INTERPRETING TIME IN TEXT
SUMMARIZING TEXT WITH TIME

JUN-PING NG

(Master of Science, NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2013

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

JUN-PING NG

30 December 2013

ACKNOWLEDGEMENTS

The guidance, care, and concern that I have received throughout my doctoral candidature has been nothing short of humbling. First and most important of all, I like to thank my advisor A/P Min-Yen Kan (or just *Min* as he prefers to be addressed) for his teachings and help through this entire period of time. Without Min, I wouldn't have achieved half of what I had. For this I am forever grateful.

I would also like to thank members of my thesis committee, including Prof Tat-Seng Chua, Prof Chew-Lim Tan, and Prof Inderjeet Mani, who have put in the time and effort to review and assess this thesis.

Next I am also thankful to my wife, Jace, and my family. The unwavering support they have given me, and the patience they have shown (the rigours of a doctoral candidature can sometimes manifest in the form of un-pleasant anti-social behavior), made all of this possible.

There are many others who have touched me in one way or another. Here I would like to extend my gratitude and thanks to my friends and colleagues including Yifan, Ziheng, Zhao Jin, Jesse, Aobo, Chen Tao, Xiangnan, Jovian, Wu Dan, Huixian, Chen Yan and Zhongyi.

CONTENTS

ABSTRACT

In this thesis, I study two key steps in building a logical representation of temporal information — a timeline — found within text from newswire articles: 1) intra-sentence event-timex (*E-T*) temporal relationship classification, and 2) article-wide event-event (*E-E*) temporal relationship classification. Events and time expressions (timexes) are basic units of temporal information in text. These two steps allow us to build an understanding of the relative ordering between these basic temporal units. For both of these classification tasks, I propose more semantically motivated features, namely the use of typed dependency parses and discourse analyses, to achieve better classification performance. This is in contrast to much work in the existing literature, which have focused on lexico-syntactic features.

Working on *E-T* temporal relationship classification, I also show that crowdsourcing is a very cost-effective and viable avenue through which a high-quality temporal corpus can be built. Making use of the structure of a sentence, I propose a unique way to identify instances which are computationally and cognitively easier. Excluding these instances from a corpus does not degrade subsequent classifier performance significantly. This allows cost savings of up to 37% when building a *E-T* temporal corpus.

Besides putting together a state-of-the-art temporal processing system, this thesis also validates the efficacy and utility of the timelines that are automatically derived. Temporal information from these timelines is incorporated into a competitive baseline multi-document summarization system. I propose several features derived from timelines and show that they lead to a 4.1% improvement in summarization performance. I also introduce a modification to the traditional Maximal Marginal Relevance (MMR) algorithm, TimeMMR. TimeMMR is shown to be useful in the summarization of some document sets. To further improve the performance gains derived from the use of temporal information, I propose a reliability filtering metric which gauges how accurate and useful a timeline is. By selectively making use of timelines guided by this reliability filtering metric, overall summarization performance is increased by a statistically significant 5.9%.

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

This chapter bootstraps the thesis by explaining the problem I am attempting to solve. I also summarize the key contributions I have achieved here.

Over the past decades, Natural Language Processing (NLP) has matured considerably. State-of-the-art systems for syntactic processing, including part-of-speech tagging and grammar parsing are readily available. Much progress has also been made for higher level semantic tasks such as discourse analysis, where the performance of automatic parsers have been improving steadily (Feng and Hirst, 2012; Hernault et al., 2010). As a result, many researchers have been empowered to explore more difficult and challenging tasks such as Machine Reading (Etzioni et al., 2006), which builds upon these advancements. Despite this progress however, there is nonetheless still much work to be done. While lexical tasks have been very well-studied, the state-of-the-art for many higher level semantic processing tasks still misses the mark.

In this thesis, I set out to examine the interpretation of time from text documents. "*Time*" is a well-known concept to many; yet it has also been the subject of a raging debate between philosophers and scientists. One line of thought, attributed to Isaac Newton and hence referred to as *Newtonian time*, sees *time* as a dimension along which events occur in sequence (Newton, 1687; Rynasiewicz, 2012). An opposing view mooted by Gottfried Leibniz (Clarke, 1717) and Immanuel Kant (Kant, 1786) holds that time does not exist in any

form, but is just simply a projection of how we logically represent things around us. In the context of this work, I will adopt Newton's view of time.

While time is sequential, the manner it is presented is not. Example 1.1 shows a paragraph of text extracted from a news report[1] about the launch of a new tablet. In the example, sentence (1) talks about the launch on "*Friday*". Sentence (2) talks about the lines on Friday being shorter than previous years, and also a projection of upcoming sales in the weekend ahead. Sentence (3) goes on to mention the differences in circumstances between this year, and the previous year when a version of the tablet also went on sale. Note that these mentions to time are not arranged linearly.

(1) The new iPad Air officially went on sale Friday.

(2) And while lines at Apple stores appeared to be shorter than in previous years, it's unclear what that actually means in terms of first weekend sales.

(3) That's because there are a number of factors that are different this year from a year ago when the new iPads went on sale.

Figure 1.1: Text extracted from a news report about the launch of a new tablet.

To properly understand this discourse and the concepts described, it is essential to know how the various events described in the sentences relate to one another temporally. This problem of interpreting time from text, together with how the gleaned information can be used, are the main foci of this thesis.

1.1 Interpreting Time From Text

The interpretation of time from text, also known as temporal interpretation or temporal processing, is really a means and not an end. Much like tasks such as part-of-speech (POS) tagging or grammar parsing, the purpose behind making sense of the temporal order of things really is to help downstream applications, or as a foundation for other more elaborate processing tasks. The goal of temporal interpretation in this thesis is therefore to derive a logical representation — a timeline — of the temporal information found within input text. Such a logical representation will allow other applications or technologies to easily leverage on

[1] *Apple Loyalists Flock To Stores For iPad Air Launch*, Los Angeles Times, 1 Nov. 2013

the temporal information found within text.

Before going on to describe what a timeline is, it is important to first explain two basic temporal units: 1) event expressions (or events for short), and 2) time expressions (or timexes for short). The definitions for these two basic temporal units follow that proposed by Pustejovsky et al. (2003a) for the standardized TimeML annotation. An event refers to an eventuality, a situation that occurs or an action. A timex is a reference to a particular date or time (e.g. "*2013 December 31*", or "*this coming Friday*").

A timeline then, allows us to relate these two basic temporal units together in a logical structure. Following from Newton's view of time, a timeline is a one-dimensional axis, along which time can be viewed as a sequential continuum. An example of a timeline is illustrated in Figure 1.2. It shows selected events from the text in Example 1.1. "*sale (2)*" denotes the occurrence of the word "sale" in sentence (2), and "*sale (3)*" denotes the occurrence of the word in sentence (3). From the timeline, it can be clearly seen that "*sale (3)*" refers to an earlier event than "*sale (2)*". It is also observed that "*appeared*" occurs together with "*sale (2)*". Chapter 3 will introduce timelines in greater detail.

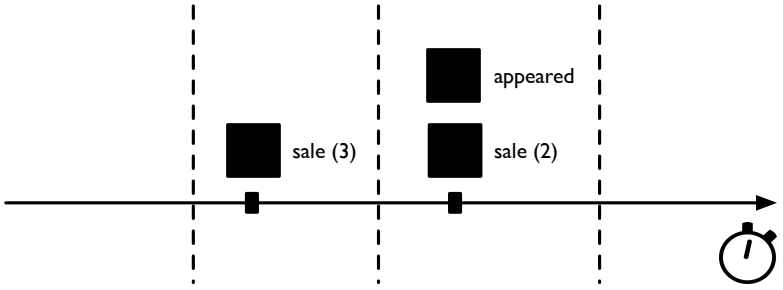


Figure 1.2: Extract of a possible timeline for Example 1.1.

To construct timelines, I adopt a sequential approach similar to that proposed by Verhagen et al. in the TempEval series of workshops (Verhagen et al., 2009, 2010). The construction makes use of results from three different temporal processing steps, including:

- Resolving a timex to an absolute timestamp
- Determining the temporal relationship between an event and a timex
- Determining the temporal relationship between two events

Later in Chapter 3, I will explain how the results from these steps can be merged to obtain a timeline.

The process of resolving a timex to an absolute timestamp is also known as *timex normalization*. Timexes can directly refer to a complete timestamp such as "*2013-December-31 05:12 +0000*", or they can refer to a relative time reference such as "*tomorrow*". Suppose a timex is created on "*2013-January-01 01:00 +0000*", then timex normalization will resolve "*tomorrow*" to "*2013-January-02*". Timex normalization is a well-studied problem. State-of-the-art systems (Bethard, 2013; Strötgen and Gertz, 2013) are capable of achieving very high accuracy rates. In this thesis I will leverage on these state-of-the-art systems.

The other two steps form the bulk of the focus of this thesis. Again, following the angle of attack used in the TempEval workshops, I frame them as classification tasks. Therefore I will refer to them henceforth as event-timex (*E-T*) temporal relationship classification and event-event (*E-E*) temporal relationship classification.

1.1.1 Temporal Relationship Classification

For *E-T* temporal relationship classification, the task is to identify the temporal relationship between an event and a timex. Accordingly, for *E-E* temporal relationship classification, the task is to identify the temporal relationship between two events.

The temporal relationships for both cases draw from a set including 1) BEFORE, 2) AFTER, and 3) OVERLAP temporal relations. Given two temporal units (without loss of generality, either of them could be events or timexes) tu_1 and tu_2, if tu_1 is BEFORE tu_2, it means that tu_1 happened before tu_2. Similarly if tu_1 is AFTER tu_2, it means that tu_1 happened after tu_2. If tu_1 OVERLAPs tu_2, it means that the two of them happened together, in the same time span.

These three core relations are the same as those defined in Verhagen et al. (2009). However in addition, they had also defined three additional relations 1) BEFORE_OR_OVERLAP, 2) OVERLAP_OR_AFTER, and 3) VAGUE. BEFORE_OR_OVERLAP represents a disjunction of BEFORE and OVERLAP, OVERLAP_OR_AFTER similarly refers to a disjunction of OVERLAP and AF-

TER. VAGUE is used to denote instances when the temporal relationship between the two vertices is not clear. These three additional classes are added after considering feedback from annotators who found them to be useful for some instances. However here, like in Denis and Muller (2011) and Do et al. (2012), I choose to focus on just the three core relations as they form up the bulk of most annotations (these three core relations form up to 89.1% of the TempEval-2 test dataset).

In the TempEval workshops, the *E-T* classification task is limited to only events and timexes found within the same sentence (i.e., intra-sentence). The *E-E* classification task is limited to only event pairs which are found within the same, or in adjacent sentences. These definitions were originally proposed as a trade-off between completeness, and the need to simplify the evaluation process (Verhagen et al., 2009). Taken in totality, they are however limiting and insufficient.

The problem is that they are insufficient to give us a complete view of the temporal information found within text. As illustrated in Figure 1.3, being able to perform only *E-E* classification on event pairs found at most in adjacent sentences does not allow us to know the temporal relationships between all event pairs. In the figure, a sentence $s3$ separates event C from D and E. It is not possible therefore to ascertain the relationship between the events in sentences $s1$ and $s2$, and those in sentence $s4$. The relationship between A and C in the figure can be determined with the use of the temporal transitivity rules (Setzer et al., 2003; Verhagen, 2005), but we cannot determine the relationship between say A and D. This presents problems downstream when one needs to construct a timeline with the results of *E-E* temporal relationship classification.

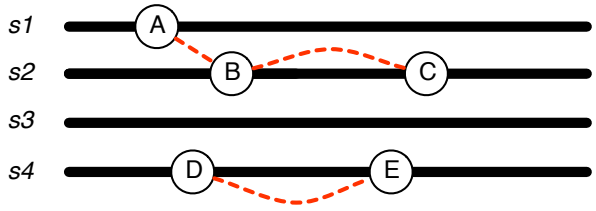


Figure 1.3: A disconnected temporal graph of events within an article. Horizontal lines depict sentences $s1$ to $s4$, and the circles identify events of interest.

Bearing this limitation in mind, this thesis will explore instead 1) intra-sentence *E-T* temporal classification, and 2) article-wide *E-E* temporal classification. By expanding the scope of *E-E* temporal classification to be able to take as input any pair of events within an article, the problem highlighted above is solved.

1.2 Making Use Of Time From Text

Having built a timeline, the last part of the thesis applies it to the problem of multi-document summarization. Very briefly, given a collection of input documents which discuss a similar topic, the goal of multi-document summarization is to generate one single summary which includes the main points from these documents, with minimal repetition of similar points from different documents. In other words, a summary should be representative, capturing as many relevant points as possible, while minimizing redundancy, that is repeated points of views or arguments. I believe that temporal information can help improve multi-document summarization precisely along these two key dimensions.

Relevancy. In Figure 1.4, the three sentences describe a recent cyclone and a previous one which happened in 1991, respectively. Recognizing that sentence (3) is about a storm that had happened in the past is important when writing a summary of the recent storm. This ensures that the content selected for the final summary will be more relevant to the reader.

(1) A fierce cyclone **packing** extreme winds and torrential rain **smashed** into Bangladesh's southwestern coast Thursday, **wiping** out homes and trees in what officials **described** as the worst storm in years.

(2) More than 100,000 coastal villagers have been **evacuated** before the cyclone made landfall.

(3) The storm matched one in 1991 that **sparked** a tidal wave that **killed** an estimated 138,000 people, Karmakar **told** AFP.

Figure 1.4: Modified extract from a news article which describes a cyclone attack. Several events which appear in Figure 1.5 are bolded.

It is reasonable to expect that a collection of documents about the recent storm will contain more references to it, compared with the earlier one that

happened in 1991. Visualized on a timeline, this will translate to more events (bolded in the Example 1.4) around the time in which the recent storm took place. There should be less events mentioned in the article for the time span in which the previous 1991 storm occurred. Figure 1.5 illustrates a possible timeline laid out with the events found in Example 1.4. The events from the more recent storm generally OVERLAP and are found together at the same time. There are less events which talk about the previous storm. By ordering the events sequentially, temporal information can help to more accurately identify relevant sentences to be included in a final summary.

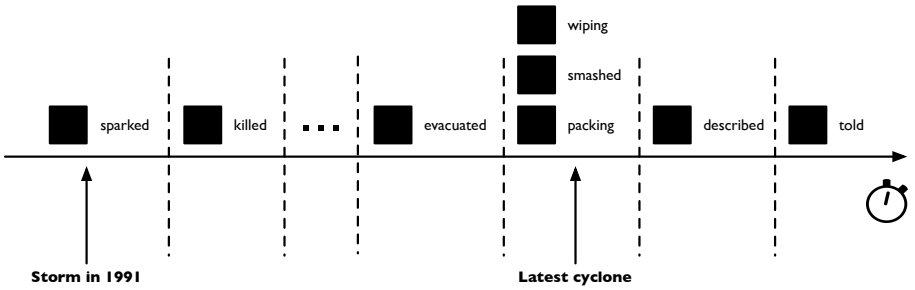


Figure 1.5: Possible timeline for events in Figure 1.4.

Redundancy. In Figure 1.6, the sentences describe many events which took place within the same time span. They describe the destruction caused by a hurricane with trees uprooted and buildings blown away. Knowing that these events occur together, a summarization system will be able to either paraphrase and group them together, or avoid selecting all of these sentences to be included in a summary. This avoids having similar pieces of information in the summary.

(1) An official in Barisal, 120 kilometres south of Dhaka, spoke of severe **destruction** as the 500 kilometre-wide mass of cloud **passed** overhead.

(2) "Many trees have been **uprooted** and houses and schools **blown** away," Mostofa Kamal, a district relief and rehabilitation officer, told AFP by telephone.

(3) "Mud huts have been **damaged** and the roofs of several houses **blown** off," said the state's relief minister, Mortaza Hossain.

Figure 1.6: Extract from a news article which describes several events (bolded) happening at the same time.

Existing approaches typically makes use of lexical-based methods to detect

such similarities (Carbonell and Goldstein, 1998; Hendrickx et al., 2009). However, in situations such as the one presented in Example 1.6, it is not trivial for a lexical approach to detect that events like "*passed*", "*uprooted*" or "*damaged*" are in fact describing the same hurricane attack. My approach in this thesis thus makes use of the time in which these events occur; the hypothesis being that events happening at the same time have a high chance of describing the same situation or incident. In this case it may not be necessary to include all of these events in the eventual generated summary.

1.3 Key Contributions

In a nutshell, my thesis examines the construction of a timeline via two important steps, 1) intra-sentence *E-T* temporal classification, and 2) article-wide *E-E* temporal classification. Improving on the state-of-the-art, my thesis then goes on to show that the obtained timeline is useful and can be effectively applied to improve multi-document summarization.

The key contributions of this thesis include:

1. eschewing the use of traditional lexico-syntactic features for *E-T* temporal classification, and showing that performance can be improved instead with the use of semantically motivated dependency parses (Ng and Kan, 2012),

2. proposing the use of discourse analysis to tackle the problem of article-wide *E-E* temporal classification achieving a 16% gain in performance over the state-of-the-art (Ng et al., 2013),

3. introducing a novel scheme to selectively annotate instances when building a dataset for *E-T* temporal classification which can shave annotation efforts by as much 37% (Ng and Kan, 2012), and

4. presenting a robust and effective scheme to integrate the results of temporal processing into multi-document summarization, leading to an improvement of 5.9% in terms of ROUGE-2 score (Ng et al., 2014) over a very competitive state-of-the-art summarization system (Ng et al., 2011, 2012).

1.4 Organization

In the next chapter, I will provide an introduction to the areas of temporal processing, as well as its application to summarization. Chapter 3 provides more details about timelines, as well as elaborates on how they can be constructed. Then in Chapters 4 and 5, I explain the work that I have done for *E-T* and *E-E* temporal classification. Building on these results, Chapter 6 talks about how temporal information can be integrated into a state-of-the-art multi-document summarization system. The last chapter concludes this thesis, highlighting opportunities for further research.

Chapter 2

Background

Before detailing the work that I have done, this chapter provides the necessary background knowledge with regards to the 1) temporal interpretation of text, and 2) use of temporal information for multi-document summarization.

Having motivated the thesis in the previous chapter, this chapter reviews the areas of 1) the temporal interpretation of text, and 2) multi-document summarization. Through this, I detail the developments of these respective fields and how they shape this thesis.

In the first section I touch on the temporal interpretation of text over three milestones: 1) early works on temporal interpretation, 2) the movement towards the development of standardized corpora, and 3) the TempEval series of evaluation workshops. I end the section with a review of the related work for temporal relationship classification, which is one of the main foci of this thesis.

Moving on to multi-document summarization, I present a brief on the various task guidelines of evaluation workshops that have shaped this area. Then I highlight key approaches that have been adopted to solve this problem, before concluding my review of the related work which applies temporal information to multi-document summarization, as is done in this thesis.

2.1 Temporal Interpretation of Text

2.1.1 Early Works On Timex Identification and Normalization

Researchers have long recognized the value of extracting temporal information from text. One of the earliest pieces of work on temporal processing was on the identification and normalization of timexes. Identification of timexes involve extracting or marking out mentions to time in a piece of text. Time mentions can include complete timestamps such as "*31 Dec 2013, 08:00 am, UTC+0800*", or relative references such as "*today*" and "*last Friday*". Normalization of timexes on the other hand refer to resolving time mentions to complete timestamps. For example, if an article is written on 31 Dec 2013 (also called the *document creation time* or DCT for short), the word "*today*" appearing in the article can be normalized to "*31 Dec 2013*". Similarly the word "*yesterday*" will normalize to "*30 Dec 2013*".

The 6th and 7th Message Understanding Conference (MUC-6, MUC-7) (Chinchor, 1998; Sundheim, 1996) included a named entity recognition task. Timexes are one out of several named entities which participants are required to extract. Correspondingly, the datasets used in MUC-6 and MUC-7 included annotations for two types of timexes, namely dates (e.g., "*December 31 2013*") and times (e.g., "*13:32:21 +0900*"). The Automatic Content Extraction (ACE) Time Expression Recognition and Normalization (TERN) 2004 shared task (ACE, 2004) expanded on the scope of MUC-6 and MUC-7 to include the normalization of identified timexes to a complete timestamp. To support the normalization task, the dataset created for ACE 2004 included normalized timestamps as well.

2.1.2 Development of Large-scale Corpora

In 2001, testament to the rising research interest in the interpretation and processing of text, one of the first workshops dedicated to temporal processing was held in conjunction with the Annual Meeting of the Association for Computational Linguistics (ACL) conference (Harper et al., 2001). Many of the published works focused on the annotation of temporal information and the compilation of datasets which can be used subsequently for machine learning (Katz and Aro-

sio, 2001; Setzer and Gaizauskas, 2001). Together with work of Ferro et al. (2000), these eventually formed the foundation of the TimeML (Pustejovsky et al., 2003a), a specification language for events and timexes.

TimeML refines upon the annotation of timexes from the earlier efforts of Ferro et al. (2000) and Setzer and Gaizauskas (2001), introducing the TIMEX3 tag to annotate explicit temporal expressions (including dates, times and durations).

Importantly, TimeML also specifies explicitly the annotation of events, which generally refer to situations that *happen* or *occur*. The definition of an *event* is encompassing without a formal definition, appealing instead to the intuition and knowledge of annotators. TimeML further included annotations for the *class*, *tense*, and *aspect* of each event. The *class* of an event refers to its type. Table 2.1 shows the identified event classes, as well as example event words associated with these classes. The *tense* of an event corresponds to the linguistic understanding of the tense of a word (i.e., *past tense*, *present tense*), while the *aspect* of an event refers to the grammatical aspect of the event (i.e. *progressive*, *perfect progressive*).

Class	**Description**
Reporting	Describes action of an entity declaring, or narrating something e.g., *say, report, announce*
Perception	Involves physical perception of another event e.g., *see, hear, watch, feel*
Aspectual	Grammatical device of aspectual predication e.g., *begin, finish, stop, continue*
I-Action[1]	Introduces an event argument describing an action which relates to the current action, e.g., *attempt, try, promise, offer*
I-State[1]	Similar to I-Action but pertains to states that refer to possible worlds e.g., *believe, intend, want*
State	Describes circumstances in which something holds e.g., *on board, kidnapped, love*
Occurrence	Events that describe something that happens or occurs e.g., *die, crash, build, merge, sell*

Table 2.1: List of TimeML event classes and representative examples of each class. This list is compiled from Pustejovsky et al. (2003a). [1] "I" here refers to "Intensional".

Another significant contribution of TimeML is the introduction of LINKS. LINKS encode the relationships that exist between temporal elements including

events and timexes, the most used being TLINKs, which capture the relationship between events (*E-E*), or between an event and a timex (*E-T*). TLINKs are typed, which means that the relationship between the two associated temporal elements are assigned to one of 13 identified scenarios (e.g., simultaneous, identical, one before the other, and so on.).

Based on TimeML, the TimeBank corpus was introduced in 2003 (Pustejovsky et al., 2003b). TimeBank consists of 300 newswire articles with TimeML prescribed annotations, including annotations for events, timexes and links between these temporal elements. Articles used in the corpus draw from various sources, including text from the Document Understanding Conference (DUC), the Automatic Content Extraction (ACE) program and also Propbank texts from the Wall Street Journal. In a revised version of the corpus released in 2006, inter-annotator agreement for events, timexes and links (TLINKs) are 78%, 83% and 55% respectively[1]. These figures are interesting because they highlight the relative difficulty of each task. Indeed as will be seen in the next sub-section, automatic system performances for event/timex extraction and TLINK classification obey the same trends described by these agreement values.

Subsequent years saw the development and availability of more temporal corpora, including three corpora compiled for the three TempEval workshops (see next sub-section). The TempEval-1 corpus, TempEval-2 corpus, and TempEval-3 corpus — as they will be referred to — form the basis for much work in this domain.

With the exception of the TempEval-3 corpus, all of these corpora (including the TimeBank corpus) are relatively small, consisting of not more than a few hundred documents. As with the development of all other corpora, putting together these corpora requires lots of costly human effort. To aid the development of large corpora, Setzer et al. (2003), and later Verhagen (2005), worked on a series of inference rules which can be used to infer new temporal relations from existing ones. One example of the rules explained in Setzer et al. (2003) include:

$$(x, y) \in B \wedge (y, z) \in O \Rightarrow (x, z) \in B \tag{2.1}$$

[1] `http://timeml.org/site/timebank/documentation-1.2.html`

where x, y, z are events, B denotes BEFORE, and O denotes OVERLAP (Setzer et al. referred to this as "*Simultaneous*" as they followed the naming used in TimeML). This rule says that if x happens BEFORE y, and y happens together with z, then it follows that x also happens BEFORE z.

Setzer et al. started with a set of seed labeled instances, and applied these transitivity inference rules to identify new temporal relations from a pool of unlabeled instances. Only unlabeled instances for which no new temporal relations can be automatically identified are then manually annotated. This reduces the effort required to create a corpus substantially. Similar schemes are subsequently used by several other researchers (Do et al., 2012; Mani et al., 2006) to expand the amount of data available for supervised machine learning.

The TempEval-3 corpus was built with a slightly different approach. A large collection of around 600K words from Gigaword (Parker et al., 2011) were collected, and annotated by automatic systems including TIPSem, TIPSem-B (Llorens et al., 2013) and TRIOS (Uzzaman and Allen, 2010). These systems are existing state-of-the-art temporal processing systems. Output from these three systems are then merged (Llorens et al., 2012) to create a "*silver*" dataset. While lacking the rigour and accuracy of human-annotated datasets, this silver dataset increased the size of the corpus significantly. The silver dataset consists of more than 600K words, while the original TimeBank corpus consists of only 61K words.

There is scope for more research in this area. Beyond just leveraging on temporal transitivity to infer new relations, the TempEval-3 corpus creatively makes use of multiple automatic systems to help annotate and build a significantly larger dataset. However completely removing humans from the annotation loop means that it is not possible to definitively know the correctness of the dataset. This thesis examines another approach — selecting only those instances which can contribute to positive classifier performance for human annotation.

