# Supervised Extractive Summarisation of News Events

Stuart William Mackie

Submitted in fulfilment of the requirements for the Degree of

*Doctor of Philosophy*

School of Computing Science

College of Science and Engineering

University of Glasgow

March 2018


University of Glasgow | School of Computing Science

# Abstract

This thesis investigates whether the summarisation of news-worthy events can be improved by using evidence about entities (i.e. people, places, and organisations) involved in the events. More effective event summaries, that better assist people with their news-based information access requirements, can help to reduce information overload in today's 24-hour news culture.

Summaries are based on sentences extracted verbatim from news articles about the events. Within a supervised machine learning framework, we propose a series of entity-focused event summarisation features. Computed over multiple news articles discussing a given event, such entity-focused evidence estimates: the importance of entities within events; the significance of interactions between entities within events; and the topical relevance of entities to events.

The statement of this research work is that augmenting supervised summarisation models, which are trained on discriminative multi-document newswire summarisation features, with evidence about the named entities involved in the events, by integrating entity-focused event summarisation features, we will obtain more effective summaries of news-worthy events.

The proposed entity-focused event summarisation features are thoroughly evaluated over two multi-document newswire summarisation scenarios. The first scenario is used to evaluate the retrospective event summarisation task, where the goal is to summarise an event to-date, based on a static set of news articles discussing the event. The second scenario is used to evaluate the temporal event summarisation task, where the goal is to summarise the changes in an ongoing event, based on a time-stamped stream of news articles discussing the event.

The contributions of this thesis are two-fold. First, this thesis investigates the utility of entity-focused event evidence for identifying important and salient event summary sentences, and as a means to perform anti-redundancy filtering to control the volume of content emitted as a summary of an evolving event. Second, this thesis also investigates the validity of automatic summarisation evaluation metrics, the effectiveness of standard summarisation baselines, and the effective training of supervised machine learned summarisation models.

# Acknowledgements

First, I would like to acknowledge the sterling efforts of my doctoral supervisors, Professor Iadh Ounis, Dr. Craig Macdonald, and Dr. Richard McCreadie. Under your supervision, throughout a B.Sc project, Honours project, Masters project, two summer internships, and a Ph.D, it's been a privilege to work with you over the past 7 years. Thanks for believing in me, for teaching me how to rigorously pursue scientific investigation, and for infusing in me an appreciation of Information Retrieval.

Thanks are also due to my examiners, Dr. Michael Oakes and Dr. Bjørn Sand Jensen, for taking time to critically engage with my research, and providing their insightful corrections.

I would also like to acknowledge the Ph.D students, post-docs, and academic staff that have contributed to my experiencing a stimulating research environment over the course of this Ph.D. In particular, past and present colleagues in the Terrier Team: Graham McDonald, Jarana Manotumruksa, Anjie Fang, Xiao Yang, Ting Su, Eric Wang, Dyaa Albakour, Nut Limosopathan, Eugene Kharitonov, Romain Deveaud, Saúl Vargas, Sean Moran, and Karin Sim Smith. My colleagues at the Glasgow Information Retrieval Group: Stewart Whiting, Rami Alkhawaldeh, James McMinn, Phil McParlane, Jesus Rodriguez, Fajie Yuan, Sean McKeown, Yashar Moshfeghi, Leif Azzopardi, Joemon Jose, and Jeff Dalton. Further, I acknowledge the comradery and high-quality banter of the following office mates: David Maxwell, Horatiu Bota, David Paule, Colin Wilkie, Fatma Elsafoury, and Luisa Rebelo Pinto.

This endeavour would have not been possible without the support from Anne and William Bremner, and Kenneth Mackie. My heartfelt thanks for being there when I needed you most.

*for Tevhide, Azra, Yusuf, and Zara*

---

I acknowledge the financial support of EPSRC Doctoral Training Grant 1509226 (£45,176), and the financial support of EC SMART Project FP7-287583 (£17,626).

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The online reporting of news events are the subject of intense interest by the general public, forming part of society's collective memory (Yeung and Jatowt, 2011). Traditionally, news reports were consumed via print, radio, and television. Recently, industry-based surveys have shown that 41% of U.K. adults and 38% of U.S. adults now access news via internet-based publications (Ofcom, 2015; Pew Research, 2016). Considering age demographics, there is a marked shift within younger generations away from print, radio, and television sources towards internet-based consumption of news, and one third of 18–24 year-olds use social media as their primary source of news (Reuters Institute, 2017). Today, there is a tremendous volume of news content being published online from a multitude of sources. For example, the Google News service provides online access to over 75,000 news sources (Google, 2016), readily accessible via a website[1] and smartphone application[2].

However, given such easy access to large volumes of news reporting, we can very quickly become overloaded by the amount of information available to us, finding ourselves with an overwhelming surplus of news content (Holton and Chyi, 2012). A consequence of information overload[3] is information fatigue (Edmunds and Morris, 2000), we may find it increasingly difficult to obtain an overview of a news event, or to follow ongoing developments within a breaking news event over time. Methods that improve news-based information access technologies, specifically automatic text summarisation (Jones, 2007; Nenkova and McKeown, 2011; Lloret and Palomar, 2012; Saggion and Poibeau, 2013; Torres-Moreno, 2014), so that

[1] news.google.co.uk

[2] https://play.google.com/store/apps/

[3] en.wikipedia.org/wiki/Information_overload

we can more effectively obtain an up-to-date overview of a news event, or more effectively track the latest developments in the evolution of a news event, are the subject of this thesis.

Specifically, this thesis investigates the task of event summarisation (Aslam et al., 2013). Event summarisation presents a challenging information access problem, where users wish to be informed about the essential details regarding news-worthy events. We loosely define an "event" as a story that is reported in the news. Given a collection of newswire articles that discuss an event, the aim of event summarisation is to derive a succinct and salient textual narrative of the important aspects of the event. Commonly, an event summary is constructed by extracting whole sentences verbatim from the news articles discussing the event, a process known as extractive multi-document newswire summarisation (Hong et al., 2014).

We address two separate event summary scenarios. First, where an event summary is based on a fixed collection of news documents (which may span several days), the aim is to provide a summary of the event to-date, i.e. an overview or retrospective summary. Second, where an event summary is based on a stream of incoming documents, the aim is to summarise the developments within an on-going news event, i.e. to provide an evolving temporal summary. In this thesis, we seek to validate our claims over both event summary scenarios.

Event summarisation systems aim to provide users with a means to digest important information about events they care about. Users should rightly expect an event summarisation system to provide high-quality event summaries. Industry-based news content providers often present journalist-curated summaries of news events to users in commercial contexts. For example, the BBC News website regularly reports events "as it happened"[1], and the Guardian website contains rolling coverage of "politics live"[2] events. However, while manually-authored event summaries may be of a high quality, the production of such summaries exhibits cost and scalability challenges. Specifically, expensive human resources are required to author the summaries, which limits the number of events that can be summarised.

In today's 24-hour news culture, given the high-volumes of online reporting of news-worthy events, automatic event summarisation systems that algorithmically summarise events have recently been offered to online news consumers. For example, smartphone applications such as Yahoo News Digest[3] aim to provide users with a concise overview of current events.

[1] bbc.co.uk/news/live/uk-39355505

[2] theguardian.com/politics/series/politics-live-with-andrew-sparrow

[3] uk.mobile.yahoo.com/newsdigest

However, the quality of summaries produced by automatic event summarisation systems is important. Users can easily switch to other websites or smartphone applications if they feel that the quality of the summaries being offered is not acceptable – potentially leading to a loss of customers or advertising revenue. This motivates us to propose algorithms that aim to produce effective summaries, i.e. summaries that users judge to be compelling digests of news events. In this thesis, we argue that standard multi-document newswire summarisation algorithms are not suitable for the task of producing effective summaries of evolving events.

As argued by Jones (1998), we make clear statements with regards to context factors – for whom we produce summaries for, and for what purpose the summaries are intended. The target audience for our event summaries is the (non-expert) general public, i.e. we do not produce summaries targeted for a specific domain such as crisis management (Carver and Turoff, 2007). Further, our purpose for producing event summaries is to inform people about an event they are interested in, i.e. we do not produce summaries intended to support processes such as complex search tasks (McLellan et al., 2001; Mani et al., 2002). For whom, and for what purpose we produce summaries, is reflected in our summarisation evaluation methodology, which is empirically validated with our stated target audience via a user-study.

In this thesis, for addressing the task of event summarisation, we investigate the application of supervised machine learning techniques (Hastie et al., 2009; Witten et al., 2016). Machine learning techniques were originally proposed for text summarisation by Kupiec et al. (1995), and machine learning techniques are known to provide an effective framework for the task of multi-document newswire summarisation (Ouyang et al., 2011a; Oliveira et al., 2016).

Our main argument is that events are about entities (i.e. people, places, and organisations). This thesis claims that effective event summaries can be constructed by leveraging information about the entities involved in the event being summarised. Such evidence includes statistics about the importance of entities within an event, the significance of interactions between entities, and the topical relevance of entities to the event. A machine learning framework provides a principled methodology to integrate such evidence (i.e. features) about entities into the summarisation process. This enables us to empirically validate our claims regarding the utility of entity-focused evidence, with respect to event summarisation effectiveness.

## 1.1 Challenges

To operationalise supervised summarisation experiments, in order to validate our claims, five specific challenges are addressed in this thesis. The first challenge we address regards experimental validity concerns relating to summarisation evaluation. Typically, automatic summarisation evaluation metrics, such as ROUGE (Lin, 2004), are used to evaluate system-produced summary text(s) by comparing them to human-authored exemplar summaries. However, automatic evaluation of summary text(s) remains a controversial topic within the summarisation community (Sjöbergh, 2007; Owczarzak et al., 2012; Rankel et al., 2013). To validate automatic evaluation methods with our target audience, we establish the correlation of automatic summarisation evaluation metrics with non-expert crowd-sourced judgements for the linguistic quality of a summary. This provides us with a measure of confidence that empirical observations obtained via automatic summarisation evaluation metrics are robust.

The second challenge we address in this thesis is the identification of baseline summarisation algorithms. We re-implement newswire summarisation algorithms from the literature (Hong et al., 2014), thoroughly exploring algorithm design choices. In this thesis, we argue that such algorithms can be improved to provide stronger baselines for use in empirical evaluations. Further, we argue that such algorithms provide discriminative features for training supervised summarisation models. In our machine learning experiments, we augment newswire summarisation features with our proposed entity-focused event summarisation features. Through experimentation, we can then observe any gains in summarisation effectiveness obtained by adding entity-focused features to baseline supervised summarisation models, which would validate our claim that utilising evidence about entities results in effective summaries of events.

The third challenge we address relates to practical matters involved in training supervised summarisation models. First, we require labelled training data. We investigate a range of methods to automatically label such training data. Supervised summarisation models, trained using different automatically induced labels, are evaluated to ascertain which labelling method(s) result in the most effective supervised summarisation models. Second, we investigate various types of learners, including regression (Hastie et al., 2009; Witten et al., 2016) and learning-to-rank (Liu, 2009).

Having validated summarisation evaluation metrics, identified suitable summarisation algorithms to provide strong baselines for evaluations and discriminative features for machine learned models, obtained high-quality labelled training data, and evaluated different types of learners, we then proceed to investigate our claims regarding the utility of entity-focused evidence, over both the retrospective summarisation task and temporal summarisation task.

Specifically, the fourth challenge we address is the formulation and evaluation of a series of entity-focused event summarisation features. Given an event that is discussed in a collection of news articles, we compute estimates of entity importance, entity–entity interaction, and entity–event relevance, with respect to the entities involved in the news event. Such entity-focused features are then evaluated within a supervised machine learned summarisation framework via a feature group ablation study. In particular, supervised machine learned summarisation models are trained using a set of baseline features. Then, we train further supervised summarisation models where the baseline features group is augmented with entity importance features, entity–entity interaction features, entity–event relevance features, and combinations thereof. We empirically validate our claim, that entity-focused event summarisation features can be used to derive effective retrospective summaries of events, if any of the models that have been augmented with entity-focused features exhibit higher summarisation effectiveness than the models trained using only baseline newswire summarisation features.

Finally, the fifth challenge we address is how to produce effective temporal summaries of evolving events. This corresponds to our second event summary scenario, where the aim is to summarise changes within an event based on a stream of incoming news articles. To test our claim that entity-focused evidence can be used to produce effective summaries of evolving news events, we propose temporal variants (i.e. time-based extensions) of our entity-focused event summarisation features. Specifically, we derive new features that represent changes (over time) in entity importance, entity–entity interaction, and entity–event relevance. We then evaluate time-based entity-focused event summarisation features using a feature group ablation study within a supervised machine learning framework. Further, in retrospective summarisation experiments the length of the summary to be generated is known ahead of time (Over et al., 2007). To effectively address the temporal summarisation task, we argue that, as real-world news events exhibit temporal patterns of activity and inactivity, to provide an effective temporal summary of an evolving event we must select a variable number

of sentences at event-determined periodic time-intervals. That is, the volume of sentences emitted over time to form the temporal summary should mirror the bursty nature of events. We investigate entity-focused anti-redundancy techniques to filter summary sentences, and control for the number of sentences emitted over time to form evolving event summaries.

The five challenges we have identified are addressed in five contributions chapters (3–7). Based on these five challenges, we now formally state our thesis and supporting hypotheses.

## 1.2 Thesis Statement

This thesis states that events are about entities, and to offer users effective temporal summaries of evolving news events we must explicitly model the importance, interactions, and relevance of named entities within the events being summarised, and use such entity-focused evidence to identify newswire sentences to include in summaries of events, and also to vary the length of the summary over time according to the entity-centric life-cycle of news-worthy events.

In particular, as news events are the subject of intense interest to the general public, and traditional multi-document summarisation approaches are not suitable for summarising evolving news events, to alleviate information overload and offer users effective summaries of events they care about, constructed over time as events develop by identifying and extracting a variable number of important and salient sentences from multiple newswire articles discussing the events, where the metrics used to measure summarisation effectiveness have been shown to correlate with the user's judgements of summary quality, and effectiveness comparisons are made to strong newswire summarisation baselines, within supervised machine learned summarisation models which are trained using high-quality automatically labelled training data, we should augment discriminative multi-document newswire summarisation features with entity-focused event summarisation features, which are derived by estimating evidence about the people, places, and organisations involved in the events being summarised.

### Hypotheses

In this thesis, based on the five challenges we have identified, we form the following five hypotheses, with each hypothesis experimentally validated in our five contributions chapters.

**Hypothesis 1.** We hypothesise that automatic summarisation evaluation metrics, which measure content coverage with respect to a gold-standard summary, exhibit strong correlation with non-expert crowd-sourced judgements for the linguistic quality of summary text(s).

**Hypothesis 2.** We hypothesise that the effectiveness of standard multi-document newswire summarisation algorithms can be improved by varying algorithm design choices.

**Hypothesis 3.** We hypothesise that supervised machine learned summarisation models based on regression techniques, that exhibit state-of-the-art effectiveness, can be trained on discriminative features, derived from standard multi-document newswire summarisation algorithms, using automatically labelled training data induced from gold-standard summaries.

**Hypothesis 4.** By learning a ranking function over newswire sentences, optimising for the importance of entities within the event, the significance of interactions between entities within the event, and the topical relevance of entities to the event, we hypothesise that the sentences that are available for inclusion into the event summary can be effectively ranked by their summary worthiness, using a supervised summarisation model trained using such entity-focused event summarisation features, augmented with document summarisation features.

**Hypothesis 5.** As real-world news events exhibit temporal patterns of activity and inactivity, reflecting ongoing developments in the evolution of the event over time, we argue that selecting a fixed number of summary sentences at pre-determined periodic time-intervals is non-optimal, and we hypothesise that entity-focused event summarisation features can be used to derive effective anti-redundancy methods, and that an effective temporal summary of an evolving event consists of a variable number of sentences selected at event-determined periodic time-intervals, mirroring event evolution over time.

## 1.3 Thesis Contributions

The contributions of this thesis are as follows:

In Chapter 3, via a crowd-sourced user-study, we confirm and quantify the validity of automatic summarisation evaluation metrics. Automatic metrics, measuring content coverage, are shown to exhibit correlation with non-expert crowd-sourced manual judgements for

the linguistic quality of a summary text. This demonstrates that automatic summarisation evaluation methods are accurately aligned with user expectations regarding summary quality.

In Chapter 4, we demonstrate that the effectiveness of standard unsupervised summarisation algorithms can be significantly improved by thoroughly exploring algorithm design choices. As such, we show that standard summarisation algorithms can still provide strong baselines for the empirical evaluation of summarisation systems. Further, we present evidence that such standard summarisation algorithms may provide a set of discriminative features for supervised machine learned summarisation models.

In Chapter 5 we demonstrate that supervised machine learned summarisation models, that exhibit state-of-the-art effectiveness for the task of multi-document newswire summarisation, can be learned using automatically induced training data. We evaluate a range of methods for automatically labelling training data, and evaluate a range of machine learning model types, forming a series of best practice recommendations. We also demonstrate that a set of standard baselines can provide effective features for supervised summarisation models.

Further, in Chapter 6 and Chapter 7, we demonstrate the utility of entity-focused event evidence for identifying important and salient event summary sentences. We propose a set of entity-focused event summarisation features, based on estimates of entity importance, entity–entity interaction, and entity-event relevance. Using evidence of the importance, significance, and relevance of entities to events, in combination with standard document summarisation features, we demonstrate that such supervised summarisation models can be used to produce effective summaries of news-worthy events.

Furthermore, in Chapter 7, we show that for the task of temporal summarisation, varying the summary length over time to reflect the bursty nature of events results in more effective summaries, compared with selecting a fixed-length summary over time. Specifically, we demonstrate the utility of entity-focused event evidence as means to perform anti-redundancy filtering, providing a means to control the volume of content emitted as a summary of an evolving event. We also show that a classifier can be trained to accurately filter (i.e. reduce) the number of sentences that are taken as input to temporal summarisation systems.

## 1.4 Origins of the Material

The material in this thesis is based on the following publications:

- Chapter 3 – The experimental framework for a crowd-sourced user-study in this chapter, to manually evaluate the quality of system-produced summary texts, is based on the work undertaken in Mackie et al. (2014b). Further, the crowd-sourced user-study undertaken in this chapter is an extension of the work reported in Mackie et al. (2016).

