the last one.
New York City have *New* and *York* related to *City* through the compound tag.

det

The relation between a noun phrase and its determiner like *the, an, a.*

dep

The generic marker between two words if a more specific dependency cannot be found between them, due to construed grammatical construction.

Table 3.1: Events Table

Column	Definition
ClusterID	The ID of the cluster(defined in chapter 4) to which this event belongs.
EventID	This is the unique ID that every event gets
Verb	The surface form of the verb as found in the text
Derived Form	The extracted form of the verb (defined in this section)
Object	The dobj (iobj) of the verb
Tense	The tense of the verb as derived from the part of speech tag of the verb
Subjposs	The root of the possessive relation of the subject
Objposs	The root of the possessive relation of the object
Mark	The dependent of the mark relation of the main verb
Timemark	The preposition corresponding to the time entity if present
Time	The Time entity as read extracted from the text
Parent	If the event belongs to the subordinate clause in a sentence. Then this denotes the eventID of the parent clause
Neg	A boolean denoting whether the main verb's predicate is negated by the presence of a negation modifier
Article	The article ID in which this sentence is from
Sentence Number	The current sentence's ID in the article from which it was taken

3.3 Rules to extract Events

In this section we discuss our hand made rules that use the heuristics based on the Stanford dependency parse to extract events and their attributes. The input to this step is a set of text documents and the output is a table of events. Each row corresponds to one event extracted. The definitions for each column in the table is given in table 3.1.

For every article we obtain a parse of every sentence which gives us the part of speech of each word and the dependencies between the words. We start from the root and add every connected clause to a queue which is visted in order after the main clause is processed. The processing of the main clause and subordinate clauses are similar. The rules are as follows:

1. Begin at the root word. If the word is a verb then save it and continue or else derive the verb form of the words as given below.
2. Extract the tense of the verb: if the main word is a noun with an auxiliary then use the tense of the auxiliary as the tense of the clause, else use the tense of the verb as derived from its parts of speech tag, as given in table 3.2.

Table 3.2: Corresponding tenses of the Verbs Parts of Speech tags

Part of Speech Tag	Tense
VB	Base Form
VBD	past tense
VBG	gerund or present participle
VBN	past participle
VBP	Verb, non-3^{rd} person singular present
VBZ	Verb, 3^{rd} person singular present

3. Traverse the `nsubj` or `nsubpass` link to obtain the subject.

4. Obtain the possessives and complete compound nouns as given below.

5. Traverse the `dobj` link to obtain the object and again complete the possessives and compound nouns.

6. If a *neg* relation exists update negation boolean.

7. Traverse all `nmods, xcomp, nmod:tmod, nummod`. Using the Named Entity Recognizer / SUTime recognizer identify the noun modifiers which are times. Add the recognized time along with the corresponding preposition to the table.

8. Traverse the `advcl, dep, conj, parataxis, ccomp, xcomp` relations and add the correponding main verbs of the derived clauses along with the connector word to the queue. And save the connections in a separate list.

9. Repeat until queue is empty and then move to next sentence in the text.

3.3.1 Resolving Verb Nouns

To resolve nouns that denote events like in, *After the king's death, his son took over.* Death should be resolved to the base verb of *die.* We use WordNet's derivationally related forms. Which are a set of lemmas of different forms of the noun. The derivationally related forms are included for the word *death, birth* and *wedding* in table 3.3

Now that we have extracted the events, the next step is to resolve the coreference of events. Coreference of events happen both before and after the event extraction where we run a preprocessing step that resolves the anaphoric references like *he, she, it etc.*, while the step

Table 3.3: Derivationally Related Forms for *Death, Birth* and *Wedding*

Word	Derivationally Related Forms
death	Lemma('die.v.01.die') Lemma('die.v.02.die') Lemma('die.v.10.die') Lemma('end.v.04.end') Lemma('end.v.02.end') Lemma('end.v.01.end') Lemma('end.v.03.end') Lemma('deadly.s.01.deathly')
birth	Lemma('be_born.v.01.be_born') Lemma('give_birth.v.01.birth') Lemma('rear.v.02.parent')
marriage	Lemma('marry.v.02.wed') Lemma('marry.v.01.marry') Lemma('marry.v.02.marry') Lemma('marriage.n.03.marriage') Lemma('married.n.01.married')

after resolves similarity and synonymous words, we discuss this in the next chapter. But before that we discuss some of the ways event extraction is carried out by others in literature.

3.4 Other extractors

There are many systems that have implemented event extractors. Some use rule based systems while some use machine learned systems. We discuss both of these methods as implemented in literature.

[15] flag and then resolve the temporal expressions based on reserved keyword such as *week, day, weekend, now, Monday, current, future,* etc. and guildlines built for English. They allow for three kinds of *points in time, durations* and *frequencies.* They also acknowledge that times can have *fuzzy boundaries* such as *morning* and *Summer*, and -specific definitions of dates like *the 80's.* They deal with the corpus at the word level. They parts-of-speech tag the words. A lookup table is used to recognize the temporal expressions and convert them to a standard form. A discourse analyser then resolves the context-dependent time expressions using syntactic clues and *Reference Time* which is current focus time.

[6] too use a parts of speech tagger along with heuristic rules to identify events. They

too describe a method of anchoring time points and dealing with vague anchors like last year etc. Before running the extractor they resolve all pronouns and their antecedents. Similar to the method described in this chapter. Most notable is the way they deal with time stamps without explicit date information:

- If the current clause refers to a point in **Present or Past Perfect** then an open-ended time interval is assigned to the event. With unknown start point and end point same as the current date or most recently assigned date.
- If the clause refers to **Future Tense** then an open ended interval is started with the end point unknown and the start point the same as the most recently assigned date or date of the article.
- **Present or Past Indefinites** are given the most recently assigned date. Leaving other tenses to further study.

[5] use Evita[16] and GUTime to indentify the event and time expressions. Evita uses a syntactic parser to extract infomation like tense, aspect, modality, polarity, event classes in addition to the event itself. The syntactic parse extracts parts of speech and phrase chunks. And a lexical lookup recognizes verbs, nouns and adjectives that can denote an event. It also looks for modals and stative predicates, with a pattern-based approach. It uses WordNet and its tree classification structure of words to find subtrees that could be canonical to events like *phenomenon*.

[17] use a novel approach to time expression extraction. They create a grammar over extreme points of the probability distribution for the occurence of the event over time. They don't take the effect of other events on this event, since their aim is to identify the meaning of words compounded together to form a larger specification for an event like *last Friday the 13*th. They also aim to remove disambiguations such as *[last Friday] [the 13*th*]* or *[last] [Friday the 13*th*]*. Though they use a gaussian distribution over the timeline, the distribution can be anything. They train a CYK parser and a version of Expectation Maximization to prune the search the space for possible interpretations.

CHAPTER 4

EVENT COREFERENCING

In this chapter we explore event coreferencing. After the steps outlined in the previous chapter we have to find and cluster all the event mentions that refer to the same real word event. Event coreferencing is similar to pronoun coreferencing, yet it is a harder problem to solve as explained in the chapter 1: To match events, both the main verb and its attributes have to match. Dictionary and rule based methods, supervised and unsupervised methods exist in literature. In our system we have used a rule based method to coreference events. In the following section we discuss the method used in the thesis followed by a section on the state of the art in literature. So that in future studies these other systems can be used in a bid to improve performance.