2.1.3 TempEval Evaluation Workshops

The availability of corpora like the TIDES temporal corpus (Ferro et al., 2000) and TimeBank corpus (Pustejovsky et al., 2003b) encouraged additional re-

search in temporal processing. To direct the efforts of the community, a series of evaluation workshops was conceived and held. The TempEval-1 (Verhagen et al., 2009), -2 (Verhagen et al., 2010), and -3 (Uzzaman et al., 2013) workshops attracted significant participation and much of existing literature on temporal processing centers around the tasks defined in the workshops.

Table 2.2 summarizes the tasks that made up each workshop. Initially in TempEval-1, events and timexes are pre-annotated. Participating systems worked on temporal relationship classification between the various temporal elements, including events, timexes, and document creation times (DCT). In TempEval-2, the scope of the workshop was expanded. New tasks were added which also required systems to perform timex identification and normalization (much like the MUC-6, MUC-7 and ACE TERN evaluations), and event identification and attributes classification. Recently in TempEval-3, with advancements in the state-of-the-art, the organizers decided to evaluate end-to-end systems, where systems have to perform the full suite of temporal processing tasks, including extracting timexes and events, and identifying the temporal relationships between them.

Recall that in TimeML, 13 TLINK relations were identified. However for TempEval-1 and -2, the organizers decided to just make use of a reduced set of three relations, i.e. core temporal relations including "BEFORE", "AFTER" and "OVERLAP". A further two relations were also subsequently added to ease annotation effort, made up of a disjunction of these three core relations, i.e. "BEFORE-OR-OVERLAP", "OVERLAP-OR-AFTER". A third relation "VAGUE" was also added in cases where no viable temporal relation could be attributed to a pair. In TempEval-3, the organizers began using the full 13 relations identified in TimeML.

2.1.4 Spotlight on Temporal Relationship Classification

Timex identification and normalization (and to some extent event identification) have been well-studied and researched on from earlier efforts in the MUCs and ACE TERN evaluations. The state-of-the-art for timex identification and normalization is a F_1 score of around 0.86 (Bethard, 2013; Strötgen and Gertz,

Task	TempEval-1	TempEval-2	TempEval-3
Timex identification and normalization		**Task A**	**Task A**
Event identification and attributes classification		**Task B**	**Task B**
E-T relationship classification (within same sentence)	**Task A**	**Task C**	
Event-DCT relationship classification	**Task B**	**Task D**	
E-E relationship classification (adjoining sentences)	**Task C**	**Task E**	
E-E relationship classification (syntactically dominated events)		**Task F**	
End-to-end system			**Task ABC**

Table 2.2: Tasks performed at different TempEval workshops.

2013). The best systems for event identification and attribute classification also perform at a F_1 score onwards of 0.8 (Grover et al., 2010; Uzzaman and Allen, 2010).

Temporal relationship classification however is still an open problem. It is a hard problem, as evident from the low annotator agreement values seen when building the TimeBank corpus.

Earlier efforts include that of Lapata and Lascarides (2006). They made use of a compiled lexicon of temporal words as clues to the correct temporal relation between two temporal elements. Han and Lavie (2004) took on a more mathematically-grounded approach. They proposed a formal representation for temporal expressions, framing the problem as a constraint-satisfaction problem. The new representation was motivated because the authors felt that *time* is sufficiently unique to warrant a separate reasoning system than a generic first-order logic system, such as that proposed by Gabbay et al. (2000).

The availability of new corpora as described above subsequently encouraged the adoption of more data-centric approaches. State-of-the-art systems now commonly adopt a variety of supervised machine learning techniques, including conditional random fields (Kolya et al., 2010), Markov logic (Ha et al., 2010; Uzzaman and Allen, 2010), maximum entropy classification (Derczynski and Gaizauskas, 2010) and convolution kernel support vector machines (Mirroshandel

et al., 2011).

These attempts have typically approached *E-T* and *E-E* temporal relationship classification as two separate classification tasks. Recognizing that they are highly co-related, Yoshikawa et al. (2009) attempted to solve these problems together with the use of a joint model, while Do et al. (2012) made use of Integer Linear Programming (ILP) with a set of global constraints. They are able to show that performing joint inference helps both classification tasks significantly.

However despite the differences in methodology or choice of machine learners, these recent works display a very strong bias towards lexico-syntactic features. It is possible to classify the features that are employed into three major feature types: 1) lexical cues, such as signal words or part-of-speech tags, 2) context, including the attributes of events and timexes, and 3) the grammatical structure of sentences obtained with the use of automatic parses. While these features are useful, system performances have however shown signs of stagnating. In this thesis thus, I propose the use of semantically motivated features to help achieve better performance.

2.2 Multi-Document Summarization

The later part of this thesis applies the results of temporal processing on multi-document summarization. In this section a quick introduction to work in this domain is given.

Broadly speaking, summarization involves identifying the key ideas expressed in a given text, and then combining these ideas into a passage which is typically much shorter than the original text. The purpose is to reduce the amount of effort and time needed by a reader. Depending on how the final summary is produced, summarization can either be 1) extractive, or 2) abstractive. In extractive summarization, sentences from the original document are used in the final summary as-is. However in abstractive summarization, text from the original document are typically paraphrased or edited before being included in the generated summary. Abstractive summarization is the harder of the two, as it involves more than just identifying salient points and issues within the input

document. As such most published results have been focusing on extractive summarization. However in recent years, researchers are also paying increasing attention to abstractive summarization, with advances in language generation, paraphrasing and sentence compression.

Earlier efforts in text summarization focused on single-document summarization (Mani, 2001), which takes as input a single document, and generates a summary for the document. Multi-document summarization on the other hand takes as input a set comprising of several documents. One output summary is to be generated for the set, presenting the main points of all the documents within the set with minimal repetition of facts.

A typical application scenario for multi-document summarization is in news summarization. Modern news aggregators like Google News (Das et al., 2007) collect news documents from various news sources. Given a collection of articles on the same news event, multi-document summarization can be used to generate a summary for the collection. This can save readers the hassle of reading through all the articles in the collection.

2.2.1 Summarization in Shared Tasks

Multi-document summarization caught on rapidly. From 2001 to 2011, it featured regularly as one of the tracks in the Document Understanding Conference (DUC) and Text Analysis Conference (TAC) workshops. These annual evaluations helped to guide and shape much of the research work in the area of text summarization. It is therefore interesting to take a look at the major developments during these series of evaluation workshops.

Multi-document summarization featured as one of the tasks in the inaugural DUC held in 2001. The task guidelines required participating systems to generate "generic" summaries of fixed target lengths, with no specific guidance given on the preferred content to include.

A notable update to the task guidelines was made in 2005. Instead of generic summarization, the focus was shifted to "*question-focused*" summarization. Participating systems now had to explicitly piece together information from an input document set to answer a question or a set of questions posed about the docu-

Title: American Tobacco Companies Overseas
Narrative: In the early 1990s, American tobacco companies tried to expand their business overseas. What did these companies do or try to do and where? How did their parent companies fare?

Figure 2.1: Example of a question posed for a document set for question-focused summarization in DUC-2005.

ment set. Figure 2.1 illustrates sample questions for a particular document set quoted from Dang (2005). These questions ask explicitly about how tobacco companies from the United States seek to expand overseas, as well as the level of success they achieved. This change in guidelines was motivated to better align the research that is being performed in this area to information needs that real users are facing.

Another important update was the introduction of "*update*" summarization in DUC-2007. As per question-focused summarization, summaries were still to be generated according to a set of questions. However each document set now consists of several clusters which are ordered sequentially based on publication dates. The idea is to progressive generate new summaries, each time highlighting new developments to the reader, assuming that previous clusters have already been read.

The summarization track in DUC was moved to the Text Analysis Conference (TAC) in 2008, where it continued to be held annually till 2011. TAC-2008 and TAC-2009 continued the update summarization task introduced in DUC-2007.

In TAC-2010 and TAC-2011, a major change to the task guidelines was made with the introduction of "*guided*" summarization. In guided summarization, each document set (also called a *topic*) to be summarized is assigned to one of several broad *categories*. These categories include 1) Accidents and Natural Disasters (Accidents), 2) Attacks, 3) Health and Safety (Health), 4) Endangered Resources (Resources), and 5) Investigations and Trials (Investigations). The abbreviations of the category names (in parentheses) are used in the rest of this thesis for brevity. For each of these categories, there is a template which contain information elements, or *aspects*, that are requested for (see Figure 2.2). Participating systems are required to generate summaries for each topic guided

by the corresponding template of aspects.

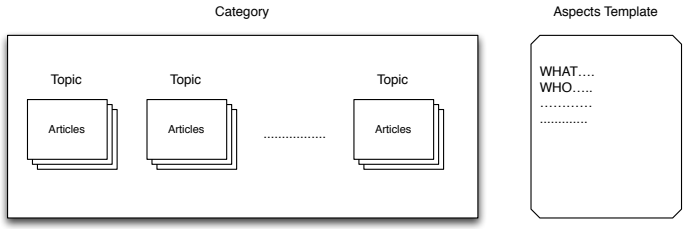


Figure 2.2: How articles, topics, categories and aspects come together.

An important element of these shared tasks is the evaluation of submitted summaries. Initially manual evaluation was carried out, with human evaluators tasked to assess the quality of automatically generated summaries. In DUC-2004, the ROUGE measure (Lin and Hovy, 2003) was introduced to complement costly human assessment. ROUGE determines the quality of a summary through overlapping units such as n-grams, word sequences, and word pairs with human written summaries. ROUGE was found to correlate well with the result of the human assessments that were made (Over and Yen, 2004).

The use of Basic Elements (Hovy et al., 2005) and Pyramids (Passonneau et al., 2005) was tried out in DUC-2005 to complement the use of ROUGE. Basic Elements (BEs) is an automatic method which evaluates the content completeness of a generated summary by breaking up sentences into smaller, more granular units of information (referred to as *Basic Elements*). Pyramids on the other hand requires a fair amount of human effort to identify units of information (called *Summary Content Units* or SCUs) from human-written model summaries. Manual judgements are then used to map content within a generated summary to these SCUs. This mapping is subsequently used in scoring the generated summary. As the mapping of content to SCUs is done at the semantic level instead of at the lexical level, Pyramids are suitable for assessing abstractive summaries as well. Abstractive summarization, by the nature of how summaries are generated, are un-fairly penalized by lexical-based evaluation metrics like ROUGE. Both Basic Elements and Pyramids have also been shown to correlate well with human assessments.

2.2.2 Main Approaches

The sheer volume of work done on multi-document summarization is so overwhelming, it is not feasible to detail all these work here. A more detailed treatment of multi-document summarization can be found in Nenkova and McKeown (2011). Here, I have identified the methodologies adopted by researchers and taxonomised them into the following key approaches: 1) concept linkage, 2) heuristics, 3) discourse analysis, 4) joint inference, 5) topic modeling, and 6) use of large data. Note however that this classification is not absolute nor complete, and definitely not mutually exclusive. In fact, many systems do make use of a combination of these approaches.

Concept Linkage. SUMMONS (McKeown and Radev, 1995) was one of the several pioneer multi-document summarization systems. It first retrieves important concepts from each input document in the input set. Then it tries to link up similar concepts together so that they need not be repeated in the final summary. The system also looks out for important phenomena, such as a change in perspective towards an issue, and tries to include these in the final summary.

Mani and Bloedorn (1997) proposed a graph-based approach. Key concepts are derived from input documents within a set, and mapped into a graph as vertices. Different types of edges link up these vertices, representing semantic associations between these vertices. For example, vertices representing the same concepts are associated with a "*co-reference*" link. An ontology is also used to build semantically-motivated links between vertices by capturing the relationship between entities (e.g., *Barack Obama* is the president of the *United States*). With such a graph, it is possible to find vertices which are common (and different). A summary can then be generated by selecting sentences which best cover vertices that are common (for similarities, as well as unique (for differences).

Heuristics. Lin and Hovy (2002) presented a influential multi-document summarization system NEATS which tapped on proven features and algorithms for single-document summarization. It is made up of a pipeline of three key stages: 1) content selection, 2) content filtering, and 3) content presentation. The features proposed for content selection including 1) term frequency, 2) sentence position, and 3) stigma words remain relevant and effective till this date.

Discourse Analysis. Radev (2000) introduced the Cross-document Structure Theory (CST) and proposed using it as the basis for multi-document summarization. CST is inspired by the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). RST is used to study and analyze text coherence, assigning pre-defined discourse relations to neighbouring units of text called "*elementary discourse units*" (EDUs). CST extends this concept to relate sets of documents. A transformation on the CST graph can then be carried out with a set of pre-identified operators to reduce the number of nodes within the graph without affecting its properties. The transformed graph forms the basis for the eventual generated summary.

Another summarizer that builds on RST is described by Marcu (1997). Most RST discourse relations differentiate between the roles of two participating argument EDUs. One of the EDUs is a "*nucleus*", while the other is a "*satellite*". The nucleus holds more importance, from the point of view of the writer, while the satellite's purpose is to provide more information to help understand the nucleus. Marcu hypothesized that nuclei EDUs are more salient, and proposed a rhetorical parsing algorithm that identifies salient EDUs to be included in the final summary.

Joint Inference. Two of the goals in multi-document summarization include maximizing the relevancy of sentences in the generated summary, while reducing the amount of redundancy. Carbonell and Goldstein (1998) proposed one of the earliest joint inferencing scheme which seeks to optimize sentence selection based on these two goals simultaneously. The proposed Maximal Marginal Relevance (MMR) algorithm is an iterative, greedy algorithm that selects sentences based on a linearly weighted sum of the importance of the sentences and their lexical similarity to the summary composed so far. McDonald (2007) expanded on this and found that a Integer Linear Programming (ILP) formulation is able to deliver better optimization results over the relatively simple linear model used in MMR.

Topic Modeling. Many researchers favored generating summaries around the topics or themes presented in source documents (Harabagiu and Lacatusu, 2005; Lin and Hovy, 2000). The motivation is that identifying the main ideas described

in the source documents will help to guide the content that needs to be included in the final summary. State-of-the-art systems built with this approach typically make use of latent Dirichlet allocation (LDA) to build an underlying topic model to describe the source documents. Haghighi and Vanderwende (2009) for example made use of this approach to build a hierarchical topic model which can be used to produce summaries covering all topics and sub-topics, as well as summaries specific to particular sub-topics.

Use of Large Data. Knowledge-poor statistical methods had also been used extensively. For example both Bysani et al. (2009) and Zhang et al. (2011) made use of supervised machine learning techniques including support vector regression and support vector machines; sentences to be included in the final summaries are derived from a ranked list of sentences which is built from the output of these machine learning algorithms.

2.2.3 Incorporating Time

Of particular relevance to this thesis, is the incorporation of the concept of *time* into multi-document summarization. *Time* can be a useful aspect especially since most of the field had so far been focused on the summarization of news article. News events are often reported chronologically, and readers following a news event often get updated on the event as it unfolds.

Barzilay et al. (1999) was one of the first pieces of work to consider the use of time for multi-document summarization. They postulated that it is important to be able to generate a summary which presents the time perspective of the summarized documents correctly. To achieve this they estimated the chronological ordering of events with a small set of heuristics, and also made use of lexical patterns to perform basic time normalization on terms like "today" relative to the document creation time. These information were then used to generate summaries which obey the chronological order set out in the original documents.

Goldstein et al. (2000) on the other hand made use of the temporal ordering of documents within a document set to be summarized. In computing the relevance of a passage for inclusion into the final summary, they considered the recency of the passage's source document. Passages from more recent documents are

deemed to be more important. Wan (2007) applied the same intuition, and considered the recency of documents in his modified TextRank algorithm. The proposed TimedTextRank algorithm gives preferential consideration to sentences from more recent documents. Demartini et al. (2010) also considered the recency and frequency of documents in a related task of entity summarization, where the goal is to retrieve a set of entities that summarizes a set of documents.

Instead of just considering the notion of recency, Liu et al. (2009) proposed an interesting approach using a temporal graph. Events within a document set correspond to vertices in their proposed graph, while edges are determined by the temporal ordering of events. From the resulting weakly-connected graph, the largest forests are assumed to contain the key topics within the document set, and used to influence a scoring mechanism to prefer sentences which touch on these topics.

Wu (2008) also made use of the relative ordering of events. He assigned absolute timestamps to events extracted from text. After laying these events out onto a timeline by making use of these timestamps, the number of events that happen within the same day is used to influence the scoring of sentences that cover these events. The underlying motivation is that days which have a large number of events should be more worthy of reporting than others.

It can be seen that prior work in applying temporal information involves either 1) sentence re-ordering, or 2) use of recency as an indicator of saliency. In sentence re-ordering, final summaries are re-arranged so that the extracted sentences that form the summary are in a chronological order. I argue that this may not be appropriate for all summaries. Depending on the style of writing or journalistic guidelines, a summary can arguably be written in a number of ways. The use of recency as an indicator of saliency is useful, yet dis-regards other pieces of information that can be had from the use of time. In fact if a summary of a whole sequence of events is desired, recency becomes less useful.

The work of Wu (2008) is closely related to what is presented in the last part of this thesis. He had also made use of temporal information to generate summaries. However his approach is guided mainly by the number of events happening within the same time span, and relies on event co-referencing. In my

work, I have simplified this idea by dropping the need for event co-referencing (thus removing one source of propagated errors), and augmented it to two other features derived from temporal information. By doing so, I am able to make better use of the available temporal information, taking into account all known events and the time spans that they occur in.

Chapter 3

Timelines

Timelines are used to represent temporal information from text in this thesis. This chapter explains what a timeline is, and details how such a timeline can be constructed.

In this chapter, I will explain the use of timelines as a means of representing temporal information from text. Timelines are well-understood constructs which have often been used for this purpose (Denis and Muller, 2011; Do et al., 2012). I will describe here the scope of the representation, as well as the construction of timelines via the merging of intermediate temporal processing steps.

3.1 Preliminaries

Figure 3.1 illustrates a typical timeline used in the rest of this thesis. The arrowed, horizontal axis is the timeline itself. The timeline can be viewed as a continuum of time, with points on the timeline referring to specific moments of time.

Small solid blocks on the timeline itself are references to absolute timestamps along the timeline. In the figure, two such examples can be seen referring to "*2013-Jan-01 01:00 +0000*" and "*2013-Feb-13 11:32 +0000*".

The black square boxes above the timeline denote events. Events can either occur at a specific instance of time (e.g., an explosion), or over a period of time (e.g. a soccer match that takes 90 minutes to play out). Generalizing, I will refer

to the time period an event takes place in as its "*time span*". This is demarcated by vertical dotted lines in the figure. Note that time spans do not necessarily correspond to specific instances of time, but instead serve mainly to demarcate BEFORE and AFTER relations between different events.

To interpret the timeline, remember that the horizontal axis is a sequential time continuum. Therefore events which appear to the left others take place earlier. If two events fall within the same time span, it means that they occur together over the same time period.

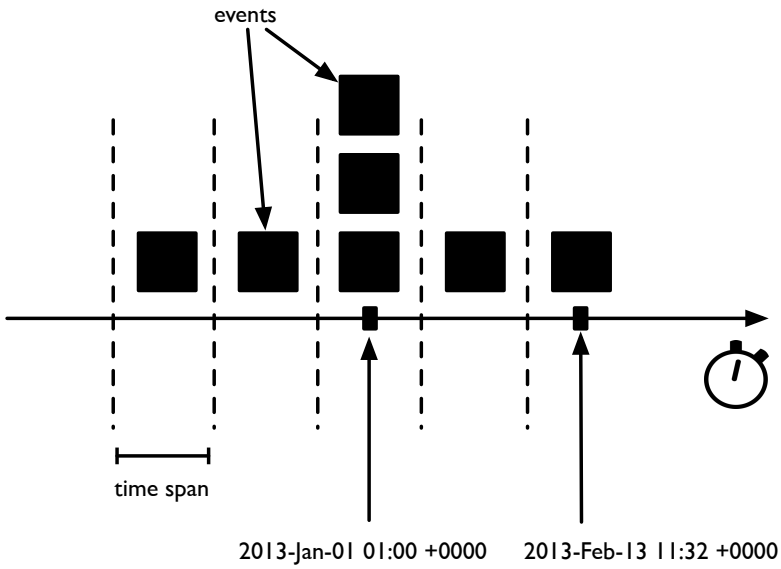


Figure 3.1: A typical timeline used throughout this thesis, showing events placed sequentially along a time continuum.

3.2 Constructing Timelines

As explained earlier in Chapter 1, this thesis breaks down the construction of a timeline into three steps:

1. Timex normalization
2. *E-T* temporal relationship classification
3. *E-E* temporal relationship classification

An overview of this is illustrated in Figure 3.2. The requisite pre-processing (i.e., event and timex extraction), and timex normalization, are integral to the

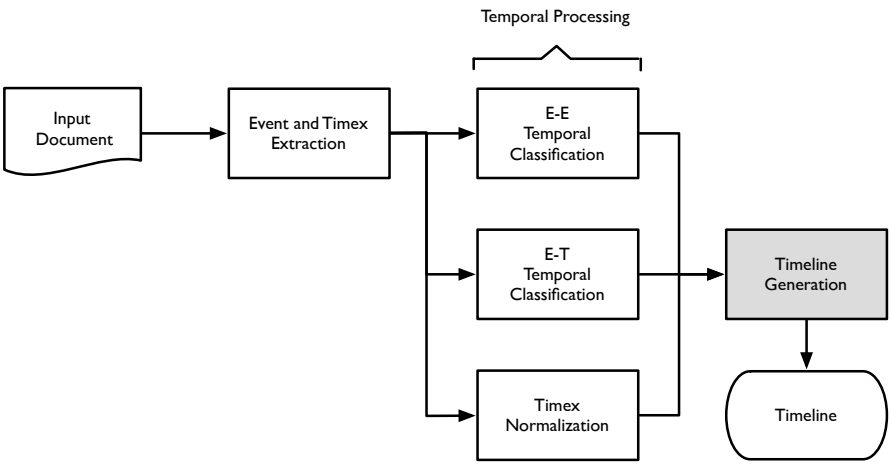


Figure 3.2: Overview of how a timeline can be constructed by merging the results from various temporal processing steps.

construction process. However, their investigation is not within the scope of this thesis. These are well-studied problems (as explained in Chapters 1 and 2) where state-of-the-art systems turn in good accuracy performances. *E-T* and *E-E* temporal relationship classification will be explained in the next two chapters. I now explain the stage shaded gray — i.e., timeline generation — which merges the results from temporal processing to obtain a timeline. Timeline generation can be broken down into three main steps:

Step 1. The first step makes use of timex normalization. In this step, all timexes which can be resolved to a particular time are identified. Note that timexes can refer to time of varying granularity. For clarity, I taxonomize them as 1) complete timestamps (e.g., "*2012-Jan-01 05:30 +0000*"), and 2) time periods (e.g., "*2012-Jan-1*", "*Friday*", "*2010*"). The difference between these two types of timexes is that the former refers to a specific time point on the timeline, while the latter refers to a continuous segment on the timeline. The timex "*2010*" for example refers to all the points that lie within "*2010-Jan-01 00:00 +0000*" and "*2010-Dec-31 23:59 +0000*" on the timeline (without loss of generality, timestamps are expressed rounded off to the nearest minute here).

Step 2. The next step makes use of information from *E-T* temporal relationship classification to place events onto the timeline. Starting from the timexes identified in Step 1, all events which OVERLAP with these timexes are

identified. The idea is to place these events onto the timeline based on the time points referred to by the corresponding timexes. Two cases are possible here. The first is when an event $e1$ OVERLAPS with timex $t1$, and $t1$ is a complete timestamp referring to say "*2012-Jan-15 05:30 +0000*". Then $e1$ is placed onto the timeline at a point corresponding to "*2012-Jan-15 05:30 +0000*". The second case arises when an event OVERLAPS with a timex which is a time period. In this situation the event is placed onto the timeline, based on the starting time of the time period. Consider event $e2$ which OVERLAPS with timex $t2$. Suppose $t2$ refers to the time period "*January 2012*" (i.e., "*2012-Jan-01 00:00 +0000*" to "*2012-Jan-31 23:59 +0000*"). Taking the starting time for $e1$ and $e2$, $e2$ is placed in front of $e1$ on the timeline (i.e., comparing "*2012-Jan-15 05:30 +0000*" and "*2012-Jan-01 00:00 +0000*").

Step 3. The last step inserts all remaining events into the timeline using information from *E-E* temporal relationship classification. For each event to be inserted, the timeline is traversed from left to right until a suitable location to insert the new event is reached (see Figure 3.3). A suitable location here means that all events to the left of the location happens BEFORE the event to be inserted; and that all events to the right of the location happens AFTER.

The complete algorithm which performs these three steps is shown in Algorithm 3.1. The algorithm makes certain important assumptions in Steps 2 and 3 described above. I will discuss the implications of these assumptions in the following section.

3.3 Timeline Limitations and Caveats

While I have chosen to adopt the divide-and-conquer approach first proposed in TempEval-1 (Verhagen et al., 2009), it is not the only way to construct a timeline. The algorithm makes several decisions (*e.g.*, the order in which OVERLAP, BEFORE and AFTER relations are considered) which can foreseeably be varied to achieve the same results. Further, it is possible to envisage the use of inter-

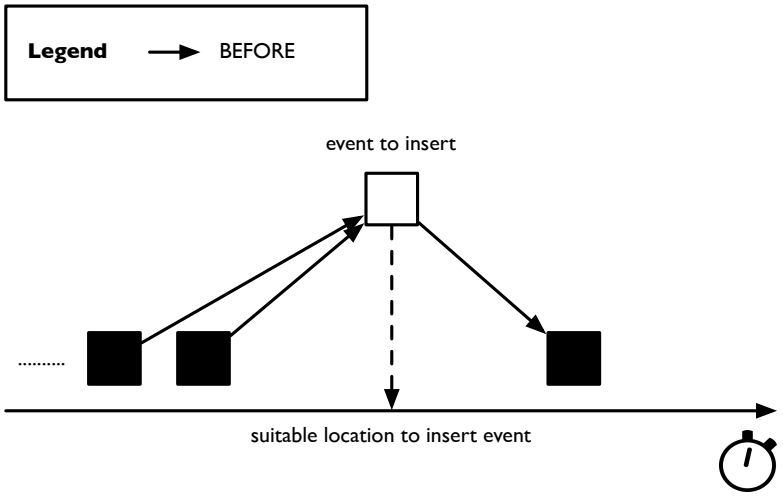


Figure 3.3: Traversing the timeline to identify a suitable point to insert new event.

mediate temporal processing steps other than what I have shown in Figure 3.2. It is instructive therefore to discuss the implications of the assumptions made in the construction algorithm. I will also discuss the the choice of using a timeline representation over a more generic temporal graph as was originally suggested in Verhagen et al. (2009).