- Chapter 4 – The reproduction and evaluation of multi-document newswire summarisation baselines presented in this chapter is an extension of the work reported in Mackie et al. (2016). A similar study was conducted in Mackie et al. (2014a), reproducing and empirically evaluating baselines within the context of microblog summarisation.

- Chapter 5 – The set of summarisation baseline algorithms used as features within supervised machine learned summarisation models were first examined in Mackie et al. (2016).

- Chapter 7 – The material presented in this chapter is an extension of work undertaken in the context of the TREC Temporal Summarisation Track (McCreadie et al., 2013, 2015).

## 1.5 Thesis Outline

The remainder of this thesis is organised as follows:

- Chapter 2 – In this chapter, we review the summarisation research literature. We begin by describing the various summarisation tasks, and discussing contextual factors that influence the design, implementation, and evaluation of summarisation systems. Next, we review the literature regarding summarisation evaluation. Following this, we review the baseline algorithms and state-of-the-art systems for multi-document news summarisation.

- Chapter 3 – In the first of five contribution chapters, we argue that for an automatic summarisation evaluation metric to be valid, it should exhibit a degree of correlation with manual summarisation evaluation judgements regarding the linguistic quality of a summary. We investigate and quantify the correlation of automatic summarisation evaluation metrics with crowd-sourced manual judgements for summary quality.

- Chapter 4 – In this chapter, we hypothesise that standard summarisation algorithms can be improved to provide stronger baselines, and further, we argue that standard summarisation algorithms can be used as discriminative features for training supervised machine learned summarisation models. As such, we re-implement and evaluate the effectiveness of several unsupervised multi-document newswire summarisation algorithms.

- Chapter 5 – In this chapter, we investigate the effective training of supervised summarisation models. We investigate various methods to automatically label training data, evaluate a range of machine learning techniques, and evaluate baseline algorithms from the previous chapter as features. We argue that effective labels, regression-based learners, and features derived from standard baselines, can be combined to train state-of-the-art models.

- Chapter 6 – In this chapter, we investigate the retrospective summarisation task. We argue that entity-focused event summarisation features can be used to derive effective summaries of events. We propose and evaluate a series of entity-focused event summarisation features for use in a supervised summarisation framework.

- Chapter 7 – In this chapter, we investigate the temporal summarisation task. We propose a set of query-based and entity-focused features specific to the nature of the task. Further, we investigate the utility of entity-evidence for performing anti-redundancy filtering.

- Chapter 8 – Finally, we highlight the contributions of this thesis, we summarise the conclusions of this thesis, and illustrate directions for future work.

# Chapter 2

# Automatic Text Summarisation

Automatically producing effective summaries of text documents is a challenging problem, with many different summarisation systems and evaluation methodologies described in the summarisation research literature (Jones, 2007; Nenkova and McKeown, 2011; Lloret and Palomar, 2012; Saggion and Poibeau, 2013; Torres-Moreno, 2014). Automatic text summarisation has been a subject of research addressed within the fields of Natural Language Processing (Manning and Schütze, 2001; Jurafsky and Martin, 2009) and Information Retrieval (Croft et al., 2010; Büttcher et al., 2010) for over 50 years (Luhn, 1958). Automatic text summarisation is by definition an information reduction process (Jones, 1998):

**Definition 2.1.** *"a reductive transformation of source text to summary text through content reduction by selection and/or generalisation on what is important in the source."*

The aim is to convey the essential information of a document, or a set of documents, by identifying the most important and salient information within the source document(s), perhaps in response to a user's specific information need (typically expressed as a query).

In this chapter, we review the automatic text summarisation literature, with a specific focus on newswire summarisation. We first introduce the main automatic text summarisation tasks, providing a taxonomy of different types of summaries. We next consider the context factors that influence automatic text summarisation. We then discuss summarisation evaluation, describing the challenges in evaluating text summaries, and provide an overview of the datasets and evaluation methodologies used to empirically evaluate automatic text summarisation in this thesis. Further, we review the baseline algorithms and state-of-the-art systems

for the task of multi-document newswire summarisation. Our literature review is concluded by examining previous work related to the specific task of summarising evolving news events.

### Chapter Outline

This chapter is organised as follows:

- Section 2.1 introduces a taxonomy of summarisation tasks: abstraction and extraction; indicative and informative summaries; single document and multi-document summaries; generic and query-biased summaries; retrospective, update, and temporal summarisation.
- Section 2.2 introduces the contextual factors that influence the design, implementation and evaluation of text summarisation systems, namely: input; purpose; and output factors.
- Section 2.3 describes the standard datasets and specific methodologies used for empirically evaluating the effectiveness automatic text summarisation systems.
- Section 2.4 reviews the baseline algorithms and state-of-the-art systems for the task of multi-document newswire summarisation, including supervised machine learned models.

## 2.1 Summarisation Tasks

We begin by introducing the various automatic text summarisation tasks, providing a taxonomy of different types of summaries. Typically, automatic text summarisation systems are designed, implemented, and evaluated for a specific summarisation task or summary style. Indeed, within the summarisation literature, a series of summarisation tasks and summary types have evolved over time. We now enumerate such tasks and styles, and discuss the particular summarisation approaches evaluated in this thesis.

Principally, automatic text summarisation systems may produce text summaries that are either abstractive or extractive in nature. While based on the input document(s) being summarised, an abstractive summarisation system generates new natural language to form the summary (e.g. McKeown and Radev, 1995; Hovy and Lin, 1998; Genest and Lapalme, 2012), using natural language generation techniques (Gatt and Krahmer, 2017). Recent abstractive summarisation systems, based on neural networks (Goodfellow et al., 2016; Goldberg, 2016),

have begun to show promising results (e.g. See et al., 2017; Li et al., 2017). However, the generation of effective abstracts remains a very challenging problem (Torres-Moreno, 2014). Other approaches to abstractive summarisation, which do not involve natural language generation, often re-use some of the original input text in some manner. For example, the following techniques are considered abstractive: sentence compression, which involves deletion of words or fragments of sentences (e.g. Zajic et al., 2007; Clarke and Lapata, 2007); sentence revision, where words or fragments of sentences are replaced with other text (e.g. Mani et al., 1999; Nenkova, 2008); and sentence fusion, that attempts to join together words or fragments of different sentences (e.g. Barzilay and McKeown, 2005; Filippova and Strube, 2008).

In contrast, an extractive text summarisation system seeks to identify the most important and salient sentences from within the document(s) being summarised, then selects and concatenates (verbatim) a subset of those sentences to form the summary text. Indeed, a majority of the state-of-the-art automatic text summarisation systems described in the summarisation literature are extractive (Nenkova and McKeown, 2011). As opposed to abstractive summarisation techniques, extractive summaries often exhibit a reasonable degree of readability and correct grammar, assuming the source documents are well-authored, whereas abstractive techniques are limited by current natural language generation technology. However, the extractive text summarisation paradigm is bounded by the input documents, i.e. it is not possible to include information in a summary that does not appear in the documents being summarised, whereas abstractive techniques could potentially generate new information for the summary, based on inference over the input documents or querying knowledge bases. In our experiments in this thesis, we focus exclusively on the extractive summarisation task.

Pioneering work by Luhn (1958) and Edmundson (1969) set the direction for automatically summarising documents via extraction. Figure 2.1 provides an illustration of the intuitions underpinning the extractive summarisation paradigm. In Figure 2.1, we show an example document from one of the standard datasets commonly used in the summarisation literature to empirically evaluate automatic text summarisation systems (Over et al., 2007). The newswire document, published by the Associate Press in October 1998, discusses the diplomatic crisis that arose from the arrest of Chilean dictator Augusto Pinochet in London. In Figure 2.1, we manually highlight some important and salient sentences, which might be suitable for inclusion into a summary of this document. When performing extractive sum-

<DOC>
<DOCNO>APW19981019.0098</DOCNO>
<DOCTYPE>NEWS</DOCTYPE>
<TXTTYPE>NEWSWIRE</TXTTYPE>
<TEXT>
Britain has defended its arrest of Gen. Augusto Pinochet, with one lawmaker saying that Chile's claim that the former Chilean dictator has diplomatic immunity is ridiculous. Chilean officials, meanwhile, issued strong protests and sent a delegation to London on Sunday to argue for Pinochet's release. The former strongman's son vowed to hire top attorneys to defend his 82-year-old father, who ruled Chile with an iron fist for 17 years. British police arrested Pinochet in his bed Friday at a private London hospital in response to a request from Spain, which wants to question Pinochet about allegations of murder during the decade after he seized power in 1973. Pinochet had gone to the hospital to have a back operation Oct. 9. "The idea that such a brutal dictator as Pinochet should be claiming diplomatic immunity I think for most people in this country would be pretty gut-wrenching stuff," Trade Secretary Peter Mandelson said in a British Broadcasting Corp. television interview Sunday. Home Office Minister Alun Michael acknowledged Sunday that Pinochet entered Britain on a diplomatic passport, but said, "That does not necessarily convey diplomatic immunity." The Foreign Office said only government officials visiting on official business and accredited diplomats have immunity. Pinochet has been a regular visitor to Britain, generally without publicity. His arrest this time appeared to reflect a tougher attitude toward right-wing dictators by Prime Minister Tony Blair's Labor Party government, which replaced a Conservative Party administration 18 months ago and promised an "ethical" foreign policy. [...]
</TEXT>
</DOC>

**Figure 2.1:** Document number "APW19981019.0098", from topic "d30003t" of the DUC 2004 dataset, which discusses the October 1998 arrest of Chilean dictator Gen. Augusto Pinochet in London. We show the first ten sentences of the document, published by the Associated Press. Further, potential summary sentences to extract are highlighted in red, such as the leading sentences, and also informative sentences from within the article.

marisation, this is the primary function of an automatic text summarisation system – to identity the most important and salient information within a document (Nenkova and McKeown, 2011). As such, the extractive summarisation task is often formulated as a sentence ranking and selection problem. In particular, there is typically some component within extractive summarisation systems that attempts to ensure that the sentences selected for inclusion into the summary do not exhibit a high-degree of textual overlap (i.e. redundancy).

Further, we note two important types of text summary, namely: indicative summaries; and informative summaries (Edmundson, 1969). Specifically, a summary may be produced to indicate what a document is about, referred to as an indicative summary. Conversely, a summary may be produced to provide an informative proxy for the original document, referred to as an informative summary. Given an indicative summary, we obtain an understanding as to what the document might be about, but would still have to read the document to understand the important aspects of the contents. Given an informative summary, it should not be necessary to read the whole document to understand the important aspects.

Furthermore, one of the main distinctions in summarisation tasks relates to how many documents are presented as input to the summarisation system. Within the context of newswire summarisation (Over et al., 2007; Hong et al., 2014), the input to the summarisation process is either a single news article, or multiple articles discussing the same (or topically related) news events. Such tasks are referred to in the summarisation literature as: single document summarisation; and multi-document summarisation.

Moreover, automatic text summaries may be general in nature, attempting to convey what is important and salient from within the input documents. For example, a generic summary may be observed via online news aggregation websites[1], where a short extract is shown to users to illustrate the contents of the newswire article. This task is referred to as generic summarisation in the literature. Alternatively, summaries may be produced in response to a specific information need (typically expressed as a query), focusing only on information about certain topics from within the input documents. The canonical example is the snippets displayed on web search engine results pages (Tombros and Sanderson, 1998). In the summarisation literature, this task is referred to as query-biased or query-focused summarisation.

The final categorisation evident in the summarisation literature relates to the temporal aspects of the source document(s) being summarised. In particular, the input documents may be from a static (i.e. historical) collection of documents that does not change over time. Specifically, a fixed batch of documents is presented as input to the summarisation system, and the process of summarisation is batch-like in nature – neither the source documents nor the summary is updated. In this thesis, we refer to this task as retrospective summarisation.

However, the summarisation task may involve a temporally dynamic collection of documents. In scenarios where the user is interested in following or tracking the evolution of information within documents over time, under the assumption that previous document summaries have been read, update summaries or temporal summaries can be produced that reflect the changes in a series of time-stamped documents. For example, when presenting a summarisation system with two document sets to summarise, where one set precedes the other in time, and the task is to summarise the new information in the second batch of documents, this task is known as update summarisation (Dang and Owczarzak, 2008). The focus is on summarising only what is new, or novel, about the subsequent batch of documents presented to the summarisation system (i.e. the user is assumed to have read the first batch).

[1] news.google.co.uk

Additionally, there are cases where multiple batches of documents are presented as input to the summarisation system over time. Within the context of news event summarisation, this task is known as temporal summarisation (Aslam et al., 2013, 2014, 2015). The key difference between update summarisation, and temporal summarisation, is that for update summarisation a system should usually always output some text as the summary. Specifically, the experimental task operates under the assumption that there is new information in the second batch of documents to summarise. Whereas, for temporal summarisation (perhaps spanning several days), the summarisation system must decide whether to output zero or more sentences at any given interval (i.e. hourly). In particular, temporal summarisation systems may implement an event tracking component (Allan, 2002), to gauge event activity over time.

For a given experimental setup, within an automatic text summarisation research project, various aspects of the above taxonomy are typically specified as part of the evaluation – i.e. the conditions described are not mutually exclusive. For example, the earliest published work on automatic text summarisation (Luhn, 1958) evaluated extractive summaries of single documents, producing generic, informative summaries, from a static collection of documents. In our experiments in this thesis, we conduct summarisation experiments over a number of conditions described in this section. Specifically, our experiments are within the extractive multi-document summarisation task, producing informative-style summaries of news events. For our experiments conducted within the retrospective summarisation task, we produce generic summaries, and produce query-biased summaries for the temporal summarisation task.

With regards to focusing on extractive summarisation, we argue that the extractive summarisation paradigm, when instantiated within a supervised machine learning framework, provides a robust and well-understood experimental setting to investigate and validate our claims in this thesis (c.f. Section 1.2). Further, as we conduct experiments within the TREC Temporal Summarisation Track[1] (Aslam et al., 2013, 2014, 2015), we are bounded to the extractive summarisation paradigm, as the task evaluation specification explicitly requires systems to output sentence identifiers (i.e. it is an extractive summarisation task).

This concludes our overview of the main summarisation tasks, and types of summaries produced by text summarisation systems. In the next section, we discuss contextual factors to consider when producing different types of summaries, for different summarisation tasks.

[1] trec-ts.org

## 2.2 Summarisation Factors

We now consider summarisation context factors. Jones (1998) defines three contextual factors that influence the design, implementation, and evaluation of automatic text summarisation systems. In particular: input factors; purpose factors; and output factors. Specifically, when producing automatic text summaries for user consumption, we should consider: the characteristics of the input document (or documents) being summarised; the purpose for producing summaries (i.e. why a user might find an automatic text summary useful in a given scenario); and also the output format required to be produced by automatic text summarisation systems.

### 2.2.1 Input Factors

The key input factors to consider are source form and scale, i.e. what is being summarised, and how much is being summarised. Specifically, text summaries can be produced from various types of input documents. In particular, the source documents being summarised could be drawn from collections of scientific literature (Teufel and Moens, 2002), web pages (Tombros and Sanderson, 1998), email (Wan and McKeown, 2004), microblog posts (Sharifi et al., 2013), or, relevant to the work in this thesis, newswire (Over et al., 2007; Hong et al., 2014).

Within each domain, documents typically exhibit genre-specific characteristics. Newswire documents, for example, often exhibit a common structure. Such articles usually begin with opening sentences that are informative of the article topic. In particular, lead-based summarisation of single newswire articles, where summaries are derived from the opening sentences, is known to be a very competitive baseline (Nenkova, 2005). In Chapter 4, we investigate the effectiveness of such lead-based baselines. Further, we investigate a lead-based feature when training supervised machine learned summarisation models, discussed in Chapter 5.

Scale is also an important input factor to consider when designing automatic text summarisation systems. Specifically, implementations of summarisation algorithms may be tailored to reflect how many documents are given as input to the summarisation process. In the case of multi-document summarisation, there often exists a degree of textual redundancy across documents that are discussing the same news event. It has been demonstrated that such cross-document redundancy can be an important summarisation feature (Erkan and Radev, 2004). Specifically, information being repeated across a number of sources can be taken

as an indication that a certain concept is important, which multi-document summarisation algorithms often seek to exploit (Radev et al., 2004; Nenkova et al., 2006).

For the multi-document newswire summarisation tasks we investigate in this thesis, typical input document batch sizes are approximately 10 newswire articles (Over et al., 2007). Further, in Chapter 7, we also conduct experiments within the context of the TREC Temporal Summarisation Track (Aslam et al., 2013, 2014, 2015), where hundreds of documents are taken as input to the summarisation process, discussing breaking news events as they evolve over time. Notably, as we demonstrated within the context of the TREC Temporal Summarisation Track (McCreadie et al., 2013, 2015), the challenges of implementing text processing pipelines, i.e. building sophisticated data structures for representing the input documents, increases when summarising hundreds of documents per-hour.

Input factors related to scale also impact the evaluation of automatic text summarisation systems. Typically, as discussed in Section 2.3, the evaluation of automatic text summarisation systems involves comparing system-produced summaries to human-authored exemplar summaries (Lloret et al., 2017). In the summarisation literature, such exemplar summaries are referred to as gold-standard summaries. In the case of single document summarisation, a human annotator is required to read one document in order to write a gold-standard summary of that document. Correspondingly, for the case of multi-document summarisation, a human annotator is required to read and comprehend multiple documents in order to summarise the most important and salient aspects of those documents – arguably a more arduous task.

Further, there is often considerable variation in the informational content selected by human annotators for inclusion into human-authored gold-standard text summaries (Rath et al., 1961; Lin and Hovy, 2002; van Halteren and Teufel, 2003; Harman and Over, 2004). As such, for newswire summarisation tasks (Over et al., 2007), current best practice is to obtain multiple gold-standard summaries, from multiple human annotators. Furthermore, as described in Section 2.3, automatic summarisation systems are typically evaluated over a number of different document sets (e.g. 50 sets of 10 input documents). As the scale of the input to the text summarisation process increases, the time and effort required to obtain gold-standard summaries for evaluating automatic text summarisation systems also substantially increases.