4.1 The Coreference Algorithm

4.1.1 Verbs Clustering

To resolve the different senses a verb may take we use a voting algorithm where the senses of the other verbs mentioned in proximity vote on the senses of the current verb it would most probably appear close with. The features of the verb that are voted on are the word's hypernym, which are a broader category under which the verb appears. We use Wordnet's hypernym-hyponym trees for our list of verb hypernyms. An example is in table 4.1.

The assumption is a verb like *fight* can have multiple meanings like *struggle* or *duel*. But if the words in the vicinity had large similarity with the *duel* sense of *fight* then their presence will increase the chances that that was indeed the sense of the verb we needed.

The algorithm follows the following steps:

1. Create a proximity graph where each node is a verb that was mentioned (multiple mentions of the same verb map to the same node) and each edge is a link between the verbs if they appear within a window of each other.

2. For every node save that node verbs hypernyms in a datastructure.

3. Once the entire document is parsed. Each node votes on the hypernyms of its neighbours weighted by the probabilities in its own hypernym set. That is, for every hypernym in the current node add a number equal to the hypernym's weight in its own hypernym list to the the hypernyms in the neighbouring node, if the hypernym exists in the neighbouring node.

4. Repeat step 3 until convergence of the probabilities. The convergence criteria is the difference in the probability distributions for all verbs from the previous iteration.

5. Given the senses and their probabilities for each verb. Choose the highest voted sense to cluster similar verbs together.

Given the verb clusters, we lexically compare the subject and object pairwise with every event using a similarity score that is the inverse of the normalized Levenshtein distance as a measure of lexical similarity. If in two sentences, the verbs are in the same cluster, and the subject and object are lexically the same we say that the events mentions are pointing at the same event.

Event coreference is a problem that is still the subject of much research. And there are many methods again ranging from rule based to machine learning in literature that we will discuss in the next section. But to highlight the shortcomings of the above method:

- Apart from the shallow lexical comparisons for nouns there are no deep dictionary based methods or word embeddings that could improve perfomance.

- Though only the effect of verbs within a window are taken into consideration during the verb sense disambiguation, the final disambugated sense is only one for the whole document for a verb.

Currently it does not do window clustering.

4.2 Event Coreferencing in Literature

In this section we discuss the state of the art in pronoun and event coreferencing in rule based and machine learned methods and show that our method is just little better than what is considered base case, yet if one of these methods were used instead then accuracy can be improved drastically.

[18] resolves nominal coreference by employing a series of increaing precision sieves upon enitities first extracted using a high recall sieve. This is has been an architecture that has been employed widely in natural language processing, and to some good results. These sieves are heuristic rules that are intuitively designed. They include the following in order. *Mention detection*, which is the high recall sieve, *speaker sieve* which matches speakers to the pronouns in their quoted speech. itshape String match which lexically checks the string strictly over many constituent words. *Relaxed string matching* which drops other words around the headword and matches the two headwords. Followed by a series of head matches and finally a pronominal coreference resolver.

[19] uses a two step bag of words model for event coreferencing. It creates templates for sentences, which put words into multiple bags such as *Action, Time, Location, Human participation, Non-Human participation* per sentence for a document. Using a supervised model trained on the bag of words feature. First a model is learned to cluster documents together if they have reference to a common entity and then the same is run for event event pairs. This attains an approximate of 70

[20] extracts the subject and the verb as attributes of an event. Named entities that are adjacent to each other are used to extend the extracted feature set to account for nouns that are joined together by conjunctions. Each word is then annotated with WordNet supersenses and BabelNet senses and hypernyms. A kernel is used to match the supersense with the word, which uses a cosine similarity between the sentence and the definition of the supersense. The

Table 4.1: Events Table

Verb	Hyponyms
fight	dispute.n.01 abstraction.n.06 psychological_feature.n.01 activity.n.01 conflict.n.01 attribute.n.02 trait.n.01 try.v.01 aggressiveness.n.01 group_action.n.01 entity.n.01 praise.v.01 military_action.n.01 boxing.n.01 contact_sport.n.01 act.v.01 disagreement.n.03 advertise.v.02 speech_act.n.01 measure.v.04 drive.n.05 contend.v.06 think.v.03 act.n.02 diversion.n.01 event.n.01 evaluate.v.02 sport.n.01 controversy.n.01
struggle	act.n.02 fight.v.03 try.v.01 abstraction.n.06 conflict.n.01 contend.v.06 group_action.n.01 activity.n.01
common(fight, struggle)	fight.n.02 abstraction.n.06 group_action.n.01 contend.v.06 act.n.02 activity.n.01

Chinese Whispers algorithm [21] is used to cluster the events. The algorithm starts by assigning every event to its own class. Then each event is randomly visited and it is assigned to the highest ranking class of its neighbours. This is repeated until convergence.

[22] use heirarchical distance dependent version of the chinese restaurant process to cluster the even mentions together. The distances are based on the lexical similarities of the words based on WordNet and other features. The heirarchical part comes in because it first clusters events within the document and then clusters within set of documents heirarchically.

CHAPTER 5

PROBABILISTIC TEMPORAL CLOSURE

This chapter describes the crux of the algorithm that is the main contribution of this thesis. Where the outputs, from the processes in the previous sections, event recognition and clustering is used

1. to find the temporal closure of the events,
2. to find a possible point or range on the timeline for the occurence of events whose times of occurence were previously unknown,
3. to tighten the range of times over which an event is said to have taken place.

In the next section we describe the method that this thesis proposes. Followed by its merits. In the last section we describe the other methods in literature that perform the probabilistic closures and discuss how they differ from the present method.

5.1 Probabilistic Closure of Event Graphs

5.1.1 Recognizing and Categorizing Event Event Discourse - Preprocessing

Discourse elements denote the event-event and event-time connections in time. For example *Elizabeth became queen after her father's death,* links the events *Elizabeth becoming queen* to the event father's death by the discourse element *after.* We have written a set of preprocessing rules that maps discourse in text to one of the following subset of Allen's rules, *(after, before,* *during, started, ended)* for some of the cases. Apart from intra-sentence discourse as denoted by discourse markers, we also take inter sentence discourse into consideration such as **narration**, consequtive sentences that have a common subject or have an object that is the subject of the next would have a temporal relation between the events which can be discerned from the tenses of the clauses under consideration.

The rules developed in this section follow from some the theory discussed in chapter 2 as a first prototype. A more detailed implementation of the theory discussed above when framing the rules would give better performance than the first prototype.

- If the discourse word is one of the keys in the left side of table 5.1, the corresponding relation it maps to is on the left hand side. Where the relation is such that the second event is the one it belonged to.
- If the first clause is **present** tense and second is**past** tense then the second event is **before** the first. *She is buying a bag. She had always wanted one.*
- Else if the first event is **past** tense and the second a **modal** then the relation is that of during. *They asked for an encore, so you should give them one.*
- Else if both the events are of **past** tense and the relation is a **gerund** then the relation is .
- Else if the first event is **past** and the second event is then the second event is after the first. *She picked up a bag then billed it.*
- Else if both events have the same tense and they are linked through inter sentential narration then the relation is such that the second event is after the first. *She picked up a bag. She walked over to the counter and billed it.*
- Else if the the event was an adjective then by default the relation is that of **during**.
- If nothing works, the default case is that discourse word does not get mapped and it is left as is.