3.3.1 Granularity of Temporal Relations

The chosen timeline representation has two main limitations: 1) the duration of events are ignored, and 2) events are ordered based on their starting times (*i.e.*, Step 2 of the construction algorithm). These boil down to the granularity of the temporal relations adopted for the temporal processing steps earlier on in the pipeline (refer to Figure 3.2). In this thesis, I had focused on the core TempEval temporal relations including 1) BEFORE, 2) AFTER, and 3) OVERLAP. This is to allow for fairer and effective comparison against existing state-of-the-art, because as reviewed earlier, much of the related work in this area stem from the TempEval evaluation workshops.

These three relations are actually derived from a complete set of 13 relations defined in Allen's interval algebra (Allen, 1983) (see Table 3.2[1]). The organisers of TempEval-1 and -2 (Verhagen et al., 2009, 2010) adopted this simplification

[1]This presentation is adapted from `http://en.wikipedia.org/wiki/Allen's_interval_algebra`

Algorithm 3.1 Obtaining a timeline from a temporal graph.

```
1: EventTimelineMap ← {} // Logical representation of timeline
2: Let timeline 𝕋 = {θ1, θ2, ...}, where each θi is a point in time
3: for each timex txi associated with an absolute time stamp tsi do
4:     Project txi onto 𝕋 by matching tsi to a corresponding θi ∈ 𝕋
5: end for
6: Let Θ = {tm1, tm2, ...} be the timexes that have been mapped onto 𝕋
7: for each event ei do
8:     for each mapped timex tmj ∈ Θ do
9:         if ei OVERLAP tmj then
10:            EventTimelineMap ← EventTimelineMap ⋃ {ei, tmj}
11:        end if
12:        if ei does not map to any tmj then
13:            UnMapped ← UnMapped ⋃ ei
14:        end if
15:    end for
16: end for
17: repeat
18:    for ei ∈ UnMapped do
19:        for each mapped event {ek, tmx} ∈ the sorted EventTimelineMap do
20:            if ei AFTER ek then
21:                continue
22:            end if
23:            if ei OVERLAP ek then
24:                EventTimelineMap ← EventTimelineMap ⋃ {ei, tmx}
25:                UnMapped ← UnMapped − ei
26:            end if
27:            if ei BEFORE ek then
28:                Create new point tmnew in Θ between tmx−1 and tmx
29:                EventTimelineMap ← EventTimelineMap ⋃ {ei, tmnew}
30:                UnMapped ← UnMapped − ei
31:            end if
32:        end for
33:    end for
34: until EventTimelineMap does not change
```

to keep the temporal classification tasks feasible and more manageable.

Table 3.1 shows a possible mapping between the three core TempEval relations and relations defined in Allen's interval algebra. The BEFORE and AFTER relations map to exactly one relation in Allen's interval algebra, while the OVERLAP relation can possibly be mapped to all remaining relations in Allen's interval algebra. A good amount of temporal information is thus potentially lost with the use of the TempEval relations. With the TempEval relations, it is possible to know the relative order of two events, if the time in which they occur do not intersect with one another. However when the times do intersect,

Relation Type	**Corresponding Relations**		
TempEval	BEFORE	AFTER	OVERLAP
Allen's	X < Y	Y > X	X o Y X s Y X f Y X d Y X = Y

Table 3.1: Possible mapping of the three core TempEval relations to the relations defined in Allen's interval algebra. For brevity, inverse relations are not shown.

it is impossible to decide how this intersection occurs.

This is the reason why the duration of events are ignored in the chosen timeline representation explained earlier. Regardless of whether two concurrent events take place over the span of seconds or hours, with the TempEval OVERLAP relation, it will not be possible to differentiate between the relative ordering of these events. This is also why events are ordered by their starting times during timeline construction. Since the OVERLAP relation cannot tell between the different intersection scenarios, the best thing is for the construction algorithm to always decide consistently to order events based on their starting times.

Having explained this, a good thought question will be how the chosen timeline representation may change, should a more fine-grained set of temporal relations be used instead. The latest iteration of the TempEval workshop (Uzzaman et al., 2013) for example, has progressed to make use of the complete set of 13 relations from Allen's interval algebra. With the use of this full set of relations, how would things change? In this case, I argue that the two limitations explained earlier will not longer be applicable, that is 1) events can be of varying durations, and 2) the relative ordering between intersecting events can be captured more accurately. This will give us timelines that are more encompassing and flexible.

Such an enhanced timeline may look like that presented in Figure 3.4. Events can be of varying durations (black boxes of different lengths), and the relative order between intersecting events can be captured in greater details (events do not need to be ordered based on starting times).

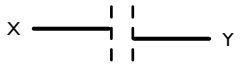
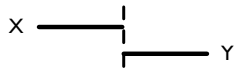
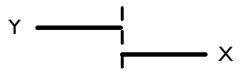
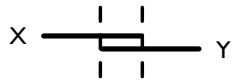
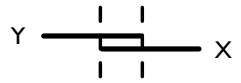
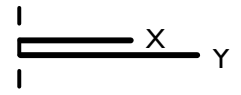
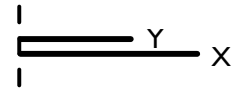
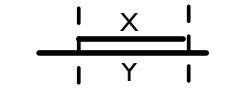
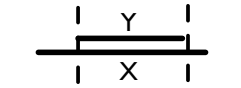
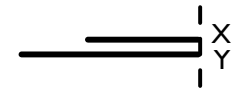
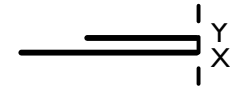
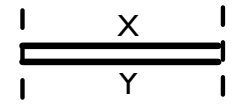
Relation	Illustration	Interpretation
X $<$ Y		X takes place before Y
X $>$ Y		
X m Y		X meets Y
X m^{-1} Y		
X o Y		X overlaps with Y
X o^{-1} Y		
X s Y		X starts Y
X s^{-1} Y		
X d Y		X during Y
X d^{-1} Y		
X f Y		X finishes Y
X f^{-1} Y		
X == Y		X is equal to Y

Table 3.2: The 13 temporal relationships described in Allen (1983), commonly known as Allen's relations. The superscript "−1" denotes an inverse function.

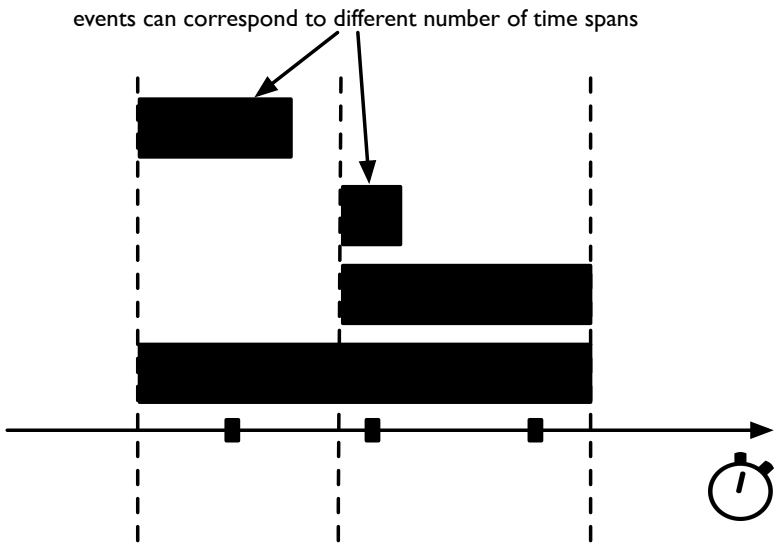


Figure 3.4: A possible enhanced timeline with events that can be of varying durations if more complex temporal relations are considered.

3.3.2 Inconsistencies in Temporal Relations

In Step 3 of the construction algorithm, an assumption that there are no conflicts in the underlying results from *E-T* and *E-E* temporal relationship classification is made. However this assumption may not always hold. The underlying automatic temporal classifiers are not perfect, and may give rise to conflicting temporal relations, such as the scenario shown in Figure 3.5. In the figure, event A is BEFORE events B and C, while both events B and C are BEFORE event Z. Using the laws of temporal transitivity (Setzer et al., 2003), event A should be BEFORE event Z. However an error in classifier output has event A happening AFTER event Z.

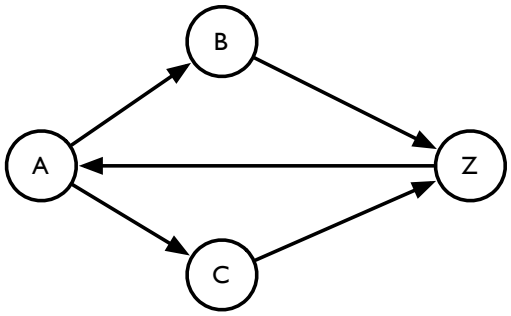


Figure 3.5: Temporal graph showing relations between events A, B, C and Z. A directed edge from event A to event B means that A takes place BEFORE B.

There are several ways to deal with this inconsistency. In the case of Algo-

rithm 3.1, I have chosen to ignore these inconsistencies. The resulting timelines may thus contain errors in them. However I argue that the underlying temporal classification systems are sufficiently robust (as will be seen later), and that errors do not render the timelines useless. Further, I also show in Chapter 6 that a reliability filtering metric can be computed. This metric can recognize timelines which are less error-prone from those which are more error-prone. With this we can make a decision to employ only timelines which are less error-prone.

An alternative approach is to attempt to correct conflicts during the construction process. Referring back to the example in Figure 3.5, we see that we have two paths from event A to event Z saying that A should be BEFORE Z, but only one path saying that A is AFTER Z. In this case we can choose to ignore the AFTER temporal relation between A and Z by adopting a majority voting scheme. This will allow us to construct a timeline with no inconsistent temporal relations. A caveat here is that this majority voting scheme is heuristic in nature. It does not guarantee the correctness of the temporal relations that are preserved in the timeline.

3.3.3 Timelines versus Temporal Graphs

Another useful issue to discuss is the choice of timelines as a logical representation for temporal information compared to full-fledged temporal graphs, as described in the TempEval series of workshops. Verhagen et al. (2009) explained that the various tasks of the TempEval workshops are designed so that they can be merged to obtain a temporal graph. In the workshops, the complex task of constructing a temporal graph is broken down into three steps:

1. Determining the temporal relationship between event and timex pairs
2. Determining the temporal relationship between two events
3. Determining the temporal relationship between an event and the document creation time (DCT)

Figure 3.6 is reproduced from Verhagen et al. (2010) to illustrate the merging process. Verhagen et al. however did not elaborate on the definition of a temporal graph. In the merged temporal graph, timexes are represented as square

vertices and events as circular vertices. The document creation time (DCT) is a complete timestamp, thus in the temporal graph it serves the role of a timex. Edges linking the vertices denote the existence of a temporal relationship between the vertices. These edges are likely typed based on the temporal relationship that exists.

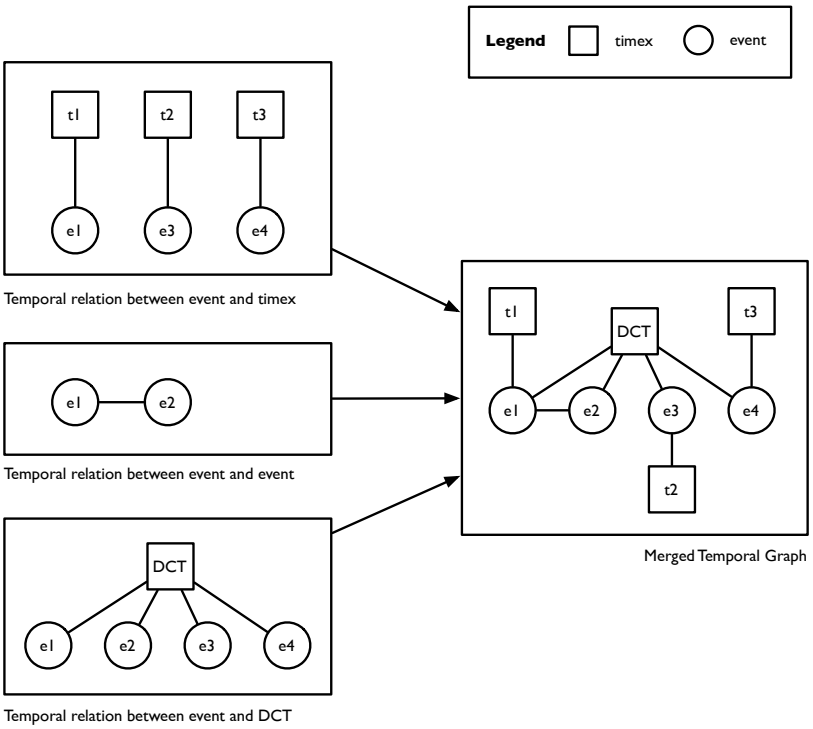


Figure 3.6: Merging of various temporal processing steps to get a temporal graph. The "DCT" node represents the "*document creation time*", *i.e.*, the time at which the document is created.

The main difference between such a temporal graph and a timeline is that the former is potentially more expressive. More information can be stored in a temporal graph than a timeline. For example, the duration of an event can be stored within each node in a temporal graph, but this is dropped in the timeline representation explained above. Also, temporal relations (*i.e.*, edges in the temporal graph) are stored as-is within a temporal graph. Some relations may be lost in a timeline representation due to the need to resolve conflicts between relations while mapping events to the time continuum.

This expressiveness comes at a cost however, as a temporal graph is potentially computationally complex. A temporal graph is essentially a hypergraph, with edges from timex vertices to event vertices forming hyperedges (the same

timex vertex can be in a same relationship with multiple event vertices). On the other hand, in a timeline, timexes are mapped onto a one-dimensional axis, reducing the dimensionality of nodes associated with it. Further, timelines are also easier to comprehend and have often been adopted in previous work (Denis and Muller, 2011; Do et al., 2012). It is for these reasons that I have decided to work with timelines in this thesis.

To end off this discussion, it is useful to note that both the temporal graph described in TempEval and my chosen timeline representation are not perfect solutions for the representation of temporal information. Hayes (1996) surveys and details several representations for times as points, intervals and even durations. In concluding, Hayes noted insightfully that there is no one temporal representation that can adequately cover all possible interpretations of time. Temporal graphs and timelines for example are unable to represent events with intermittent intervals (i.e., "every Friday night"). While a perfect representation might be possible, I posit that any adopted logical representation of time and events needs to take into consideration the application at hand.

Chapter 4

Event-Timex Temporal Classification

One of the steps in building a timeline involves classifying the temporal relationships between an event and a timex. This chapter explains my work in improving event-timex temporal relationship classification.

Classifying the temporal relationships between pairs of event and timexes is a critical step in building a timeline for a text. In this chapter, I describe my two-pronged approach (Ng and Kan, 2012) to tackle this problem:

1. Eschewing the use of traditional lexico-syntactic features in favour of more semantically motivated ones, and
2. Making use of crowdsourcing as a cost-effective, viable avenue to increase the amount of training data available to help train more effective classifiers.

Data Sparseness. A substantial body of work in this area is found in the TempEval series of evaluation workshops (Uzzaman et al., 2013; Verhagen et al., 2009, 2010). The top-performing teams have typically employed supervised machine learning systems, including support vector machines (SVM) (Bethard and Martin, 2007), conditional random fields (CRF) (Kolya et al., 2010) and Markov Logic Networks (MLN) (Ha et al., 2010; Uzzaman and Allen, 2010). These approaches make use of a variety of lexical and syntactic features which can be

summarized as: 1) lexical cues, such as signal words and part-of-speech tags; 2) context, including attributes of events and timexes, and; 3) the grammatical structure of sentences, obtained with the use of automatic parsers. Possibly since these approaches all use similar features, they achieve similar levels of performance of about 65% in their judgements.

The problem with this suite of features is that the feature types can take on many different values, and thus represent potentially a large feature space. Given the small size of training data available (*e.g.* the TempEval-2 dataset consists of only 959 instances), it is likely that the learned models suffer from data sparseness.

4.1 Reducing Dimensionality of Feature Space

The first approach to tackling this problem of data sparseness is to reduce the dimensionality of the input feature space. I propose to achieve this by dropping the use of typical lexico-syntactic features which as explained earlier is a cause for the sparseness.

Instead, I will make use of a semantically motivated approach, and adopt features derived from dependency parses of sentences. Typically sentences are composed following well-established grammar rules. It follows thus that constituents within sentences with similar grammatical structures share similar temporal relations. As an example, consider the two phrases in Example 4.1:

(1) ... left for Europe on Sunday ...
(2) ... went to America on Monday ... (4.1)

The constituent grammar parses of these two phrases are given in Figure 4.1. Note that they are constructed in a similar way with the same grammar rules. In phrase (1), the event "**left**" and the timex "**Sunday**" happened in the same time span, so we say that there is a OVERLAP temporal relation between them. Phrase (2) is similarly constructed, and the event "**went**" and the timex "**Monday**" are also related by the OVERLAP temporal relation.

Choice of Parse Trees. This observation suggests that the grammatical structure of text can be very useful in deciphering event-timex temporal relations.

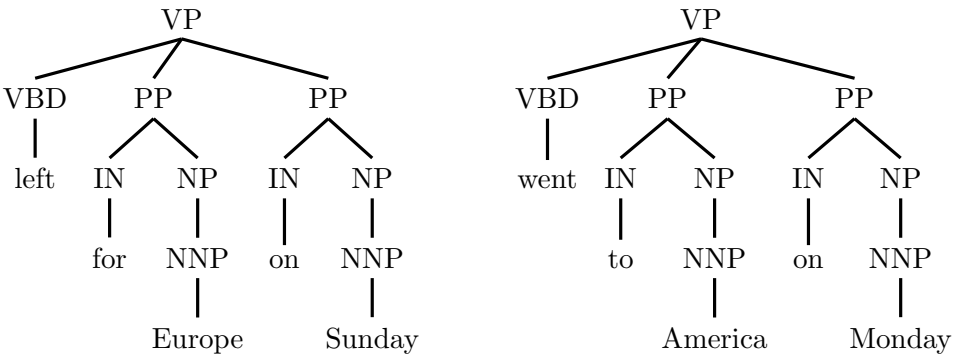


Figure 4.1: Phrases with similar grammatical structure. Note the similar temporal relations shared between events and timexes within each phrase.

There are two main types of grammar parses that are commonly in use: 1) constituent grammar parses, and 2) dependency parses. Previous work such as that of Mirroshandel et al. (2011) have centered around the use of the former. However, as constituent parses create internal phrasal nodes for every semantic constituent, such parse trees are often deep and overly detailed. Paths in such trees are fine-grained, capturing nuances (e.g., intervening finite verb phrase nodes), and as such may not generalize well when used to compute tree or path similarities.

To avoid the problems of constituent grammar parses, I study the use of dependency parses instead. Dependency parses are generally more compact than constituent grammar parses because they have no immediate phrasal nodes. This translates into a more compact feature space, which directly addresses data sparseness. As such, dependency parses should generally be more useful than constituent parses for this task.

Path Feature. I compute two features based on dependency parses to capture the grammatical structure of each event-timex pair. To help illustrate how these features are extracted, Figure 4.2 shows an extract of a dependency parse for this sentence fragment: ... *met with his friends early December* ....

1. **Dependency path from event to timex.** Starting from a full dependency parse of a sentence, I identify the vertices representing the event and timex. The shortest path from the timex vertex to the event vertex is located and used as a feature. Referring to Figure 4.2, the event "**met**" and the timex "**early December**" are bolded. The shortest path between

them includes the dependency relations "*root*" and "*tmod*". The feature value in this case is a path $\{tmod \rightarrow root\}$.

2. **Dependency path of timex.** Timexes can range from single word tokens to multi-word phrases, with vastly different semantics. For example, "*Friday*", "*last Friday*" and "*next Friday*" convey very different meanings. To capture this, I extract a sub-tree from the full dependency parse consisting of all the vertices and edges related to the time expression and use this as a feature. Referring again to Figure 4.2, the timex "**early December**" is a multi-word phrase. The feature value in this case is the path $\{amod \rightarrow tmod\}$.

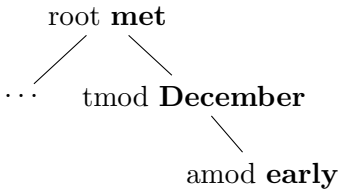


Figure 4.2: Extract of dependency parse to illustrate feature extraction.

Convolution Kernels. The next issue is how the similarity between two grammar structures can be computed. A typical approach to do this is to engineer flat representations of the structures through feature engineering. This process is time consuming and often requires good knowledge of the problem structure to decide which features are more discriminating than others.

Convolution kernels (Collins and Duffy, 2001) on the other hand model this form of structure similarity well. Convolution tree kernels take as input tree structures and calculate a degree of similarity between the two trees. In its simplest form, similarity is computed by recursively counting the number of identical sub-trees that appear in both input instances. With this structural similarity measure, we can do away with the need to "flatten" the structure with hand-devised representations. For these reasons, I use a support vector machine (SVM) for supervised classification, together with a convolution kernel as its kernel function (Moschitti, 2006b).

Pipelined System. Fitting these pieces together, a complete *E-T* temporal classification is obtained. Figure 4.3 illustrates the main stages that make up

the pipeline of the system. The input to the system will be a sentence which contains a pre-identified event and timex pair. Both the proposed path features are derived from a dependency parse of the input sentence. Therefore in the first stage I make use of the Stanford Parser (De Marneffe et al., 2006) to get a dependency parse of the input sentence. From the dependency parse, feature extraction is then carried out to derive the path features explained earlier. These features are then fed into three separate one-vs-all SVM classifiers, one for each temporal relation (i.e., BEFORE, AFTER, OVERLAP).

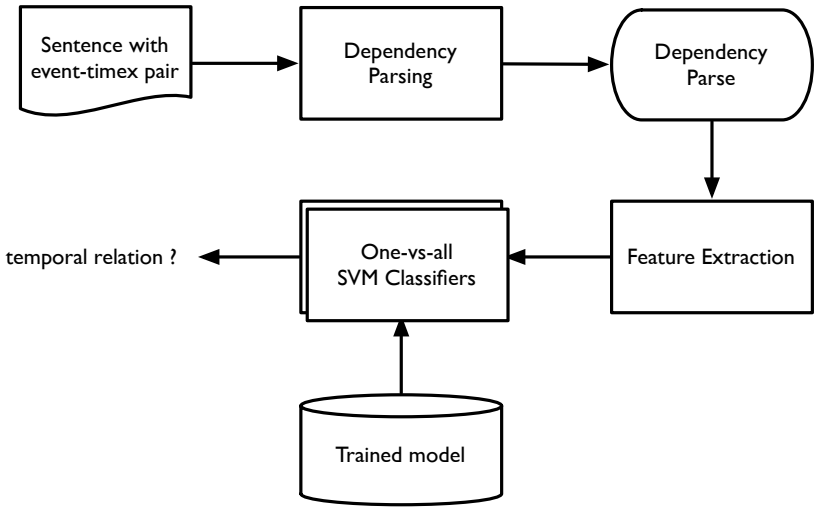


Figure 4.3: Overview of event-timex temporal relation classification system built with a SVM classifier using convolution kernels.

4.1.1 Experiments

I tested the proposed system against prior work on the TempEval-2 dataset. This dataset was assembled for TempEval-2, and comprises of 959 training instances together with 138 testing instances. Table 4.1 gives the performance of the classifier which I refer to as `SVMConvoDep` vis-à-vis the top performing systems in TempEval-2. The same *accuracy* metric used in the TempEval-2 task is adopted, which is the number of correct answers divided by the number of answers.

Macro-averaged precision, recall, and F_1 measures are also listed in the table. As will be explained later, the dataset has a skewed distribution of labels, with OVERLAP instances forming the majority. Micro-averaged measures give more weight to the most common label in such skewed datasets, but it is important for systems to be able to perform well across all the temporal labels. Thus,

macro-averaged measures are more appropriate than micro-averaged ones for this task.

From the results, the proposed classifier outperforms all previous temporal relation classifiers. While this performance gain is probably not statistically significant (as I have no access to the participating systems' individual judgments, it is impossible to check for statistical significance), these are impressive results as the classification input is decidedly simple (just two sub-trees derived from dependency parses as features).

System	**Accuracy (%)**	**Precision**	**Recall**	**F_1**
`SVMConvoDep`	67.4	0.828	0.512	0.633
TRIOS	65.0	Not Available		
JU_CSE	63.0			
NCSU-indi	63.0			
NCSU-joint	63.0			
TRIPS	63.0			
USFD2	63.0			

Table 4.1: Performance on TempEval-2 testing set. Results for TempEval-2 systems are cited from Verhagen et al. (2010).

4.2 Building Larger Data Sets

A second approach to alleviate the problem caused by data sparseness is to increase the amount of training data available for supervised machine learning. Developing a large, suitably annotated dataset is expensive — both in terms of time and monetary costs. Recent work in natural language processing suggests that crowdsourcing annotations from the untrained public can provide annotated data at similar annotation quality as expert annotators, but for a fraction of the cost (Hsueh et al., 2009; Sheng et al., 2008). It is useful thus to explore if the same findings apply to building a temporal dataset which is inherently complex (Setzer and Gaizauskas, 2000) and which requires a better understanding of the target language.

4.2.1 Task Setup

I set up a crowdsourcing task for this purpose in CrowdFlower[1]. CrowdFlower has access to a large user base (it uses Amazon Mechanical Turk to find workers), and adds an extra validation layer to attempt to address quality concerns as this has been an issue in many applications of crowdsourcing in natural language annotation tasks (Callison-Burch and Dredze, 2010; Mason and Watts, 2009).