Once the process of manually authoring summaries of multiple documents, by multiple human annotators, and for multiple document sets, has been completed, we should seek to

maximise such manual annotation investment. As described in Section 2.3, it is common in the summarisation literature to use cheap and repeatable automatic summarisation evaluation methodologies (Lin, 2004) – once we have obtained manual summaries of large collections of documents. However, we should ensure that such automatic summarisation evaluation metrics accurately reflect human judgements for the quality of automatic text summaries, for the specific summarisation task currently being evaluated. In Chapter 3, we conduct a user-study to validate that the automatic evaluation metrics we use in this thesis are aligned with summarisation judgements from (crowd-sourced) human annotators. Further, in Chapter 5, we reuse manual gold-standard summaries, which have been produced by human annotators, to train supervised machine learned summarisation models.

### 2.2.2 Purpose Factors

Arguably, the most important factors are related to the purpose for producing text summaries. Understanding why a document is to be summarised, i.e. for what reason a user might find that summary useful, is central to the design, implementation, and evaluation of automatic text summarisation systems. Indeed, Jones (1998) argues that we cannot properly evaluate a summary unless we know for whom, and for what purpose, the summary was produced. As such, purpose factors are closely linked to the task that a user is attempting to accomplish, where the user's performance on that task could increase given a more effective summary.

For example, given a user who's current task is searching the web, search engine results pages often display short snippets of web documents highlighting portions of the document relevant to the user's query (Wang et al., 2007; Metzler and Kanungo, 2008). This is not (primarily) intended as an informative summary, i.e. the most important and salient aspects of the document are not summarised. However, such summaries provide users with an indication as to what documents in the search engine results page might be about. We can use the summaries of web documents provided by search engines (i.e. indicative snippets) to decide if we wish to read particular documents. Such a task, deciding if a document is relevant, based on search engine result page snippets, would be easier to complete given more effective indicative summaries (Tombros and Sanderson, 1998).

Whether the purpose is to produce an indicative summary, or to produce an informative summary, should be taken into account when evaluating automatic text summarisation sys-

tems. Specifically, purpose factors should influence evaluation methodology. For example, when producing indicative summaries for search engine results pages, the methodology for evaluating search snippets could be task-oriented (Savenkov et al., 2011), where an evaluation considers the ability of a user to complete their search tasks more effectively. Further, when producing an informative summary, any evaluation of summarisation effectiveness should seek to measure the extent to which a given summary conveys the most important and salient aspects of the original document(s). This necessitates that a human annotator has read the document(s) being summarised, and identified what the most important and salient aspects are. In particular, the gold-standard to evaluate informative summaries should be an informative (i.e. not indicative) gold-standard summary.

In this thesis, we aim to produce informative summaries of news events. The purpose for producing informative summaries of news events (i.e. our task) is such that the most important and salient aspects of a given news event can be understood without consuming multiple news articles. Further, our intended audience is the general public. In particular, we do not claim to produce summaries that are effective for use by specific target users undertaking complex tasks within specialised domains, such as crisis management (Carver and Turoff, 2007). As such, we do not evaluate summarisation effectiveness within such scenarios. As previously stated in Section 2.2.1, in Chapter 3 we explicitly validate the methodology used to evaluate summaries, for our stated target purpose, and with our stated target audience.

### 2.2.3 Output Factors

The requirements on presentational aspects of system-produced text summaries also has an impact on the design, implementation, and evaluation of automatic summarisation systems. The key output factors to consider when producing automatic text summaries are the summary format, and the summary length. Such output factors may be interpreted as constraints on what a summary should look like, and how long the summary should be. For example, the summary may be required to be presented as a series of natural language sentences, where the maximum number of words in the summary is specified in advance (e.g. 100 words).

When implementing text summarisation systems, summarisation formatting constraints influence the design and implementation of algorithms and data structures. For example, in the context of newswire summarisation (Over et al., 2007; Hong et al., 2014), a common re-

quirement is that the summary should be formatted as a series of natural language sentences. In such cases, where the required output format is whole sentences, an implementation of a summarisation system would typically need to store and manipulate statistics about sentences. Simpler types of summaries, e.g. key-word summaries or word-clouds (Viégas and Wattenberg, 2008), that exhibit fewer constraints on the required output format would typically only need to store and manipulate statistics about individual words. However, in certain domain-specific contexts, e.g. summarising medical documents (Afantenos et al., 2005), further external resources might be required to meet more complex output format constraints. For example, when summarising patient records, if the summary output should preferably not contain domain-specific terms, a lexical ontology or knowledge base may be required for automatically expanding medical acronyms or replacing medical terminology with terms that are more easily understood by (non-clinician) lay-persons (Demner-Fushman et al., 2009).

Constraints on output format should also be considered when evaluating automatic text summarisation systems. In particular, to ensure robust measurement of summarisation effectiveness, any constraints placed on summarisation output should be accounted for in the evaluation methodology. For example, when specifying that a summary should contain natural language sentences, the evaluation methodology might consider the lexical qualities (e.g. readability) of such sentences (Pitler et al., 2010; Ellouze et al., 2016). Further, when evaluating summarisation output via comparison of system-produced summaries to human-authored gold-standard summaries, it would be necessary that the output formats match. Specifically, if the output required is a natural language summary of 10 newswire articles, system-produced summaries should be compared to human-authored natural language summaries of those 10 articles. Comparing a sequence of natural language sentences (produced by an automatic summarisation system) to a human-authored word-cloud, for example, would not provide an accurate measurement of a system's ability to produce natural language summaries.

The second key output factor to consider is length. In particular, an automatic text summarisation system may be constrained to output a summary that is a maximum of 100 words in length (i.e. a summary length limit), or 10% of the original document (i.e. an explicit compaction ratio). Further, the required length may be task-constrained, such as for the newswire headline generation task (Witbrock and Mittal, 1999; Banko et al., 2000; Dorr et al., 2003). Such specific length constraints impact both the design and implementation of summarisation

systems, and further, should be taken into account when evaluating summarisation systems.

Given the nature of automatic text summarisation, where the aim is to produce concise representations of documents, the main impact of the length output factor on the design and implementation of summarisation systems concerns anti-redundancy filtering. Specifically, within the short space available in which to convey the most important and salient information from the summarised document(s), there should ideally not be any repeated information (i.e. redundancy) expressed within the summary text. For example, it is common in multi-document newswire summarisation systems (Hong et al., 2014) that there exists a component of the system that explicitly attempts to reduce redundancy in the system-produced summary text(s). In this thesis, in Chapter 4, we describe and evaluate a range of commonly used anti-redundancy filtering methods, that seek to minimise the amount of repeated information over the sentences that are selected for inclusion into the summary text.

With regards to the evaluation of automatic text summarisation systems, the main impact of the length output factor relates to an issue of fairness when evaluating the output from two summarisation systems. Specifically, we cannot reliably evaluate summary texts of different lengths, where the output has been constrained to be a specific length (e.g. 100 words). This would introduce a bias in empirical observations towards the longer summary, as it would have more opportunity to convey more information from the summarised document(s). Indeed, it is typical in the experimental setup of summarisation evaluations, where a length constraint is imposed by the summarisation task, to truncate the summary text(s) under evaluation to equal lengths (Over et al., 2007). In particular, system-produced summary text(s) that exceed a specified length limit are simply truncated (with any text over the length limit simply ignored).

This completes our overview of the input, purpose, and output factors that influence the design, implementation, and evaluation of automatic text summarisation systems. We now describe the datasets and summarisation evaluation methodologies used in this thesis.

## 2.3 Summarisation Evaluation

The evaluation of system-produced summaries of text documents is a crucial part of the automatic text summarisation task. Summarisation evaluation methodologies aim to provide a quantification of the effectiveness of system-produced summaries. Evaluation is conducted

to ensure that the quality of summaries produced by automatic text summarisation systems meets user expectations. In this section, we discuss summarisation evaluation methodologies. We begin with a discussion of the challenges involved in evaluating summarisation systems. We then introduce the datasets used in our experiments in this thesis, which contain the input documents to be summarised, and also describe the specific metrics we use for evaluation.

### 2.3.1 Challenges

While producing effective summaries is a challenging problem, effectively evaluating text summarisation is equally challenging (Lloret et al., 2017). The difficulties of evaluating text summarisation systems arise because the output of automatic text summarisation systems is natural language text, and natural language is highly ambiguous (Manning and Schütze, 2001; Jurafsky and Martin, 2009). Thus, evaluating the output of natural language processing systems, including summarisation systems, is inherently difficult (Galliers, 1997).

It has been shown that there is often considerable variation in the content selected for inclusion into a summary by human annotators (e.g. when writing gold-standard summaries), and there is often measurable disagreement in the judgements provided by human assessors regarding the quality of text summaries (Rath et al., 1961; Lin and Hovy, 2002; van Halteren and Teufel, 2003; Harman and Over, 2004). This indicates that summarisation is subjective, as the use and interpretation of natural language itself is subjective. In particular, for any given set of input documents, there is no single "correct" answer to the text summarisation problem. For example, in the extractive summarisation setting, there are numerous combinations of sentences that, when extracted to form summaries, could be judged as somewhat effective.

Therefore, to make progress in automatic text summarisation research, and to potentially improve commercial summarisation software products, there exists important limitations within the methodologies used to evaluate text summarisation systems, discussed below.

#### Intrinsic vs. Extrinsic Evaluation

Within the summarisation evaluation literature, there exists a clear distinction between extrinsic and intrinsic evaluation (Lloret et al., 2017). Extrinsic evaluation involves evaluating summarisation systems in the context of where they are used. This is referred to as task-oriented evaluation (e.g. Mani et al., 1999; Teufel, 2001; Mani et al., 2002; McKeown et al.,

2005), e.g. search tasks (Tombros and Sanderson, 1998; McLellan et al., 2001; Savenkov et al., 2011). In an extrinsic summarisation evaluation, users are performing a task that requires a summary, and user performance on that task depends on the quality of the summary. Specifically, given two summarisation systems, $A$ and $B$, if task performance across a sample of users increases when using summaries from system $A$, compared to when using summaries from system $B$, the evaluation result is that summarisation system $A$ is more effective.

Extrinsic task-based summary evaluations are the most realistic method to measure text summarisation effectiveness (Galliers, 1997). However, such evaluations are also expensive and time-consuming, and it is thus not common to find this style of evaluation reported in the summarisation literature (Nenkova and McKeown, 2011). Specifically, it is not feasible (nor desirable) to conduct extrinsic user evaluations during summarisation system development, e.g. to evaluate slight algorithm changes or conduct parameter sweeps. As such, one limitation of summarisation evaluation is that evaluation is typically intrinsic in nature.

#### Coverage, Linguistic Quality, and Responsiveness

Intrinsic evaluation assesses the effectiveness of a summary by examining the summary text directly (Lloret et al., 2017). The primary methodologies used are: measuring the coverage of a summary; measuring the linguistic quality of a summary; and measuring the responsiveness of a query-biased summary to the given query. Each property is measured independently. This is a further limitation of summarisation evaluation (Conroy and Dang, 2008).

Coverage measures the extent to which a summary text conveys the most important and salient aspects of the summarised document(s). To measure coverage, the summary text is examined to determine the extent to which it matches: a set of manually identified informational units; gold-standard summary text(s) authored by human annotators; or the original documents that were summarised. Manually identified informational units are known as: factoids (Teufel and van Halteren, 2004); summary content units (Nenkova and Passonneau, 2004); basic elements (Hovy et al., 2006); or nuggets (Ekstrand-Abueg et al., 2016). A summary is judged to be effective based on the number of such gold-standard informational units it contains. Further, coverage comparison with respect to a gold-standard summary is typically with respect to $n$-gram overlap (Lin, 2004). A more effective summary contains more $n$-grams from the gold-standard summary text(s). Furthermore, it has been proposed that the

language model of an effective summary does not diverge from the language model of the summarised documents (Saggion et al., 2010; Louis and Nenkova, 2013).

Linguistic quality measures the readability of the summary. To measure linguistic quality, the summary text is examined and judged on a specific set of linguistic quality criteria. Such criteria may include coherence, conciseness, redundancy, grammar and formatting[1]. These criteria are formally defined in Section 3.2.1 (c.f. Figure 3.1). Typically, human annotators provide judgements on a numerical-scale (e.g. [1..5]), for each of the linguistic quality criteria (Over et al., 2007; Dang and Owczarzak, 2008). Linguistic quality evaluation is conducted with respect to the system-produced summary in isolation – i.e. this method does not rely on manually produced gold-standard summaries. Due to the complexity of assessing linguistic quality, such evaluations are not commonly reported (Nenkova and McKeown, 2011).

Responsiveness measures the extent to which a query-biased summary answers the information need expressed in the given query (Dang, 2005). To measure responsiveness, a human assessor reads a topic statement, describing the underlying information need (which is expressed as a short text query, and given as input to the summarisation system). Then, the human assessor will read the query-biased summary, and judge to what extent the summary text answers the query. Judgements are typically on a numerical-scale (e.g. [1..5]). This evaluation method does not require a gold-standard, as the summary text is evaluated in isolation.

#### Manual Evaluation vs. Automatic Evaluation

The above intrinsic summarisation evaluation methods (i.e. coverage, linguistic quality, and responsiveness) can be undertaken as either manual or automatic procedures (Lloret et al., 2017). In particular, when judgements with regards to summarisation quality are obtained from human annotators, this is referred to as manual evaluation. When judgements with regards to summarisation quality are obtained via evaluation software toolkits, this is referred to as automatic evaluation. As manual summarisation evaluation methods are expensive and time-consuming to undertake, there is a clear preference to report automatic evaluation results in the literature (Nenkova and McKeown, 2011). However, this is perhaps one of the most controversial limitations of the evaluation of text summarisation systems, as automatic evaluation is known to be problematic (Sjöbergh, 2007; Graham, 2015; Schluter, 2017).

[1] duc.nist.gov/duc2004/quality.questions.txt

While there are some research proposals that investigate the automatic evaluation of summary readability (e.g. Lapata and Barzilay, 2005; Pitler et al., 2010; Ellouze et al., 2016), automatic summarisation evaluation is primarily limited to evaluating summary coverage. For example, the ROUGE[1] (Lin, 2004) and FRESA[2] (Saggion et al., 2010) automatic summarisation evaluation software toolkits are commonly used to report results in the literature. ROUGE is the current de-facto standard (Lloret et al., 2017). ROUGE requires comparison to human-authored gold-standard summary text(s), whereas FRESA permits a model-free style of evaluation – measuring summary text(s) to the original documents that were summarised. Such automatic summarisation evaluation methods are described in detail in Section 3.1.

The use of crowd-sourcing platforms, e.g. MTurk[3] or CrowdFlower[4], to obtain manual judgements of summary quality has been investigated in the summarisation literature (e.g. Gillick and Liu, 2010; Lloret et al., 2013). In this thesis, we conduct such an evaluation, in Chapter 3, to assess the linguistic quality of text summaries. However, manual evaluation procedures, undertaken to measure summarisation coverage, readability, and responsiveness, where assessments of summary quality are obtained from expert human assessors, are typically conducted within the context of formally organised workshops, as described below.

### Summarisation Evaluation Workshops

In order to alleviate and control for the effects of the above assumptions and limitations within summarisation evaluation methodologies, there exist several standardised frameworks for evaluating summarisation in practice. Specifically, the summarisation research community has organised a series of summarisation evaluation workshops, where common datasets, gold-standards, and evaluation metrics have evolved over time. In particular, given a summarisation task, e.g. generic summarisation, query-biased summarisation, update summarisation, or temporal summarisation, a set of documents is collected from a specific domain (such as the newswire domain), to be provided as input to automatic text summarisation systems. Expert human annotators read and produce gold-standard summaries of those documents, or, identify a set of gold-standard informational nuggets that reflect what information automatic

[1] berouge.com
[2] fresa.talne.eu
[3] mturk.com
[4] crowdflower.com

**Table 2.1:** Multi-document newswire summarisation evaluation frameworks used in this thesis.

| Framework | Year | Topics | Task | Evaluation | Experiments |
|---|---|---|---|---|---|
| DUC | 2001 | 59 | Generic retrospective summarisation | ROUGE w.r.t. gold-standard | Chapters 3, 4, 5, 6 |
| DUC | 2002 | 59 | | | |
| DUC (Task 2) | 2003 | 30 | | | |
| DUC (Task 2) | 2004 | 50 | | | |
| TREC-TS | 2013 | 9 | Query-biased temporal summarisation | Nuggets-based evaluation | Chapter 7 |
| TREC-TS | 2014 | 15 | | | |
| TREC-TS | 2015 | 26 | | | |

summaries should convey. Manual and automatic evaluation procedures are then undertaken, to evaluate the text summaries produced by different automatic text summarisation systems.

Many summarisation evaluation workshops have been organised over the past 20 years, e.g.: SUMMAC[1] (1998); the Document Understanding Conferences[2] (2001–2007); the Text Analysis Conference Summarisation Tracks[3] (2008–2011); TREC Temporal Summarisation[4] (2013–2015); TREC Real-Time Summarisation[5] (2016–2017); and MultiLing[6] (2011–2017). One of the key outcomes of each evaluation workshop is a reusable summarisation evaluation framework, which is often used to report results in the summarisation literature long after the workshop has taken place. For example, 10 years after the completion of the DUC 2004 summarisation evaluation workshop[7], the documents, gold-standard summaries, and evaluation methodology are still being used to report evaluations in the literature (c.f. Hong et al., 2014).

In this thesis, we conduct experiments within two such established summarisation evaluation frameworks. Specifically, we reuse documents, gold-standard summary annotations, metrics, and methods from: the 2004 Document Understanding Conference (Task 2); and the 2013–2015 TREC Temporal Summarisation track. The details of these summarisation evaluation frameworks are described below, and summarised in Table 2.1.

[1] www-nlpir.nist.gov/related_projects/tipster_summac
[2] duc.nist.gov
[3] tac.nist.gov
[4] trec-ts.org
[5] trecrts.github.io
[6] http://multiling.iit.demokritos.gr/
[7] duc.nist.gov/duc2004

### 2.3.2 DUC 2004 Task 2

The Document Understanding Conferences ran from 2001 through 2007 (Nenkova, 2005; Over et al., 2007). Various tasks related to the summarisation of newswire articles were investigated, including generic and query-biased single- and multi-document summarisation. In this thesis, we report experimental results on the DUC 2004 Task 2 dataset, using previous years (DUC 2001–2003) as training and validation (i.e. development) data. The DUC 2004 Task 2 dataset is used to evaluate the retrospective summarisation of news events, specifically, it is a generic multi-document newswire summarisation task. The datasets are organised into topics, which are collections of one or more input documents to be summarised, and a corresponding set of one or more exemplar summaries of those documents (i.e. human-authored gold-standard summaries). Table 2.1 lists the number of topics in each annual dataset.