Table 5.1: Events Table

Discourse Item	Relation
while	during
that	during
after	after
since	after
before	before
during	during
for	during
to	during
and	during
on	during
from	after
by	before
until	before

5.2 Creating the Graph

5.2.1 Initialization

The temporal graph consists of two types of nodes: events and relations. An event node is connected to another event node only through a relation node. Each node has a distribution over time that is a skewed-normal distribution. The distribution over time for an event node is the distribution over **time of occurence** for the event. While the distribution for the relation node is a distribution over *delta time* between the two events, i.e. how far apart in time did the two events occur. A skewed normal distribution takes the form

$$p(x) = 2 \times pdf(x) \times cdf(\alpha x) \tag{5.1}$$

which is a normal distribution when $\alpha = 0$ and a half normal distribution when $|\alpha| = \infty$ and *pdf* and *cdf* are the probability density function and the cummulative density function of the normal distribution with mean μ and standard deviation σ respectively. We shall see the significance in the choice of distribution next.

The three main relation shapes are skewed to the left, skewed to the right and not skewed which can be seen in figures 5.1, 5.2 and 5.3 with examples being *before, after, during* respectively. The full relations mapping is in table 5.2.

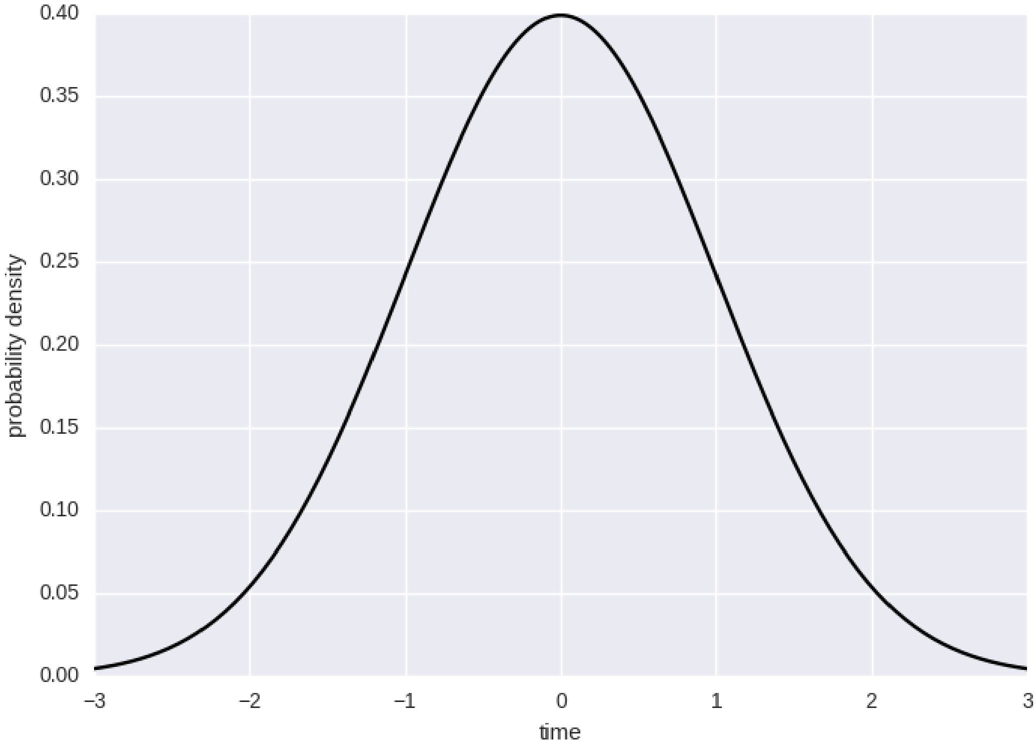

Figure 5.1: PDF of the relation node for *during*.

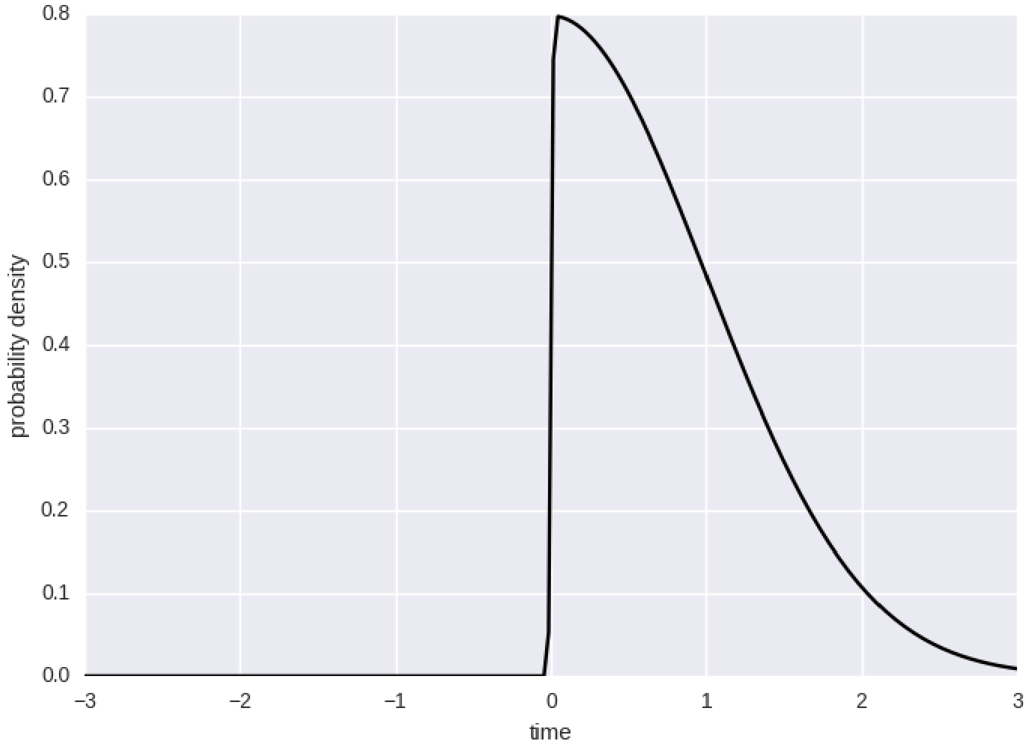

Figure 5.2: PDF of the relation node for *after*.