Each annotation instance consists of a single sentence. I pre-processed the sentence to highlight one event expression and one timex found within it. Annotators were tasked to choose from five (`OVERLAP`, `AFTER`, `BEFORE`, `NOT-RELATED`, `BAD-SENTENCE`) possible temporal relationships between the marked event and timex. An additional choice for `BAD-SENTENCE` was included to allow annotators to indicate if there had been problems with the automatic pre-processing. Such instances (67 in this study) were discarded. The instructions provided to the annotators are shown here in Figure 4.4.

At least three judgments were requested from different annotators for each annotation instance. Majority voting was then used to decide on a final label for each annotation. To ensure the quality of the judgments that were obtained, I made use of a validation facility provided by CrowdFlower. Pre-annotated gold instances were mixed together with unlabeled instances. These gold instances were used to validate the annotations made by each annotator. Annotators were not informed which were the gold instances. During the annotation process, annotators who failed to label these gold instances correctly were stopped from proceeding with the task, and the annotations they made were discarded.

Raw Data. To ready a set of unlabeled instances for annotation, news articles on several news web sites, including Wall Street Journal, New York Times, CNN, and Channel News Asia, were crawled from 2 June to 8 July 2012. The sentence splitting module from the Apache OpenNLP[2] library was used to obtain individual sentences from these news articles.

Event and Timex Extraction. I built a CRF-based event and timex extractor (`CRFEventTimexExt`) which is able to automatically identify events and timexes

[1] `http://www.crowdflower.com`
[2] `http://opennlp.apache.org`

```
Within a sentence, there is an "event" marked out within
asterisks (**).  There is also a time expression marked out
within percentages (%%).  Give a judgement on the temporal (time)
relationship between the event and time expression.

Example 1:  I **told** Peter %%today%% that I visited Europe
last week.

In this example, "told" is the event, and "today" is the time
expression.  The event happens today (i.e.  I told Peter today),
so the relationship between them is OVERLAP.

Example 2:  I **told** Peter today that I visited Europe %%last
week%%.

In this example, "told" is the event, and "last week" is the
time expression.  I visited Europe last week but only told Peter
about it today, so the relationship between them is AFTER.

Valid judgements include:

a.  OVERLAP - event and time expression happen within same time
period
b.  AFTER - event happens after time expression
c.  BEFORE - event happens before time expression
d.  NOT-RELATED - there is no relationship between the event and
the time expression
e.  BAD-SENTENCE - the sentence is not processed correctly.  It
is not a complete sentence, or the event and time-expression is
not identified correctly.
```

Figure 4.4: Annotation instructions shown to CrowdFlower participants.

in each of the collected sentences. For event extraction, part-of-speech (POS) tags were the main features employed. For timex extraction, a compiled lexicon of the days in a week and months of a year was used on top of POS tags. The performance of the extractor is compared against top participating systems in TempEval-2 in Table 4.2. The results only included systems which are able extract both event and time expressions as it is not required for systems to be able to do both in TempEval-2.

System	Event			Timex		
	Precision	**Recall**	$\mathbf{F_1}$	**Precision**	**Recall**	$\mathbf{F_1}$
`CRFEventTimexExt`	0.81	0.82	0.82	0.76	0.62	0.68
Edinburgh	0.75	0.85	0.80	0.85	0.82	0.84
TIPSem	0.81	0.86	0.83	0.92	0.80	0.85
TRIPS	0.55	0.88	0.68	0.85	0.85	0.85
TRIOS	0.80	0.74	0.77	0.85	0.85	0.85

Table 4.2: Comparison of event and time expression identification with TempEval-2 systems. Performance of TempEval-2 systems are cited from Verhagen et al. (2010).

`CRFEventTimexExt` turns in competitive F_1 scores for event expression extraction, but less so for timex extraction due mainly to poor recall. The precision scores of 0.81 and 0.76 for event and time expression identification respectively are sufficiently high, considering it is used only for pre-processing data for annotation. Incorrectly identified events and timexes can be labeled as so by annotators, and these precision scores will minimize the occurrences of such mistakes. To address the problems with recall, more sentences were collected to increase the number of time expressions available for annotation.

4.2.2 Initial Annotations

I piloted a first set of annotations to investigate how a similarly sized crowdsourced corpus compares with the manual, expert annotations from TempEval-2. For this reason, I extracted an initial batch of 1,000 tuples (close in size to the TempEval-2's training data of 959 instances) of events and timexes from the sentences that had been crawled. Annotations for these 1,000 tuples were made and collected with the help of CrowdFlower. I will refer to this dataset as `d-1000`. The distribution of the temporal relationship labels within this collected set is

compared against that in the TempEval-2 training and testing sets in Table 4.3. The skewed distribution in the TempEval-2 datasets are similarly observed in the collected dataset, with the OVERLAP label taking up more than 50% of the whole dataset.

Dataset	Size	Distribution of label (%)			
		OVERLAP	**BEFORE**	**AFTER**	**Others**
TempEval-2 training set	959	53.8	18.0	23.0	5.2
TempEval-2 testing set	138	55.1	14.5	19.6	10.9
`d-1000`	1000	70.0	12.2	17.8	0.0

Table 4.3: Distribution of labels in different datasets.

A new classifier was trained using the same features as `SVMConvoDep` with the crowdsourced annotations `d-1000`. The performance of this trained classifier, `CF-1000`, on the TempEval-2 testing set is shown in Table 4.4. Macro-averaged precision, recall and F_1 measures are also included in the table.

System	Accuracy (%)	Precision	Recall	$\mathbf{F_1}$
`SVMConvoDep`	67.4	0.828	0.512	0.633
`CF-1000`	65.2	0.578	0.535	0.556
`CF-1000+TE`	71.7	0.726	0.598	0.656

Table 4.4: Classifier performance on TempEval-2 testing set.

`CF-1000` did not do as well as `SVMConvoDep` in terms of accuracy and F_1. This is not surprising for two reasons. First the annotators recruited for the annotation task are not domain experts, and it is fair to expect mistakes in the annotations obtained. Second, `SVMConvoDep` is trained on the TempEval-2 training set. This dataset is prepared in similar fashion together with the TempEval-2 testing set on which our evaluation is performed. It is fair to expect the two TempEval datasets to share more similar attributes and characteristics than the crowdsourced `d-1000` set. The content of the TempEval-2 datasets and `d-1000` span a different time period, are sourced from different sources, and possibly drawn from different domains and categories.

I tried combining the TempEval-2 training set with `d-1000`. With this combined dataset, I trained another classifier `CF-1000+TE`. The performance of this classifier is also reported in Table 4.4. The system improves results in both accuracy and F_1 scores. This improvement is significant, with $p < 0.05$ when tested

with the one-tailed paired Student's t-test.

There are two important conclusions that can be drawn from these results. First, the difference in performance between `SVMConvoDep` and `CF-1000` is slight. So while the the novices recruited via crowdsourcing may not have been domain experts, they are able to generate a dataset that is comparable to an expert-curated one. Then, the improvement to the performance of `CF-1000+TE` shows that despite the possible differences in time span, source, and categorization of `d-1000` from the TempEval-2 dataset as I have suggested, there is value to the crowdsourced dataset. Doubling the amount of training data available by putting `d-1000` and the TempEval-2 training set together rewards performance.

4.2.3 Selective Annotations

With crowdsourcing, the costs associated with generating temporal datasets are lowered. However it is important to do this efficiently with minimal wastage. Building on the earlier reported results, I will explain how we can selectively acquire annotations to further reduce the annotation costs involved without affecting the efficacy of the data collected.

This is best done through an illustrative example. It is often not easy to decide on the relationship between an event and timex. For example, let us take a look at Sentence 4.2[3] below:

> Two top aides to Netanyahu , political adviser Uzi Arad and Cabinet Secretary Danny Naveh, **left** for Europe on *Sunday* , apparently to **investigate** the Syrian issue , the newspaper **said**. (4.2)

Within the sentence there are several events. Let us focus on two of them: 1) "**left**", and 2) "**said**". One immediate observation is that timexes are commonly adjuncts (in this case, a prepositional phrase (PP)) that attach to a verb phrase (VP). This implies that timexes often directly modify only the head it is attached to. In such cases, it is usually easy to identify the temporal relationship between the event and timex. Looking at "**left**" and "**Sunday**", it is quite straightforward to determine that they take place within the same time span.

[3]Extracted from document APW19980301.0720 of the TempEval-2 dataset

However, it gets significantly more difficult when we look at "**said**" and "**Sunday**". We have to read through more of the sentence to build up an understanding of the relation between "**left**", "**investigate**", and "**said**" before we can conclude that "**said**" takes place after "**Sunday**".

The key insight to more efficient data acquisition is to leverage this observation — that some instances are both computationally and cognitively easier than others. The hypothesis is that less training data is needed for easier instances. Instead annotation effort should be focused on harder instances so that more of such training instances can be obtained. If there is a way to identify easier instances from harder ones, then annotators can just be tasked to handle the latter. This will make it more cost-effective when building the desired corpus.

Definitions. I define here a few terms that will be used in the rest of this section. Let the input be a collection $\mathbb{S}$ of sentences. Each sentence s is composed of one or more word tokens, i.e. $s = w_1 w_2 \ldots$. Let s^* be the set of all possible subsequences of s. For each s, I can define a set of unigram word tokens $\epsilon_s = \{w, w \in s\}$ that are events. I can also define a set $\Theta_s = \{\theta, \theta \in s^*\}$ which includes all the timexes within s. The problem then is to define some function $f : s, e, t \rightarrow R, s \in \mathbb{S}, e \in \epsilon_s, t \in \Theta_s$ where R is the temporal relationship between the event e and timex t.

Identifying Harder Instances for Annotation. I want to be able to partition a set of unlabeled instances so that the harder, more complex instances are separated from the easier ones. Such a partitioning scheme can be built around the ordering of the elements in ϵ_s given a timex $t \in \Theta_s$.

To do this, I first build a dependency parse[4] of the input sentence s, where each $w_i \in s$ forms a vertex within the dependency parse tree. A portion of the dependency parse tree for Sentence 4.2 is shown in Figure 4.5.

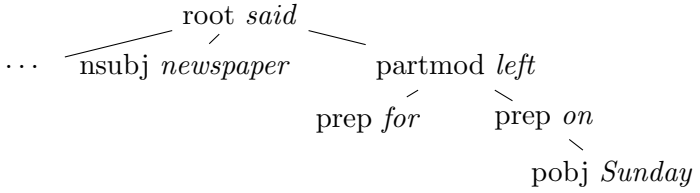


Figure 4.5: Excerpt of the dependency parse for Example 4.2.

[4]Dependency parse composed of Stanford dependencies (Klein and Manning, 2003).

In the figure, the governing dependency relation of a word is shown as a vertex. The associated word is also shown in *italics* to illustrate which part of the sentence this parse is about.

For a given dependency parse and a target time expression t, a total order O_t on ϵ_s can be defined. O_t is defined by arranging each $w \in \epsilon_s$ in ascending order of their respective distances from the timex vertex. Distance in this case is defined as the number of edges that needs to be traversed to reach the timex vertex.

Imposing the total order O_t on ϵ_s gives a totally ordered set, i.e. $(\mathrm{O}_t, \epsilon_s) = \{e_0, e_1, \ldots\}$. From this, for every input sentence s and its associated event (e) and timex (t), the tuple $< s, e, t >$ can be placed into a partition $\mathcal{P}_i$, where i is the index of e within $(\mathrm{O}_t, \epsilon_s)$.

Referring to Example 4.2 as an illustration, <"**left**", "**Sunday**"> will be placed in $\mathcal{P}_0$ because "**left**" is the nearest event expression to "**Sunday**" in the dependency parse in Figure 4.5. <"**said**","**Sunday**"> will be placed in $\mathcal{P}_1$ as "**said**" is the next nearest event expression. For convenience, I also refer to $\mathcal{P}_i$ as the set of Level-i instances.

This partitioning scheme is premised on the intuition that it requires more effort to understand the temporal relationship between events and timexes which are both structurally and semantically further away from each other. Higher level instances thus should be more complex than their lower level peers.

Following this scheme, I separated the TempEval-2 testing set into different partitions. A breakdown of the performance of `SVMConvoDep` on each of the partitions $\mathcal{P}_i$ is shown in Table 4.5. Accuracy drops steadily from Level-0 instances to Level-2 instances. This provides support for the intuition that higher level instances are harder to classify accurately.

Given that there are only 10 Level-3 instances and 1 Level-4 instance, the variance in measurements can potentially be very wide. There are too few Level-3 and Level-4 instances for their results to be analyzed reliably.

With the high prediction accuracy on Level-0 instances, I argue that it is not necessary to obtain more annotations for them. Instead the focus should be on the higher level instances. By opting not to annotate additional Level-0

Accuracy (%)				
Level-0 (59)	**Level-1 (47)**	**Level-2 (21)**	**Level-3 (10)**	**Level-4 (1)**
84.5	66.0	42.9	30.0	100.0

Table 4.5: Breakdown of performance of `SVMConvoDep` on partitions of TempEval-2 testing data. The number of instances for each partition is indicated in parentheses.

instances, substantial cost savings can be made as seen from Table 4.6. It shows a breakdown of the relative size of each partition to its entire dataset. `d-full` is a set of 8,851 tuples of events and timexes that were extracted from the same set of sentences I had crawled earlier. As seen from the table, Level-0 instances consistently form a large part of the various datasets. Not annotating Level-0 instances will directly lead to a cost savings of at least 37%.

Dataset	**Relative size of partition (%)**			
	Level-0	**Level-1**	**Level-2**	**Others**
TempEval-2 Training Set	40.9	35.2	15.1	8.8
TempEval-2 Testing Set	41.4	34.3	15.7	8.6
`d-full`	37.0	34.3	17.5	11.2

Table 4.6: Breakdown of partition sizes of different datasets.

Experiments. I collected annotations for all the tuples in `d-full` via CrowdFlower in a similar way to what was done earlier. From this `d-full` collection of 8,851 annotations, I removed all Level-0 instances to create a subset of 5,576 annotations which I will call `d-nolevel0`.

Using `d-full` and `d-nolevel0`, two new classifiers `CF-Full` and `CF-NoLevel0` are trained. Table 4.7 shows the performance of these two new classifiers with that of `SVMConvoDep` when tested with the TempEval-2 testing set.

From the results, it is seen that `CF-NoLevel0` is able to deliver a significant performance gain of about 8.6% over `SVMConvoDep` ($p < 0.05$) even though it only made use of part of the full data set that was collected. Further, it is able to match the performance of `CF-Full` which was trained over all collected instances (`d-full`). The performances of `CF-NoLevel0` and `CF-Full` are not significantly different.

These results are illuminating. The proposed partitioning scheme is able to reliably identify unlabeled instances that will not be able to contribute to better

System	Accuracy (%)	Precision	Recall	$\mathbf{F_1}$
SVMConvoDep	67.4	0.828	0.512	0.633
CF-NoLevel0	73.2	0.659	0.643	0.651
CF-Full	73.2	0.660	0.647	0.653

Table 4.7: Performance on TempEval-2 test set.

classifier performance. By focusing our annotation efforts solely on harder, more complex instances, data acquisition costs are cut by a large amount (37%) with no adverse impact on classifier performance.

4.2.4 Analysis and Discussion

It is useful to analyze the experimental results reported so far to get a better understanding of the performance of the temporal classifier, as well as identify possible directions for further research and development.

As a recap, Table 4.8 puts together the results obtained from the above reported experiments. Augmenting the original TempEval-2 training dataset with an initial batch of 1,000 annotated instances, CF-1000+TE gave a performance gain of about 6.4% in accuracy. By making use of the complete set of 8,851 crowdsourced annotations, a 8.6% improvement over SVMConvoDep is obtained by CF-Full. Interestingly, stripping away Level-0 instances from these crowdsourced annotations does not hurt the performance of CF-NoLevel0.

System	Accuracy (%)	Precision	Recall	$\mathbf{F_1}$
SVMConvoDep	67.4	0.828	0.512	0.633
CF-1000	65.2	0.578	0.535	0.556
CF-1000+TE	71.7	0.726	0.598	0.656
CF-NoLevel0	73.2	0.659	0.643	0.651
CF-Full	73.2	0.660	0.647	0.653

Table 4.8: Recap of the results achieved by the different E-T temporal classifiers introduced so far.

Selective Data Acquisition. Why does CF-NoLevel0 work as well as CF-Full despite the reduction in the amount of training data? Table 4.9 shows a more concise breakdown of the precision, recall and F_1 scores of both classifiers when tested with the TempEval-2 testing set. Both classifiers achieve similar performance across all three labels. Performance for the OVERLAP label is better, possibly because there are more training instances for it in the training dataset.

Classifier	OVERLAP			BEFORE			AFTER		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
`CF-NoLevel0`	0.72	0.96	0.82	0.56	0.45	0.50	0.70	0.52	0.60
`CF-Full`	0.72	0.95	0.81	0.57	0.40	0.47	0.70	0.60	0.64

Table 4.9: Performance measures on TempEval-2 testing set broken down by individual labels.

Table 4.10 illustrates the distribution of the AFTER and BEFORE labels in the first three partitions of the test data. The labels make up a larger part of the annotations at Level-1 and Level-2 than at Level-0.

Label	Distribution of labels (%)		
	Level-0	**Level-1**	**Level-2**
AFTER	10.1	21.2	23.6
BEFORE	5.1	13.7	16.1

Table 4.10: Distribution of labels in each partition.

Putting the pieces of the puzzle together, it is likely that the training instances in Level-0 are more useful for the OVERLAP label than the other two. `CF-NoLevel0` is already able to classify these instances quite well, so not having access to Level-0 training instances does not affect it adversely.

The performance of the classifiers broken down by each partition of the test data in Figure 4.6. As expected, without access to Level-0 training instances, `CF-NoLevel0` is slightly outperformed (but the difference is not statistically significant) by `CF-Full` for Level-0 test instances. Interestingly, the performance of `CF-NoLevel0` for Level-2 and Level-3 instances are also behind that for `CF-Full`. This suggests that having additional Level-0 training instances can have a positive effect on classifier performance for Level-2 and Level-3 test instances. This is worthy and interesting to re-visit in future work.

Common Errors. Another useful analysis to perform is to study how close the current results are to the theoretical upper-bound. Typically, such an upper-bound is derived from the inter-annotator agreement for the training set. The inter-annotator agreement for the annotations we have collected for `d-full` is 78.8%. Considering that the participants of the crowdsourcing exercise are not domain experts, the actual upper-bound could be higher. Nonetheless, this pessimistic value suggests that significant improvements can still be made to the

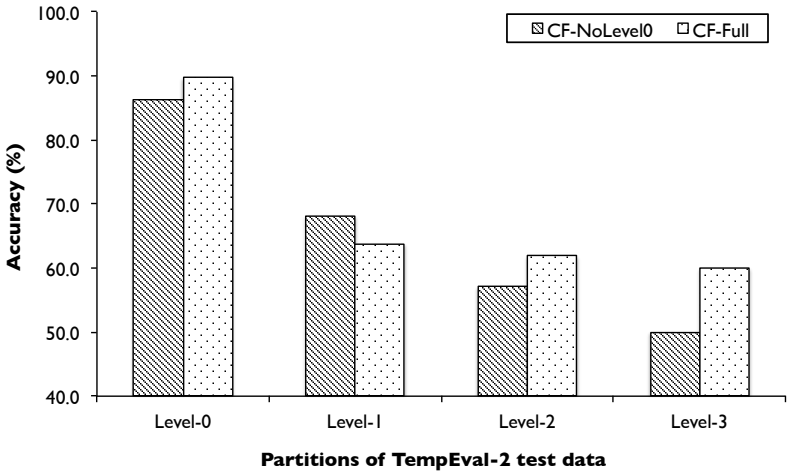


Figure 4.6: Breakdown of performance across different partitions.

Actual Label	Predicted label		
	OVERLAP	**BEFORE**	**AFTER**
OVERLAP	78	2	1
BEFORE	7	9	4
AFTER	13	0	14

Table 4.11: Confusion matrix for `CF-NoLevel0`.

existing results.

The best performing classifier presented in this work has an accuracy of 73.2%. To get some insight into what improvements could be made, the errors and mis-classifications of the current systems are studied. The confusion matrix for `CF-NoLevel0` is shown in Table 4.11. From the table, there are two large clusters of errors.

First, BEFORE and AFTER instances are often mis-classified as OVERLAP. Reflecting on this, I believe the skewed training dataset where OVERLAP instances form a majority is one of the reasons why.

Considering the lower recall scores we get for BEFORE and AFTER labels, the mis-classifications are likely a direct result of a lack of suitable training instances within the training dataset to better identify BEFORE and AFTER instances.

Looking closer, the performance on BEFORE instances is slightly lower than the performance for AFTER instances. This can be related to the smaller number of training instances available for the BEFORE label as seen from Table 4.10.

In future work it will be useful to verify if increasing the number of training

instances available can help to improve the recall of the classifier for instances of these two labels, thereby eradicating this cluster of errors.

The next major cause of mistakes is the misclassification of BEFORE instances as AFTER. One possible explanation is that the dependency parse features that were used did not consider modal or copular modifications to the event expressions. For example, take a look at Example 4.3[5]. The relationship between "**added**" and "**early November**" should have been BEFORE but was incorrectly classified as AFTER. The relevant portions of the dependency parse extracted as a feature for this instance is shown in Figure 4.7.

> He **added** that final guidelines to be published in **early November** will determine whether the bank is in compliance. (4.3)

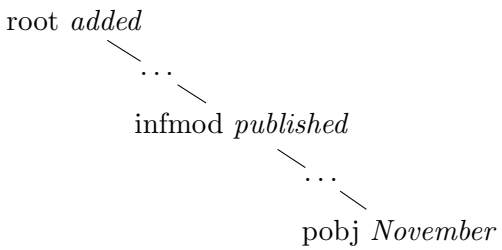


Figure 4.7: Dependency parse of Sentence 4.3.

The dependency parse feature that is extracted is the shortest path between the vertices for "**added**" and "**November**". The key to interpreting the temporal relationship in this example however is the copula modifier "*to be*" in front of "**published**".

Without capturing these modifiers, the sentence will appear to read as "He **added** that final guidelines published in **early November** will determine ...". In this reading, "**added**" takes place AFTER the time span indicated by "**early November**", which explains the mis-classification by the classifier. With this in mind, it can be useful to examine if including auxiliary modifiers of event expressions into the parse features used can help improve classifier performance.

[5]Extracted from document wsj_0527 of the TempEval-2 dataset.

4.3 Conclusion

The main problem which I believe is hampering better performance for *E-T* temporal relationship classification is a lack of sufficient training data. I adopt a two-pronged approach to tackle this problem: 1) by simplifying the feature space with the use of dependency parses as features to a convolution kernel support vector machine, and 2) by leveraging on crowdsourcing to get more data to support supervised machine learners.

I show that the classifier design I had adopted is competitive when pitted against classifiers which make use of far more complex mechanics and features. With this as a starting point, I went on to expand the training data that is available for use via crowdsourcing. Despite the complexity of the annotation task, novice annotators are able to generate a dataset that helps improve classifier performance significantly.

Building on my insight of the clausal structure inherent to event and time expressions, I suggest an effective way to selectively acquire annotations. The proposal reduces the amount of data to be annotated by up to 37% without sacrificing classifier performance. I achieved a classification accuracy of 73.2%, which represents a 8.6% improvement over a very competitive baseline.

Chapter 5

Event-Event Temporal Classification

In this chapter I will explain how discourse analysis can be useful in solving the event-event temporal relationship classification problem.

Another piece of the puzzle towards obtaining a timeline for a piece of text is to classify the temporal relationships between pairs of events. As explained in Chapter 1, to solve possible coverage problems caused by intra-sentence *E-E* classification, I choose to work on an article-wide variant of the problem, i.e. I am seeking to determine the temporal relationship between two events found *anywhere* within a piece of text. This chapter details my novel approach (Ng et al., 2013) which leverages on discourse analysis to solve the problem.

5.1 Making Use Of Discourse

Chapter 2 explained that the state-of-the-art for intra-sentence *E-E* temporal relationship classification focused largely on the use of lexico-syntactic surface features. In a closely related piece of work, Do et al. (2012) studied the problem of article-wide *E-E* temporal classification as part of a joint inference scheme with *E-T* temporal classification. However they have similarly made use of popular surface features.

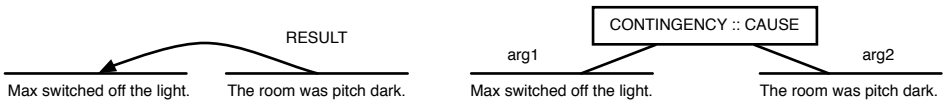


Figure 5.1: RST and PDTB discourse structures for sentence [B] in Example 5.1. The structure on the left is the RST discourse structure, while the structure on the right is for PDTB.

To highlight the deficiencies of surface features, this is an example from Lascarides and Asher (1993):

[A] Max opened the door. The room was pitch dark.
[B] Max switched off the light. The room was pitch dark. (5.1)

The two sentences [A] and [B] in Example 5.1 have similar syntactic structure. Given only syntactic features, we might conclude that they share similar temporal relationships. However in [A], the events temporally OVERLAP, while in [B] they do not. Clearly, syntax alone is not going to be useful to help us arrive at the correct temporal relations.

If existing surface features are insufficient, what is sufficient? Given a *E-E* pair which crosses sentence boundaries, how can we determine the temporal relationship between them? Informed by the work of Lascarides and Asher (1993), I postulate that discourse relations hold the key to interpreting such temporal relationships. I make use of a suite of discourse analysis studies, including 1) the Rhetorical Structure Theory (RST) discourse framework, 2) Penn Discourse Treebank (PDTB)-styled discourse relations based on the lexicalized Tree Adjoining Grammar for Discourse (D-LTAG), and 3) topical text segmentation, and validate their effectiveness for *E-E* temporal classification.