The DUC 2001[1] and DUC 2002[2] documents are sourced from: the Wall Street Journal (1987–1992); the Associated Press (1989–1990); the San Jose Mercury News (1991); the Financial Times (1991–1994); the LA Times (1989–1990); and the U.S.-based Foreign Broadcast Information Service (1996). The DUC 2001 dataset is accompanied with 400-, 200-, 100-, and 50-word gold-standard summaries. The DUC 2002 dataset is accompanied with 200-, 100-, 50-, and 10-word gold-standard summaries. The DUC 2001 dataset was originally distributed as 60 topics: 30 training topics, and 30 test topics. We combine these sets, and with topic "d31" being withdrawn at source, DUC 2001 thus has 59 topics in total. The DUC 2001 training topics have a single gold-standard summary, whereas the test topics contain three gold-standard summaries per-topic. The DUC 2002 dataset has two gold-standard summaries per-topic, and we use the abstractive summaries in our experiments (specifically, not the extractive summaries which were also produced as part of the DUC 2002 evaluations).

The DUC 2003[3] and DUC 2004[4] documents are sourced from: the Associated Press (1998); and the New York Times (1998). The DUC 2003 and DUC 2004 datasets are accompanied with 100-word gold-standard summary texts, for which each topic has four gold-standard summaries. The DUC 2003 and DUC 2004 documents are focused by events (Over et al., 2007), being drawn from clusters of documents generated by topic detection and track-

[1] duc.nist.gov/guidelines/2001.html
[2] duc.nist.gov/guidelines/2002.html
[3] duc.nist.gov/guidelines/2003.html
[4] duc.nist.gov/guidelines/2004.html

ing systems (Allan, 2002). The DUC 2001–2002 documents primarily describe news events, but also contain a non-event articles (e.g. biographical and opinion).

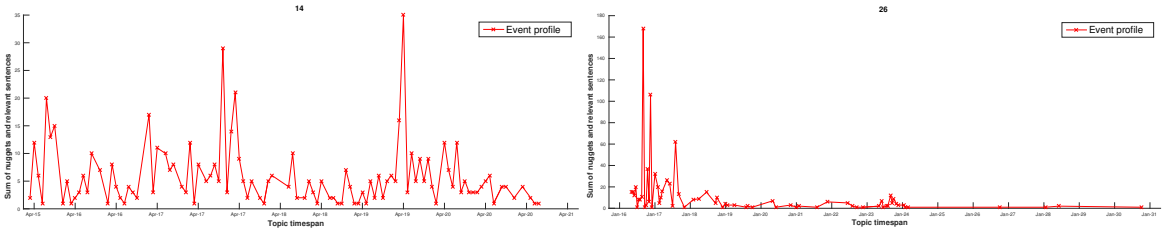
For the test set that we report results on, DUC 2004 Task 2, the task is to produce 100-word summaries of multiple news stories, where each topic (i.e. document set) contains 10 newswire articles (with scores averaged over the 50 document sets). At the DUC 2004 workshop, for Task 2 runs, in addition to automatic evaluation using ROUGE (Lin, 2004), manual evaluation of coverage and linguistic quality was undertaken (Over and Yen, 2004). In our experiments in Chapters 4, 5, and 6, over the DUC 2004 Task 2 dataset, automatic coverage evaluation of test runs is undertaken using ROUGE (i.e. no manual evaluation is undertaken).

### 2.3.3 TREC-TS 2013–2015

Further to our experiments over DUC 2004, we also conduct experiments within the context of the 2013–2015 TREC Temporal Summarisation Track (Aslam et al., 2013, 2014, 2015). The aim of this evaluation workshop is to promote research into systems that can emit relevant and novel sentences regarding a breaking news event. The events to be summarised include, for example, the 2012 "Buenos Aires Rail Disaster", and the 2013 "Boston Marathon bombings". This is a temporal summarisation task, requiring the summarisation of evolving news events over a multi-day time-period. Additionally, this is a query-biased summarisation task, as each event is described by a short text query. As such, the summarisation task primarily necessitates reporting important, salient, and relevant information, but also emphasises the reporting of novel information. The TREC Temporal Summarisation (TREC-TS) evaluation campaign offers an advanced form of summary evaluation that is specifically tailored to the summarisation of evolving news events. Indeed, TREC-TS is currently the primary evaluation framework for researching the state-of-the-art for the temporal summarisation of news events.

The summarisation task requires extracting sentences from a large corpus of sequentially time-stamped documents[1], to construct a summary of an evolving news event. The documents are crawled from the web, from the time-period of December 2011 through May 2013, and contain primarily news articles, and blog posts. Systems may iterate over the corpus in real time, i.e. document-by-document, or process the documents within the corpus in batches, e.g. hourly. The time periods of news events in the corpus often last several days in duration.

[1] s3.amazonaws.com/aws-publicdatasets/trec/kba/index.html




**(a)** Topic 14: "boston marathon bombings". **(b)** Topic 26: "vauxhall helicopter crash".

**Figure 2.2:** Example events within the TREC-TS dataset, showing the different nature of evolving news events. We show the volume of time-stamped gold-standard nuggets and relevant sentences over the event period, which indicates the bursty nature of the activity (i.e. sub-events) within different evolving news events over time.

There are 10 types of events in the corpus, specifically: "accident"; "bombing"; "conflict"; "earthquake"; "hostage"; "impact event"; "protest"; "riot"; "shooting"; and "storm". In particular, there is a distribution over known/expected events, and unknown/unexpected events.

Figure 2.2 provides an illustration of one of the events in the TREC-TS corpus: topic 14, the "Boston Marathon bombings"[1]. In Figure 2.2, on the $x$-axis we show the timespan of the event, and on the $y$-axis we show the volume of time-stamped gold-standard nuggets and relevant sentences, indicating the activity profile of the event over time. From Figure 2.2, we can observe that events exhibit a bursty nature. This illustrates one of the key differences in the summarisation task as compared to DUC 2004, which involved generating a 100-word summary of a given set of documents. For the TREC-TS task, systems must track events over time, and decide to emit zero, one, or more summary sentences at any given time-period – depending on whether there is any important, salient, and relevant information in the corpus at that particular time. As such, the temporal summarisation task also includes an element of topic detection and tracking (Allan, 2002). Specifically, the length of the summary is not defined a priori, and an effective temporal summarisation system might decide to emit no content in any given time-period to reflect the bursty nature of events.

The summarisation evaluation approach used at TREC-TS is nuggets-based (Ekstrand-Abueg et al., 2016). Specifically, for each event, a series of discrete units of information about the event are manually identified. For example, an arrest of a suspect, or the number of people injured, would be examples of information nuggets that are relevant to news events. For evaluation purposes, such nuggets are manually derived from the revision history of Wikipedia

[1] en.wikipedia.org/wiki/Boston_Marathon_bombing

articles related to the events in the corpus. Summary effectiveness is measured with respect to the extent that system-produced event summary texts cover the essential nuggets about the event, with discount factors penalising latency and verbosity.

The TREC Temporal Summarisation metrics are defined to capture the precision and comprehensiveness of a summary. The precision metric, referred to as *expected gain*, is the sum of the relevance of each nugget that an update is matched to. For a summarisation system producing an update stream $\mathscr{S}$, gain is computed as:

$$ExpectedGain(\mathscr{S}) = \frac{1}{|\mathscr{S}|} \sum_{u \in \mathscr{S}} \sum_{n \in \mathbf{M}(u)} \mathbf{g}(u,n) \tag{2.1}$$

where $\mathbf{M}(u)$ is the set of nuggets matching update $u$ and $\mathbf{g}(u,n)$ measures the utility of matching update $u$ with nugget $n$. On the other hand, the comprehensiveness metric, referred to as *comprehensiveness*, is the proportion of all nuggets matched by the system updates,

$$Comprehensiveness(\mathscr{S}) = \frac{1}{|\mathscr{N}|} \sum_{u \in \mathscr{S}} \sum_{n \in \mathbf{M}(u)} \mathbf{g}(u,n) \tag{2.2}$$

where $\mathscr{N}$ is the set of nuggets for the current event. Between them, these metrics capture precision, comprehensiveness and brevity. To provide a target metric, an $F$-like measure is also defined, referred to as *combined*, or $\mathscr{H}$. This the harmonic mean of $\mathbf{G}$ and $\mathbf{C}$,

$$Combined(\mathscr{S}) = 2 * \frac{\mathbf{C}(\mathscr{S}) * \mathbf{G}(\mathscr{S})}{\mathbf{C}(\mathscr{S}) + \mathbf{G}(\mathscr{S})} \tag{2.3}$$

In order to reward novelty within a summary, a summary only recieves gain the first time they return an update matching a nugget. Matches to updates later in the summary are ignored when computing Equations 2.1 and 2.2.

This concludes our overview and discussion of the evaluation of summarisation systems. We now provide a review of the baseline algorithms, and state-of-the-art systems, for the task of generic extractive multi-document newswire summarisation (Section 2.4).

## 2.4 Multi-document Newswire Summarisation

Automatic text summarisation is performed by text summarisation systems, i.e. implementations of text summarisation algorithms. There have been numerous automatic text summarisation algorithms proposed in the summarisation literature, and named entities are a well

known summarisation feature (Jones, 2007; Nenkova and McKeown, 2011; Lloret and Palomar, 2012; Saggion and Poibeau, 2013; Torres-Moreno, 2014). In this thesis, we focus on the task of extractive summarisation, specifically, extractive multi-document newswire summarisation. In this section, we discuss a range of summarisation systems that have been proposed for this specific task. In particular, we discuss the newswire summarisation approaches highlighted by the recent study of Hong et al. (2014), where the standard baselines, and state-of-the-art systems, for the task of generic extractive multi-document newswire summarisation were identified.

The three main approaches to extractive summarisation (Nenkova and McKeown, 2011) include: rank-then-select; combinatorial optimisation; and supervised machine learning. For example, in the rank-then-select approach, the task of extractive summarisation is composed of two steps. First, candidate summary sentences (i.e. the sentences to be summarised) are scored according to their summary worthiness. Sentence scores are determined by a summarisation algorithm, based on computing estimates of sentence importance and salience (e.g. Erkan and Radev, 2004; Nenkova et al., 2006). This scoring process enables a summarisation system to then rank candidate summary sentences by their preference for inclusion into the summary text. Then, the highest-ranked sentences are iteratively selected for inclusion into the summary text, subject to a summary length limit (e.g. 100 words). Commonly, an anti-redundancy component is used to skip sentences (from the ranking) if they are textually similar to the sentences that were previously selected for inclusion into the summary. We formally define such anti-redundancy components in Section 4.2.2.

Further, the extractive summarisation task can be formulated as a combinatorial optimisation problem (e.g. McDonald, 2007; Gillick and Favre, 2009). Specifically, the candidate summary sentences are taken as a set, and the aim is to select a globally optimal subset of the input sentences. Proposed approaches under the optimisation paradigm are thus required to define what an optimal summary should be. This is achieved by defining values for sentences, or values for sub-sentence elements (e.g. $n$-grams), and maximising an objective function over such values. Exact or approximate solutions to such $NP$-hard (Khuller et al., 1999) combinatorial subset optimisation problems can be obtained by expressing the problem as an Integer Linear Program (ILP), for which open-source solvers[1] are readily available.

[1] gnu.org/software/glpk/

**Table 2.2:** ROUGE (Lin, 2004) results, over DUC 2004 Task 2, for the baseline algorithms and state-of-the-art systems for the task of extractive generic multi-document newswire summarisation (c.f. Hong et al., 2014)

| **Baseline Algorithms** | **Year of Publication** | **ROUGE-1** | **ROUGE-2** |
|---|---|---|---|
| *LexRank* | Erkan and Radev (2004) | 36.00 | 7.51 |
| *Centroid* | Radev et al. (2004) | 36.42 | 7.98 |
| *FreqSum* | Nenkova et al. (2006) | 35.31 | 8.12 |
| *TsSum* | Conroy et al. (2006) | 35.93 | 8.16 |
| *Greedy–KL* | Haghighi and Vanderwende (2009) | **38.03** | **8.56** |
| **State-of-the-art Systems** | **Year of Publication** | **ROUGE-1** | **ROUGE-2** |
| *CLASSY 04* | Conroy et al. (2004) | 37.71 | 9.02 |
| *CLASSY 11* | Conroy et al. (2011) | 37.21 | 9.21 |
| *Submodular* | Lin and Bilmes (2012) | 39.23 | 9.37 |
| *DPP* | Kulesza and Taskar (2012) | **39.84** | 9.62 |
| *OCCAMS_V* | Davis et al. (2012) | 38.50 | 9.75 |
| *RegSum* | Hong and Nenkova (2014) | 38.60 | 9.78 |
| *ICSISumm* | Gillick and Favre (2009) | 38.44 | **9.81** |

Furthermore, the extractive summarisation task can be formulated as a supervised machine learning problem (e.g. Kupiec et al., 1995; Teufel and Moens, 1997; Aone et al., 1998). Specifically, a machine learned model (Hastie et al., 2009; Witten et al., 2016) is trained to predict scores for candidate summary sentences, from which a set or ranking of sentences is then induced. Typically, sentences are selected for inclusion into the summary by passing the set or ranked list of candidate summary sentences to an anti-redundancy filtering component. However, extractive summarisation under the supervised paradigm necessitates obtaining training data, defining features, and model selection – problems that we thoroughly examine in Chapter 5.

In Table 2.2, we reproduce the results presented by Hong et al. (2014) over the 50-topic DUC 2004 Task 2 dataset, where the task is to produce 100-word summaries of 10 newswire documents (i.e. multi-document summarisation). Table 2.2 shows the ROUGE-1 recall and ROUGE-2 recall scores of 5 baseline algorithms, and 7 state-of-the-art systems, and the year of publication for each summarisation approach. The ROUGE scores in Table 2.2 are com-

puted via SumRepo[1], a repository of system-produced summary texts from various summarisation systems. In Table 2.2, following the recommendations of Owczarzak et al. (2012), summaries evaluated using ROUGE are stemmed with stopwords retained, as this particular setting was shown to exhibit high correlation with manual summarisation evaluation methods. In general, ROUGE scores are in the range $[0..1]$, however, the ROUGE scores in Table 2.2 are multiplied out by a factor of 100 for readability.

All of the baseline algorithms show in Table 2.2 are unsupervised, and assign summary worthiness scores to candidate summary sentences using a single feature (i.e. there is no feature combination in the baseline algorithms). However, such baseline algorithms have model parameters and anti-redundancy threshold values that should be learned via a process of cross-validation, or learned on development/validation data (e.g. DUC 2003). The baseline algorithm results, in Table 2.2, are often reported in the summarisation literature, where they are used for comparison purposes in experimental evaluations of newly proposed summarisation approaches. In our experiments in this thesis, we use such baselines when conducting a user-study to examine the validity of automatic evaluation metrics (Chapter 3). Further, we investigate various algorithm design choices when re-implementing such baselines (Chapter 4). Furthermore, we propose to reuse such baseline algorithms as features within supervised summarisation models (Chapters 5, 6, and 7).

The state-of-the-art systems use more advanced techniques such as supervised learning (e.g. regression) and combinatorial optimisation (e.g. integer linear programming). RegSum (Hong and Nenkova, 2014) is closest to the work we conduct in this thesis, using supervised regression techniques. CLASSY04 (Conroy et al., 2004) was the most effective system at DUC 2004. The CLASSY04 summarisation system is supervised (trained on DUC 2003), and utilises a Hidden Markov Model (HMM). The CLASSY11 (Conroy et al., 2011) system uses non-negative matrix factorisation. Submodular (Lin and Bilmes, 2012) formulates the summarisation problem as an optimisation problem (using submodular shells). Further, DPP (Kulesza and Taskar, 2012) uses determinantal point processes (a distribution over finite subsets), and is the most effective state-of-the-art system under the ROUGE-1 metric, with a score of 39.84. OCCAMS_V (Davis et al., 2012) is another approach from the optimisation family of systems. ICSISumm (Gillick and Favre, 2009) is another optimisation algorithm,

[1] www.seas.upenn.edu/~nlp/corpora/sumrepo.html

and is the most effective state-of-the-art system, under the ROUGE-2 metric, with a score of 9.81. The state-of-the-art results, in Table 2.2, are often reported in the literature to base claims about newly proposed summarisation approaches with respect to the state-of-the-art. In our experiments, we use such systems to assess summarisation effectiveness with respect to the state-of-the-art, when reporting summarisation results over the DUC 2004 Task 2 dataset.

### 2.4.1 Baseline Algorithms

We now describe each of the baseline algorithms from Table 2.2. Formal definitions are provided in Chapter 4, where we discuss the re-implementation of such algorithms.

**LexRank** (Erkan and Radev, 2004) – The LexRank algorithm scores candidate summary sentences by projecting sentences into a graph-based structure, and computing sentence centrality within the graph. Specifically, given a set of sentences from multiple documents, each sentence is represented as a vertex in a graph. The sentence representation is a vector of tf.idf term weights (Croft et al., 2010; Büttcher et al., 2010). A completely connected un-directed graph links all vertices. The edges in the sentence-graph are weighted by the cosine similarity between the connecting vertices (i.e. sentences). A parameter is introduced, a cosine similarity threshold, that is used to remove edges in the graph. Edges with weights that fall below the threshold are removed, disconnecting some of the vertices. A graph centrality algorithm is then applied to score the vertices, such as PageRank (Page et al., 1999). The resulting vertex (i.e. sentence) scores are then used to rank the candidate summary sentences. Sentences are selected for inclusion into the summary by applying an anti-redundancy filtering component. In Section 4.2.2 we discuss a range of such anti-redundancy filtering methods.

**Centroid** (Radev et al., 2004) – The Centroid algorithm scores candidate summary sentences by their similarity to the centroid of a cluster of all input sentences. Specifically, given a set of input sentences, tf.idf term vector weights are computed for each sentence. Then, a centroid pseudo-vector is computed over the cluster of input sentences. Each sentence is scored according to the cosine similarity to this centroid vector. Sentences are then selected for inclusion into the summary via an anti-redundancy filtering component.