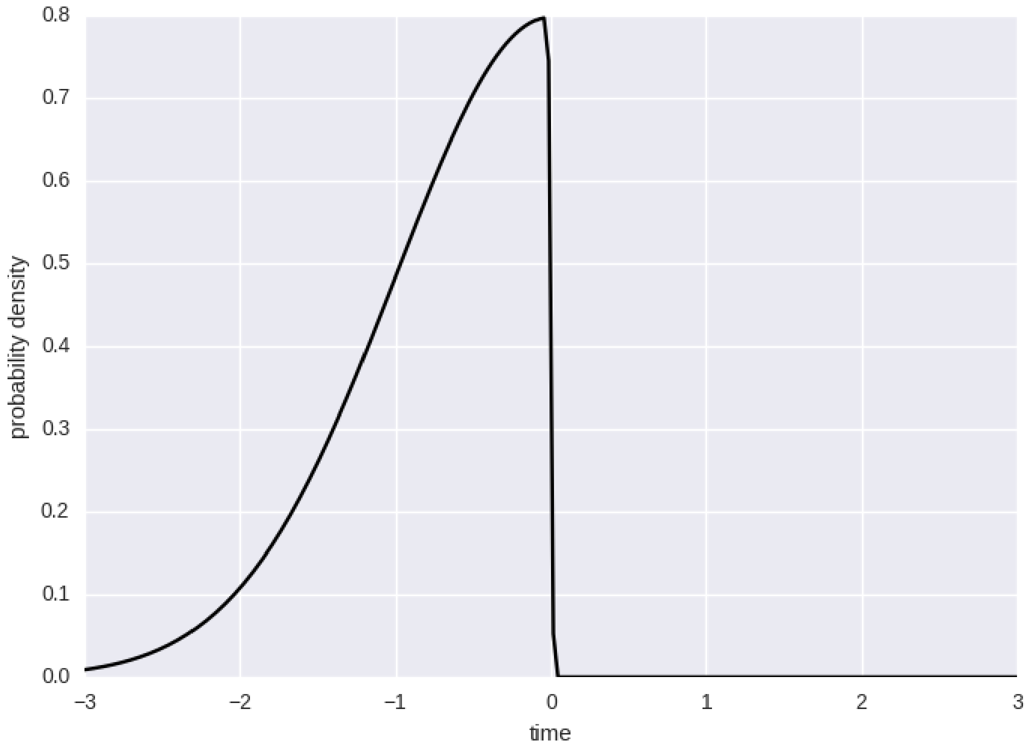

Figure 5.3: PDF of the relation node for *before*.

Table 5.2: Shapes of the Relations

Relation	Shape
before	skewed to left
after	skewed to right
during	not skewed
starts	skewed to right with center at start point
ends	skewed to left with center at end point
simultaneous	not skewed

Table 5.3: μ and σ depending on the specificity of time mentioned

Most specific time	μ	σ
dd-mm-yyyy	dd-mm-yyyy	1/6
mm-yyyy	15-mm-yyyy	30/6
yyyy	1-7-yyyy	365/6

The creation and subsequent closure of the graph takes place in two steps. A first run through the events list created in the previous two steps creates nodes for the events while simultaneously creating the relation nodes, initializing the distributions in each using event-time and event-event relation information respectively, and linking the event nodes through the relation node.

The nodes are initialized by assigning values to the parameters of the skew normal distribution, μ, σ and α. The granularity of the time axis is *one day* and months are represented as 30 days and years as 365 days. Given a time for an event the μ is found given the most specific time. That is it is specific to a day if day information is given, else to the 15^{th} of the month if day is not specified yet month is, or to the 1^{st} July if the month and day are not given yet the year is. σ is again dependent on the specificity of the time. If the specificity is a day then sigma is 0.66, else if specificity is a month then it is 30/6, and if it is a year then it is 365/6, such that the 3 σ's to the left and right of μ fall inside the full range of a day, month and year respectively. The same can be extended to other units of time such as week, fornight etc. This is tabularized in table 5.3. Each node also has attributes *left-limit* and *right-limit*, which designates the significant portion of the node. In the beginning the left-limit and right-limit are 3σ's to the left and to the right of μ.

The initial distributions for the relation nodes have μ's that depend on the relation specified in the event-event context. If a linear shift in date such as *5 days later* is specified then the μ would be 5. The σ is chosen according to table 5.3 based on the specificity of the event that anchors the relation. The α is chosen according to table 5.2

5.2.2 Closure

After the graph is formed, each node that has an anchoring in time is iterated over and its influence is propagated through the relation onto its neighbouring node in the graph. The

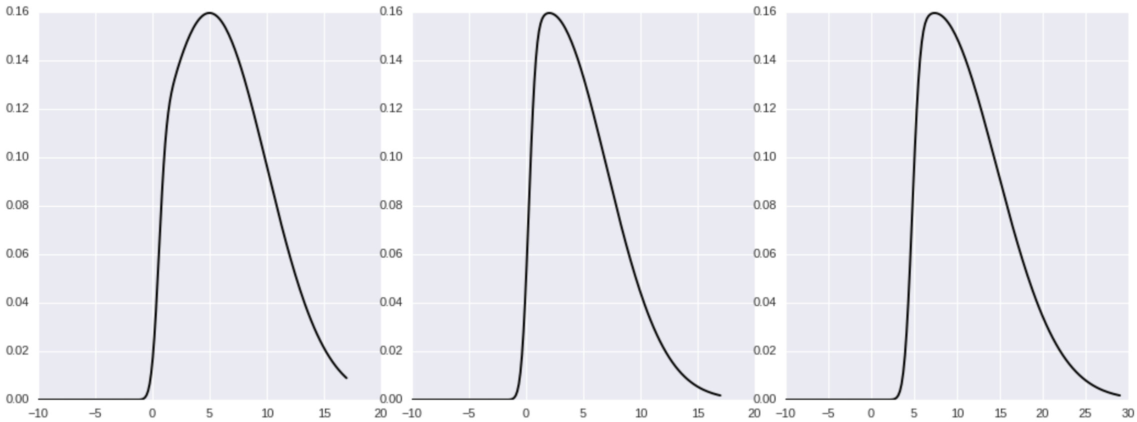

Figure 5.4: a) Is the distribution over time for the first event, b) is the distribution over time for the difference in time of occurence between event 1 and event 2, c) is the distribution over time for the occurence of event 2.

method followed for the influence promulgation is an emperical one where the mode of the time of occurence for the current node is taken as a bias for the relation node. That is, anchoring zero of the relation node's distribution at the mode, the values of the distribution are sampled at regular intervals and added to the values of the corresponding time points in the neighbouring node. The distribution is then fit to a skewed normal using a curve fitting tool. After all the nodes with anchoring in times have been visited every node including the new nodes anchored in time due to the influence of nodes in the previous iteration are visited again and its affect is propogated onto its neighbours. This process repeats until there are no new nodes that are being anchored in time, and there is no significant change in the distributions of the nodes.

In figure 5.4 the distribution of event 1 in subplot a, the relation between event 1 and event 2 in sublot b, and event 2 in subplot c. Assume that event 2 is currently not anchored in time. The mode of event 1 is at 5. Thus the distribution of event 2 is the distribution of the relation biased by the mode of event 1. If event 2 were already anchored in time then biased relation will be added to the current distribution and it will be fit to find the parameters.