RST Discourse Framework. RST (Mann and Thompson, 1988) is a well-studied discourse analysis framework. In RST, a piece of text is split into a sequence of non-overlapping text fragments known as elementary discourse units (EDUs). Neighbouring EDUs are related to each other by a typed relation. Most RST relations are *hypotactic*, where one of the two EDUs participating in the relationship is demarcated as a *nucleus*, and the other a *satellite*. The nucleus holds more importance, from the point of view of the writer, while the satellite's purpose is to provide more information to help with the understanding of the

nucleus. Some RST relations are however *paratactic*, where the two participating EDUs are both marked as nuclei. A discourse tree can be composed by viewing each EDU as a leaf node. Nodes in the discourse tree are linked to one another via the discourse relations that hold between the EDUs.

RST discourse relations capture the semantic relation between two EDUs, and these often offer a clue to the temporal relationship between events in the two EDUs too. As an example, let us refer once again to Example 5.1. Recall that in the second line of text "**switched off**" happens BEFORE "**dark**". The RST discourse structure for the second line of text is shown on the left of Figure 5.1. We see that the two sentences are related via a "*Result*" discourse relation. This is in-line with our intuition: when there is causation, there should be a BEFORE/AFTER relationship. The RST discourse relation in this case is very useful in helping us determine the relationship between the two events.

PDTB-styled Discourse Relations. Another widely adopted discourse relation annotation is the PDTB framework (Prasad et al., 2008). Unlike the RST framework, the discourse relations in PDTB build on the work on D-LTAG by Webber (2004), a lexicon-grounded approach to discourse analysis. Practically, this means that instead of starting from a pre-identified set of discourse relations, PDTB-styled annotations are more focused on detecting possible connectives (can be either explicit or implicit) within the text, before identifying the text fragments which they connect, and how they are related to one another.

Applied again to the second line of text we have in Example 5.1, we get a structure as shown on the right side of Figure 5.1. From the figure it can be seen that the two sentences are related via a "*Cause*" relationship. Similar to what was explained earlier for the case of RST, the presence of a causal effect here strongly hints that events in the two sentences share a BEFORE/AFTER relationship.

At this point, it is worthwhile to note the differences between the use of the RST framework and PDTB-styled discourse relations in the context of this work. The theoretical underpinnings behind these two discourse analysis are very different, and they can be complementary to each other. First, the RST framework breaks up text within an article linearly into non-overlapping EDUs. Relations

can only be defined between neighboring EDUs. However this constraint is not found in PDTB-styled relations, where a text fragment can participate in one discourse relation, and a subsequence of it participate in another. PDTB relations are also not restricted only to adjacent text fragments. In this aspect, the flexibility of the PDTB relations can complement the seemingly more rigid RST framework.

Second, with PDTB-styled relations not every sentence needs to be in a relation with another as the PDTB framework does not aim to build a global discourse tree that covers all sentence pairs. This is a problem for a article-wide analysis. The RST framework does not suffer from this limitation however as it is possible to build up a discourse tree connecting all the text within a given article.

Topical Text Segmentation. A third, complementary type of inter-sentential analysis is topical text segmentation. This form of segmentation separates a piece of text into non-overlapping segments, each of which can span several sentences. Each segment represents passages or topics, and provides a coarse-grained study of the linear structure of the text (Hearst, 1994; Skorochod'Ko, 1972). The transition between segments can represent possible topic shifts which can provide useful information about temporal relationships.

> (The Davao Medical Center, a regional government hospital, recorded 19 deaths with 50 wounded. Medical evacuation workers however said the injured list was around 114, spread out at various hospitals.)$_1$
> (A powerful bomb tore through a waiting shed at the Davao City international airport at about 5.15 pm (0915 GMT) while another explosion hit a bus terminal at the city.)$_2$ (5.2)

In Example 5.2[1], the lines of text have been delimited into segments with parentheses along with a subscript. Segment (1) talks about the casualty numbers seen at a medical centre, while Segment (2) provides background information that informs us a bomb explosion had taken place. The segment boundary hints at a possible temporal shift and can help us to infer that the bombing event took place BEFORE the deaths and injuries had occurred.

[1]From article AFP_ENG_20030304.0250 of the ACE 2005 corpus.

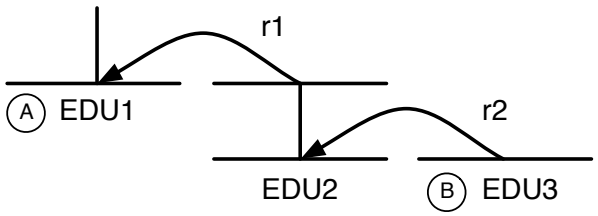


Figure 5.2: A possible RST discourse tree. The two circles denote the two relevant events A and B.

5.2 Methodology

Having motivated the use of discourse analysis, I will now explain how we can make use of them for temporal classification. The different facets of discourse analysis that are being explored in this work are structural in nature. RST and PDTB discourse relations are commonly represented as graphs, and the output of text segmentation can also be viewed as a graph with individual text segments forming vertices, and the transitions between them forming edges.

Considering this, as was done earlier for E-T temporal classification where the features adopted are similarly structural in nature, a good approach would be to similarly use a support vector machine (SVM) classifier with a convolution kernel (Collins and Duffy, 2001) for its kernel function (Moschitti, 2006a; Vapnik, 1999). As noted in Chapter 4, the use of convolution kernels makes it possible to do away with the extensive feature engineering typically required to generate flat vectorized representations of features. This process is time consuming and demands specialized knowledge to achieve representations that are discriminating, yet are sufficiently generalized.

RST Discourse Framework. Recall that the RST framework results in a discourse tree for an entire input article. In recent years several automatic RST discourse parsers have been made available. In our work, we first make use of the parser by Feng and Hirst (2012) to obtain a discourse tree representation of our input. To represent the meaningful portion of the resultant tree, I make use of path information between the two sentences of interest.

Figure 5.2 illustrates an example discourse tree. EDUs including $EDU1$ to $EDU3$ form the vertices while discourse relations $r1$ and $r2$ between the EDUs

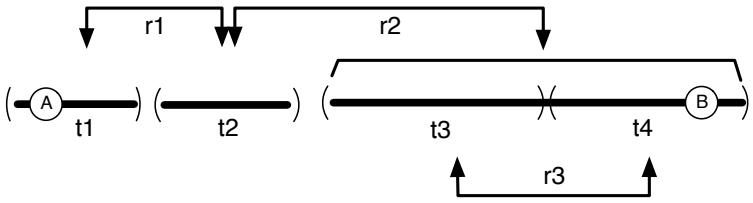


Figure 5.3: A possible PDTB-styled discourse annotation where the circles represent the events of interest.

form the edges. For an *E-E* pair, $\{A, B\}$, I obtain a feature structure by first locating the EDUs within which A and B are found. A is found inside $EDU1$ and B is found within $EDU3$. The shortest path between $EDU1$ and $EDU3$ will be the feature structure for the *E-E* pair, *i.e.* $\{r1 \rightarrow r2\}$.

PDTB-styled Discourse Relations. I make use of the automatic PDTB discourse parser from Lin et al. (2013) to obtain the discourse relations over an input article. Similar to how the feature for the RST discourse framework is built, for a given *E-E* pair, I retrieve the relevant text fragments and use the shortest path linking the two events as a feature structure.

An example of a possible PDTB-styled discourse annotation is shown in Figure 5.3. The horizontal lines represent different sentences in an article. The parentheses delimit text fragments, $t1$ to $t4$, which have been identified as arguments participating in discourse relations, $r1$ to $r3$. For a given *E-E* pair $\{A, B\}$, the shortest path between them *i.e.* $\{r1 \rightarrow r2\}$ is used as a feature structure.

There is a need to take special care to regularize the input (as, unlike EDUs in RST, arguments to different PDTB relations may overlap, as in $r2$ and $r3$). This is done by modeling each PDTB discourse annotation as a graph before employing Dijkstra's shortest path algorithm. The graph resulting from the annotation in Figure 5.3 is given in Figure 5.4. Each text fragment t_i maps to a vertex n_i in the graph. PDTB relations between text fragments form edges between corresponding vertices. As $r2$ relates $t2$ to both $t3$ and $t4$, two edges link up $n2$ to the corresponding vertices $n3$ and $n4$ respectively. By doing this, Dijkstra's algorithm will always find the desired shortest path.

Topical Text Segmentation. Taking as input a complete text article, I make use of the state-of-the-art text segmentation system from Kazantseva and Sz-

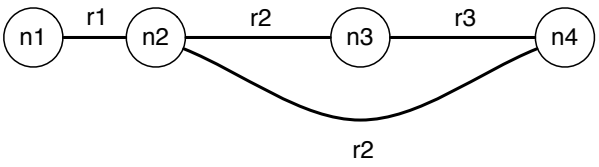


Figure 5.4: Graph derived from discourse annotation in Figure 5.3.

pakowicz (2011). The output of the system is a series of non-overlapping, linear text segments. To refer to these segments, I number them sequentially in the order they occur.

In Figure 5.5 the horizontal lines represent sentences $s1$ to $s4$. Parentheses with subscripts mark out the segment boundaries. We can see two segments *seg1* and *seg2* here. Given a target *E-E* pair $\{A, B\}$ (represented as circles inside the figure), the segment number of the corresponding segment in which each of A and B is found is identified. A feature structure is built with the identified segment numbers, *i.e.* $\{seg1 \rightarrow seg2\}$ to capture the segmentation. The directionality of the feature denotes the sequential nature of linear text segmentation.

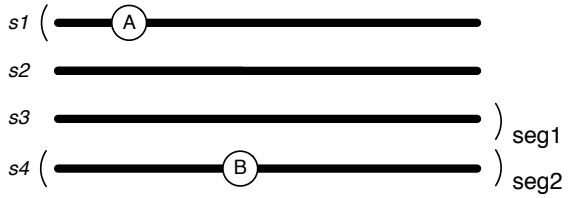


Figure 5.5: A possible segmentation of four sentences into two segments.

5.3 Experiments and Results

5.3.1 Dataset

In the previous chapter, the TempEval-2 dataset was used for my experiments and analysis. While the TempEval-2 dataset also contains annotations for *E-E* temporal relation classification, these are restricted to just intra-sentence event pairs, or event pairs found in adjacent sentences. It is thus not suitable here for the evaluation of article-wide *E-E* classification.

Instead I make use of the dataset built by Do et al. (2012). As part of their

work covers article-wide *E-E* classification too, making use of the same dataset also allows for comparative evaluation.

The dataset consists of 20 newswire articles which originate from the ACE 2005 corpus (ACE, 2005). Initially, the dataset consists of 324 event mentions, and a total of 375 annotated *E-E* pairs. The same temporal saturation step as described in Do et al. (2012) is performed, and a total of 7,994 *E-E* pairs[2] are obtained. Human annotation is costly and it is not easy to obtain large numbers of *E-E* temporal annotations. The temporal saturation step allows us to make up for this by leveraging on the transitivity properties of temporal relations to generate new annotations that are inferred from existing ones.

A breakdown of the number of instances by each temporal class is shown in Table 5.1. Unlike earlier data sets such as that for TempEval-2 where more than half (about 55%) of test instances belong to the OVERLAP class, OVERLAP instances make up just 10% of the data set.

This difference is due mainly to the fact that the dataset consists not only of intra-sentence *E-E* pairs, but also of article-wide *E-E* pairs. Figure 5.6 shows the number of instances for each temporal class broken down by the number of sentences (*i.e.* sentence gap) that separate the events within each *E-E* pair. We see that as the sentence gap increases, the proportion of OVERLAP instances decreases. The intuitive explanation for this is that when event mentions are very far apart in an article, it becomes more unlikely that they happen within the same time span.

Class	AFTER	BEFORE	OVERLAP
# *E-E* pairs	3,588 (45%)	3,589 (45%)	815 (10%)

Table 5.1: Number of event pairs in data set attributable to each temporal class. Percentages shown in parentheses.

[2]Though the data set was obtained from the original authors, there was a discrepancy in the number of *E-E* pairs. The original paper reported a total of 376 annotated *E-E* pairs. Besides this, I also repeated the saturation steps iteratively until no new relationship pairs are generated. This is an enhancement as it ensures that all inferred temporal relationships are generated.

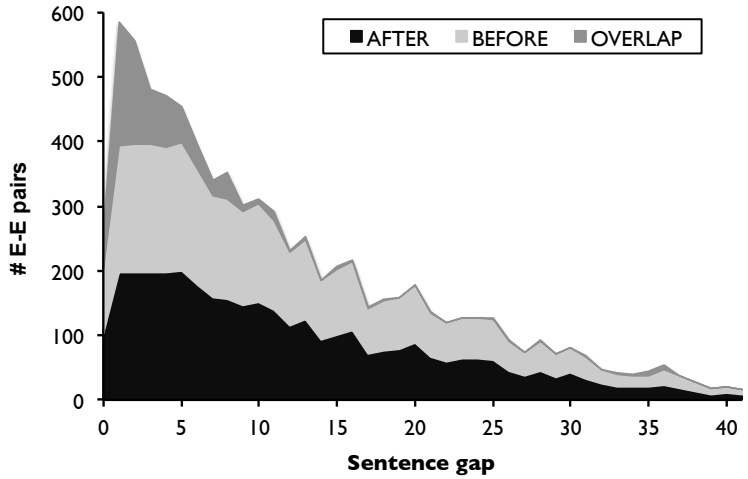


Figure 5.6: Breakdown of number of event pairs for each temporal class based on sentence gap.

	System	Precision	Recall	F_1
(1)	Do2012	43.86	52.65	47.46
(2)	Base	59.55	38.14	46.50
(3)	Base + Rst + Pdtb + TopicSeg	71.89	41.99	53.01
(4)	Base + Rst + Pdtb + TopicSeg + Coref	75.23	43.58	55.19
(5)	Base + O-Rst + Pdtb + O-TopicSeg + O-Coref	78.35	54.24	64.10

Table 5.2: Macro-averaged results obtained from our experiments. The difference in F_1 scores between each successive row is statistically significant, but a comparison is not possible between Rows 1 and 2.

5.3.2 Experiments

The work done in Do et al. (2012) is highly related to my experiments, and so the relevant results for local *E-E* classification are reported in Row 1 of Table 5.2 as a reference. While largely comparable, note that a direct comparison is not possible because 1) the number of *E-E* instances I have is slightly different from what was reported, and 2) I do not have access to the exact partitions they have created for 5 fold cross-validation.

Instead I have implemented a baseline adopting similar surface lexico-syntactic features used in previous work (Bethard and Martin, 2007; Do et al., 2012; Mani et al., 2006; Ng and Kan, 2012), including 1) part-of-speech tags, 2) tenses, 3) dependency parses, 4) relative position of events in article, 5) the number of sentences between the target events and 6) VerbOcean (Chklovski and Pantel, 2004) relations between events. This baseline system, and the subsequent systems described here, comprises of three separate one-vs-all classifiers for each of

the temporal classes. The result obtained by the baseline system is shown in Row 2 (*i.e.* BASE) in Table 5.2. The results show that this baseline is competitive and performs similarly the system described by Do et al. (2012) in terms of F_1. However a test for statistical significance between the results is not possible without access to the raw judgments from Do's system.

I also implemented the proposed discourse-based features and show the results obtained in the remaining rows of Table 5.2. In Row 3, RST denotes the RST discourse feature, PDTB denotes the PDTB-styled discourse features, and TOPICSEG denotes the text segmentation feature. Compared to the baseline system in Row 2, there is a relative increase of 14% in F_1, which is statistically significant when verified with the one-tailed paired Student's t-test ($p < 0.01$).

Event Co-reference. In addition, Do et al. (2012) have shown the value of event co-reference. Event co-reference refers to the detection of references to the same event in text. Intuitively, this can help temporal classification because it is fair to expect that references to the same event are going to occur in the same time span (i.e., they OVERLAP one another). Therefore this feature has also been included by making use of an automatic event co-reference system by Chen et al. (2011). The result obtained after adding this feature (denoted by COREF) is shown in Row 4. The relative increase in F_1 of about 4% from Row 3 is statistically significant ($p < 0.01$) and affirms that event co-reference is a useful feature to have, together with our proposed features. The complete system in Row 4 gives a 16% improvement in F_1, relative to the reference system DO2012 in Row 1.

Oracular Discourse Features. To get a better idea of the performance that can be obtained if oracular versions of the discourse features are available, the table also shows the results obtained if hand-annotated RST discourse structures, text segments, as well as event co-reference information were used. Annotations for the RST discourse structures and text segments were performed by me (RST annotations were made following the annotation guidelines given by Carlson and Marcu (2001)). Oracular event co-reference information was already included in the dataset that was used.

In Row 5 the prefix O denotes oracular versions of the discourse features.

From the results we see that there is a marked increase of over 15% in F_1 relative to Row 4. Compared to Do's state-of-the-art system, there is also a relative gain of at least 35%. These oracular results further confirm the importance of non-local discourse analysis for temporal processing.

Ablation test. An ablation test is additionally performed to help affirm the efficacy of the proposed discourse features. Starting from the full system, each discourse feature was dropped in turn to see the effect this has on overall system performance. This test is performed over the same data set, again with 5 fold cross-validation. The results in Table 5.3 show a statistically significant (based on the one-tailed paired Student's t-test) drop in F_1 in each case, which proves that each of the proposed features is useful and required.

From the ablation tests, it is also observed that the RST discourse feature contributes the most to overall system performance while the PDTB discourse feature contributes the least. However it is premature to conclude that the former is more useful than the latter; as the results are obtained using parses from automatic systems, and are not reflective of the full utility of ground truth discourse annotations.

Ablated Feature	**Change in $\mathbf{F_1}$**	**Sig**
−RST	-9.03	**
−TOPICSEG	-2.98	**
−COREF	-2.18	**
−PDTB	-1.42	*

Table 5.3: Ablation test results. '**' and '*' denote statistically significant differences against the full system with $p < 0.01$ and $p < 0.05$, respectively.

5.4 Discussion

Useful Relations. The ablation test results reveal that discourse relations (in particular, RST discourse relations) are the most important in our system. Earlier in this chapter I had also motivated my approach with the intuition that certain relations such as the RST "*Result*" and the PDTB "*Cause*" relations provide very useful temporal cues. Let us now study the effectiveness of using these discourse relations.

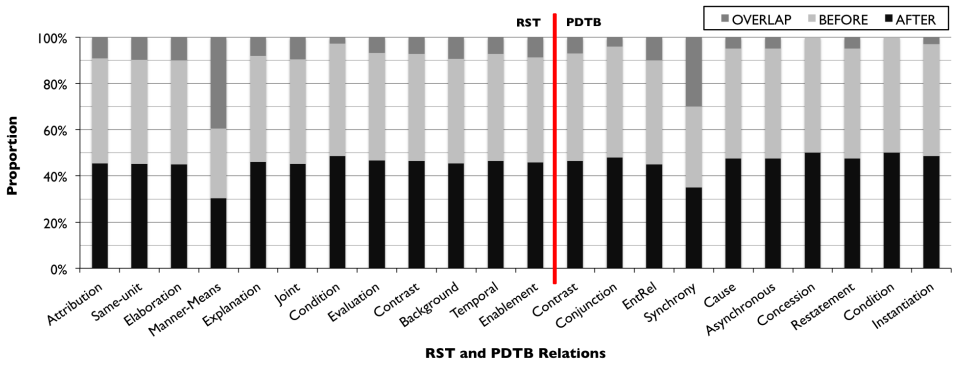


Figure 5.7: Proportion of occurrence in temporal classes for every RST and PDTB relation.

Figure 5.7 illustrates the relative proportion of temporal classes in which each RST and PDTB relation appear. If the relations are randomly distributed, one would expect their distribution to follow that of the temporal classes as shown in Table 5.1. However it can be seen that many of the relations do not follow this distribution. For example, several relations such as the RST "*Condition*" and PDTB "*Cause*" relations are almost exclusively found within AFTER and BEFORE event pairs only, while the RST "*Manner-means*" and PDTB "*Synchrony*" relations occur in a disproportionately large number of OVERLAP event pairs. These relations are likely useful in disambiguating between the different temporal classes.

I further examine the convolution tree fragments that lie on the support vector of our SVM classifier. The work of Pighin and Moschitti (2010) in linearizing kernel functions makes it possible to take a look at these tree fragments. Applying the linearization process leads to a different classifier from the one used in the earlier experiments. The identified tree fragments are therefore just an approximation to the actual tree fragments. However, this analysis still offers an introspection as to what relations are most influential for classification.

Table 5.4 shows a subset of the top RST discourse fragments identified for the BEFORE and OVERLAP one-vs-all classifiers. The list is in line with what can be expected from Figure 5.7. The former consists of fragments containing relations such as "*Temporal*" and "*Condition*", while the latter has a sole fragment containing "*Manner-Means*".

To illustrate what these fragments may mean, Example 5.3 shows several

BEFORE		OVERLAP	
BEF1	(Temporal ...	OLP1	(Manner-means ...
BEF2	(Temporal (Elaboration ...		
BEF3	(Condition (Explanation ...		
BEF4	(Condition (Attribution ...		
BEF5	(Elaboration (Background ...		

Table 5.4: Subset of top RST discourse fragments on support vectors identified by linearizing kernel function.

example sentences extracted from the experimental dataset. The corresponding discourse structures for both sentences [A] and [B] are also illustrated in the top and bottom half of Figure 5.8 respectively. The discourse structure for sentence [A] consists of the tree fragment *BEF1* (i.e., "*(Temporal...*"). This fragment indicates (correctly) that the event "**wielded**" happened BEFORE Milosevic was "**swept out**" of power. It is also seen that the discourse structure for sentence [B] consists of the tree fragment *OLP1* (i.e., "*(Manner-means...*"). As with the previous example, the fragment suggests (correctly) that there should be a OVERLAP relationship for the "**requested** – **said**" event pair.

From these analysis, it can be seen that 1) some discourse relationships are indeed salient indicators of temporal relationships, and 2) the SVM classifiers are able to identify the discourse relationships that matter.

> [A] Milosevic and his wife **wielded** enormous power in Yugoslavia for more than a decade before he was **swept out** of power after a popular revolt in October 2000.
>
> [B] The court order was **requested** by Jack Welch's attorney, Daniel K. Webb, who **said** Welch would likely be asked about his business dealings, his health and entries in his personal diary.

(5.3)

Segment Numbers. From the ablation test results, text segmentation is the next most important feature after the RST discourse feature. Example 5.4 illustrates an instance where text segmentation is very helpful in disambiguating between temporal relations. The example shows an extract[3] from the experimental dataset, as well as the actual segmentation produced by the text segmentation system. The two sentences are grouped into two different segments (i.e., *SEG1* and *SEG2*). These segment numbers are marked at the beginning of each sen-

[3]Drawn from article AFP_ENG_20030319.0879.

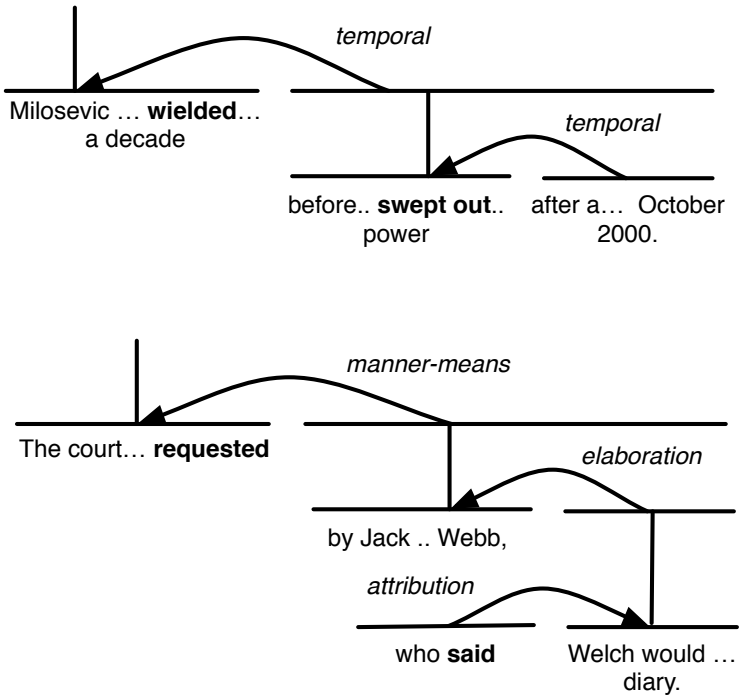


Figure 5.8: RST discourse structures for sentences [A] (top half) and [B] (bottom half) in Example 5.3.

tence, and I will refer to the sentences with the segment numbers directly for brevity. Note the two events which are bolded inside *SEG1*. They share an OVERLAP relationship. The same can be said of the two events bolded inside *SEG2*. However if we look at the event "**reviewing**" from *SEG1*, and the event "**sale**" from *SEG2*, the sale actually actually took place some time ago. There is a AFTER relationship between the "**reviewing** – **sale**" event pair. The segment boundary (and lack of) between *SEG1* and *SEG2* is clearly effective in demarcating the different temporal relations between these sets of events.

> [SEG1] "Now that Vivendi Universal has begun a formal process in **reviewing** options for its entertainment assets, it is appropriate to **step** aside from any direct management responsibility.".
>
> [SEG2] As part of the 11-billion-dollar **sale** of USA Interactive's film and television operations to the French media company in December 2001, USA Interactive **received** 2.5 billion dollars in preferred shares in Vivendi Universal Entertainment. (5.4)

As described earlier, the feature design makes use of actual segment numbers. Intuitively, one might expect that this may be detrimental to performance. The

learnt structures may not generalize well, especially if articles are of different lengths, as each article may have vastly different number of segments. The transition across segments may also not carry the same semantic significance for different articles.

Despite this however the feature has been shown to be helpful. This is due possibly to two reasons. First, the default settings of the text segmentation system that was used prefer precision over recall (Kazantseva and Szpakowicz, 2011, p. 292). There is thus just an average of between two to three identified segments per article. Second, the style of writing in newswire articles in the experimental dataset generally follows common journalistic guidelines. The semantics behind the transitions across the coarse-grained segments that were identified are thus likely to be of a similar nature across many different articles.