**FreqSum** (Nenkova et al., 2006) – The FreqSum algorithm scores candidate summary sentences by a summation over the collection frequency of each term in the sentence, normalising for sentence length. Specifically, given a set of input sentences, a uni-gram language model

is computed over all input words (over all sentences). A sentence score is taken as the average probability of the words in the sentence. Summary sentence selection can be made via applying a cosine similarity anti-redundancy method (Hong et al., 2014).

**TsSum** (Conroy et al., 2006) – The TsSum algorithm scores candidate summary sentences by a computing the ratio of topic words (Lin and Hovy, 2000) that a sentence contains. Topic words are words within the input documents that occur with a higher probability than when compared to a background corpus. The log-likelihood ratio (LLR) test is applied to determine if a word from the input documents is a topic word, with a threshold parameter introduced to provide a cut-off (distinguishing non-topic words). A sentence is scored by computing the ratio of unique topic words to all unique words in the sentence. Such scores are used to rank the sentences, and summary sentence selection is via a cosine similarity threshold.

**Greedy–KL** (Haghighi and Vanderwende, 2009) – The most effective baseline algorithm is Greedy–KL (Haghighi and Vanderwende, 2009), with a ROUGE-1 score of 38.03, and a ROUGE-2 score of 8.56. The Greedy–KL algorithm scores candidate summary sentences by their Kullback–Leibler divergence to all other sentences. A uni-gram language model is computed over all the input sentences, and also for each sentence individually. Then, sentences are greedily selected for inclusion into the summary, based on the criteria that they minimise the Kullback–Leibler divergence between the input sentences and the set of summary sentences previously selected. This algorithm does not employ a cosine similarity threshold.

This concludes our review of the summarisation research literature. We now begin addressing our research challenges outlined in Section 1.1.

# Chapter 3

# On the Validity of Automatic Summarisation Evaluation Metrics

In this chapter, we address our first challenge, regarding experimental validity concerns relating to summarisation evaluation. As discussed in Section 2.3, automatic summarisation evaluation metrics, such as ROUGE (Lin, 2004), are commonly used to evaluate system-produced summary text(s) by comparing them to human-authored exemplar summaries. In particular automatic evaluation provides an inexpensive and repeatable compliment to expensive and time-consuming manual evaluation procedures, and are often used during the system development and experimentation phase (Nenkova and McKeown, 2011).

However, automatic evaluation of summary text(s) remains a controversial topic within the summarisation community (Sjöbergh, 2007; Owczarzak et al., 2012; Rankel et al., 2013). To validate automatic evaluation methods, we should verify that automatic summarisation evaluation metrics exhibit strong rank correlation with manual evaluation judgements. This would provide us with a measure of confidence that empirical observations obtained via automatic summarisation evaluation metrics are robust.

Hence, in this chapter, we present a crowd-sourced user-study to validate that automatic summarisation evaluation metrics are aligned with non-expert manual judgements of summary quality. Specifically, as we assert in our Thesis Statement (Section 1.2), to verify empirical observations regarding summarisation effectiveness, which have been obtained using automatic summarisation evaluation metrics, it is required that we validate automatic evaluation metrics against our manual evaluation procedure.

As discussed in Section 2.3, the gold-standard procedure for manual summarisation evaluation is the expert human assessments conducted at the Document Understanding Conference (DUC), Text Analysis Conference (TAC) summarisation track, and the Text Retrieval Conference (TREC) temporal summarisation track. However, in experiments outwith the DUC/TAC/TREC evaluation cycles, to manually evaluate the quality of summary text(s) in this thesis, we crowd-source non-expert judgements of summary quality. Crowd-sourcing manual summarisation evaluations remains a relatively expensive and time-consuming process, therefore, we would also seek to use automatic evaluation metrics.

As such, we require validation that automatic summarisation evaluation metrics accurately reflect non-expert crowd-sourced judgements for summary quality. Validation provides confidence in the empirical observations obtained via automatic evaluation metrics, with respect to the effectiveness of summarisation algorithms. To establish the validity of an automatic summarisation evaluation metric, the current best practice (Louis and Nenkova, 2013; Graham, 2015) is to observe the correlation between system rankings obtained via manual judgements for summary quality, and system rankings obtained via automatic metrics. For example, when first proposed by Lin (2004), the ROUGE automatic evaluation metric was validated by measuring the correlation of ROUGE scores with DUC expert manual scores.

This chapter is based on the following publications: Mackie et al. (2014b, 2016).

## Chapter Outline

This chapter is organised as follows:

- Section 3.1 defines the automatic summarisation evaluation metrics we investigate, namely, ROUGE (Section 3.1.1), ROUGE-WE (Section 3.1.2), and FRESA (Section 3.1.3).
- Section 3.2 describes the procedure to manually evaluate summary quality, with respect to specific quality guidelines (Section 3.2.1), via a crowd-sourced user-study (Section 3.2.2).
- Section 3.3 evaluates the validity of automatic summarisation evaluation metrics, by measuring the correlation between non-expert manual judgements for summary quality and automatic metrics, for the task of generic multi-document newswire summarisation.

## 3.1 Automatic Summarisation Evaluation Metrics

Given the overview of automatic summarisation evaluation, in Section 2.3, we now provide a further discussion of the ROUGE (Lin, 2004), ROUGE-WE (Ng and Abrecht, 2015), and FRESA (Saggion et al., 2010) automatic summarisation evaluation metrics. For each metric, we discuss the basic intuitions, and formally state how scores are assigned to summaries.

### 3.1.1 Recall-Oriented Understudy for Gisting Evaluation

The ROUGE metric was proposed by Lin (2004), with a reference implementation publicly available[1]. ROUGE was introduced as an official metric at the DUC 2004 summarisation evaluation campaign[2]. Effectively, ROUGE is the *de facto* standard metric used in the summarisation literature to report summarisation results (Nenkova and McKeown, 2011). As the name of the metric illustrates, ROUGE evaluations are recall-oriented. The basic premise is that ROUGE quantifies the amount of overlapping informational content between a summary text under evaluation, and one or more gold-standard summary texts. As discussed in Section 2.3, a gold-standard summary is an exemplar summary, typically authored by an expert human annotator, and this style of evaluation is known as content coverage (Over et al., 2007).

Within ROUGE, there are various methods of quantifying content coverage between a summary text and gold-standard summary text(s). The variants of ROUGE are: ROUGE-N (*n*gram co-occurrence); ROUGE-L (longest common subsequence); ROUGE-W (weighted longest common subsequence); ROUGE-S (skip-bigram); and ROUGE-SU (skip-bigram plus unigram co-occurrence). However, for evaluating the task of multi-document newswire summarisation, the current best practice is to report ROUGE-N recall scores (Hong et al., 2014), due to the reported agreement with expert manual evaluation scores (Owczarzak et al., 2012).

Specifically, ROUGE-N measures the degree to which the summary being evaluated contains the same *n*grams as the gold-standard summary. Formally, where $N$ is the number of *n*grams, ROUGE-1 and ROUGE-2 (i.e. unigram and bigram) recall and precision are defined:

$$\text{ROUGE-N Recall} = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \tag{3.1}$$

[1] berouge.com

[2] duc.nist.gov/duc2004/tasks.html

$$\text{ROUGE-N Precision} = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{CandidateSummary\}} \sum_{gram_n \in S} Count(gram_n)} \tag{3.2}$$

When evaluating information retrieval systems (Croft et al., 2010; Büttcher et al., 2010), the recall metric quantifies the number of relevant documents that are retrieved, and the precision metric quantifies the number of retrieved documents that are relevant. In the context of summarisation evaluation with ROUGE, the ROUGE-N recall metric quantifies the number of gold-standard *n*grams in the summary text, whereas the ROUGE-N precision metric quantifies the number of *n*grams in the summary text that appear in the gold-standard text(s).

However, unlike ad-hoc retrieval evaluation, typically the results (i.e. the summary) under evaluation is required to be of a specific fixed length, e.g. 100 words in the DUC 2004 Task 2 evaluation. As such, it is much less common to report ROUGE-N precision in the literature. This is because, in the summarisation task, there is no equivalent case of returning every document in the collection. That is, by definition, a summary cannot contain all the content of the original text(s). Subsequently, this means we do not observe the recall–precision trade-off, evident in ad-hoc retrieval evaluation, when evaluating fixed-length summary texts.

### 3.1.2 A Version of ROUGE Extended with Word Embeddings

The ROUGE-WE metric was proposed by Ng and Abrecht (2015), with a reference implementation publicly available[1]. The metric is implemented as a direct extension of ROUGE, and functions in a similar manner (c.f. Equation 3.1 and 3.2). As a recent proposal, it is yet to gain traction in the summarisation literature (to report results). However, ROUGE-WE is an attempt to overcome a perceived shortcoming of ROUGE. When computing *n*gram overlap, ROUGE performs exact string matching between the summary and the gold-standard summary text(s). If summary *A* contains the unigram "football", summary *B* contains the unigram "soccer", and the language used in the gold-standard mentions only "football", under ROUGE summary *B* will not get any credit despite the semantic synonymy. ROUGE-WE moves beyond exact string matching in the ROUGE *n*gram co-occurrence function by utilising word embeddings (Mikolov et al., 2013).

[1] github.com/ng-j-p/rouge-we

Specifically, in place of ROUGE exact *n*gram matching, $Count_{match}(gram_n)$ in Equations 3.1 and 3.2, ROUGE-WE sums real-valued semantic similarity scores between *n*grams in the summary and the gold-standard text(s). Equation 3.3 shows the ROUGE *n*gram matching function, and Equation 3.4 shows the ROUGE-WE *n*gram similarity function.

$$f_R(w_1, w_2) = \begin{cases} 1, & \text{if } w_1 = w_2 \\ 0, & \text{otherwise} \end{cases} \quad (3.3) \qquad f_{WE}(w_1, w_2) = \begin{cases} 0, & \text{if } v_1 \text{or } v_2 \text{ are OOV} \\ v_1 \cdot v_2, & \text{otherwise} \end{cases} \quad (3.4)$$

Given content (unigrams or bigrams) being compared, $(w_1, w_2)$, the ROUGE matching function (Equation 3.3) returns a score of 1 if there is an exact lexical match, or 0 otherwise. The ROUGE-WE similarity function (Equation 3.4), however, will return a semantic similarity score based on the word embeddings of $(w_1, w_2)$, i.e. the dot product of $(v_1, v_2)$.

### 3.1.3 Framework for Evaluating Summaries Automatically

The FRESA metric was proposed by Saggion et al. (2010), with a reference implementation publicly available[1]. FRESA was used in the INEX[2] Question Answering track, and Tweet Contextualization track. FRESA differs from ROUGE-based metrics in two important points. First, Jensen–Shannon divergence (Lin, 1991) is used to measure the content coverage between the summary text being evaluated and the gold-standard text(s). Second, FRESA also permits a model-free style of evaluation. That is, FRESA can evaluate without a summarisation gold-standard, by comparing a summary text to the original input document(s).

Specifically, given two probability distributions, $P$ and $Q$, that represent unigram or bigram language models of the texts being evaluated, Jensen–Shannon divergence is defined:

$$D_{JS}(P\|Q) = \frac{1}{2}D_{KL}(P\|M) + \frac{1}{2}D_{KL}(Q\|M) \quad (3.5)$$

Where $M = \frac{1}{2}(P+Q)$, and $D_{KL}(P\|Q) = \sum_i P(i)\log_2\frac{P(i)}{Q(i)}$ (i.e. defining Jensen–Shannon divergence in terms of Kullback–Leibler divergence (Kullback and Leibler, 1951), $D_{KL}$). Summary text(s) are scored as either the Jensen–Shannon divergence of the summary text

[1] fresa.talne.eu

[2] inex.mmci.uni-saarland.de

and the gold-standard text(s), or the Jensen–Shannon divergence of the summary text and the original document(s). When a unigram language model is used, for representing text(s) under evaluation, the metric is known as FRESA-1, and FRESA-2 when a bigram language model is used. For empirical observations, lower values are better, as FRESA measures divergence.

In our later experiments, in Section 3.3, we refer to ROUGE-1 recall as "R1R", ROUGE-1 precision as "R1P", ROUGE-2 recall as "R2R", ROUGE-2 precision as "R2P", and denote the word embeddings version of ROUGE using "(WE)", e.g. ROUGE-WE unigram recall is referred to as "R1R(WE)". When using a gold-standard with FRESA, we denote FRESA-1 and FRESA-2 as "F1(GS)" and "F2(GS)". When evaluating without a gold-standard using FRESA, we denote FRESA-1 and FRESA-2 as "F1(MF)" and "F2(MF)" (i.e. model-free).

## 3.2 Manual Judgements for Summary Quality

As discussed in Section 2.3, to manually evaluate the quality of summary text(s) a criteria for distinguishing high-quality and low-quality summary text(s) must be defined. Further, a method for soliciting judgements of summary quality, from human annotators, must be instantiated. We describe each of these below, where we first define the DUC linguistic quality criteria, which provides a set of guidelines to assist human annotators in assessing the quality of summary text(s). We then describe a crowd-sourced user-study, in order to obtain summary quality judgements from non-expert annotators.

### 3.2.1 Linguistic Quality Criteria

Throughout the Document Understanding Conference (DUC) summarisation evaluation campaigns, the linguistic quality of system-produced summaries was manually evaluated using a specific set of criteria (Over et al., 2007). The DUC linguistic quality criteria are designed to assist a human annotator in providing their assessment of the quality of summary text(s). The particular qualities under consideration are the readability and fluency of a given summary. The DUC linguistic quality criteria, when used to measure summary quality, provide a quantification of summary readability that is in sharp contrast to automatic summarisation evaluation metrics, which are restricted to measuring content coverage with respect to a gold-standard or the summarised document(s). However, obtaining such nuanced assessments of

1. **Grammaticality** – "The summary should have no datelines, system-internal formatting, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read."
2. **Non-redundancy** – "There should be no unnecessary repetition in the summary. Unnecessary repetition might take the form of whole sentences that are repeated, or repeated facts, or the repeated use of a noun or noun phrase (e.g., "Bill Clinton") when a pronoun ("he") would suffice."
3. **Referential clarity** – "It should be easy to identify who or what the pronouns and noun phrases in the summary are referring to. If a person or other entity is mentioned, it should be clear what their role in the story is. So, a reference would be unclear if an entity is referenced but its identity or relation to the story remains unclear."
4. **Focus** – "The summary should have a focus; sentences should only contain information that is related to the rest of the summary."
5. **Structure and Coherence** – "The summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to sentence to a coherent body of information about a topic."

**Figure 3.1:** The DUC linguistic quality criteria, used to evaluate summary text(s).

the linguistic quality of summary text(s) requires human assessors to read the summary text. The DUC linguistic quality criteria are reproduced verbatim[1] in Figure 3.1.

Figure 3.1 illustrates several desirable characteristics of a summary text. In particular, a summary text should not contain basic formatting errors or partial sentence snippets that inhibit readability. Further, given the short amount of text available in which to express information, a key characteristic of a summary is that information should not be repeated. Furthermore, unresolved anaphora harm summary readability, e.g. "she said" or "they did", where the summary text does not actually define who "she" is, or who "they" are. The sentences within a summary should also be on-topic, i.e. only contain salient information. Finally, an ideal summary text should exhibit structure and coherence. This can be a particular problem for multi-document newswire summarisation, if the summary is a non-ordered collection of sentences and the aim is to convey a sequence of sub-events (Mishra and Berberich, 2017).

Manual evaluation of the linguistic quality of a summary text provides a useful counterbalance to automatic summarisation evaluation methods (i.e. content coverage). In partic-

[1] www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt

ular, under a ROUGE-based evaluation, it is possible to obtain what appear to be effective evaluation scores, while producing summary text(s) that are actually unreadable for humans. By simply constructing a summary that consists of a non-ordered bag-of-words, where the ngrams in the summary happen to frequently occur in the gold-standard summary text(s), it has been shown that ROUGE can be fooled into returning high recall scores (Sjöbergh, 2007). Subsequently, manually evaluating a summary, by asking a human to read the summary text, acts as an important safety check against the limitations of automatic evaluation.

### 3.2.2 Crowd-sourced User-study

Given a specific set of criteria, shown in Figure 3.1, for evaluating the linguistic quality of a summary text, we now describe a method for soliciting summary quality judgements from non-expert annotators. In our user-study, we operationalise manual summarisation evaluation via crowd-sourcing. In order to perform manual summarisation evaluation in a crowd-sourcing environment, we have to formulate the procedure as a micro-task. A micro-task is typically a small unit of recompensed, easily comprehensible work, which should only require a short-term time commitment of the user engaging in the task (Kittur et al., 2008). To obtain crowd-sourced judgements of summary quality, we require that human annotators: (1) understand the task they have been asked to complete; (2) read a summary text; and (3) provide a judgement regarding the linguistic quality of the given summary text.

For the first requirement, crowd-workers are asked to read the DUC linguistic quality criteria. This set of guidelines, which evolved over several years of manual evaluation initiatives (Over et al., 2007), provides a robust mechanism to train human assessors in the task of summarisation evaluation. For the second requirement, crowd-workers are shown a summary text. The summary is displayed in a natural language format, i.e. the text is not decomposed into bullet points, sentence segmented, or annotated in any way. Only the summary text is shown, the original document(s) are not supplied, and the gold-standard summary text(s) are withheld. For our third requirement, a web interface control is used to allow the crowd-worker to provide a judgement on the quality of the summary. Specifically, a radio box is displayed showing a 10-point scale, with clear labelling of the scale (low vs. high quality). The interface for soliciting summary quality assessments is shown in Figure 3.2. The experimental setup of our user-study is described in Section 3.3.2.