5.3 Event-Event Relation Extraction and Temporal Closures in Literature

[23] uses a Naive Bayes and Maximum Entropy classifier to find the following attributes of the events: **tense, aspect, class, modality, polarity**. The features they use are the parts of

speech tags, lemmas of the event words , WordNet synsets and the appearance of auxiliaries (*is, are*) and modals(*should, would*) and the presence of a *not*. They find the appropriate relation from a reduced version of Allen's set for each pair of events in the corpus by training and running on an SVM classifier, Naive Bayes and Maximum Entropy classifier. The features they use are the event specific features that we just described, the Parts of Speech tags of the event, two tokens preceding, and one token following, Event-Event Syntactic properties, that is if one event dominates the other, Prepositional Phrases, and Disourse structure. [24] too trains an SVM to recognize the event-event relations. The features they use are text of the main word itself, part of speech tag of the word, the morphological stem of the word, auxiliaries present, modal words present, whether it was a state or reporting, the tense of the event, aspect (progressive or perfective), polarity of the event. The attributes of the connective word between events are also used as features: the text of the word, whether it was of type after or before, the path on the syntactic parts of speech parse, between the word and complement, the bag of words of between the two events and the function words between the events. It was trained and tested against the TimeBank corpus gaining 87

[5, 25] deal with time intervals through their end points.

[5] uses GUTime and Evita to identify events and extract their attributes. It then uses syntactic dependency parse to find the dependencies between the words in a sentence. If any word in the textual representation of the event in the text has a dependency with any word in the textual representation of the other event then a dependency is created between the events. The dependencies are labeled with relations from Allens set like *after, before*. Thus a large network of events that have dependencies between them are formed, each dependency may have multiple labels. These labels are then pruned using *Markov Logic Networks* which is a probabilistic graph structure resembling a Markov Network atomic logic expressions forming the nodes. The conditional probabilities between the atomic expression, i.e.then the implication, if atomic expression A is true then atomic expression B is true, forms the transition between the nodes, and the uncertainity of this implication forms the transition probability between the two atomic statements.

TempEval 2007 [26] was a shared task in which the participants had to identify the pairwise temporal relations between events in a news corpus. There were three tasks **A)** Given a corpus annotated with identified events, the links between event and time -TLINKS - had to be learned. The training set had TLINKS, making supervised learning easier. **B)** Finding the relations between events and the document creation time. **C)** Find the temporal relation between the main event of adjacent sentences.

CU-TMP [27] outperformed the other participants at the 2007 TempEval. It trains an SVM with the following features of the events in text: parts of speech tags of the words, the preposition governing the event, he governing or main verb, the parts of speech tag of the governing verb (gerund or tense), the presence of auxiliary adverbs modifying the verbs. The next few take information from the gold standard annotations of the TimeBank text: modal, stem of the verb in the gold standard, parts of speech, class, tense, aspect, polarity, whether it were date or time, the temporal location of the event.of the gold standards. For task A they used these extra features: the count of time expressions between the event and the time, the path on the syntactic tree between the event and the time, the path in the the previous feature not broken into three parts, with one part from first event to least common ancestor (LCA), the LCA itself and the path from the LCA to the second event. The number of clauses between the two events. For task C they used rules about tenses like past events happened before present events and present events happen before future events are scheduled to take place. For task A and C the output of task B that is the connection resolution of event times given the document creation time.

[7, 8] use integer linear programming to find the contraints largest set of orderings predicates between events are true. The objective function in [8] is

$$max_{ij} p_{ij} x_{ij}$$

and constraints:

$$\forall i \forall j x_{ij} 0, 1 \tag{5.2}$$

$$\forall i x_{i1} + x_{i2} + ... + x_{im} = 1 \tag{5.3}$$

Where x_{ij} is an indicator variable for the jth relation of m relations. Given n pairs of events there will be $n \times m$ variables, and p$_{ij}$ is the probability that pair i is classified with relation j. The second equation makes sure that the indicator variable can only be a 1 or 0 and the third equation makes sure that only one relation is true for an event at a given time. And equation

$$x_{ia} + x_{jb} - x_{kc} \leq 1 \tag{5.4}$$

which ensures transitivity as a,b and c denote event pairs. If a has a relation and b has a relation then c has a relation.

[28] too uses Markov logic Networks to find the best set of event-event relations. One of their features isa derivation of the relationships between verbs queried from the VERBOCEAN [29] database. Which has *similarity, strength, antonymy, enablement, happens-before* relations. [30, 31] uses conditional random fields as the structure that they train in event relation extraction.

CHAPTER 6

EXPERIMENT AND RESULTS

The rules for event extraction were built heuristically with an open set of new articles based on *Apple Inc.* This is set was the training set of TempEval 15. The evaluation was carried out on another set of 30 articles from the test set of TempEval 15. These articles were about *Boeing* and *Airbus.* and are from the 2000-2010 decade.

The run of the program itself is breakable into individual parts that are otherwise serially runnable. The parts of the program and its output after the various sections in the run are enumerated in the following.

1. As a preprocessing step, Stanford Natural Language Toolkit's coreference resolver is used and identified pronouns are replaced with the non-pronounial reference to the entity. Noun phrases that are partials of a larger descriptor for that entity are also resolved and replaced with the largest descriptor for that entity.

2. Every verb (and verbal noun), whether it belongs to an actual event or belongs to a phrase that is a statement of a fact is identified. For every verb identified above its attributes as extracted by the syntactic dependency parser is identified. These include *subject, object, time-phrase,* and *preposition.* If the nouns in these phrases are compound nouns, then they are resolved. Complex sentences with multiple clauses are split into the individual parts anchored at their respective root verbs. These pairs of words are recorded as a relation along with the word that connected the two. The verbs and attributes are printed out. Each identified verb has a unique ID number assigned to it.

3. Using the procedure layed out in section 4.1, verbs with similar word senses are grouped

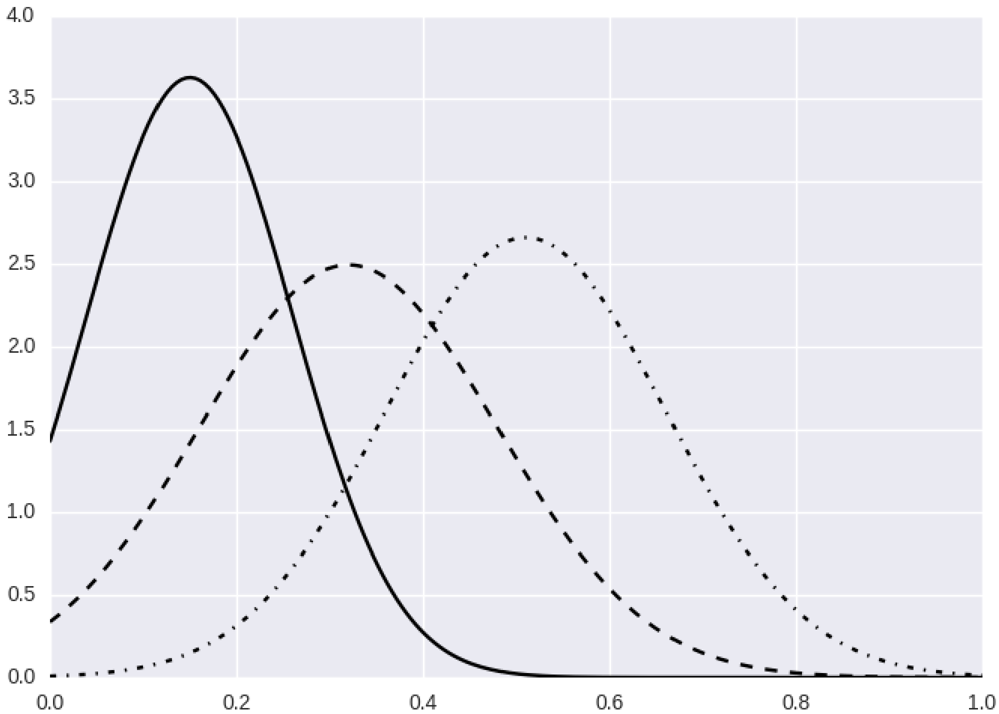

Figure 6.1: The mean and spread of the per article normalized counts for the 3 categories of accuracy. The solid line represents Category 1, the dash line is Category 2, and the dot-dash line is Category 3.

together and the events are clustered and the clusters are assigned cluster identifiers. Each verb mention is now assigned the cluster ID of the cluster to which belongs. Each verb retains its orginal ID number too.