Error Analysis. Besides examining the discourse features that were used, it is also instructive to get a better idea of the errors made by the system. Recall that there are separate one-vs-all classifiers for each of the temporal classes, so each of the three classifiers generates a column in the aggregate confusion matrix shown in Table 5.5. In cases where none of the SVM classifiers return a positive confidence value, no temporal class (captured as column **NONE**) is assigned. The high number of event pairs which are not assigned to any temporal class explains the lower recall scores that were observed earlier in Table 5.2.

Actual	**Predicted**			
	OVERLAP	**BEFORE**	**AFTER**	**NONE**
OVERLAP	119	114	104	474
BEFORE	19	2067	554	928
AFTER	16	559	2046	947

Table 5.5: Confusion matrix obtained for the full system.

Additionally, an interesting observation is the low percentage of OVERLAP instances that the classifier managed to predict correctly. About 57% of BEFORE and AFTER instances are classified correctly (i.e., 2067 and 2046 out of 3568 respectively), however only about 15% (119 out of 811) of OVERLAP instances are correct.

Figure 5.9 offers more evidence to suggest that the classifier works better for the BEFORE and AFTER classes than the OVERLAP class. We see that

as sentence gap increases, the classifier achieves a fairly consistent performance for both BEFORE and AFTER instances. It does not fare well for OVERLAP instances however, with the best accuracy figure coming in below 30%. Although not definitive, this may be because the dataset that is used consists of much fewer OVERLAP instances than the other two classes. This bias may have led to insufficient training data for accurate OVERLAP classification. It will be useful in future work to investigate if using a more balanced data set for training can help overcome this problem.

We note that the under-performance for OVERLAP instances does not have a significant negative impact on overall system performance, as might be suspected. This is because there are much fewer OVERLAP instances than BEFORE and AFTER instances, and they are found mainly where sentence gap is less than 7. The 0% accuracy figures for the right end of the graph for OVERLAP instances are thus not un-expected.

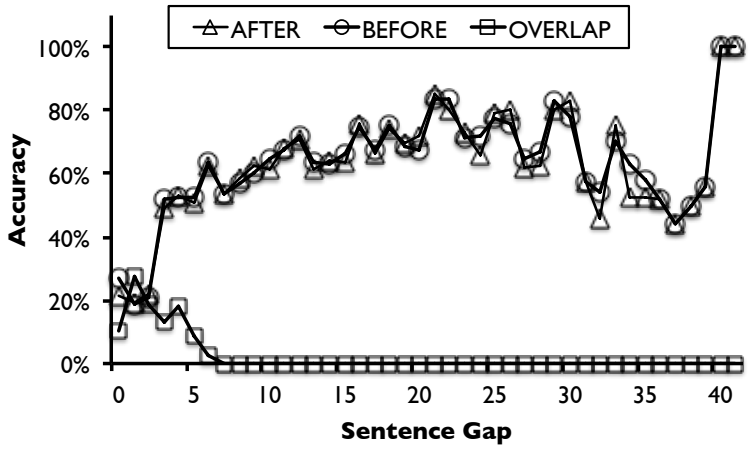


Figure 5.9: Accuracy of the classifier for each temporal class, plotted against the sentence gap of each event pair.

5.5 Conclusion

In this chapter, I have worked on article-wide *E-E* temporal relation classification. While most related work have been focused on the relationship between event pairs found in the same sentence, or in adjacent sentences, this enlarged scope is necessary to allow me to construct a more complete timeline subsequently.

To deal with this problem, traditional lexico-syntactic features employed by most state-of-the-art systems would likely have been insufficient. In their place I propose the use of features derived from discourse analysis. Several discourse analysis frameworks are used and shown to be effective for classifying the relationships between article-wide *E-E* pairs. These proposed features are robust and work well, even though automatic discourse analysis is noisy. Further experiments show that improvements to these underlying discourse analysis systems will directly impact and benefit system performance.

In future work, it will be useful to explore how to better exploit the various discourse analysis frameworks for temporal classification. For instance, RST relations are either *hypotactic* or *paratactic*. Marcu (1997) made use of this to generate automatic summaries by considering EDUs which are nuclei to be more salient. It is interesting to examine how such information can help.

Further having demonstrated the utility of discourse features in this section, it will be useful to study the use of these features in the context of a global inferencing system (Do et al., 2012; Yoshikawa et al., 2009). It is highly likely that such analyses will also benefit these systems as well.

Chapter 6

Summarization

Having presented a fully automatic pipeline to construct a timeline, this chapter looks at how it can be used to improve multi-document summarization.

Much of previous literature as reviewed in Chapter 2 have been focused on temporal processing, without much attempt to study how it can be exploited for downstream applications. In this chapter, I address this gap and study how temporal information can be effectively used in multi-document summarization.

The approach I have taken is to first implement a state-of-the-art multi-document summarization system, `SWING` (Ng et al., 2012). Then I incorporate elements of timelines into this system to obtain a significant improvement (Ng et al., 2014) as measured by ROUGE (Lin and Hovy, 2003).

This work is significant because it introduces 1) three novel features derived from a timeline which can help improve multi-document summarization, 2) a modification to the traditional MMR algorithm that goes beyond analyzing lexical similarities, and 3) a metric which can help decide automatically when timelines can be usefully employed for summarization.

6.1 The Notion of "*Temporal Summarization*"

So far, I have deliberately avoided the use of the term "*temporal summarization*". There is a lack of community consensus of what constitutes temporal summariza-

tion, unlike well-defined problems like *multi-document summarization* or *guided summarization*. It is loosely taken to mean any methodology or system which factors in a notion of time, and which alters the original input information such that it is presented in a more compact form.

For example, the Temporal Summarization track at the Text Retrieval Conference is a new shared task in 2013[1]. Given a large stream of data in real-time, the purpose of the Temporal Summarization track is to look out for a query event, and retrieve specific details about the event over a period of time. If the event is a plane crash, then details to be retrieved may include the number of casualties involved, and the location of the crash. Systems are also expected to identify the source sentences from which these details are retrieved. In some sense, this task resembles the temporal slot-filling task (Ji et al., 2010) of TAC, but with an additional requirement to deal with a real-time evolving data feed.

Georgescu et al. (2013) on the other hand describes another temporal summarization system which attempts to solve a different problem. Making use of the edit history of Wikipedia articles, peaks of update activity for a given entity are first identified using burst detection. Events pertaining to this entity are then extracted from these edits with a supervised classifier. These events can be clustered to identify which of them are unique. They can then be plotted along a timeline to help present a visual summary of the events which are relevant to the entity.

It is understandable why these authors have chosen to label their respective works as *temporal summarization*. However the described systems are very different in terms of their functionalities and end-goals. As such to avoid any confusion, I have decided to avoid the use of this term. I will refer to my work as "*Timeline-Assisted Summarization*" to more accurately reflect what is involved.

6.2 SWING: A Competitive Summarization Testbed

As I am seeking to show that temporal information can be useful for text summarization, an important first step is to implement a state-of-the-art summarization

[1] `http://www.trec-ts.org`

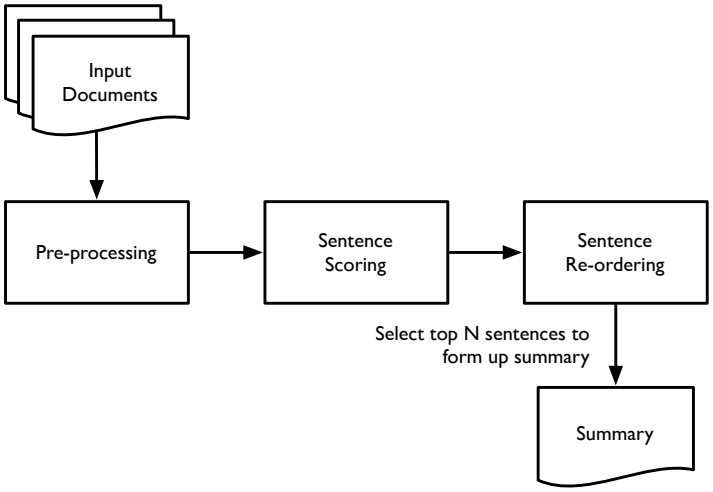


Figure 6.1: Pipeline of the SWING text summarization system.

system to serve as a comparative benchmark. In this section I will describe SWING that I have implemented for this purpose.

SWING is fundamentally based on a supervised learning framework. A set of features is derived for each sentence in the input documents to measure their importance. The top-ranked sentences are then selected to form the eventual summary. The main stages in the SWING pipeline are illustrated in Figure 6.1.

In the initial stage, *Pre-processing* is carried out. This include performing stop word removal and stemming.

Then in the *Sentence Scoring* stage, a set of features is used to assign a score to every sentence from the input documents. To combine the scores from individual features together, a set of weights derived from support vector regression (SVR) (Gunn, 1998) is used, following the methodology described in Bysani et al. (2009). Data from TAC-2010 is used as the training corpus, and the trained regression model is used to predict the saliency scores of each sentence in the TAC-2011 dataset.

Finally in the *Sentence Re-ordering* stage, the Maximal Marginal Relevance (MMR) algorithm (Carbonell and Goldstein, 1998) is used to perform sentence re-ranking and selection. MMR is a greedy algorithm, iteratively selecting a sentence with the highest score for incorporation into the final summary. Sentences are re-ranked based their feature scores, as well as their similarity to sentences which have already been selected to be in the final summary. In SWING, the

MMR of a sentence s is computed as:

$$MMR(s) = Score(s) - R2(s, S) \tag{6.1}$$

where $Score(s)$ is the score predicted by the regression model, S is the set of sentences already selected to be in the summary from previous iterations, and $R2$ is the predicted ROUGE-2 (R-2) score of the sentence under consideration (s) with respect to the selected sentences (S).

6.2.1 Features

The features used in `SWING` include 1) sentence position, 2) sentence length, and 3) a modified version of document frequency which calculates the relevance of a sentence. These features have previously been shown to be effective for text summarization in related literature.

1. Sentence position (Edmundson, 1969) is a popular feature used in summarization, especially in the news domain. The intuition behind the feature is that leading sentences in a news article usually contain important, summary-worthy information owing to typical journalistic guidelines. Accordingly, the score of this feature is gradually decreased from the first sentence to the last sentence in a document based on its position.

2. Sufficient sentence length is a binary feature that helps in avoiding noisy short text in the summary. The value of this feature is 1 if the length of sentence is at least 10, and zero otherwise. The value 10 is empirically determined during system tuning.

3. Interpolated N-gram Document Frequency (INDF) is an extended formulation of the popular document frequency (DF) measure. The efficacy of DF in summarization has been previously demonstrated by Schilder and Kondadadi (2008) and Bysani et al. (2009). It computes the importance of a token as the ratio of the number of documents in which it occurred to the total number of documents within a topic. The use of DF is extended from unigrams to bigrams. INDF is the weighted linear combination of

the DF for unigrams and bigrams of a sentence. Since bigrams encompass richer information and unigrams avoid problems with data sparseness, a combination of both is chosen. The INDF of a sentence s, is computed as:

$$INDF(s) = \frac{\alpha(\sum_{w_u \in s} DF(w_u)) + (1-\alpha)(\sum_{w_b \in s} DF(w_b))}{|s|} \quad (6.2)$$

where w_u are the unigram and w_b are the bigram tokens in sentence s. α is a weighting factor that is set to 0.3 empirically.

6.2.2 Performance

In this section and the rest of this thesis, summarization evaluation is done using ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) (Lin and Hovy, 2003). ROUGE is used here for two key reasons:

1. R-2 and R-SU4 have previously been shown to co-relate well with human assessment (Lin, 2004) and are often used to evaluate automatic text summarization.

2. As will be explained later, I am using the shared task dataset from the Text Analysis Conference (TAC) in the experiments. Using ROUGE, which was used during the shared task, allows me to compare and evaluate my work with existing state-of-the-art systems easily and fairly.

The use of ROUGE as a measure of summarization performance is not perfect. This is why human assessments are often conducted, especially to obtain judgements on the readability and responsiveness (Dang, 2006) of generated summaries. But while readability and responsiveness are important aspects to consider when evaluating summaries, I am trying to show that temporal information is effective in guiding content selection. ROUGE is a n-gram based metric which scores automatic summaries based on their lexical-likeness to human-written ones. In this sense it is effective as a gauge of how close the content in the automatic and manual summaries are. This is because vocabulary overlap between a target and a candidate summary is arguably a good hint of similarity.

There have been attempts to improve on the automatic ROUGE measures, notably the Pyramid method (Passonneau et al., 2005) which has been used

extensively in the TAC evaluation workshops. However this is a semi-automatic method and requires human effort to identify snippets of information referred to as "nuggets". Summaries participating in the same evaluation are scored collectively against these nuggets. It is not possible nor fair to re-evaluate new summaries from previously identified nuggets because 1) the annotators are not the same, and 2) new nuggets that are possibly found in the new summaries would not have been included in the collectively identified pool of nuggets we are measuring against, leading to a bias in scores against the new summaries.

Several iterations of the Automatically Evaluating Summaries of Peers (AESOP) task (Dang and Owczarzak, 2009) have also been held in TAC, with the goal of discovering newer and better automatic measures to improve on ROUGE. However despite these community efforts, there has yet to be a consensus on an effective, automatic evaluation method, and ROUGE remains widely in use.

Configuration	**R-2**	**R-SU4**	**Sig**
SWING	0.1339	0.1651	NA
CLASSY	0.1278	0.1581	-
POLYCOM	0.1227	0.1595	**

Table 6.1: ROUGE scores over the TAC-2011 dataset. Results for CLASSY and POLYCOM are reported after the jackknifing procedure, as released by the shared task organizer. '**' denotes a statistically significant difference in R-2 relative to SWING with $p < 0.05$.

The results obtained through combining the above described features for SWING are shown in Table 6.1. The results of two reference systems CLASSY (Conroy et al., 2011) and POLYCOM (Zhang et al., 2011) are also included as benchmarks. CLASSY and POLYCOM are the second and third best performing systems at TAC-2011. The top performing system is a derivative of SWING which makes use of category-specific information, targeting guided summarization (Ng et al., 2011). In the table, it is seen that the performance of SWING is competitive, out-performing both CLASSY and POLYCOM. In fact the difference in performance between SWING and POLYCOM is also statistically significant with $p < 0.05$ (one-tailed paired Student's t-test).

6.3 Timeline-Assisted Summarization

With SWING as the foundation, the next step I took is to examine the use of temporal information to enhance the summarization process. As motivated earlier in Chapter 1, I seek to do this by targeting the goals of increasing relevancy, and minimizing redundancy. The approach I adopt is to 1) temporally process the input documents to be summarized as described in the previous two chapters, 2) construct timelines for the input documents, and 3) inject these timelines into the sentence scoring and re-ordering stages.

This workflow is summarized in Figure 6.2. The input to the system is a collection of documents for which we want to generate one single summary for. The top part of the figure shows the steps involved in the temporal processing of these input documents. The stages involved correspond to those shown earlier in Figure 3.2. Note that in here, one timeline is generated for each input document within the input collection. The bottom half of the figure is the original pipeline for SWING shown earlier in Figure 6.1.

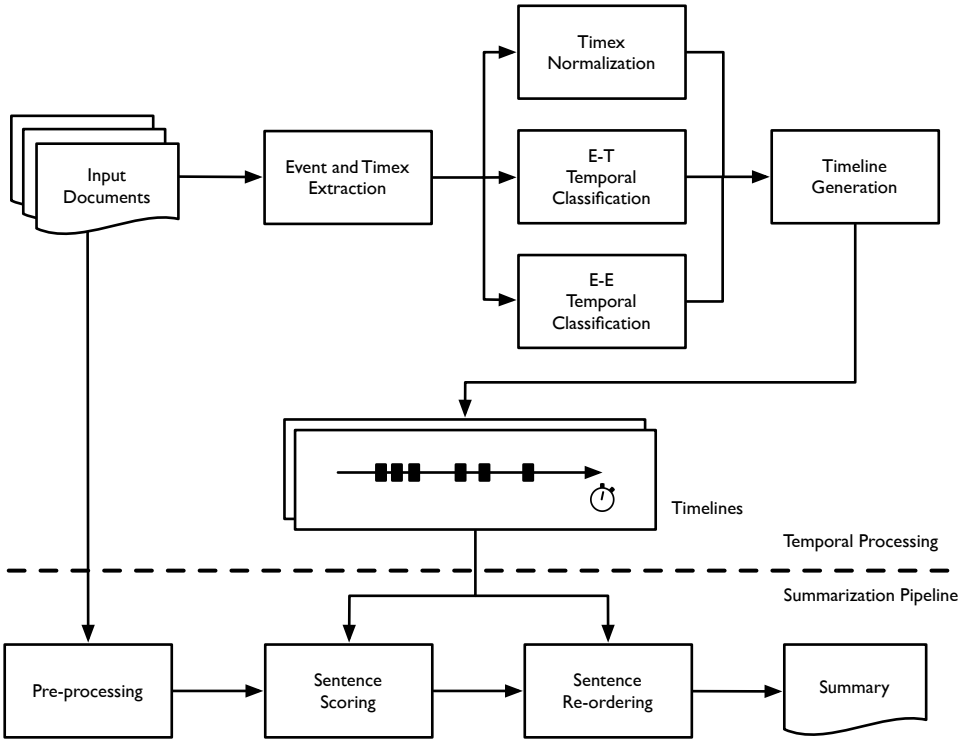


Figure 6.2: Overview of how temporal information is incorporated into SWING.

The key here is how information from timelines can be injected into the *Sentence Scoring* and *Sentence Re-ordering* stages. More details on this are given

in the rest of this section; but to summarize, I extract a set of three features from the timeline. This is used in tandem with the original features in `SWING` to score and rank sentences from the input documents. This updated scoring mechanism helps identify more relevant sentences to include in the eventual summary. Then the MMR algorithm used in the re-ordering process is modified to also take in time span similarities. This allows the MMR algorithm to identify redundant events and promote diversity in the selected sentences.

6.3.1 Timeline Features for Sentence Scoring

I will now explain the three timeline features that feed into the *Sentence Scoring* stage of `SWING`. The common motivation behind these features is the hypothesis that temporal information has an influence on the saliency of a sentence. Thus incorporating temporal information into sentence scoring should help improve the relevancy of the eventual generated summary.

Figure 6.3 shows a simplified timeline, along with annotations that will be referenced in this section to help explain how the timeline features are derived. Time spans have been demarcated by vertical dotted lines. Solid blocks on the time axis represent time spans that can be mapped to an absolute timestamp. Events are represented as squares in each time span. Events within the same time span temporally overlap. Events to the left of other events happen BEFORE the latter. For the avoidance of doubt, do note that events that fall within the same time span are not necessarily referencing the same event occurence. It is also useful to keep in mind that sentences are not directly represented in the timeline. However sentences are related to the timeline in the sense that events referenced in sentences are placed along the timeline.

1. ***Time Span Importance.*** Time span importance (TSI) captures the saliency of a particular time span along the timeline. Given the many events that are extracted from the input documents, some of them will be more important than others. I hypothesize that when more events happen within a particular time span, that time span is potentially more relevant for summarization. An example of this was shown earlier in Figure 1.4, the timeline of which is re-produced here in Figure 6.4 for convenience. In the

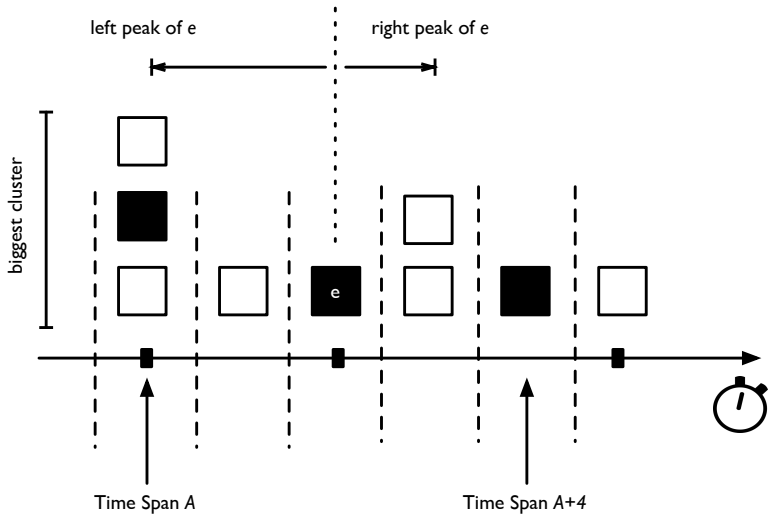


Figure 6.3: A simplified timeline illustrating how the various timeline features can be derived.

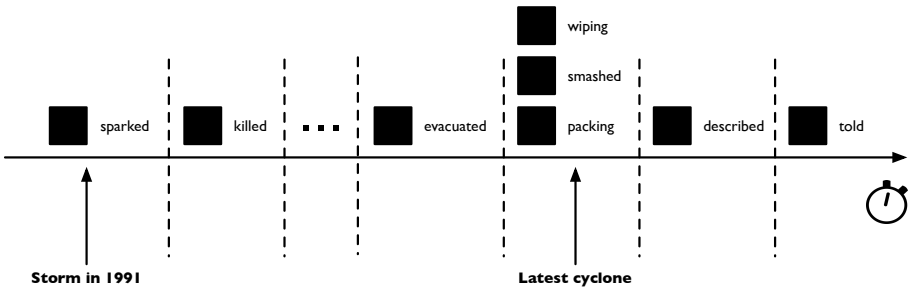


Figure 6.4: Possible timeline for events in Figure 1.4. (Reproduced from Figure 1.5 for convenience).

timeline, the time span with the most number of events refer to when the latest cyclone made landfall. Sentences which contain events in this time span are going to be more important for a summary about the cyclone.

This is similar to the concept of event-based summarization presented in Wu (2008). Events are laid out on a timeline, and weights are assigned to sentences based on the frequencies of these events, and the number of days the events span. In this formulation, it is necessary to know all references to the same event to be able to compute the number of days it spans. My approach here differs in that scoring is based solely on the number of events happening in each time span. This in line with the motivation behind this feature, and I believe its simplicity is an advantage here. Using Wu's approach, it would not have been easy to retrieve all references to the same event (i.e., event co-reference).

Let the time span with the largest number of events in a timeline be TS_L. Then the importance of a time span TS_i can be computed by normalizing the number of events in TS_i against the number of events in TS_L to get a real value between 0 and 1, inclusive. The time span importance TSI of a sentence s is then computed as:

$$TSI(s) = \frac{\sum_{w \in s} \frac{|TS_w|}{|TS_L|}}{|s|} \tag{6.3}$$

where TS_w denotes the time span which a word w is associated with, and $|TS_w|$ is the number of events within the time span. This feature essentially says that if a sentence contains events which OVERLAP with many other events in the same time span, then the sentence may be more important.

2. ***Contextual Time Span Importance.*** Contextual time span importance (CTSI) is based on the intuition that the importance of a time span does not depend solely on the number of events that happen within it. If it is near time spans which are "important" (i.e., one that has a large number of events), it should also be of relative importance. A more concrete illustration of this is seen in Example 1.4. The corresponding timeline has been re-produced earlier in Figure 6.4. Sentence (2) from Example 1.4 explains that a lot of people have been evacuated prior to the cyclone making landfall. It is imaginable that this can be potentially useful information to be included in a summary, even though from looking at the corresponding timeline in Figure 6.4, the "*evacuated*" event falls in a time span with a low importance score (i.e., the time span only has one event). CTSI seeks to promote sentences such as this. Since the "*evacuated*" event happens in a time span preceding the actual cyclone attack, its CTSI is higher than other time spans, improving the chances of its source sentence being included in the eventual summary.

   The CTSI of a sentence is derived by first computing the contextual importance of words in the sentence. The contextual importance of a word found in time span TS_i is defined as a weighted sum of the time span importance of the two nearest peaks TS_{lp} and TS_{rp} found to the left and right of TS_i,

respectively. Referring to Figure 6.3, taking reference from event e (shaded in black), the left peak to the time span e is in happens to be time span A, while the right peak is time span $A + 4$. The contribution of each peak to the weighted sum is decayed by its distance from TS_i. Formally, the contextual importance of a word w can be expressed as:

$$ctsi(w) = \alpha \left(\frac{I_{lp}}{|TS_w - TS_{lp}|} \right) \times \beta \left(\frac{I_{rp}}{|TS_{rp} - TS_w|} \right) \tag{6.4}$$

where TS_w is the time span associated with w. I_{lp} and I_{rp} are the time span importance of the peaks to the left and right of TS_w respectively, while $|TS_w - TS_{lp}|$ and $|TS_{rp} - TS_w|$ are the number of time spans between the left and right peaks of TS_w respectively. α and β denotes the weights attributed to the importance of the left and right peaks. For now it is intuitive to set $\alpha = \beta = 0.5$.

The contextual time span importance score attributable to a sentence $CTSI(s)$ is then computed as:

$$CTSI(s) = \frac{\sum_{e \in \mathbb{E}_s} ctsi(e)}{|\mathbb{E}_s|} \tag{6.5}$$

where $\mathbb{E}_s$ denotes the set of events words in s.

3. ***Sentence Temporal Coverage Density.*** To understand temporal coverage density, let me first explain what I define as the "*temporal coverage*" of a sentence. Suppose a sentence contains events which are associated with time spans TS_a, TS_b, TS_c. The time spans are ordered in the sequence they appear on the timeline. Then the temporal coverage of a sentence is defined as the number of time spans between the earliest time span TS_a and the latest time span TS_c. Referring to Figure 6.3, suppose a sentence contains the three events which have been shaded black. The temporal coverage in this case includes all the time spans from time span A to time span $A + 4$ inclusive.

   Given that there is a constraint on the number of sentences that can be included in a summary, it is important to be able to select compact sentences

which contain as many relevant facts and information nuggets as possible. Traditional lexical measures may attempt to achieve this by computing the ratio of keyphrases to the number of words in a sentence (Gong and Liu, 2001). The idea being that if two sentences are of the same length, then the one with more keyphrases should likely contain more useful facts and information nuggets.