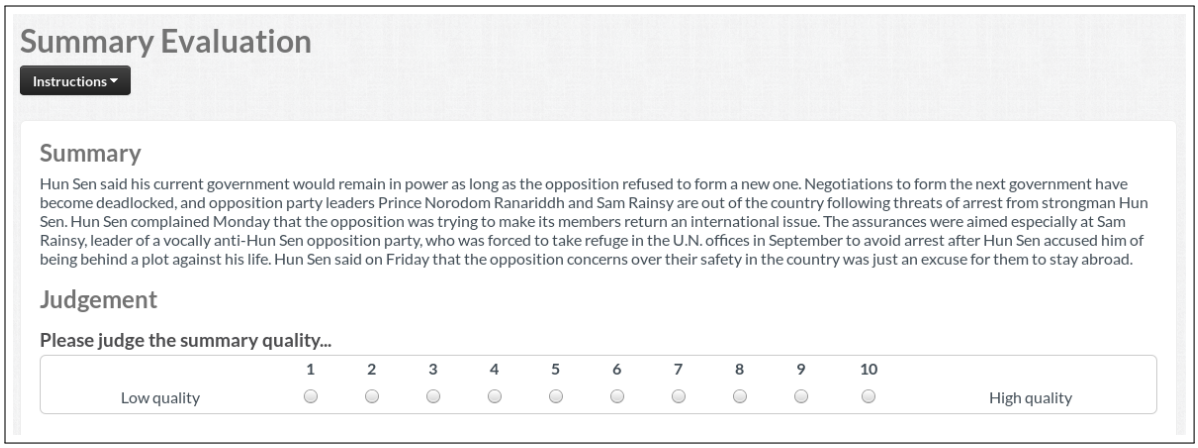



**Figure 3.2:** The interface for our user-study, for soliciting judgements for the linguistic quality of summary text.

## 3.3 Evaluation

In this section, we conduct a crowd-sourced user-study examining the validity of automatic evaluation metrics for multi-document newswire summarisation. We begin by stating our research questions, then describe our experimental setup. Results are provided over the DUC 2004 Task 2 dataset, for the task of generic extractive multi-document newswire summarisation. Finally, we discuss and analyse our empirical observations.

### 3.3.1 Research Questions

In our Thesis Statement (Section 1.2), we formed Hypothesis 1:

> We hypothesise that automatic summarisation evaluation metrics, which measure content coverage with respect to a gold-standard summary, exhibit strong correlation with non-expert crowd-sourced judgements for the linguistic quality of summary text(s).

To validate Hypothesis 1, we address the following research questions:

**Research Question 3.1.** Are automatic summarisation evaluation metrics aligned with non-expert crowd-sourced judgements of summary quality, with respect to the categorisation of summarisation baselines and state-of-the-art systems?

**Research Question 3.2.** Are automatic summarisation evaluation metrics correlated with non-expert crowd-sourced judgements of summary quality, with respect to system rankings?

Both of our research questions investigate the relationship of automatic summarisation evaluation metrics with non-expert crowd-sourced manual judgements for summary quality. For our first research question, we investigate the broad alignment of automatic metrics with non-expert manual judgements. We would expect that automatic summarisation evaluation metrics generally agree with crowd-sourced manual judgements of summary quality, with respect to the categorisation of baseline algorithms and state-of-the-art systems. For our second research question, we formally quantify the correlation of automatic metrics with non-expert judgements. In particular, as we assert in our Thesis Statement (Section 1.2), a valid automatic summarisation evaluation metric should provide a measurement of summary quality that is aligned with non-expert judgements of summary quality. We would expect that, if an automatic metric is valid, the system ranking obtained via automatic evaluation is correlated with the system ranking obtained via manual evaluation.

### 3.3.2 Experimental Setup

To answer our research questions, a user-study is conducted via the CrowdFlower[1] platform. We manually evaluate 12 summarisation algorithms (5 baseline systems, and 7 state-of-the-art systems) over the 50-topic DUC 2004 dataset using summary texts from SumRepo[2]. Assessors are provided with evaluation criteria by which judgements of summary quality are to be made, specifically, the DUC linguistic quality criteria (Section 3.2.1). Due to the nature of the task, providing judgements on the linguistic quality of English-language newswire text, we restrict the pool of crowd-workers to English-speaking countries. Following the recommendations of Owczarzak et al. (2012), summaries under evaluation with automatic evaluation metrics are stemmed, and stopwords are not removed. Summary text(s) shown to crowd-workers for linguistic quality assessment are not subjected to stemming or stopword removal. For each of the 12 systems, over each of the 50 topics, the system-produced summary is judged by 5 unique crowd-workers. In total, 412 crowd-workers participated in the user-study (inter-annotator agreement is reported in Section 3.3.3). Via CrowdFlower, we obtained 3,000 assessments (12 systems $*$ 50 topics $*$ 5 assessors) for a cost of $109.74.

[1] crowdflower.com

[2] www.seas.upenn.edu/~nlp/corpora/sumrepo.html

To quantify the alignment of automatic summarisation evaluation metrics with non-expert quality judgements, system rankings based on automatic evaluation are compared with a system ranking based on crowd-sourced quality judgements. We report Spearman (1904) $\rho$ and Kendall (1938) $\tau$ rank correlation coefficients. In the discussion of the results from our correlation analysis, we qualitatively interpret correlation coefficients as follows: $>$ .10 weak; $>$ .30 moderate; $>$ .50 strong; and $>$ .70 very strong (Rosenthal, 1996). Further, we use the Fisher (1921) and Williams (1959) tests to assess the statistical significance between pairs of metric correlations (i.e. metric vs. metric), reporting $p$-values. Our sample size is $N = 12$.

### 3.3.3 Experimental Results

#### Crowd-sourced User-study

In this section, we present empirical observations over the DUC 2004 Task 2 dataset, for the task of generic multi-document newswire summarisation. We begin by providing the results of our user-study, evaluating the linguistic quality of summary text(s). Then, based on the results from the user-study, we address our two research questions.

Table 3.1 provides the results for our crowd-sourced user-study. In Table 3.1, we show the per-system linguistic quality scores for each of the 12 summarisation systems under evaluation. The per-system judgements provided by the crowd-workers are first aggregated at the topic level (i.e. mean of the 10-point scale judgements from 5 different assessors), and then aggregated at the dataset level (i.e. mean over the 50 topics of DUC 2004). Further, in Table 3.1, we quantify the per-system inter-annotator agreement (i.e. inter-rater reliability) using Krippendorff's $\alpha$ (Artstein and Poesio, 2008). From Table 3.1, we first observe that the linguistic quality scores for all 12 systems can be used to establish a ranking of systems. Specifically, the crowd-sourced linguistic quality evaluation has returned a system ranking of: [ ICSISumm $>$ GreedyKL $>$ RegSum $>$ DPP $>$ Submodular $>$ CLASSY11 $>$ OCCAMS_V $>$ LexRank $>$ Centroid $>$ TsSum $>$ CLASSY04 $>$ FreqSum ]. We investigate the alignment and correlation of this manual ranking with system rankings established via automatic summarisation evaluation metrics in research question 3.1 and research question 3.2.

We now consider the linguistic quality assessments in more detail. From the results in Table 3.1, we observe that the range of scores for linguistic quality assessments is between 7.16 (min) and 8.10 (max), with a mean of 7.70, and a standard deviation of $\sigma = 0.26$. Further,

**Table 3.1:** Manual summarisation evaluation results, reporting crowd-sourced linguistic quality scores, for SumRepo's 5 standard baselines and 7 state-of-the-art systems, over the DUC 2004 Task 2 dataset. We report mean Linguistic Quality (LQ), and Krippendorff's $\alpha$ (measuring inter-annotator agreement), ordered by LQ.

| System | LQ | $\alpha$ |
|---|---|---|
| *FreqSum* | 7.16 | 0.26 |
| *CLASSY04* | 7.36 | 0.21 |
| *TsSum* | 7.60 | 0.21 |
| *Centroid* | 7.64 | 0.19 |
| *LexRank* | 7.66 | 0.26 |
| *OCCAMS_V* | 7.70 | 0.29 |
| *CLASSY11* | 7.71 | 0.16 |
| *Submodular* | 7.75 | 0.19 |
| *DPP* | 7.80 | 0.19 |
| *RegSum* | 7.85 | 0.18 |
| *GreedyKL* | 8.05 | 0.29 |
| *ICSISumm* | 8.10 | 0.23 |

while evaluating the linguistic quality of a summary text is a subjective task, we observe that there is measurable per-system agreement (Krippendorff's $\alpha$) among the assessments provided by the crowd-workers. However, the magnitude of $\alpha$ (measuring inter-annotator agreement) also indicates a level of disagreement in the crowd-sourced judgements.

We investigate such agreement and disagreement in Table 3.2, using LexRank as an example. Table 3.2 shows the per-topic linguistic quality assessments provided by 5 different assessors, and the standard deviation of those assessments. In particular, we show the five topics where the standard deviation between assessments is lowest, and show the five topics where the standard deviation between assessments is highest. In our study, agreement means that two or more crowd-workers have assigned the exact same value to a particular summary text, where linguistic quality assessments are based on a 10-point numerical-scale (i.e. $[1..10]$). The magnitude of disagreement in such assessments is important. For example, for topic "d30022" where we obtain judgements of $[8,8,9,9,10]$, there exists disagreement, but generally the assessors agree that this particular summary text is of a high linguistic quality. However, for topic "d30056", where we obtain judgements of $[1,7,7,9,10]$, the magnitude

**Table 3.2:** Per-topic linguistic quality assessments from 5 assessors, for the LexRank system, over DUC 2004. We show the 5 topics where the standard deviation is lowest, and 5 topics where the standard deviation is highest.

| Topic | LQ1 | LQ2 | LQ3 | LQ4 | LQ5 | $\sigma$ |
|---|---|---|---|---|---|---|
| d30022 | 8 | 8 | 9 | 9 | 10 | 0.8367 |
| d31026 | 8 | 9 | 10 | 10 | 10 | 0.8944 |
| d31043 | 8 | 9 | 10 | 10 | 10 | 0.8944 |
| d30017 | 8 | 9 | 10 | 10 | 10 | 0.8944 |
| d30024 | 8 | 10 | 10 | 10 | 10 | 0.8944 |
| **Topic** | **LQ1** | **LQ2** | **LQ3** | **LQ4** | **LQ5** | $\sigma$ |
| d31022 | 3 | 4 | 4 | 9 | 10 | 3.2404 |
| d31031 | 2 | 7 | 8 | 10 | 10 | 3.2863 |
| d31038 | 2 | 8 | 8 | 10 | 10 | 3.2863 |
| d30002 | 2 | 8 | 9 | 10 | 10 | 3.3466 |
| d30056 | 1 | 7 | 7 | 9 | 10 | 3.4929 |

of disagreement is more substantial. From this, we can conclude that not all topics are equal, i.e. that some topics are harder to judge than others, and that such differences in the ability of assessors to judge specific topics is the key source of disagreement in our user-study. Another potential source of disagreement is the choice of a 10-point numerical-scale. Modifying the user-study by soliciting assessments on a 5-point scale, or using a binary scale (such as "low linguistic quality" vs. "high linguistic quality") may have reduced disagreement.

To conclude our analysis of the crowd-sourced linguistic quality user-study, we provide the results of statistical significance tests in Table 3.3. We report the pair-wise statistical significance between systems, using the Student's t-test (two-tailed, paired sample, 95% confidence level). From Table 3.3, we observe that there exists statistically significant differences between the linguistic quality scores assigned to systems by crowd-sourced annotators. For example, the most effective system, ICSISumm, is significantly more effective than 9 other systems. Further, we observe that 10 systems are significantly more effective than FreqSum, 5 systems are significantly more effective than CLASSY04, and 2 systems are significantly more effective than TsSum. Given that we have quantified that there exists agreement between workers in our user-study, and that there are statistically significant differences between sys-

**Table 3.3:** Statistical significance tests over linguistic quality scores from our user-study. 10 systems are significantly more effective than FreqSum, 5 systems are significantly more effective than CLASSY04, 2 systems are significantly more effective than TsSum, and ICSISumm is significantly more effective than 9 other systems.

| System | *FreqSum* | *CLASSY04* | *TsSum* | *Centroid* | *LexRank* | *OCCAMS_V* | *CLASSY11* | *Submodular* | *DPP* | *RegSum* | *GreedyKL* | *ICSISumm* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *FreqSum* | – | | | | | | | | | | | |
| *CLASSY04* | | – | | | | | | | | | | |
| *TsSum* | ✔ | | – | | | | | | | | | |
| *Centroid* | ✔ | | | – | | | | | | | | |
| *LexRank* | ✔ | | | | – | | | | | | | |
| *OCCAMS_V* | ✔ | | | | | – | | | | | | |
| *CLASSY11* | ✔ | | | | | | – | | | | | |
| *Submodular* | ✔ | ✔ | | | | | | – | | | | |
| *DPP* | ✔ | ✔ | | | | | | | – | | | |
| *RegSum* | ✔ | ✔ | | | | | | | | – | | |
| *GreedyKL* | ✔ | ✔ | ✔ | | | | | | | | – | |
| *ICSISumm* | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | | – |

tems, we now analyse the manual ranking of systems with respect to system rankings obtained by automatic summarisation evaluation metrics.

### Research Question 3.1

We now address research question 3.1, where we seek to establish if automatic summarisation evaluation metrics are aligned with non-expert crowd-sourced judgements of summary quality, with respect to the categorisation of summarisation baselines and state-of-the-art systems. To address research question 3.1, we refer to Table 3.4, which presents summarisation evaluation results for both manual and automatic procedures, for 5 baseline algorithms, and 7 state-of-the-art systems, ordered by ROUGE-2 recall (R2R).

In Table 3.4, we show scores obtained via 12 automatic evaluation metrics, from ROUGE, ROUGE-WE, and FRESA (Section 3.1), and linguistic quality scores obtained via our crowd-sourced user-study (Section 3.2). The designation of summarisation approaches as "baselines" or "state-of-the-art", shown in the upper and lower sections of Table 3.4, follows the taxonomy of Hong et al. (2014), which we have previously discussed in Section 2.4. For each of the 12 automatic metrics, and the linguistic quality evaluation, we annotate the 7 (numerically) most effective results in bold. We would expect that, if automatic metrics are aligned with manual judgements, in terms of the 5 baseline algorithms and 7 state-of-the-art systems, the pattern of bold annotations observed for non-expert crowd-sourced linguistic quality (LQ) evaluation will be reflected across the observations for automatic evaluation metrics.

**Table 3.4:** Summarisation evaluation results, reporting crowd-sourced linguistic quality results, and 12 different automatic evaluation metrics, for SumRepo's 5 standard baselines and 7 state-of-the-art systems, over DUC 2004. We report results for Linguistic Quality (LQ), ROUGE, ROUGE extended with word embeddings (WE), FRESA when using a gold-standard (GS), and FRESA when not using a gold-standard, i.e. model-free (MF). We evaluate: ROUGE-1 recall (R1R), ROUGE-1 precision (R1P), ROUGE-2 recall (R2R), ROUGE-2 precision (R2P), FRESA-1 (F1), and FRESA-2 (F2). Per measure, the 7 most (numerically) effective systems are shown in bold. For LQ, ROUGE, and ROUGE-WE, higher is better, for FRESA, lower is better (measuring divergence).

| | | ROUGE | | | | ROUGE (WE) | | | | FRESA (GS) | | FRESA (MF) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Baselines (5)** | **LQ** | **R1R** | **R1P** | **R2R** | **R2P** | **R1R** | **R1P** | **R2R** | **R2P** | **F1** | **F2** | **F1** | **F2** |
| *LexRank* | 7.66 | 36.00 | 35.94 | 7.51 | 7.49 | 21.41 | 21.37 | 4.57 | 4.55 | 13.81 | 2.90 | 4.26 | 0.93 |
| *Centroid* | 7.64 | 36.42 | 35.95 | 7.98 | 7.87 | 21.59 | 21.31 | 4.58 | 4.51 | 13.68 | 2.64 | 4.07 | 0.98 |
| *FreqSum* | 7.16 | 35.31 | 34.93 | 8.12 | 8.02 | 21.01 | 20.78 | 4.74 | 4.69 | 13.59 | 2.61 | 4.33 | 0.94 |
| *TsSum* | 7.60 | 35.93 | 35.63 | 8.16 | 8.09 | 21.05 | 20.87 | 4.81 | 4.76 | 13.71 | **2.43** | **3.86** | **0.84** |
| *GreedyKL* | **8.05** | **38.03** | **37.60** | 8.56 | 8.46 | **22.63** | **22.38** | 5.01 | 4.95 | **11.09** | 2.46 | **3.47** | **0.88** |
| **SotA (7)** | **LQ** | **R1R** | **R1P** | **R2R** | **R2P** | **R1R** | **R1P** | **R2R** | **R2P** | **F1** | **F2** | **F1** | **F2** |
| *CLASSY04* | 7.36 | **37.71** | 37.33 | **9.02** | **8.92** | **22.19** | 21.97 | **5.15** | **5.10** | **11.34** | **2.20** | **3.87** | 1.06 |
| *CLASSY11* | **7.71** | 37.21 | **37.43** | **9.21** | **9.26** | 21.90 | **22.03** | **5.24** | **5.26** | 13.69 | 2.57 | 4.08 | **0.92** |
| *Submodular* | **7.75** | **39.23** | **39.30** | **9.37** | **9.38** | **23.19** | **23.22** | **5.29** | **5.29** | **13.23** | **2.40** | 3.90 | 0.99 |
| *DPP* | **7.80** | **39.84** | **39.75** | **9.62** | **9.59** | **23.52** | **23.47** | **5.62** | **5.61** | **12.03** | **2.09** | **3.71** | **0.83** |
| *RegSum* | **7.85** | **38.60** | **38.30** | **9.78** | **9.70** | **22.47** | **22.29** | **5.49** | **5.44** | **12.38** | **1.72** | **3.83** | **0.81** |
| *OCCAMS_V* | **7.70** | **38.50** | **38.36** | **9.75** | **9.72** | **23.14** | **23.06** | **5.61** | **5.58** | **11.58** | **2.36** | **3.65** | **0.83** |
| *ICSISumm* | **8.10** | **38.44** | **38.61** | **9.81** | **9.86** | **22.35** | **22.45** | **5.57** | **5.59** | **10.22** | **1.98** | **3.58** | **0.73** |

Considering both the manual and automatic evaluation results, from Table 3.4, we first observe that the majority of the most effective scores (shown in bold) are in the lower half of the table, which is aligned with the "state-of-the-art" categorisation of Hong et al. (2014). A notable exception is GreedyKL, which manual evaluation judgements, and 7 out of 12 automatic metrics, have determined is an effective summarisation algorithm. Further, we observe that TsSum performs effectively under the FRESA automatic metrics.