4. Every relation that was recognized in 2 is bucketed into one of *after, before, during, started, ended*. These also include the temporal relations between events due to narration. These relations are printed in an relations file.

5. . For every event cluster identified in step 3 an event node is created in the graph and for every relation identified in step 2 a relation node is created linking the clusters to which the events of the identified relation are a part of. And for every cluster for which an anchoring to the timeline is known its effect is propogated to the neighbouring clusters. After convergence, the timeline graphs of the clusters are drawn.

The following were the evaluations carried out on results that were output at every level of the program run:

Table 6.1: Precision of the event extractor per article. Category 1 are those events that were adjudged as having wrong attributes extracted. The mean of the normalized count is 0.15 with spread 0.11. Category 2 are those events that do not have wrong attributes extracted but have incomplete information extracted in one of more attributes. The mean of normalized counts is 0.32 with spread 0.16. Category 3 are those events which have all attributes extracted correctly. The mean of normalized counts is 0.51 with spread 0.15.

	1		2		3	
Article ID	Count	Normalized Count	Count	Normalized Count	Count	Normalized Count
1173	19	0.22	26	0.3	39	0.46
3307	2	0.11	5	0.27	11	0.61
3385	4	0.3	5	0.38	4	0.3
4764	8	0.32	6	0.24	11	0.44
5403	3	0.16	4	0.22	11	0.61
8951	0	0.0	12	0.6	8	0.4
8983	5	0.17	20	0.68	4	0.13
11714	2	0.14	3	0.21	9	0.64
11716	0	0.0	7	0.87	1	0.12
25501	5	0.25	3	0.15	12	0.6
41882	0	0.0	6	0.33	12	0.66
51682	3	0.2	2	0.13	10	0.66
61389	1	0.04	4	0.17	18	0.78
62929	0	0.0	10	0.37	17	0.62
71426	5	0.12	14	0.35	21	0.52
71526	3	0.11	10	0.38	13	0.5
73808	1	0.03	10	0.35	17	0.6
78496	4	0.23	5	0.29	8	0.47
82548	2	0.2	2	0.2	6	0.6
82597	2	0.2	1	0.1	7	0.7
87805	12	0.25	9	0.18	27	0.56
100911	14	0.19	19	0.26	39	0.54
102977	3	0.18	7	0.43	6	0.37
106653	6	0.14	11	0.26	24	0.58
108294	7	0.41	4	0.23	6	0.35
128016	0	0.0	7	0.58	5	0.41
129865	2	0.04	15	0.33	28	0.62
145930	3	0.21	4	0.28	7	0.5
149731	7	0.38	4	0.22	7	0.38
276079	2	0.05	7	0.2	25	0.73

Table 6.2: Recall of the event extractor per article. The overall mean is 0.61 and standard deviation 0.1

Article ID	System Recognized Verbs	Evaluator Recognized Verbs	Precision
1173	64	97	0.65
3307	18	23	0.78
3835	10	19	0.52
4764	20	29	0.68
5403	18	26	0.69
8951	18	33	0.54
8983	20	27	0.74
11714	13	24	0.54
11716	7	17	0.41
25501	19	35	0.54
41882	18	31	0.58
51682	14	26	0.53
61389	19	27	0.7
62929	23	39	0.58
71426	35	53	0.66
71526	23	43	0.53
73808	27	44	0.61
78496	17	40	0.42
82548	9	18	0.5
82597	10	14	0.71
87805	46	80	0.57
10091	72	138	0.52
102977	15	20	0.75
106653	31	42	0.73
108294	15	27	0.55
128016	12	18	0.66
129865	40	55	0.72
145930	11	16	0.68
149731	17	30	0.56
276079	28	41	0.68

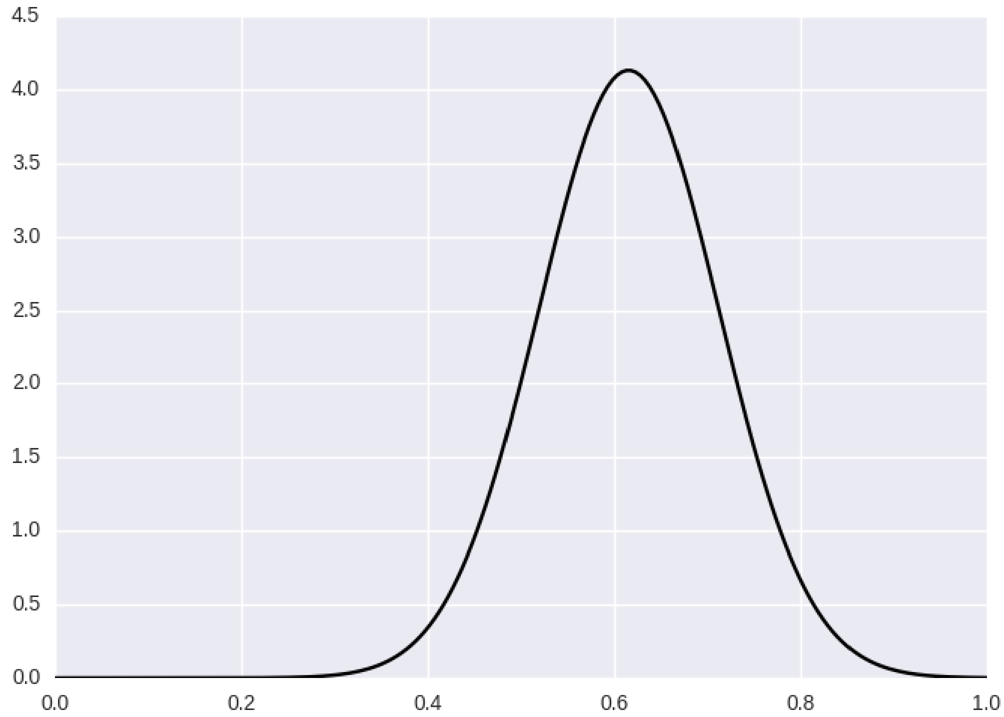

Figure 6.2: The mean and spread of the per article recall on the Boeing and Airbus test set.