Sentence temporal coverage density parallels this idea with the use of temporal information, i.e. by preferring sentences which contain more events, given their temporal coverage. The intuition is that if two sentences are of the same temporal coverage, then the one with more events should carry more useful facts and information nuggets.

Formally, if a sentence s contains events $\mathbb{E}_s = \{e_1, \ldots, e_n\}$, where each event is associated with a time span TS_i, the temporal coverage density TCD is computed using:

$$TCD(s) = \frac{|\mathbb{E}_s|}{|TS_n - TS_1|} \tag{6.6}$$

where $\mathbb{E}_s$ is the number of events found in s, and $|TS_n - TS_1|$ is the temporal coverage of the sentence as explained earlier.

6.3.2 TimeMMR — Considering Time Span Similarity with MMR

In the sentence re-ordering stage of the `SWING` pipeline, the MMR algorithm is used to adjust the score of a candidate sentence, s, based on Equation 6.1. Effectively, the score of a sentence is penalized if it is lexically similar to other sentences that have already been selected to form the eventual summary $S = \{s_1, s_2, \ldots\}$.

Information from a timeline can potentially improve this re-ordering process as it allows us to look beyond lexical similarity. To promote diversity, I propose further penalizing the score of s if it contains events that happen in similar time spans as those contained in sentences within S. Referring to this as TimeMMR, formally:

$$TimeMMR(s) = Score(s) - \alpha R2(s, S) - \beta \mathcal{T}(s, S) \tag{6.7}$$

where, as in Equation 6.1, $Score(s)$ is the score predicted by the regression model, S is the set of sentences already selected to be in the summary from previous iterations, and $R2$ is the predicted R-2 score of the sentence under consideration (s) with respect to the selected sentences (S). α and β are weighting parameters which have been empirically set to 0.9 and 0.1, respectively. $\mathcal{T}$ is the proportion of events in s which happen in the same time span as another event in any other sentence in S.

Suppose s contains events $\mathbb{E}_s = \{qe_1, \ldots, qe_n\}$. Also, for every $s_i \in S$, let the events contained in each s_i be the set $\mathbb{E}_{s_i} = \{se_{i1}, \ldots\}$, and $\mathbb{E}_S = \{\mathbb{E}_{s1}, \ldots\}$. Then,

$$\mathcal{T}(s, S) = \frac{1}{n} \times \sum_{i=1}^{n} \mathcal{I}(qe_i, \mathbb{E}_S) \tag{6.8}$$

$\mathcal{I}(qe_i, \mathbb{E}_S) = 1$ if $\exists i, x, se_{ix} \in \mathbb{E}_{s_i}, \mathbb{E}_{s_i} \in \mathbb{E}_S$ such that se_{ix} and qe_i are in the same time span. To compute $\mathcal{I}$, there is a need to 1) associate events to specific timestamps, and 2) decide if one time span is contained within another.

Associating Events to Timestamps. This is achieved by combining information from the earlier *E-T* and *E-E* temporal relationship classification systems with a timex normalizer, such as HeidelTime (Strötgen and Gertz, 2013). HeidelTime associates timexes to complete timestamps as far as possible. Events which have an OVERLAP relationship with a timex can then be associated to the timestamp as well. Similarly, events which OVERLAP with other events that have been associated with a timestamp can then be associated with the same timestamp. Algorithm 6.1 describes this process.

Ascertaining Time Span Similarity. Timex normalizers such as HeidelTime cannot always resolve timexes to timestamps of the same granularity. Often timexes get resolved to timestamps of much coarser granularity such as a week of the year, or a particular month. Given two events e_1 and e_2, and their associated timestamps T_{e_1} and T_{e_2}, without loss of generality, we say that e_1 is within e_2 if e_1 is at least as fine grained as e_2, and falls within the time spans specified by e_2.

So if we are comparing the year '1999' with a date '1999 Oct 1', '1999 Oct 1' is more fine-grained than '1999', and falls within '1999'. On the other hand

Algorithm 6.1 Associating events to specific timestamps.

```
1: Map ← {} // Holds the association between events and timestamps
2: UnMapped ← {}
3: for each event e_i do
4:     for each timex tx_j do
5:         if e_i OVERLAP tx_j then
6:             Let tm_j be the time point tx_j is mapped to based on Algorithm 3.1
7:             Map ← Map ⋃ {e_i, tm_j}
8:         end if
9:     end for
10:    if e_i is not mapped to any timex then
11:        UnMapped ← UnMapped ⋃ {e_i}
12:    end if
13: end for
14: repeat
15:    for each event e_i ∈ UnMapped do
16:        for each {e_j, tm_x} in Map do
17:            if e_i OVERLAPS e_j then
18:                Map ← Map ⋃ {e_i, tm_x}
19:                UnMapped ← UnMapped − e_i
20:            end if
21:        end for
22:    end for
23: until Map does not change
```

if we compare two dates '2000 Jan 31' and '2001 Jan 31', they are of the same granularity and do not overlap, so neither is within the other.

6.3.3 Overcoming Propagated Errors with Reliability Filtering

The best results obtained over our experimental dataset for *E-T* and *E-E* temporal classification is around 0.656 and 0.552. Together with the simplifying assumptions that were made in timeline construction in Algorithm 3.1, the timelines that are constructed are likely to carry errors. Mis-classifications of temporal relationships will have an impact on the correctness of the order of the events that are laid out onto the timeline. Making use of these timelines for summarization may then not be helpful, as the errors will propagate throughout the entire pipeline.

With this in mind, I propose selectively employing timelines to generate summaries only when we are reasonably confident of their accuracy or what I term "*reliability*". Reliability filtering involves computing a metric which can be used to decide whether or not temporal information is to be used.

The length of a timeline can be a useful metric for reliability filtering. With the currently obtainable accuracy rates for *E-T* and *E-E* temporal classification, there is likely to be a fair amount of errors within the generated timelines. In longer timelines, to which more events are mapped, these errors are spread over the timeline, and do not over-power any useful signal that can be obtained with the timeline features outlined earlier. When a timeline is short, these errors are very easily propagated into summary generation, leading to less useful results.

The mechanics involved in reliability filtering are as follows: Given an input document set (which in our case, consists of 10 documents), the average size of all the timelines for each of these 10 documents is computed. If this is larger than some threshold value, then timeline information is used. Otherwise, the timelines are deemed to be too inaccurate, and thus not employed.

Noteworthy here is that the reliability filtering's purpose is really to mitigate possible propagated errors caused by current state-of-the-art in *E-T* and *E-E* temporal classification. As the performance of these underlying classification systems improve, reliability filtering may be less useful and eventually retired.

6.3.4 Experiments and Results

The proposed timeline features and TimeMMR algorithm were implemented on top of `SWING`. Repeating the same experimental settings from earlier, the ROUGE scores that are obtained are given in Table 6.2. In the table, each row refers to a specific summarization system configuration. The result for `SWING` in row R is re-produced from Table 6.1 for reference. TSI refers to the time span importance feature; CTSI to the contextual time span importance feature; and TCD to the sentence temporal coverage density feature. Statistical significance in the two "**Sig**" columns are computed with respect to Row R using the one-tailed paired Student's t-test. Rows 9 to 16 repeat the system configurations in Rows 1 to 8, applying reliability filtering. In these experiments, the threshold for filtering is set to be the average of all the timeline sizes over the whole input dataset (i.e., 42.68). In a production environment where this assumption may not hold, this threshold could be set by empirical tuning over a development set.

Row 1 shows the value of the three proposed timeline-based features. A

	Configuration	R-2	Sig	R-SU4	Sig
(R)	SWING	0.1339	NA	0.1651	NA
Without Filtering					
(1)	SWING + TSI + CTSI + TCD	0.1394	*	0.1688	*
(2)	SWING + TSI + CTSI	0.1372	-	0.1681	-
(3)	SWING + TSI + TCD	0.1372	-	0.1673	-
(4)	SWING + CTSI + TCD	0.1387	*	0.1673	-
(5)	SWING + TSI + CTSI + TCD + TIMEMMR	0.1389	-	0.1688	-
(6)	SWING + TSI + CTSI + TIMEMMR	0.1374	-	0.1686	*
(7)	SWING + TSI + TCD + TIMEMMR	0.1343	-	0.1659	-
(8)	SWING + CTSI + TCD + TIMEMMR	0.1363	-	0.1665	-
With Filtering					
(9)	SWING + TSI + CTSI + TCD	0.1418	**	0.1695	**
(10)	SWING + TSI + CTSI	0.1378	**	0.1677	**
(11)	SWING + TSI + TCD	0.1389	**	0.1677	**
(12)	SWING + CTSI + TCD	0.1401	**	0.1681	**
(13)	SWING + TSI + CTSI + TCD + TIMEMMR	0.1402	**	0.1678	-
(14)	SWING + TSI + CTSI + TIMEMMR	0.1397	**	0.1693	**
(15)	SWING + TSI + TCD + TIMEMMR	0.1376	*	0.1665	-
(16)	SWING + CTSI + TCD + TIMEMMR	0.1390	**	0.1672	*

Table 6.2: Resulting ROUGE scores obtained after incorporating temporal information into SWING. '**' and '*' denotes statistically significant differences with respect to Row R with $p < 0.05$ and $p < 0.1$ respectively.

statistically significant improvement is obtained with the use of all three features. An even better improvement is obtained when reliability filtering is performed in Row 9.

The ablation test results in Rows 2 to 4 show a drop in R-2 and R-SU4 each time a feature is left out. With the exception of Row 4 where the time span importance feature is dropped, removing the other features has such an impact that the resulting R-2 measures are no longer significantly different from SWING. Rows 9 to 12 show the same system configurations with reliability filtering added in. The same observations hold. Removing any one feature causes a drop in ROUGE measures. There are two possibilities: 1) each feature is inherently weak, however they synergize well collectively to give a result that is more than the sum of its parts, or 2) the features are useful, yet are hampered by the performance of the underlying *E-T* and *E-E* temporal classification systems that generated the timelines used.

Rows 5 to 8 and Rows 13 to 16 show the effect of TIMEMMR. While the results do not uniformly show that the proposal is effectively, TIMEMMR can

be helpful such as when comparing Rows 2 and 6, or Rows 10 and 14. Both R-2 and R-SU4 improves marginally with the use of TimeMMR.

An important observation is that the use of reliability filtering consistently improves ROUGE scores. The purpose of reliability filtering is to try and avoid the use of timelines that may be too inaccurate to be useful when applied to text summarization. In this sense reliability filtering is successful. In fact with the use of reliability filtering, good improvements in ROUGE can be obtained when compared to `SWING`. Importantly, the differences between corresponding system configurations with and without reliability filtering (e.g., Rows 1 and 9, or Rows 2 and 10) are not statistically significant. This shows that reliability filtering by itself is not the reason for the good improvements obtained over `SWING`. Instead it complements the proposed timeline features, and helps identify cases when the timeline features are effective for summarization.

To help visualize what the differences in these ROUGE scores mean, Figure 6.5 shows two summaries generated for document set D1117C of the TAC-2011 dataset. The left one ($\mathbb{L}1$ to $\mathbb{L}3$) is produced by the configuration in Row 9, and the right one ($\mathbb{R}1$ to $\mathbb{R}4$) is produced by `SWING` without the use of any temporal information. Note that this summary and the others that follow in the rest of this thesis are truncated to fit within the 100-word target length as per the TAC-2011 guidelines.

The higher ROUGE score obtained by the summary on the left (0.0873) compared to the one on the right (0.0723) suggests that the use of timeline features can help to identify salient sentences more accurately. As an illustration, let us take a closer look at sentences ($\mathbb{L}2$) and ($\mathbb{R}2$). ($\mathbb{L}2$) achieved a R-2 score of 0.0424 while ($\mathbb{R}2$) achieved 0.0249. ($\mathbb{L}2$) is favoured over ($\mathbb{R}2$) when timeline features are used. Figure 6.6 shows a breakdown of the raw feature scores achieved by both of these sentences. SP, Length and INDF refer to the `SWING` features of sentence position, sentence length, and interpolated n-gram document frequency respectively. It can be seen that both sentences achieved similar scores for the `SWING` features, except for SP where ($\mathbb{R}2$) does better. The scores for all three timeline features are however higher for ($\mathbb{L}2$) than ($\mathbb{R}2$).

R-2: 0.0873		R-2: 0.0723
(𝕃1) The Army's surgeon general criticized stories in The Washington Post disclosing problems at Walter Reed Army Medical Center, saying the series unfairly characterized the living conditions and care for soldiers recuperating from wounds at the hospital's facilities.	==	(ℝ1) The Army's surgeon general criticized stories in The Washington Post disclosing problems at Walter Reed Army Medical Center, saying the series unfairly characterized the living conditions and care for soldiers recuperating from wounds at the hospital's facilities.
(𝕃2) Defense Secretary Robert Gates says people found to have been responsible for allowing substandard living conditions for soldier outpatients at Walter Reed Army Medical Center in Washington will be "held accountable," although so far no one in the Army chain of command has offered to resign.	≠≠	(ℝ2) A top Army general vowed to personally oversee the upgrading of Walter Reed Army Medical Center's Building 18, a dilapidated former hotel that houses wounded soldiers as outpatients.
(𝕃3) Top Army officials visited Building 18, the decrepit former hotel housing more than 80 recovering soldiers, outside	≠≠	(ℝ3) "I'm not sure it was an accurate representation," Lt. Gen. Kevin Kiley, chief of the Army Medical Command which oversees Walter Reed and all Army health care, told reporters during a news conference.
	>>	(ℝ4) The Washington

Figure 6.5: Generated summaries for document set D1117C from the TAC-2011 test set. The summary on the left (i.e., 𝕃1 to 𝕃3) is generated by SWING+TSI+CTSI+TCD with filtering, while the summary on the right (i.e., ℝ1 to ℝ4) is by SWING.

6.4 Discussion

Following the experimental results, let us examine the proposed 1) timeline features, 2) TimeMMR algorithm, and 3) reliability filtering metric, in greater detail to get some insight into their efficacy and utility.

6.4.1 A Closer Look at Timeline Features

The proposed timeline features are motivated along the lines of relevancy and redundancy. It is useful to re-visit the intuitions behind these features, and perform a micro-analysis of the actual results from the summarization system.

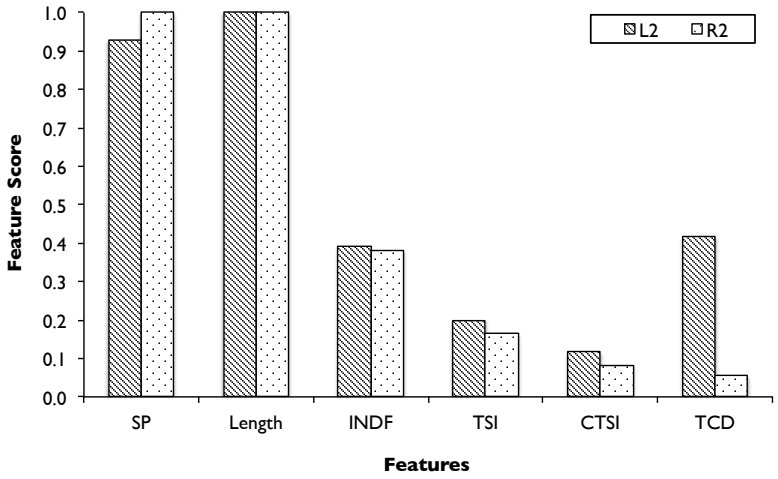


Figure 6.6: Breakdown of raw feature scores for sentences (𝕃2) and (ℝ2) from Figure 6.5.

The purpose is to examine the differences between the summaries generated and understand the actual improvement these features bring to the final summaries. Since the ROUGE metric is used to evaluate these generated summaries, it will be used as a proxy for relevancy and redundancy here.

Time Span Importance. Figure 6.7 shows the last two sentences from a pair of summaries generated with and without the use of time span importance. The other sentences in the summaries are otherwise exactly the same. The summary on the left has a higher R-2 score of 0.1683, compared to 0.1533 for the one on the right.

R-2: 0.1683		R-2: 0.1533
...		...
(𝕃1) A piece of steel fell and sheared off one of the ties holding it to the building, causing it to detach and topple, said Stephen Kaplan	≠≠	(ℝ1) About 19 of the 44 stories of the crane had been erected and it was to be extended when a piece of steel fell and sheared

Figure 6.7: Extract from summaries for document set D1137G from the TAC-2011 test set. The extract on the left (i.e., 𝕃1) is generated by `SWING`+TSI+CTSI+TCD, while the summary on the right (i.e., ℝ1) is by `SWING`+CTSI+TCD.

The original source articles for this document set describe an industrial accident where casualties were suffered when a crane toppled onto a building. It is easy to see why (𝕃1) scores higher — it describes the cause of the accident just

R-2: 0.1215		R-2: 0.0861
(𝕃1) Caribbean coral species essential to the region's reef ecosystems are at risk of extinction as a result of climate change.	==	(ℝ1) Caribbean coral species essential to the region's reef ecosystems are at risk of extinction as a result of climate change.
(𝕃2) But destructive fishing methods and over-harvesting have reduced worldwide catches by 90 percent in the past two decades.	≠≠	(ℝ2) The Coral Reef Task Force, created in the Clinton administration, regularly assesses coral health.
(𝕃3) Scientists warn that up to half of the world's coral reefs could disappear by 2045.	≠≠	(ℝ3) With a finished necklace retailing for up to 20,000 dollars (15,000 euros), red corals are among the world's most expensive wildlife commodities.
...		...

Figure 6.8: Extract from summaries for document set D1131F from the TAC-2011 test set. The extract on the left (i.e., 𝕃1 to 𝕃3) is generated by SWING+TSI+CTSI+TCD, while the summary on the right (i.e., ℝ1 to ℝ3) is by SWING+TSI+TCD.

as it occurred. (ℝ1) however talks about how much of the tower had already been erected and the plans to extend it, events which happened before the accident itself. In this case time span importance is able to correctly guide summary generation by favoring time spans containing events related to the actual crane toppling.

Contextual Time Span Importance. Building on top of TSI, CTSI recognizes that events which happen along the fringes of a big cluster of other events can potentially be important too. The benefits of this feature can be most clearly seen in Figure 6.8 which shows extracts of the summaries generated with and without the use of CTSI. The summary of the left achieved a R-2 score of 0.1215 while the one on the right achieved 0.0861. (𝕃2) and (𝕃3) were both boosted by the use of the CTSI feature.

Figure 6.9 shows an extract of the timeline generated for the source document from which (𝕃3) is extracted. The two events inside (𝕃3) fall in time spans A and B marked in the figure. Their proximity to the peak between them gives the sentence a higher score for CTSI. This boosts the total score attributed to the sentence sufficiently such that it gets selected to be included in the final

summary. This sentence was lifted exactly in one of the model summaries for this document set, resulting in a very good R-2 score when CTSI is used.

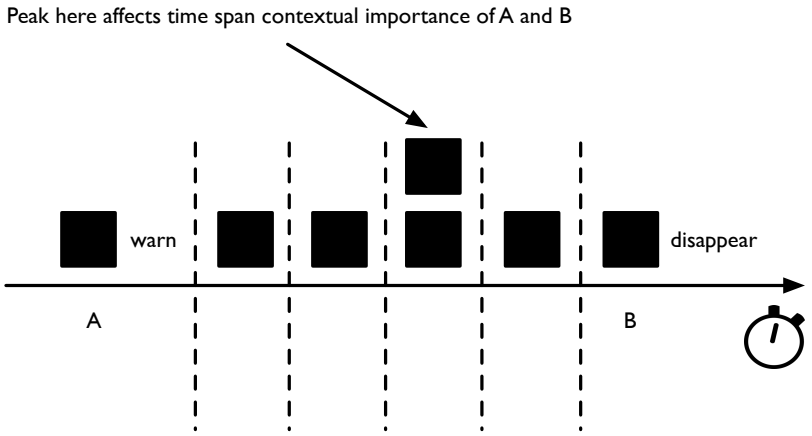


Figure 6.9: Extract of timeline generated for document APW_ENG_20070615.0356 from the TAC-2011 testing dataset.

Temporal Coverage Density. This feature promotes sentences which contain a larger number of events given the number of time spans covered. The intuition behind this is that if the temporal coverage of a sentence is large, it should correspondingly reference more events. Otherwise it could be that the sentence is just un-necessarily long with superfluous syntactic constructs.

Figure 6.10 shows two sentences from the final summary generated for document set D1113C. ($\mathbb{L}1$) is selected to be part of the summary when the temporal coverage density feature is used, replacing ($\mathbb{R}1$). Using the temporal coverage density feature results in a higher R-2 score (0.1163 vs 0.0646). One of the human-written model summaries for this document set contains a very similar sentence to ($\mathbb{L}1$). This validates that the feature is able to identify sentences which human assessors find salient too.

6.4.2 Is TimeMMR Useful?

The experimental results do not conclusively affirm the usefulness of TimeMMR. However it could be because the evaluation metric that is used (i.e., ROUGE scores) is not the most suitable to evaluate TimeMMR. Recall that TimeMMR seeks to eliminate redundancy based on time span similarities and not lexical likeness. ROUGE however measures the latter. While ROUGE generally co-relates well with human assessments, it is not perfect for precisely this reason.

R-2: 0.1163		R-2: 0.0646
...		...
(𝕃1) When a shipping container was seized in Singapore four years ago carrying more than SIX tons of elephant ivory inside, conservation and law enforcement agencies realized that they had intercepted the largest shipment of the contraband material since its international trade was banned in 1989.	$\neq\neq$	(ℝ1) A team of US scientists has used DNA testing to identify the geographic origin of poached elephant tusks – research they hope may help curb the illegal ivory trade.
	>>	(ℝ2) When a shipping container was seized in Singapore four years ago carrying more than SIX tons

Figure 6.10: Extract from summaries for document set D1113C from the TAC-2011 test set. The extract on the left (i.e., 𝕃1) is generated by SWING+TSI+CTSI+TCD, while the summary on the right (i.e., ℝ1 to ℝ2) is by SWING+TSI+CTSI.

That is also why measures like Pyramids have been often used alongside ROUGE in the TAC summarization track.

An interesting case in point is given in Figure 6.11. The summary on the left is generated with the use of TimeMMR, while the one on the right is generated without the use of TimeMMR. The summary on the right achieved a higher ROUGE score, suggesting that TimeMMR is not helpful for this document set.

The key difference in the two summaries is (ℝ3). (𝕃3) is the equivalent of (ℝ4), while (𝕃4) is the full version of the truncated (ℝ5). TimeMMR down-weights (ℝ3) and it is easy to see why. (ℝ3) reports that the shoe-throwing incident happened as the U.S. President Bush appeared together with the Iraqi Prime Minister Nouri al-Maliki. However their joint appearance is already reported in (ℝ1) (and similarly (𝕃1)). (ℝ3) is just repeating what had been presented earlier. Since (ℝ1) and (ℝ3) talks about the same time span, TimeMMR does what it is designed to do and down-weights (ℝ3). I argue that this is for the better, however the ROUGE scores indicate otherwise.

So is TimeMMR useful? This example suggests so despite the lowered ROUGE scores. Experimental results from Table 6.2 also seem to support the use of TimeMMR (albeit weakly). However there are several other factors clouding

R-2: 0.2643		R-2: 0.2772
(𝕃1) – An Iraqi reporter threw his shoes at visiting U.S. President George W. Bush and called him a "dog" in Arabic during a news conference with Iraqi Prime Minister Nuri al-Maliki in Baghdad	==	(ℝ1) – An Iraqi reporter threw his shoes at visiting U.S. President George W. Bush and called him a "dog" in Arabic during a news conference with Iraqi Prime Minister Nuri al-Maliki in Baghdad
(𝕃2) "All I can report is it is a size 10,.	==	(ℝ2) "All I can report is it is a size 10,.
(𝕃3) Muntadhar al-Zaidi, reporter of Baghdadiya television jumped and threw his two shoes one by one at the president, who ducked and thus narrowly missed being struck, raising chaos in the hall in Baghdad's heavily fortified green Zone.	≠≠	(ℝ3) The incident occurred as Bush was appearing with Iraqi Prime Minister Nouri al-Maliki.
(𝕃4) The president lowered his head and the first shoe hit the American and Iraqi flags behind the two leaders.	≠≠	(ℝ4) Muntadhar al-Zaidi, reporter of Baghdadiya television jumped and threw his two shoes one by one at the president, who ducked and thus narrowly missed being struck, raising chaos in the hall in Baghdad's heavily fortified green Zone.
(𝕃5) The	≠≠	(ℝ5) The president lowered his head and the

Figure 6.11: Generated summaries for document set D1126E from the TAC-2011 test set. The summary on the left (i.e., 𝕃1 to 𝕃5) is generated by SWING+TSI+CTSI+TCD+TimeMMR, while the summary on the right (i.e., ℝ1 to ℝ5) is by SWING+TSI+CTSI+TCD.

the issue here, including 1) the accuracy of the timelines that are automatically generated, 2) potential differences in what the human-written summaries find to be salient and 3) the automatic ROUGE measures. More experimentation is definitely going to be required to either support or disprove the value of TimeMMR, alongside the use of a more suitable evaluation metric.

6.4.3 Reliability Filtering

To shed some insight into reliability filtering, Table 6.3 shows the effect of varying the filtering threshold on R-2 for the best performing configuration from Table 6.2 (i.e., SWING+TSI+CTSI+TCD). The result obtained in Row 9 using a threshold of 42.68 is also re-produced for reference. The column "**# Temp**" denotes the

number of times (out of the 44 test document sets) temporal information is used for the corresponding threshold value.

Threshold	R-2	Sig	# Temp
0	0.1394	*	44
10	0.13820	-	43
20	0.13768	-	41
30	0.1393	**	35
40	0.1426	**	22
42.68	0.1418	**	21
50	0.1386	**	13
60	0.1361	*	7
70	0.1351	-	3
80	0.1351	-	2
90	0.1353	-	1
100	0.1339	-	0

Table 6.3: Effect of varying the reliability filtering threshold on R-2 for the configuration `SWING`+TSI+CTSI+TCD. '**' and '*' denotes a statistically significant difference from `SWING` of $p < 0.05$ and $p < 0.1$ respectively.