Considering the crowd-sourced linguistic quality results, from Table 3.4, we observe that, with respect to the Hong et al. (2014) categorisation, the non-expert crowd-worker assessors have agreed with the categorisation of 4 out of 5 approaches as "baseline" algorithms, and agreed with the categorisation of 6 out of 7 approaches as "state-of-the-art" systems. The exceptions, where non-expert crowd-sourced manual evaluation disagrees with Hong et al. (2014), are GreedyKL, which the crowd-workers have collectively rated 8.05 (second best),

and CLASSY04, which has been rated 7.36 (second worst). This manually derived system ranking is generally aligned with the Hong et al. (2014) categorisation, for baseline algorithms and state-of-the-art systems.

From the results in Table 3.4, we can answer research question 3.1. We conclude that automatic summarisation evaluation metrics are generally aligned with non-expert manual crowd-sourced judgements of summary quality, for the task of generic multi-document newswire summarisation, with respect to the categorisation of baselines and state-of-the-art systems.

**Research Question 3.2**

Given that we have observed such general alignment, we now formally quantify this alignment via a rank correlation analysis. We seek to understand if automatic summarisation evaluation metrics are correlated with non-expert crowd-sourced judgements of summary quality. As such, multiple system rankings, established via different automatic evaluation metrics, are compared to a single system ranking, established via manual evaluation. The expectation is that, if an automatic metric provides a useful proxy for manual evaluation, the system ranking obtained via that metric will exhibit correlation with the reference system ranking obtained via manual judgements.

To address research question 3.2, we refer to Figure 3.3, and Table 3.5. In Figure 3.3, we visualise the alignment of crowd-sourced linguistic quality (LQ) assessments with 12 automatic evaluation metrics. In Table 3.5, we quantify the correlation between manual judgements and automatic evaluation metrics, reporting Spearman's $\rho$ and Kendall's $\tau$.

Figure 3.3 plots the 12 summarisation systems under evaluation on the $x$-axis, and the (standardised) summarisation evaluation scores on the $y$-axis. The ordering of the summarisation approaches along the $x$-axis follows the system ranking established via non-expert crowd-sourced manual judgements. We plot the linguistic quality scores as a line. We also plot lines for the 3 metric variants, from each of ROUGE, ROUGE-WE, and FRESA, that exhibit the highest correlation with manual linguistic quality judgements (c.f. Table 3.5). Further points on the plot illustrate the behaviour of the other metrics variants within the 3 metric groups. From Figure 3.3, we can clearly visualise that the automatic summarisation evaluation metrics appear to exhibit a certain degree of correlation with manual judgements for summary quality. In Table 3.5, we formally quantify this correlation via a rank correlation analysis.

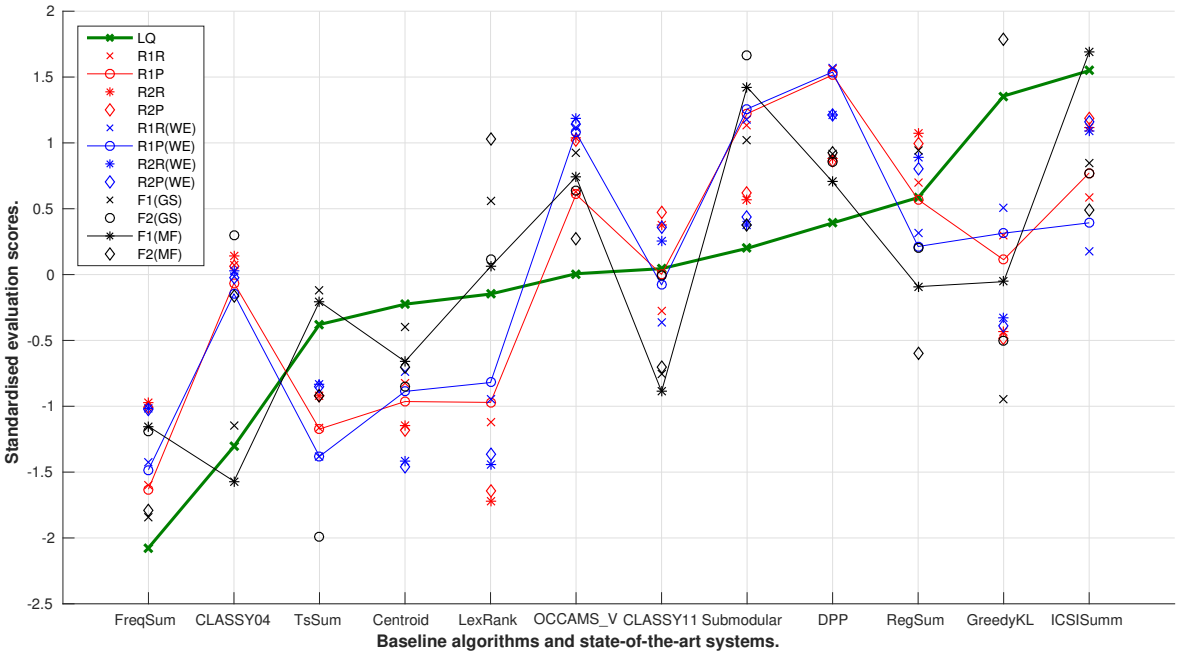



**Figure 3.3:** Distribution of (standardised) summarisation evaluation scores, for crowd-sourced linguistic quality (LQ) evaluation and 12 automatic summarisation evaluation metrics, for SumRepo's 5 standard baselines and 7 state-of-the-art systems, over DUC 2004. Visually, automatic metrics appear to be correlated with LQ scores.

Table 3.5 shows the rank correlation coefficients between crowd-sourced linguistic quality (LQ) assessments and automatic evaluation metrics, for SumRepo's 5 standard baselines and 7 state-of-the-art systems, over DUC 2004. We report Spearman's $\rho$ and Kendall's $\tau$ rank correlation coefficients, with $p$-values. For each metric group (ROUGE, ROUGE-WE, and FRESA), we annotate in bold the metric variant that exhibits the highest numerical correlation with manual summarisation evaluation, under both measures of rank correlation. Further, for the most effective metric variant, we provide statistical significance tests against the other metric variants within that metric group. The statistical difference between pairs of metric correlations is assessed using the Fisher (1921) and Williams (1959) tests, reporting $p$-values.

From Table 3.5, we first observe that under both Spearman's $\rho$ and Kendall's $\tau$, all automatic summarisation evaluation metrics exhibit at least moderate correlation with non-expert crowd-sourced manual judgements for the linguistic quality of summary text(s). Specifically, under Spearman's $\rho$, all metrics exhibit at least strong correlation, with 3 metrics exhibiting very strong correlation. Under Kendall's $\tau$, all metrics exhibit at least moderate correlation, with 5 metrics exhibiting strong correlation. Further, from the results in Table 3.5, we observe that both measures of rank correlation ($\rho$ and $\tau$) agree on which metric variant, in each of the

**Table 3.5:** Rank correlation coefficients between crowd-sourced linguistic quality (LQ) assessments and automatic evaluation metrics, for SumRepo's 5 standard baselines and 7 state-of-the-art systems, over DUC 2004. We report Spearman's $\rho$ and Kendall's $\tau$, with $p$-values. For ROUGE, ROUGE-WE, and FRESA, we annotate in bold the metric variant that exhibits the highest correlation with non-expert linguistic quality (LQ) assessments. Additionally, we report $p$-values for the Fisher and Williams tests for significant differences between correlations, with respect to the (ROUGE, ROUGE-WE, and FRESA) metrics that exhibit highest correlation.

| | **Spearman** | | **Sig. Difference** | | **Kendall** | | **Sig. Difference** | |
|---|---|---|---|---|---|---|---|---|
| **LQ vs.** | $\rho$ | $p$**-value** | **Fisher** | **Williams** | $\tau$ | $p$**-value** | **Fisher** | **Williams** |
| **R1R** | 0.7063 | 0.0133 | 0.7687 | 0.1413 | 0.5152 | 0.0210 | 0.8546 | 0.3312 |
| **R1P** | **0.7692** | 0.0053 | – | – | **0.5758** | 0.0088 | – | – |
| **R2R** | 0.6713 | 0.0204 | 0.6633 | 0.2362 | 0.4848 | 0.0311 | 0.7878 | 0.3559 |
| **R2P** | 0.6434 | 0.0280 | 0.5894 | 0.1773 | 0.4545 | 0.0447 | 0.7251 | 0.3066 |
| **R1R(WE)** | 0.6783 | 0.0188 | 0.7870 | 0.1523 | 0.4848 | 0.0311 | 0.7128 | 0.1869 |
| **R1P(WE)** | **0.7413** | 0.0082 | – | – | **0.6061** | 0.0054 | – | – |
| **R2R(WE)** | 0.5944 | 0.0458 | 0.5684 | 0.1330 | 0.3939 | 0.0863 | 0.5436 | 0.1775 |
| **R2P(WE)** | 0.6434 | 0.0280 | 0.6878 | 0.2256 | 0.4545 | 0.0447 | 0.6523 | 0.2512 |
| **F1(GS)** | -0.5245 | 0.0839 | 0.5671 | 0.1630 | -0.3939 | 0.0863 | 0.7449 | 0.3152 |
| **F2(GS)** | -0.5175 | 0.0887 | 0.5533 | 0.1977 | -0.3636 | 0.1160 | 0.6889 | 0.3065 |
| **F1(MF)** | **-0.6923** | 0.0159 | – | – | **-0.5152** | 0.0210 | – | – |
| **F2(MF)** | -0.6410 | 0.0247 | 0.8445 | 0.3968 | -0.5038 | 0.0278 | 0.9739 | 0.4849 |

metric groups, exhibits the numerically highest correlation with non-expert crowd-sourced judgements for the linguistic quality of summaries. In particular, as shown in bold, under the ROUGE-based metrics, ROUGE-1 precision exhibits the highest correlation with manual judgements, and under FRESA, FRESA-1 without a gold-standard (i.e. mode-free) exhibits the highest correlation with manual judgements. Specifically, R1P exhibits very strong correlation under Spearman's $\rho$ and strong correlation under Kendall's $\tau$. Similarly, R1P(WE) exhibits very strong correlation under Spearman's $\rho$ and strong correlation under Kendall's $\tau$. Further, F1(MF) exhibits strong correlation under both Spearman's $\rho$ and Kendall's $\tau$.

We now consider each of the 3 metric groups in turn. For ROUGE and ROUGE-WE, we examine results in terms of ROUGE-1 vs. ROUGE-2 (i.e. unigram vs. bigram), and in terms of recall vs. precision. For FRESA, we examine results in terms of FRESA-1 vs. FRESA-2

(i.e. unigram vs. bigram), and in terms of evaluating with or without a gold-standard. Considering the ROUGE metric, from the results in Table 3.5, we observe that ROUGE-1 (unigram) exhibits higher correlations than ROUGE-2 (bigram). Further, under ROUGE-1, precision exhibits higher correlations than recall, but under ROUGE-2, recall exhibits higher correlations than precision. Considering the ROUGE-WE metric, from the results in Table 3.5, we observe that ROUGE-WE-1 (unigram) exhibits higher correlations than ROUGE-WE-2 (bigram). Unlike ROUGE, under both ROUGE-WE-1 and ROUGE-WE-2 precision exhibits higher correlations than recall. Considering the FRESA metric, from the results in Table 3.5, we observe that FRESA-1 (unigram) exhibits higher correlations than FRESA-2 (bigram). Further, evaluating using FRESA's model-free style of evaluation exhibits higher correlation with manual judgements than when evaluating using a gold-standard with FRESA.

From the results in Table 3.5, we can now answer research question 3.2. We conclude that, for the ROUGE metric all variants are correlated with manual judgements, with the highest correlations observed for the unigram precision variant. Similarly for ROUGE-WE, we conclude that all variants are correlated with manual judgements, with the unigram precision variant exhibiting the highest correlation with manual judgements. For FRESA, we again conclude that all variants are correlated with manual judgements, with the unigram model-free variant exhibiting the highest correlations with manual judgements.

### 3.3.4 Discussion & Analysis

Having addressed our research questions, from Section 3.3.1, we now discuss and analyse our empirical observations, positioning our empirical results within this chapter with respect to the summarisation evaluation literature. We discuss: our observations regarding the numerically higher correlations with manual evaluation for unigram-based evaluation metrics; observations regarding numerically higher correlations when evaluating without a gold-standard using FRESA; and the difficulty in justifiably selecting a particular automatic metric based on our observations regarding statistical significance testing of metric–metric correlations.

To begin, from the results in Table 3.5, we note that the numerically highest correlations with non-expert manual evaluation are exhibited by automatic evaluation metrics that are unigram-based, specifically ROUGE-1, ROUGE-WE-1, and FRESA-1. For example, under ROUGE, R1R ($\rho = 0.7063, \tau = 0.5152$) and R1P ($\rho = 0.7692, \tau = 0.5758$) exhibit numer-

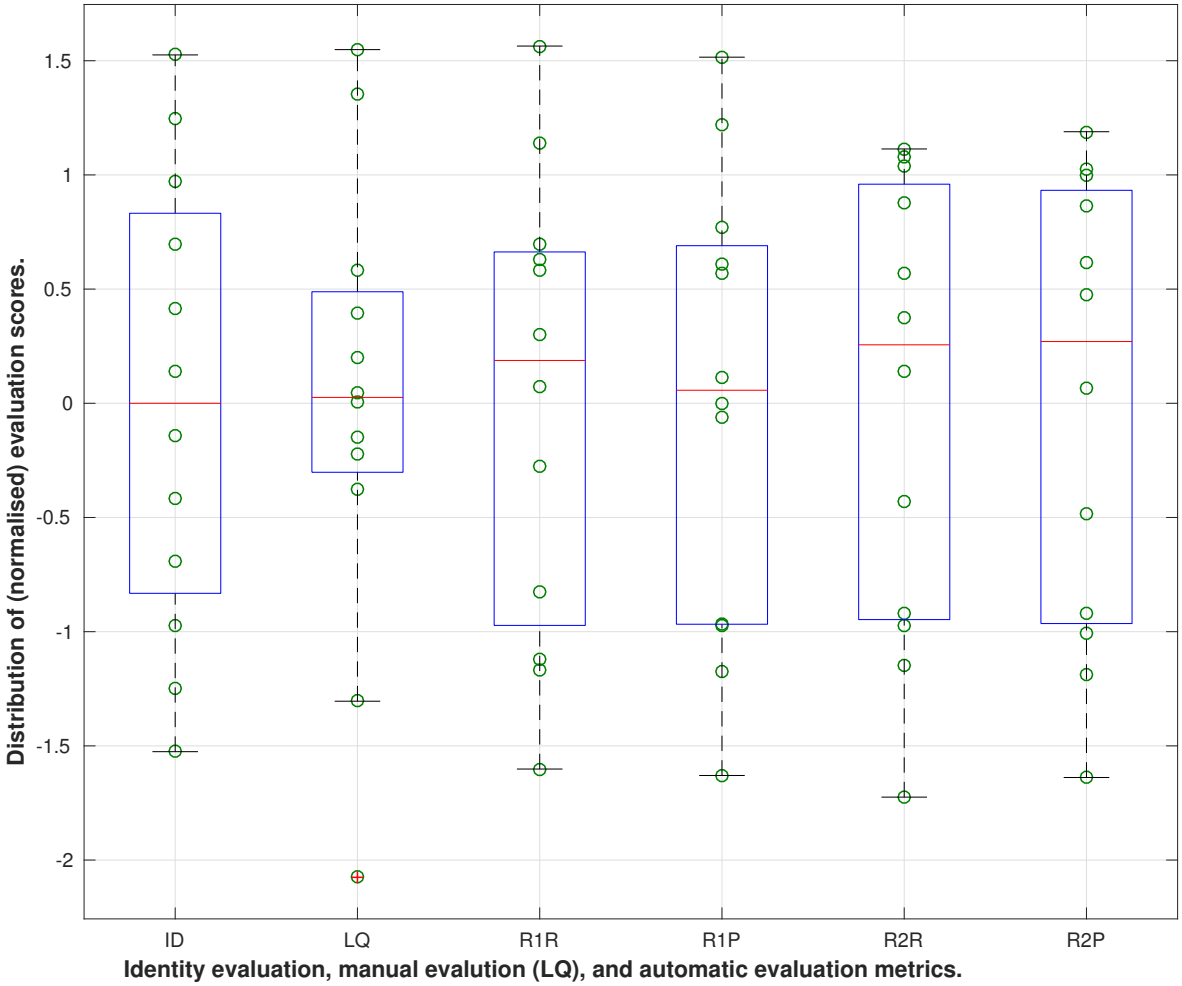



**Figure 3.4:** Boxplots of (standardised) summarisation evaluation scores, for crowd-sourced linguistic quality (LQ) evaluation and ROUGE evaluation metrics, for SumRepo's 5 standard baselines and 7 state-of-the-art systems, over DUC 2004. Additionally, we show an identity evaluation, where the 12 systems are assigned sequential scores ([1..12]). Further, we annotate each metric's boxplot with the 12 evaluation scores.

ically higher correlation with manual judgements than R2R ($\rho = 0.6713, \tau = 0.4848$) and R2P ($\rho = 0.6434, \tau = 0.4545$). We now examine this empirical observation, by analysing the scoring of summarisation systems under the ROUGE metric in more detail.

Figure 3.4 shows the distribution of standardised summarisation evaluation scores, for our non-expert crowd-sourced linguistic quality (LQ) evaluation, and ROUGE automatic summarisation evaluation metrics, for SumRepo's 5 standard baselines and 7 state-of-the-art systems, over DUC 2004. We additionally show an identity evaluation (ID), where a hypothetical evaluation metric has assigned sequential scores with a uniform magnitude in difference (i.e. [1..12]) between the 12 systems under evaluation. Further, we annotate the boxplot for each evaluation method with the 12 standardised evaluation scores (shown as circles).

From Figure 3.4, we first observe the distribution of scores in the case of simply assigning each system a sequential score in the range $[1..12]$. Specifically, for the ID boxplot, we see that the mean is zero, and that the upper-quartile, maximum, lower-quartile, and minimum regions are equally distributed, i.e. there is no skew in the data points, as the magnitude of difference in evaluation scores is uniform. With respect to the identity evaluation, for the LQ boxplot, we see that there is a contraction in the inter-quartile range, and also observe an outlier beyond the minimum observation point. Further, the distribution of scores in the LQ boxplot, for non-expert crowd-sourced linguistic quality (LQ) evaluation, illustrates that the manual evaluation procedure resulted in the identification of 2 systems that were clearly more effective than the others, and 2 systems that were clearly less effective than the others. Referring back to Figure 3.3, the two more effective systems are GreedyKL and ICSISumm, and the two less effective systems are FreqSum and CLASSY04.