1. To evaluate event recognition, a human evaluator adjudged ratings on each event that was output at step 2, while having access to the original text. The events were given one of the three ratings: *1* if there was a mismatch between the actual part of speech of the word recognized and the part of a speech to which it was assigned. That is the verb recognized were not a verb or verbal noun or if the nouns such as subject and object were in fact verbs or adjectives. This rating is also assigned if the attributes assigned to the main verb do not in fact belong to the main word. *2* is assigned, if the actual part of speech and the part of speech assigned are consistent but the all the information was not captured, such as the compound part of the nouns or if passive voice were recognized as active or vice versa. *3*, if all the information has been captured correctly, including parts of speech and the correct corresponding attributes. The results for this are shown in table 6.2 and figure 6.1. This represents the *precision* in event recognition. To measure *recall* the verbs and noun-verbs were marked in the corpus by a human evaluator, the recall is measured as the ratio of the number of these events that are recognized by the program to the total number of events marked by the human evaluator. The results for this are shown in table 6.1 and figure 6.2.

2. To evaluate the relations extracted from the text, a human evaluator marked (gold standard) relations between events as one of *before, after, starts, ends, during.* Due to the size of the corpus, only verbs that refer to actual events that took place are marked, and verbs that are parts of factual clauses are excluded. Of those event pairs that are marked in the gold standard, the number of event pairs marked as having a relationship between them is counted. The ratio of the correctly labeled relations to the total relations in the gold standard is taken as precision. A precision of 0.66 was achieved in this step.

3. Finally, after the convergence of step 5 the anchoring of each event on the timeline is reported as graph. These are marked by a human evaluator, on the accuracy of the location and the spread of the distribution. That is the **location** of the distribution is correct if the event is bounded on both sides and the actual location of the event is within the distribution reported by the system, or else, if the event is bounded only on one side then the actual event should lie on the unbounded side. The spread is evaluated as follows. If the actual spread of the event as marked by the human evaluator is more specific than actually mapped then the relation is marked as incomplete. After evaluation it was seen that propagation of the inefficiencies of the previous steps gave the final outcome completely incorrect answers. Therefore, until the performance of the previous steps are improved the final outcome cannot be fully evaluated. Some possible steps for improvement are mentioned in section 8. However, in order to make a case for the strength of this method a human evaluator marked actual events in 20 of the 30 articles in the test set, which were all related to *Airbus A380s* or *Boeing 747s.* Of these events marked the evaluator partioned the set into those events which had a relation to time and therefore had a direct(explicit) anchoring to a point on the timeline, and those events for which at first an anchoring on the timeline wasn't known but was known at the end of the process. Readers are refered to Table 6.3 for the counts.

Table 6.3: Number of events per article for which the anchoring to the time line is explicitly mentioned and those that is implicitly derived from its relation with other events. These are marked by the evaluator.

Article ID	Explicit	Implicit	Total
3307	3	9	12
3835	4	1	5
4764	6	1	7
5403	3	3	6
8951	3	13	26
8983	3	11	14
11714	3	11	14
11716	4	7	11
25501	5	15	20
41882	6	12	18
51682	2	3	5
61389	2	5	7
62929	10	3	13
73808	7	2	9
78496	1	10	11
82548	4	1	3
82597	3	1	4
128016	4	3	7
145930	2	3	5
276079	3	8	11

CHAPTER 7

FURTHER READING

The four TempEval shared tasks in the last decade helped formalize the problem of temporal information extraction.

The first TempEval that happened in 2007. EmpEval-2 part of the SemEval 2010 shared tasks extended the previous iteration. In this edition more languages were included and the tasks were both further subdivided and extended. The subtasks were

1. Determine the extent of the time expressions as defined by TimeML Timex3 tag.
2. Determine the extend of the events in a text as defined by the TimeML EVENT tag and determine the *Class, Tense, Aspect, Polarity*, and *Modality*
3. Determine the temporal relation betwen event and time expression in the same sentence.
4. Deterimine the temporal relation between and event and document creation time.
5. Determine the temporal relation between two main events in consequtive sentences
6. Determine the temporal relation between two events where one event syntactically dominates the other

8 teams took part that year. In 2013 the TempEval-3 [32] task was more or less a repeat of the previous iteration, however, the corpus upon which the tasks were carried out was enlarged. And unlike in the previous iteration, participants were asked to make an end to end temporal relations processor. That is, raw text to annotated relations. There were 9 participants that year. Many of whose systems are discussed in the previous section. For the

Timex extraction there were data driven machine learning approaches, rule based approaches and approaches that were a hybrid of the previous two. With the rule based approaches out performing the data driven approaches. Event extraction had mostly data driven with one entrant working on a rule based approach. With relation classification being approach entirely with only data driven approaches.

SemEval 2015 Task 4 [33], extends the previous TempEvals. This was a pilot task that built on the output of the previous TempEvals. Given a set of documents and target entities a timeline related to each entity had to be built, i.e., detecting, anchoring the event in time, and ordering the events in which the target entity was involved. Entity coreferencing was therefore part of the task. Only four teams took part, for a comparison of their results with ours and a discussion of the task dataset refer chapter 6.

CHAPTER 8

MAIN CONTRIBUTIONS AND FUTURE WORK

This thesis layed out the broad steps that are requisite for end to end processing of temporal information in text. It followed the theoritical aspects of temporal information in discourse and speech and used that to build a first model which follows these steps. Though the final outcome of the current prototype is not complete nor favourable, by analysing the results of even the preliminary methods used at each step of the procedure promise for better results given better models and methods at each step.

The current procedure currently identifies only events that are anchored by a main word that is a verb, not nouns-verbs even though a provision was included for it. The verbs recognized by the dependency parser that anchor the clause also include copula verbs like *be, is, are, etc.* These are normally used in sentences that are stating a fact or state, these facts can be about events or true only during a certain period, in such cases it is intuitive that the fact can be anchored to the timeline in such a way that is true during the period it is anchored to. However, some facts are true always, in the sense, it is a constant fact, such cases should be correctly recognized and ignored. In another case, facts mentioned in new articles normally have an implicit period during which it is implicitly understood to be true to the reader due to her past knowledge about the world, this time range, however may not be that clear in a stand alone article, at least the shallow way in which it has been approached here, a much deeper semantic understanding would be required.

Another case of events not handled fully here nor in literature are the cases of events that are modal in nature, projecting events that are yet to happen, either anticipated, planned or not, and those events which were mentioned to have happened in quoted text. For example, in our test case, the news articles were reporting about the anticipated release date of the Boeing 787 and projected it to a certain date, however, later articles refered to the same release date of the aircraft, however, the date was now released. This mismatch of the dates will be represented as an inaccuracy of the information extraction process not accounting for the case that the event it self was not a surety in actuallity. The same conendrum stands for quoted speech since events reported as happened by one may not match with those reported by another even though they may be refering to the same event. A second level of uncertainity will have to accounted for for these events that is not currently present.

In section 2.5 the comparison between statives and procedurals were discussed and in section 2.4 the coercion of one form to the other was discussed. However, the latter only treated the coercion in the syntantic form of how to convert the verb to a procedural if it is a stative and vice-versa. A similar concept even if at a deeper level of understanding of the event itself is resolving cases like *The event has started* and *The event is going on now.* In terms of our method the effect of one statement being true will have to be propogated on the truthiness of the other event, which is currently possible if there is a relation node between the two. Such a relation cannot be created by a sheer lexical process and needs deeper semantic understanding of these words. Results with wordvecs and sentence vecs promise some progress in this kind of deeper understanding of words in text given a well trained model.