Note that a threshold of 0 effectively means no filtering is done and temporal information is used for all document sets. A threshold of 100 in this case means that no temporal information is used at all because the length of the longest timeline is less than 100. So accordingly, the first row with a threshold value of 0 corresponds to row 1 in Table 6.2, and the last row with a threshold of 100 corresponds to row R.

As the threshold value increases from 0 to around 40 and 50, we see an improvement in summarization performance. The number of document sets in which temporal information is employed also reduces. This can be interpreted that filtering is successful in identifying timelines that are not accurate enough such that the use of which affects summarization performance.

Beyond 60, the R-2 scores are still higher than that obtained by `SWING`, but no longer significantly different. At these higher thresholds, temporal information is still able to help get an improvement in R-2. However as this affects only very few out of the 44 document sets, statistical variances means that these R-2 scores are no longer significant from that produced by `SWING`. This is understandable.

6.5 Conclusion

Starting from a competitive baseline `SWING`, I have shown in this chapter how temporal information in the form of timelines can be incorporated into automatic text summarization. Three features are proposed which can be extracted from every timeline. These features include 1) time span importance, 2) contextual time span importance, and 3) temporal coverage density. They are premised on the intuition that temporal information can impact sentence saliency. With these features, an improvement of 4.1% is obtained in R-2.

Also, a modification, TimeMMR, was proposed to the MMR algorithm used in `SWING` so that it further incorporates temporal information to reduce the amount of redundant text in the generated summaries. I argue that the ROUGE metric is not the most suitable to evaluate the efficacy of TimeMMR. However despite this, experimental results still show that TimeMMR can be useful in certain situations.

The underlying *E-T* and *E-E* temporal classification systems are not very accurate, and this likely affected the quality of the timelines that are generated. To overcome this, I next proposed a reliability filtering metric, which can be used to automatically decide when temporal information should be used for summarization. The use of reliability filtering helps boost R-2 scores by a further 1.7%, leading to an overall 5.9% gain in R-2 over the competitive `SWING` baseline.

Chapter 7

Conclusion

This concluding chapter summarizes the work that was done for this thesis, and explores further research directions that are worth pursuing based on what has been achieved.

As the capability to process lexical and syntactic properties of text improve, researchers are increasingly focused on tackling the harder but more rewarding semantical aspects of text. This thesis examines one such aspect — the interpretation of time in text. The ability to process and understand the temporal information that is found inside text has great potential to improve many natural language processing applications, ranging from text summarization to question-answering. The increasing community attention to this domain of work is further validation of the utility and value of temporal interpretation.

7.1 Future Work

This thesis answers many important questions, including the use of crowdsourcing to build a cheaper, yet effective temporal corpus, as well as the use of temporal information to benefit multi-document summarization. However, many other questions worthy of further exploration remain.

Better Features for* E-T *Temporal Relation Classification. The performance obtained for intra-sentence *E-T* temporal relation classification can be improved. Besides the findings noted from the error analysis in Section 4.2.4,

such as the need to focus on copula modifiers (*e.g.*, "*to be*), I believe that it is important to study the use of additional semantic cues. A possible direction is to use semantic role labeling. Knowing the roles of a sentence's constituents may help overcome many of the problems associated with syntax.

Intra-sentence* E-E *Temporal Relation Classification. In this thesis I have shown that discourse analysis is a great help to article-wide *E-E* temporal relation classification, of which the intra-sentence variant of the problem is a subset. The results and analysis that was performed hints that discourse analysis is less useful for the intra-sentence variant. This is likely because discourse analysis captures longer distance relationships better than the short distance ones needed for effective intra-sentence classification. Much of the existing literature has focused on the intra-sentential case, so it will be exciting to combine these efforts together with my work in article-wide classification. Done correctly, the resulting system will be able to handle both article-wide event-event pairs and intra-sentence pairs well, giving better overall classification performance.

Enhancing Timeline with Richer Temporal Relations. Earlier in Chapter 3, I have noted that the use of a richer set of temporal relations can help enhance the information that can be captured in timelines. One key enhancement is allowing events to last across multiple time spans, potentially overlapping one another. This thesis has already shown the usefulness of minimizing information redundancy through considering time span overlaps. Having access to an enhanced timeline which more accurately captures the temporal ordering of events will likely further boost this utility. It will be interesting to relax the constraints on the temporal relations supported by the underlying temporal relation classification systems and examine the opportunities and possibilities that enhanced versions of timelines can bring.

Summarization of Text With Time. In this thesis, temporal information was used for summarization by way of vectorized features extracted from a timeline. Other possibilities and solutions are definitely worth exploring too. For example, since temporal information can encoded in a temporal graph (Verhagen et al., 2010), perhaps these can be better incorporated into a graph-based text summarization system, such as that proposed by Mani and Bloedorn (1997).

Vectorizing a timeline into several features like what had been done here requires a fair amount of effort in feature engineering. If this feature engineering process can be removed by using temporal graphs with graph-based summarization, it may be possible to expect better summarization performance.

It is also important to explore the use of alternative evaluation measures besides ROUGE. As we depart from the traditional use of frequency statistics for summarization in this thesis, the deficiencies of a solely lexical-based measure such as ROUGE become increasingly glaring. I have argued earlier that ROUGE is unlikely to be the most suitable evaluation measure of heuristics such as TimeMMR which targets temporal similarity. It will be instructive to re-evaluate my work with a more relevant metric.

Enhancing Question-Answering with Time. Besides text summarization, another often cited use of temporal information is to enhance question-answering. There is a significant body of research exploring this, including Schockaert et al. (2006) who have tried to combine the notion of Allen's interval algebra (Allen, 1983) with facts which are lexically extracted from large resources including Wikipedia and commercial search engines. Using a probabilistic reasoning framework, they apply the facts that are gathered to try and answer temporal questions automatically. Adding a timeline to question-answering will bring about new opportunities and methodologies. For example, it could be used for answer verification, where timelines from various sources are constructed and used to check whether potential answer candidates agree with one another.

Reliability Filtering Metrics. Reliability filtering at the moment plays a big role in improving the results of timeline-assisted summarization. It is an effective stop-gap that can enable the use of temporal information for text summarization, even as the performance of the underlying temporal classification systems are not stellar as yet. It may also potentially be beneficial when attempting to integrate temporal information with other applications like question-answering as explained above. Besides the length of timelines, which relies on the premise that more inaccurate timelines tend to have larger variances in terms of accuracies, it will be useful to study if other such metrics can be derived. For example, can we make use of the transitive properties of time (Setzer and Gaizauskas, 2000; Ver-

hagen, 2005) to compute a measure of the possible number of mis-classifications in timelines?

7.2 Highlights and Summary

In this thesis, I have motivated the importance of interpreting temporal information from text. It allows us to have a better grasp of the semantic underpinnings of text, and has great potential to help enable downstream applications. With continued progress and success, it is not hard to imagine that temporal processing will become another key pillar of natural language processing, just like technologies including part-of-speech tagging or grammar parsing have become.

Building on this motivation, this thesis sets out to explore two important steps in the construction of timelines:

1. intra-sentence *E-T* temporal relation classification
2. article-wide *E-E* temporal relation classification

Combining the results from these two steps with those from timex normalization, timelines which describe the temporal relationships between basic temporal units including events and timexes are obtained.

To plug a gap in existing literature, where not much attention has been paid to the exploitation of temporal interpretation, this thesis goes on to apply the use of time in an important application — multi-document summarization. A competitive testbed that was the basis for the best performing entry to the summarization track of TAC-2011 was developed. By incorporating features derived from timelines into this testbed, together with proposed modifications to the traditional MMR algorithm, better summarization performance is observed. The timelines are derived from fully automatic systems, but inaccuracies can cause errors to be propagated down the pipeline, affecting the quality of the generated summaries. Therefore I also propose a reliability filtering metric which can help automatically decide whether the use of timelines will potentially benefit summarization. Combining these technologies, this thesis shows the efficacy and utility of applying temporal information to text summarization.

Highlighting again the key contributions in this thesis:

1. I showed that *E-T* and *E-E* temporal relationship classification can benefit from the use of more semantically-rooted features, such as dependency parsing and discourse analysis. By making use of these features for intra-sentence *E-T* and article-wide *E-E* temporal classification respectively, I developed a state-of-the-art temporal processing system which automatically derives a timeline that depicts the temporal relationships between basic temporal units in a piece of text.

2. Targeting the building and development of a event-timex temporal corpus, I demonstrated that crowdsourcing can be exploited cost-effectively for this purpose. Further, I identified a link between the structure of a sentence and how easy it is computationally and cognitively to process a event-timex pair. The easiest to process of such instances can be left out of the annotation for huge savings in annotation efforts.

3. I made effective use of automatically generated timelines to improve a state-of-the-art multi-document text summarization system. Through three innovative timeline features, as well as a modification to the traditional MMR algorithm, temporal information is shown to enhance sentence scoring and re-ordering. To deal with possible inaccuracies of the generated timelines, I also proposed an effective reliability filtering metric. The metric is used to decide whether or not temporal information should be incorporated when generating a summary.

The last section of this thesis also discussed briefly future research directions that will be both exciting and interesting to explore. I have identified possible enhancements to the proposals in this thesis to further improve the state-of-the-art for temporal processing. Beyond applying temporal information to text summarization, it is also exciting to integrate it with other applications including question-answering.

With rising community interest, as evident from the recent TempEval series of evaluation workshops, as well as the varied applications of temporal information, work on temporal processing can only get more important.

Bibliography

ACE. The ACE 2004 Evaluation Plan. `http://www.itl.nist.gov/iad/mig/tests/ace/2004/doc/ace04-evalplan-v7.pdf`, July 2004.

ACE. The ACE 2005 (ACE05) Evaluation Plan. `http://www.itl.nist.gov/iad/mig/tests/ace/ace05/doc/ace05-evalplan.v3.pdf`, October 2005.

James F Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.

Regina Barzilay, Kathleen McKeown, and Michael Elhadad. Information Fusion in the Context of Multi-document Summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics (ACL)*, pages 550–557, 1999.

Steven Bethard. A Synchronous Context Free Grammar for Time Normalization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 821–826, October 2013.

Steven Bethard and James H. Martin. CU-TMP: Temporal Relation Classification Using Syntactic and Semantic Features. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval)*, pages 129–132, June 2007.

Praveen Bysani, Vijay Bharath Reddy, and Vasudeva Varma. Modeling Novelty and Feature Combination using Support Vector Regression for Update Summarization. In *Proceedings of the 7th International Conference on Natural Language Processing (ICON)*, 2009.

Chris Callison-Burch and Mark Dredze. Creating Speech and Language Data with Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12, 2010.

Jaime Carbonell and Jade Goldstein. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 335–336, 1998.

Lynn Carlson and Daniel Marcu. Discourse tagging manual. Technical Report ISI-TR-545, Information Sciences Institute, University of Southern California, July 2001.

Bin Chen, Jian Su, Sinno Jialin Pan, and Chew Lim Tan. A Unified Event Coreference Resolution by Integrating Multiple Resolvers. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 102–110, November 2011.

Nancy Chinchor. Overview of MUC-7/MET-2. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, 1998.

Timothy Chklovski and Patrick Pantel. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 33–40, July 2004.

Samuel Clarke. *A Collection of Papers, Which Passed Between the Late Learned Mr. Leibnitz and Dr. Clarke, in the Years 1715 and 1716: Relating to the Principles of Natural Philosophy and Religion.* 1717.

Michael Collins and Nigel Duffy. Convolution Kernels for Natural Language. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, volume 14, pages 625–632, 2001.

John M. Conroy, Judith D. Schlesinger, Jeff Kubina, Peter A. Rankel, and Dianne P. O'Leary. CLASSY 2011 at TAC: Guided and Multi-lingual Summaries and Evaluation Metrics. In *Proceedings of the Text Analysis Conference (TAC)*, 2011.

Hoa Trang Dang. Overview of DUC 2005. In *Proceedings of the Document Understanding Conference Workshop on Text Summarization*, October 2005.

Hoa Trang Dang. Overview of DUC 2006. In *Proceedings of the Document Understanding Conference Workshop on Text Summarization*, October 2006.

Hoa Trang Dang and Karolina Owczarzak. Overview of the TAC 2009 Summarization Track. In *Proceedings of the Text Analysis Conference (TAC)*, November 2009.

Abhinandan Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google News Personalization: Scalable Online Collaborative Filtering. In *Proceedings of the 16th International World Wide Web Conference (WWW)*, pages 271–280, 2007.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the 5th International Conference on Lnaguage Resources and Evaluation (LREC)*, pages 449–454, May 2006.

Gianluca Demartini, Malik Muhammad Saad Missen, Roi Blanco, and Hugo Zaragoza. Entity Summarization of News Articles. In *Proceedings of the 33rd Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 798–796, 2010.

Pascal Denis and Philippe Muller. Predicting Globally-Coherent Temporal Structures from Texts via Endpoint Inference and Graph Decomposition. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAL)*, July 2011.

Leon Derczynski and Robert Gaizauskas. USFD2: Annotating Temporal Expressions and TLINKs for TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval)*, pages 337–340, July 2010.

Quang Xuan Do, Wei Lu, and Dan Roth. Joint Inference for Event Timeline Construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*, pages 677–689, July 2012.

Harold P. Edmundson. New Methods in Automatic Extracting. *Journal of ACM*, 16:264–285, April 1969.

Oren Etzioni, Michele Banko, and Michael J Cafarella. Machine Reading. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 1517–1519, July 2006.

Vanessa Wei Feng and Graeme Hirst. Text-level Discourse Parsing with Rich Linguistics Features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 60–68, July 2012.

Lisa Ferro, Laurie Gerber, Inderjeet Mani, Beth Sundheim, and George Wilson. Instruction Manual for the Annotation of Temporal Expressions. Technical report, The MITRE Corporation, 2000.

D.M. Gabbay, I. Hodkinson, M. Reynolds, and M. Finger. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Clarendon Press, 2000.

Mihai Georgescu, Dang Duc Pham, Nattiya Kanhabua, Sergej Zerr, Stefan Siersdorfer, and Wolfgang Nejdl. Temporal Summarization of Event-related Updates in Wikipedia. In *Proceedings of the 22nd International World Wide Web Conference (WWW)*, pages 281–284, 2013.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. Multi-document Summarization by Sentence Extraction. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, volume 4, pages 40–48, 2000.

Yihong Gong and Xin Liu. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *Proceedings of the 24th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 19–25, September 2001.

Claire Grover, Richard Tobin, Beatrice Alex, and Kate Byrne. Edinburgh-LTG: TempEval-2 System Description. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval)*, pages 333–336, July 2010.

Steve R. Gunn. Support Vector Machines for Classification and Regression. *ISIS Technical Report*, 14, 1998. URL `http://eprints.ecs.soton.ac.uk/6459/`.

Eun Young Ha, Alok Baikadi, Carlyle Licata, and James C. Lester. NCSU: Modeling Temporal Relations with Markov Logic and Lexical Ontology. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval)*, pages 341–344, July 2010.

Aria Haghighi and Lucy Vanderwende. Exploring Content Models for Multi-Document Summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 362–370, June 2009.

Benjamin Han and Alon Lavie. A Framework for Resolution of Time in Natural Language. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1):11–32, 2004.

Sanda Harabagiu and Finley Lacatusu. Topic Themes for Multi-document Summarization. In *Proceedings of the 28th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 202–209, August 2005.

Lisa Harper, Inderjeet Mani, and Beth Sundheim, editors. *Proceedings of the ACL Workshop on Temporal and Spatial Information Processing*, September 2001.

Pat Hayes. A Catalog of Temporal Theories. Technical Report UIUC-BI-AI-96-01, University of Illinois, 1996.

Marti A. Hearst. Multi-Paragraph Segmentation of Expository Text. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9–16, June 1994.

Iris Hendrickx, Walter Daelemans, Erwin Marsi, and Emiel Krahmer. Reducing Redundancy in Multi-document Summarization using Lexical Semantic Similarity. In *Proceedings of the Workshop on Language Generation and Summarisation (UCNLG+Sum)*, pages 63–66, August 2009.

Hugo Hernault, Helmut Prendinger, Mitsuru Ishizuka, et al. HILDA: A Discourse Parser using Support Vector Machine Classification. *Dialogue and Discourse*, 1(3), 2010.

Eduard Hovy, Chin-Yew Lin, and Liang Zhou. Evaluating DUC 2005 using Basic Elements. In *Proceedings of the Document Understanding Conference Workshop on Text Summarization*, October 2005.

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35, 2009.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. Overview of the TAC 2010 Knowledge Base Population Track. In *Proceedings of the 3rd Text Analysis Conference (TAC)*, 2010.

Immanuel Kant. *Metaphysische Anfangsgründe der Naturwissenschaft*. 1786.

Graham Katz and Fabrizio Arosio. The Annotation Of Temporal Information In Natural Language Sentences. In *Proceedings of the ACL Workshop on Temporal and Spatial Information Processing*, September 2001.

Anna Kazantseva and Stan Szpakowicz. Linear Text Segmentation Using Affinity Propagation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 284–293, July 2011.

Dan Klein and Christopher D. Manning. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*, volume 1, pages 423–430, 2003.

Anup Kumar Kolya, Asif Ekbal, and Sivaji 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)*, pages 345–350, July 2010.

Mirella Lapata and Alex Lascarides. Learning Sentence-internal Temporal Relations. *Journal of Artificial Intelligence Research*, 27(1):85–117, 2006.

Alex Lascarides and Nicholas Asher. Temporal Interpretation, Discourse Relations and Commonsense Entailment. *Linguistics and Philosophy*, 16(5):437–493, 1993.

Chin-Yew Lin. Looking for a Few Good Metrics: ROUGE and its Evaluation. In *Working Notes of the 4th NTCIR Workshop Meeting*, June 2004.

Chin-Yew Lin and Eduard Hovy. The Automated Acquisition of Topic Signatures for Text Summarization. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 495–501, August 2000.

Chin-Yew Lin and Eduard Hovy. From Single to Multi-document Summarization: A Prototype System and its Evaluation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 457–464, July 2002.

Chin-Yew Lin and Eduard Hovy. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL)*, volume 1, pages 71–78, May 2003.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. A PDTB-styled End-to-End Discourse Parser. *Natural Language Engineering*, FirstView:1–34, February 2013.

Maofu Liu, Wenjie Li, and Huijun Hu. Extractive Summarization Based on Event Term Temporal Relation Graph and Critical Chain. In *Information Retrieval Technology*, volume 5839 of *Lecture Notes in Computer Science*, pages 87–99. Springer Berlin Heidelberg, 2009.

Hector Llorens, Naushad UzZaman, and James F Allen. Merging Temporal Annotations. In *Proceedings of the 19th International Symposium on Temporal Representation and Reasoning (TIME)*, pages 107–113, 2012.

Hector Llorens, Estela Saquete, and Borja Navarro-Colorado. Applying Semantic Knowledge to the Automatic Processing of Temporal Expressions and Events in Natural Language. *Information Processing and Management*, 49(1):179–197, January 2013.

Inderjeet Mani. *Automatic Summarization*, volume 3. John Benjamins Publishing Company, 2001.

Inderjeet Mani and Eric Bloedorn. Multi-document Summarization by Graph Search and Matching. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI)*, pages 622–628, July 1997.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. Machine Learning of Temporal Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 753–760, July 2006.

William C. Mann and Sandra A. Thompson. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8(3):243–281, 1988.

Daniel Marcu. From Discourse Structures to Text Summaries. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, volume 97, pages 82–88, July 1997.

Winter Mason and Duncan J. Watts. Financial Incentives and the Performance of Crowds. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 77–85, 2009.

Ryan McDonald. *A Study of Global Inference Algorithms in Multi-document Summarization*, volume 4425 of *Lecture Notes In Computer Science*, pages 557–564. Springer, 2007.

Kathleen McKeown and Dragomir R. Radev. Generating Summaries of Multiple News Articles. In *Proceedings of the 18th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 74–82, 1995.

Seyed Abolghasem Mirroshandel, Gholamreza Ghassem-Sani, and Mahdy Khayyamian. Using Syntactic-Based Kernels for Classifying Temporal Relations. *Journal of Computer Science and Technology*, pages 68–80, January 2011.

Alessandro Moschitti. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In *Proceedings of the 17th European Conference on Machine Learning (ECML)*, September 2006a.

Alessandro Moschitti. Making Tree Kernels Practical for Natural Language Learning. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, volume 6, pages 113–120, 2006b.

Ani Nenkova and Kathleen McKeown. Automatic Summarization. *Foundations and Trends in Information Retrieval*, 5:103–233, 2011.

Isaac Newton. *Philosophie Naturalis Principia Mathematica*. July 1687.

Jun-Ping Ng and Min-Yen Kan. Improved Temporal Relation Classification using Dependency Parses and Selective Crowdsourced Annotations. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 2109–2124, December 2012.

Jun-Ping Ng, Praveen Bysani, Ziheng Lin, Min-Yen Kan, and Chew-Lim Tan. SWING: Exploiting Category-Specific Information for Guided Summarization. In *Proceedings of the Text Analysis Conference (TAC)*, December 2011.

Jun-Ping Ng, Praveen Bysani, Ziheng Lin, Min-Yen Kan, and Chew-Lim Tan. Exploiting Category-Specific Information for Multi-Document Summarization. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 2093–2108, December 2012.

Jun-Ping Ng, Min-Yen Kan, Ziheng Lin, Wei Feng, Bin Chen, Jian Su, and Chew-Lim Tan. Exploiting Discourse Analysis for Article-Wide Temporal Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12–23, October 2013.

Jun-Ping Ng, Yan Chen, Min-Yen Kan, and Zhoujun Li. Exploiting Timelines to Enhance Multi-document Summarization. In *Proceedings of the 52nd Annual Meeting on Association for Computational Linguistics (ACL) (to appear)*, June 2014.

Paul Over and James Yen. An Introduction to DUC 2004 Intrinsic Evaluation of Generic New Text Summarization Systems. In *Proceedings of the Document Understanding Conference Workshop on Text Summarization*, May 2004.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. *English Gigaword Fifth Edition*. Linguistic Data Consortium, 2011.

Rebecca J. Passonneau, Ani Nenkova, Kathleen McKeown, and Sergey Sigelman. Applying the Pyramid Method in DUC 2005. In *Proceedings of the Document Understanding Conference Workshop on Text Summarization*, October 2005.

Daniele Pighin and Alessandro Moschitti. On Reverse Feature Engineering of Syntactic Tree Kernels. In *Proceedings of the 14th Conference on Natural Language Learning (CoNLL)*, August 2010.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, May 2008.

James Pustejovsky, José Castano, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proceedings of the 5th International Workshop on Computational Semantics (IWCS)*, 2003a.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. The TIMEBANK corpus. In *Proceedings of Corpus Linguistics*, pages 647–656, March 2003b.

Dragomir R. Radev. A Common Theory of Information Fusion from Multiple Text Sources Step One: Cross-Document Structure. In *Proceedings of the 1st SIGdial Workshop on Discourse and Dialogue*, volume 10, October 2000.

Robert Rynasiewicz. Newton's Views on Space, Time, and Motion. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2012 edition, 2012.

Frank Schilder and Ravikumar Kondadadi. FastSum: Fast and Accurate Query-based Multi-document Summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 205–208, June 2008.

Steven Schockaert, David Ahn, Martine De Cock, and Etienne E Kerre. *Question Answering with Imperfect Temporal Information*, pages 647–658. Springer, 2006.

Andrea Setzer and Robert Gaizauskas. Building a Temporally Annotated Corpus for Information Extraction. In *Proceedings of the LREC-2000 Workshop on Information Extraction Meets Corpus Linguistics*, 2000.

Andrea Setzer and Robert Gaizauskas. A Pilot Study on Annotating Temporal Relations In Text. In *Proceedings of the ACL Workshop on Temporal and Spatial Information Processing*, September 2001.

Andrea Setzer, Robert Gaizauskas, and Mark Hepple. Using Semantic Inferences for Temporal Annotation Comparison. In *Proceedings of the 4th International Workshop on Inference in Computational Semantics (ICoS)*, September 2003.

Victor S. Sheng, Foster Provost, and Panos G. Ipeirotis. Get Another Label? Improving Data Quality and Data Mining using Multiple, Noisy Labelers. In *Proceeding of the 14th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 614–622, 2008.

Eduard F. Skorochod'Ko. Adaptive Method of Automatic Abstracting and Indexing. In *Proceedings of the International Federation for Information Processing Congress (IFIP)*, pages 1179–1182, 1972.

Jannik Strötgen and Michael Gertz. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013.

Beth Sundheim. Overview of results of the MUC-6 evaluation. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, pages 423–442. Association for Computational Linguistics, 1996.

Naushad Uzzaman and James F. Allen. TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval)*, pages 276–283, July 2010.

Naushad Uzzaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James F. Allen, and James Pustejovsky. SemEval-2013 Task 1: TEMPEVAL-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval)*, June 2013.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*, chapter 5. Springer, 1999.

Marc Verhagen. Temporal Closure in an Annotation Environment. *Language Resources and Evaluation*, 39(2-3):211–241, 2005.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Jessica Moszkowicz, and James Pustejovsky. The TempEval Challenge: Identifying Temporal Relations in Text. *Language Resources and Evaluation*, 43(2): 161–179, 2009.

Marc Verhagen, Roser Saurí, Tommaso Caselli, and James Pustejovsky. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval)*, pages 57–62, July 2010.

Xiaojun Wan. TimedTextRank: Adding the Temporal Dimension to Multi-Document Summarization. In *Proceedings of the 30th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 867–868, July 2007.

Bonnie Webber. D-LTAG: Extending Lexicalized TAG to Discourse. *Cognitive Science*, 28(5):751–779, 2004.

Mingli Wu. *Investigations on Temporal-Oriented Event-Based Extractive Summarization*. PhD thesis, Hong Kong Polytechnic University, 2008.

Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. Jointly Identifying Temporal Relations with Markov Logic. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL) and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (AFNLP)*, pages 405–413, August 2009.

Renxian Zhang, You Ouyang, and Wenjie Li. Guided Summarization with Aspect Recognition. In *Proceedings of the Text Analysis Conference (TAC)*, November 2011.