Other methods in literature for the indiviual steps in the process were discussed in the respective chapters. However, as a matter of future treatment for these, the use of reinforcement learning, and deep learning can be used to improve the event coreferencing. In our system we hand wrote heuristics for the event and event relation extraction, however these modern machine learning tools can learn these rules from data. This work however is not fruitless since, as a first look into any new problem, efficient machine learned models are possible only with a detailed understanding of the first principles of the problem itself. Intricate problems like those covered in the beginning of this section stand as testimony to this.

Bibliography

[1] Z. Vendler, "Verbs and times. section 51 in linguistics in philosophy, pp97-121," 1967.

[2] H. Reichenbach, "Verbs and times. chapter 4 in element of symbolic logic, pp287-98," 1947.

[3] J. Pustejovsky, "The syntax of event structure," *Cognition*, vol. 41, no. 1, pp. 47–81, 1991.

[4] D. R. Dowty, "The effects of aspectual class on the temporal structure of discourse: semantics or pragmatics?," *Linguistics and philosophy*, vol. 9, no. 1, pp. 37–61, 1986.

[5] X. Ling and D. S. Weld, "Temporal information extraction," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pp. 1385–1390, AAAI Press, 2010.

[6] E. Filatova and E. Hovy, "Assigning time-stamps to event-clauses," in *Proceedings of the Workshop on Temporal and Spatial Information Processing - Volume 13*, TASIP '01, (Stroudsburg, PA, USA), pp. 13:1–13:8, Association for Computational Linguistics, 2001.

[7] P. Bramsen, P. Deshpande, Y. K. Lee, and R. Barzilay, "Inducing temporal graphs," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, (Stroudsburg, PA, USA), pp. 189–198, Association for Computational Linguistics, 2006.

[8] N. Chambers and D. Jurafsky, "Jointly combining implicit constraints improves temporal ordering," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, (Stroudsburg, PA, USA), pp. 698–706, Association for Computational Linguistics, 2008.

[9] K. Yoshikawa, S. Riedel, M. Asahara, and Y. Matsumoto, "Jointly identifying temporal relations with markov logic," in *Proceedings of the Joint Conference of the 47th Annual*

Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1, ACL '09, (Stroudsburg, PA, USA), pp. 405–413, Association for Computational Linguistics, 2009.

[10] M. Moens and M. Steedman, "Temporal ontology and temporal reference," *Comput. Linguist.*, vol. 14, pp. 15–28, June 1988.

[11] J. F. Allen, "Towards a general theory of action and time," *Artificial intelligence*, vol. 23, no. 2, pp. 123–154, 1984.

[12] J. Pustejovsky, J. M. Castaño, R. Ingria, R. Saurí, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev, "Timeml: Robust specification of event and temporal expressions in text," in *New Directions in Question Answering*, 2003.

[13] M.-C. De Marneffe and C. D. Manning, "The stanford typed dependencies representation," in *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, pp. 1–8, Association for Computational Linguistics, 2008.

[14] M.-C. De Marneffe and C. D. Manning, "Stanford typed dependencies manual," tech. rep., 2008.

[15] G. Wilson, I. Mani, B. Sundheim, and L. Ferro, "A multilingual approach to annotating and extracting temporal information," in *Proceedings of the workshop on Temporal and spatial information processing-Volume 13*, p. 12, Association for Computational Linguistics, 2001.

[16] R. Saurí, R. Knippen, M. Verhagen, and J. Pustejovsky, "Evita: A robust event recognizer for qa systems," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, (Stroudsburg, PA, USA), pp. 700–707, Association for Computational Linguistics, 2005.

[17] G. Angeli, C. D. Manning, and D. Jurafsky, "Parsing time: Learning to interpret time expressions," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 446–455, Association for Computational Linguistics, 2012.

[18] H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, and D. Jurafsky, "Deterministic coreference resolution based on entity-centric, precision-ranked rules," *Computational Linguistics*, vol. 39, no. 4, pp. 885–916, 2013.

[19] A. Cybulska and P. Vossen, “bag of events approach to event coreference resolution. supervised classification of event templates,” *IJCLA*, p. 11, 2015.

[20] D. Rusu, J. Hodson, and A. Kimball, “Unsupervised techniques for extracting and clustering complex events in news,” *ACL 2014*, p. 26, 2014.

[21] C. Biemann, “Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems,” in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, (Stroudsburg, PA, USA), pp. 73–80, Association for Computational Linguistics, 2006.

[22] B. Yang, C. Cardie, and P. Frazier, “A hierarchical distance-dependent bayesian model for event coreference resolution,” *arXiv preprint arXiv:1504.05929*, 2015.

[23] N. Chambers, S. Wang, and D. Jurafsky, “Classifying temporal relations between events,” in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pp. 173–176, Association for Computational Linguistics, 2007.

[24] S. Bethard, J. H. Martin, and S. Klingenstein, “Timelines from text: Identification of syntactic temporal relations,” in *Semantic Computing, 2007. ICSC 2007. International Conference on*, pp. 11–18, IEEE, 2007.

[25] M. T. Vicente-Díez, J. M. Schneider, and P. Martínez, “Uc3m system: Determining the extent, type and value of time expressions in tempeval-2,” in *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval ’10, (Stroudsburg, PA, USA), pp. 329–332, Association for Computational Linguistics, 2010.

[26] M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky, “Semeval-2007 task 15: Tempeval temporal relation identification,” in *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 75–80, Association for Computational Linguistics, 2007.

[27] S. Bethard and J. H. Martin, “Cu-tmp: Temporal relation classification using syntactic and semantic features,” in *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval ’07, (Stroudsburg, PA, USA), pp. 129–132, Association for Computational Linguistics, 2007.

[28] E. Y. Ha, A. Baikadi, C. Licata, and J. C. Lester, "Ncsu: Modeling temporal relations with markov logic and lexical ontology," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 341–344, Association for Computational Linguistics, 2010.

[29] T. Chklovski and P. Pantel, "VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations," in *Proceedings of EMNLP 2004* (D. Lin and D. Wu, eds.), (Barcelona, Spain), pp. 33–40, Association for Computational Linguistics, 2004.

[30] A. K. Kolya, A. Ekbal, and S. Bandyopadhyay, "Ju_cse_temp: A first step towards evaluating events, time expressions and temporal relations," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, (Stroudsburg, PA, USA), pp. 345–350, Association for Computational Linguistics, 2010.

[31] M. Filannino, G. Brown, and G. Nenadic, "Mantime: Temporal expression identification and normalization in the tempeval-3 challenge," *arXiv preprint arXiv:1304.7942*, 2013.

[32] N. UzZaman, H. Llorens, L. Derczynski, J. Allen, M. Verhagen, and J. Pustejovsky, "Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 1–9, Association for Computational Linguistics, 2013.

[33] A.-L. Minard, M. Speranza, E. Agirre, I. Aldabe, M. van Erp, B. Magnini, G. Rigau, R. Urizar, and F. B. Kessler, "Semeval-2015 task 4: Timeline: Cross-document event ordering," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 778–786, 2015.