If we examine the boxplots for the ROUGE metrics, we first observe that R2R and R2P (i.e. the bigram variants) exhibit a marked contraction in their 4th quartile whiskers, whereas R1R and R1P (i.e. the unigram variants) do not. Further, for R2R and R2P we also observe a group of 4 systems at or below the $-1$ point on the $y$-axis, whereas for R1R and R1P we observe a group of 3 systems at or below the $-1$ point. Additionally, the mean is slightly higher for the bigram variants. To conclude this point, we interpret the observations from Figure 3.3 as indicating that the bigram variants of ROUGE are less discriminative than the unigram variants at identifying systems at (particularly) the upper- and lower-ends of the effectiveness scale, and hence, are less correlated with manual evaluation than unigram ROUGE variants.

For our next discussion point, we refer back to results in Table 3.5. In particular, from our experiments we have observed that, under the FRESA automatic evaluation metric, the model-free (MF) variants exhibit numerically higher correlations with manual evaluation than when evaluating summaries using gold-standard (GS) summary text(s). Specifically, when evaluating using FRESA with a gold-standard, we observe correlations (with manual evaluation) of F1(GS) ($\rho = -0.5245, \tau = -0.3939$) and F2(GS) ($\rho = -0.5175, \tau = -0.3636$), which are lower than F1(MF) ($\rho = -0.6923, \tau = -0.5152$) and F2(MF) ($\rho = -0.6410, \tau = -0.5038$), when evaluating without a gold-standard. This means that when evaluating a summary text, under the FRESA automatic summarisation evaluation metric, it is not required that we have an expert annotator author exemplar summaries. The expectation may have been that in re-

moving the traditional gold-standard summary text(s) from the summarisation evaluation procedure, the model-free variant of FRESA would have exhibited less correlation with manual judgements. Indeed, to conclude this point, evaluating without a gold-standard, by comparing system-produced summaries to the original input document(s), is more closely aligned with non-expert crowd-sourced manual evaluation for the linguistic quality of summary text(s).

For our final discussion point, we examine the possibility of forming a justified selection of a particular automatic summarisation evaluation metric, based on our experiments in Section 3.3. As shown in Table 3.5, the results from performing statistical significance tests on the differences between metric–metric correlations are inconclusive – i.e. we fail to reject the null hypothesis that the difference in correlations is zero. Specifically, we observe that under both the Fisher and Williams tests, none of the metrics that exhibit the highest correlation with manual judgements (shown in bold in Table 3.5) are statistically significantly more correlated with manual judgements than the other metrics within the same metric group. Based on the interpretation of such significance testing, it is not possible to conclude that one metric is significantly better than another, for the 12 summarisation systems we investigated.

Further, from Table 3.6, we can see that several of the variants of automatic summarisation evaluation metrics are correlated with each other. In Table 3.6, we show the full matrix of automatic evaluation metric–metric correlations between the ROUGE, ROUGE-WE, and FRESA metric variants. In the upper-right section of the table, we report Spearman's $\rho$, and report Kendall's $\tau$ in the lower-left section of the table. From Table 3.6, with observed correlations such as R1R vs. R1P ($\rho = 0.9650, \tau = 0.8788$) and R2R vs. R2P ($\rho = 0.9930, \tau = 0.9697$), it is clear that a recommendation for recall variants over precision variants can not be made. Similarly, with observed correlations such as R1R vs. R2R ($\rho = 0.7902, \tau = 0.5455$) and R1P vs. R2P ($\rho = 0.8182, \tau = 0.6364$), it is also difficult to justify a recommendation for unigram variants over bigram variants. To conclude this final point, based on the observed non-significant differences in correlations (with manual evaluation) between metric variants, and then also the high correlations observed among the automatic metric variants themselves, it is not possible to state that one particular automatic evaluation metric is significantly better than another, based on our experiments in Section 3.3.

**Table 3.6:** Rank correlation coefficients between automatic evaluation metrics, for SumRepo's 5 standard baselines and 7 state-of-the-art systems, over DUC 2004. We report Spearman's $\rho$ (upper right) and Kendall's $\tau$ (lower left). For ROUGE, ROUGE-WE, and FRESA,, we annotate in bold the metric variant that exhibits the highest correlation with another metric in the same group.

| $\tau$ \ $\rho$ | ROUGE | | | | ROUGE (WE) | | | | FRESA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **R1R** | **R1P** | **R2R** | **R2P** | **R1R** | **R1P** | **R2R** | **R2P** | **F1(GS)** | **F2(GS)** | **F1(MF)** | **F2(MF)** |
| **R1R** | – | 0.9650 | 0.7902 | 0.7832 | 0.9650 | 0.9441 | 0.8531 | 0.8392 | -0.5734 | -0.7343 | -0.6014 | -0.3678 |
| **R1P** | 0.8788 | – | 0.8112 | 0.8182 | 0.9441 | **0.9790** | 0.8741 | 0.8741 | -0.5944 | -0.6853 | -0.6224 | -0.4168 |
| **R2R** | 0.5455 | 0.6061 | – | **0.9930** | 0.6923 | 0.7483 | 0.9371 | 0.9441 | -0.6434 | **-0.8951** | -0.6434 | -0.6480 |
| **R2P** | 0.5152 | 0.6364 | **0.9697** | – | 0.7063 | 0.7692 | 0.9510 | 0.9510 | -0.6573 | -0.8671 | -0.6573 | -0.6375 |
| **R1R(WE)** | **0.9091** | 0.8485 | 0.4545 | 0.4848 | – | 0.9650 | 0.8042 | 0.7762 | -0.6084 | -0.6154 | -0.6503 | -0.2942 |
| **R1P(WE)** | 0.8485 | **0.9091** | 0.5152 | 0.5455 | 0.8788 | – | 0.8462 | 0.8462 | -0.5804 | -0.6084 | -0.6294 | -0.3853 |
| **R2R(WE)** | 0.6364 | 0.6970 | 0.8485 | 0.8788 | 0.6061 | 0.6667 | – | **0.9860** | -0.6084 | -0.8392 | -0.6364 | -0.5919 |
| **R2P(WE)** | 0.5758 | 0.6970 | 0.8485 | 0.8788 | 0.5455 | 0.6667 | **0.9394** | – | -0.6084 | **-0.8531** | -0.6294 | -0.6235 |
| **F1(GS)** | -0.3333 | -0.3939 | -0.4848 | -0.5152 | -0.3636 | -0.3636 | -0.4545 | -0.4545 | – | 0.6713 | **0.7762** | 0.3187 |
| **F2(GS)** | -0.5455 | -0.4848 | **-0.7576** | -0.7273 | -0.4545 | -0.3939 | **-0.6667** | **-0.6667** | 0.4848 | – | 0.6713 | 0.5674 |
| **F1(MF)** | -0.3939 | -0.4545 | -0.4848 | -0.5152 | -0.4848 | -0.4242 | -0.4545 | -0.4545 | **0.6364** | 0.4848 | – | 0.6550 |
| **F2(MF)** | -0.2595 | -0.3206 | -0.5038 | -0.4733 | -0.1679 | -0.2901 | -0.3817 | -0.4428 | 0.1679 | 0.4733 | 0.4122 | – |

With respect to the literature on summarisation evaluation, we now have a fuller understanding of the correlations among the various evaluation paradigms. In particular, our experiments demonstrate that automatic summarisation evaluation metrics are correlated with non-expert crowd-sourced manual judgements for the linguistic quality of a summary, over the DUC 2004 Task 2 dataset (generic multi-document newswire summarisation). As reported by each of Lin (2004), Ng and Abrecht (2015), and Saggion et al. (2010), where the ROUGE, ROUGE-WE, and FRESA metrics were introduced, these automatic evaluation metrics have previously been reported to be correlated with expert (i.e. DUC/TAC) manual judgements.

Further, based on the scores from the manual summarisation evaluation experiments reported by Gillick and Liu (2010), we can derive a quantification as to the agreement of non-expert crowd-sourced manual judgements with expert manual judgements provided by TAC assessors. In particular, over the TAC 2008 dataset, based on scores assigned by crowd-workers from Mechanical Turk[1], we compute the correlation of non-expert crowd-sourced linguistic quality evaluation and expert TAC linguistic quality evaluation ($\rho = 0.7381, \tau = 0.6429$). As such, we can now conclude that: (1) automatic metrics are correlated with expert manual judgements; (2) automatic metrics are also correlated with non-expert manual judgements; and (3) non-expert manual judgements are correlated with expert judgements.

[1] mturk.com

Where we diverge from previous results in the literature, addressing the evaluation of summarisation, is in our observation that unigram-based automatic evaluation metrics (i.e. ROUGE-1) exhibit numerically higher correlations with non-expert manual judgements than bigram-based metrics. Whereas, for example, Owczarzak et al. (2012) reports that ROUGE-2 recall agrees best with expert manual evaluation. Further, Graham (2015) finds that higher-order ROUGE variants (i.e. ROUGE-2, ROUGE-3, and ROUGE-4) agree best with expert manual evaluation. Future work should investigate this discrepancy, between the correlation of automatic metrics with expert judgements and with non-expert judgements. For example, a better understanding of how crowd-workers manually evaluate summaries could help to illicit more effective crowd-sourcing evaluation protocols (Gillick and Liu, 2010). Further, a better understanding of crowd-based summary annotations could lead to obtaining more accurate large-scale training data, from non-expert crowd-sourced summary evaluations, which in turn could assist in the training of supervised models for automatically evaluating the linguistic quality of a summary text (Pitler et al., 2010; Ellouze et al., 2016).

From our experiments in Section 3.3, we demonstrate that when evaluating using the FRESA metric, scoring summaries with respect to a gold-standard exhibits numerically lower correlations with non-expert manual judgements than when evaluating using FRESA's model-free style of evaluation. In the summarisation evaluation literature, it has been shown that this model-free summarisation evaluation paradigm exhibits correlation with expert manual evaluation judgements (Saggion et al., 2010; Louis and Nenkova, 2013), when implementing model-free evaluation as the Jensen–Shannon divergence (JSD) between the original document(s) and the summary. However, neither Saggion et al. (2010) nor Louis and Nenkova (2013) explicitly quantify the correlation of JSD-based model-free metrics vs. JSD-based gold-standard metrics. From our experiments, we now have such a quantification, illustrating that JSD-based model-free evaluation is numerically more correlated with non-expert manual judgements than JSD-based gold-standard evaluation. This provides more confidence in the empirical results obtained when using a model-free style of summarisation evaluation.

With regards to the difficulties in the statistical significance testing of the difference in correlations between specific automatic summarisation evaluation metrics and manual summary judgements, we find ourselves in broad agreement with the literature. From our experiments in Section 3.3, we observed that under ROUGE, while R1P ($\rho = 0.7692, \tau = 0.5758$) exhibits

numerically higher correlation with manual judgements than R1R ($\rho = 0.7063, \tau = 0.5152$), the difference in correlation coefficients was not statistically significant under the Fisher or Williams tests. Indeed, Graham (2015) reports similar results regarding the statistical significance testing of the difference in correlation (vs. manual) between automatic evaluation metrics. Specifically, the experimental setup of Graham (2015) exhaustively examined 192 different parameter settings of ROUGE before statistical differences in correlation were observed, with a large sample size based on every summarisation system submitted for evaluation at DUC 2004. Given the small sample size in our experiments, where we evaluate the system ranking of 12 summarisation approaches, and the small sample size of the experiments of Gillick and Liu (2010), where 8 summarisation approaches are examined, it is difficult to make conclusions regarding the statistical significance between the performance of different automatic metrics to accurately reproduce manual evaluation rankings.

## 3.4 Chapter Summary

In this chapter, we investigated automatic summarisation evaluation metrics. We provided experimental results to empirically validate Hypothesis 1 from our Thesis Statement (Section 1.2). We validated our claim that automatic summarisation evaluation metrics, which measure content coverage with respect to a gold-standard summary, exhibit strong rank correlation with non-expert crowd-sourced judgements for the linguistic quality of summary text(s). We investigated the alignment of automatic summarisation evaluation metrics with non-expert crowd-sourced manual summarisation judgements. By answering Research Question 3.1, we established that automatic summarisation evaluation metrics generally agree with non-expert crowd-sourced manual summarisation judgements with respect to the categorisation of standard baselines and state-of-the-art systems. Further, by answering Research Question 3.2, we observed that system rankings obtained via automatic evaluation metrics are correlated with the system ranking obtained via non-expert crowd-sourced manual summarisation evaluation.

In conclusion, having validated automatic evaluation metrics against our manual evaluation procedure, we establish confidence in the empirical observations that are obtained via automatic summarisation evaluation metrics. Hence, in subsequent chapters, we use auto-

matic evaluation metrics in several of our experiments. In particular, in Chapter 4, we use automatic metrics to evaluate various baseline summarisation algorithms, in Chapter 5, we use automatic metrics as a means to label training data for supervised summarisation, in particular ROUGE-*n* precision, and for evaluating the effectiveness of learned models based on this training data. Further, in Chapter 6 and in Chapter 7, we use automatic metrics to evaluate our proposed entity-focused event summarisation features.

# Chapter 4

# On the Effectiveness of Unsupervised Summarisation Baselines

In this chapter, we address our second challenge regarding the identification of high-quality (i.e. effective) baseline summarisation algorithms. We re-implement newswire summarisation algorithms from the literature (Hong et al., 2014), thoroughly exploring algorithm design choices. In this thesis, we argue such algorithms, when improved, can provide strong baselines for use in empirical evaluations. We also claim that such algorithms can provide discriminative features for supervised summarisation models, investigated in Chapter 5.

Specifically, we claim that the effectiveness of standard multi-document newswire summarisation algorithms can be improved by varying algorithm design choices. We re-implement several variations of standard algorithms from the literature, revisiting and exploring assumptions regarding implementation details. We also propose to use our re-implementations of the standard baseline summarisation algorithms as features in a supervised summarisation setting. Specifically, we seek to identify a set of discriminative features for use when training supervised machine learned summarisation models. Identifying suitable features is one key criteria of a robust supervised framework. We further address this point in Chapter 5, where we investigate a variety of methods to obtain training data for supervised summarisation.

This chapter is based on the following publications: Mackie et al. (2014a, 2016).

## Chapter Outline

This chapter is organised as follows:

- Section 4.1 discusses the random (Section 4.1.1) and lead (Section 4.1.2) baselines, for establishing a lower-bounds on the effectiveness of extractive summarisation of newswire.
- Section 4.2 examines core components of the baseline summarisation algorithms, in particular, summary sentence ranking (Section 4.2.1) and anti-redundancy filtering (Section 4.2.2).
- Section 4.3 evaluates the effectiveness of unsupervised summarisation algorithms, providing experimental observations over the DUC 2004 Task 2 dataset (generic summarisation).

## 4.1 Establishing a Lower-bounds on Effectiveness

In this section, we describe two commonly used methods for establishing a lower-bounds on extractive summarisation effectiveness, particularly when summarising newswire articles. First, we describe a stochastic summary sentence selection method. In selecting random sentences for the summary, the minimum expected effectiveness that should be achieved by any reasonably effective summarisation algorithm under the extractive summarisation paradigm can be established. We argue that the random baseline should always be reported while conducting extractive summarisation experiments. Second, we describe a proposed improvement to the standard lead summarisation baseline used at the Document Understanding Conference (DUC). Specifically, we define a method of lead-based summary sentence selection whereby the interleaving of multiple leading sentences is passed through an anti-redundancy filter. Further, we argue that the random and lead summarisation baselines also facilitate an analysis of the documents sets to identify "easy" and "hard" summarisation topics.

### 4.1.1 Randomly Extracting Summary Sentences

The random baseline, while not universal, is often used while reporting results in the summarisation literature (Radev and Tam, 2003). Given a single document or multiple documents that are to be summarised, the random baseline simply extracts a unique set of random sentences, given a fixed summary length (e.g. 5 sentences). In the case of single document summarisation, the set of random sentences is drawn from one document, and across multiple documents for the case of multi-document summarisation. The set of random sentences are evaluated, and a summarisation effectiveness score is recorded for that particular random

sample. The process of sampling and evaluating random summary sentences is repeated for a number of trials (e.g. 100 samples). The scores for each of the trials are then averaged over the number of samples taken. Given the probability of obtaining more-effective and less-effective random summaries by chance, repeated sampling and averaging in this manner leads to a convergence on a robust final evaluation score for a randomly extracted summary.

Due to the nature of extractive summarisation, i.e. selecting whole sentences verbatim from the input document(s), evaluating random samples of sentences (over multiple trials) provides a robust estimate of the lower-bounds on the expected effectiveness of extractive summarisation algorithms. Any effective summarisation algorithm, that can successfully identify important and salient sentences, should out-perform a randomly generated summary in terms of summarisation effectiveness. Moreover, within the natural language processing pipeline we find many confounding variables, such as: sentence segmentation; tokenisation; stopword removal (with numerous stopword lists possible); and stemming or lemmatisation (again, with numerous algorithms possible). Given such variation in the experimental setup of summarisation evaluations, reporting random summary evaluation scores alongside the scores for particular summarisation algorithm(s) being evaluated serves to control for such experimental variability. Hence, as the random extraction of summary sentences provides a useful worst-case bounds on expected performance, given a particular experimental setup, we argue that it should always be reported as a baseline for the task of extractive summarisation.

Further, we argue that the random baseline enables a useful identification of "easy" and "hard" topics, where high evaluation scores for the random baseline indicate an "easy" topic, and low evaluation scores indicate a "hard" topic. Specifically, we can analyse the effectiveness of the random baseline on a per-topic basis, as demonstrated in Figure 4.1. Given random baseline evaluation scores (shown on the $y$-axis) for the 50 topics of a hypothetical summarisation dataset (shown on the $x$-axis), we can form two hypotheses regarding the nature of the document sets with respect to how difficult each topic is to summarise. A first hypothesis is that all topics may be equal in terms of difficulty, illustrated as a horizontal line ("Random $x$") on Figure 4.1. A second hypothesis is that the topics may exhibit observable variability in difficulty, illustrated as a diagonal line ("Random $y$") on Figure 4.1. We test these hypotheses in Section 4.3. Analysing the random baseline in this manner can give an indication of which particular topics should be the focus of failure analysis, for example. Additionally, we

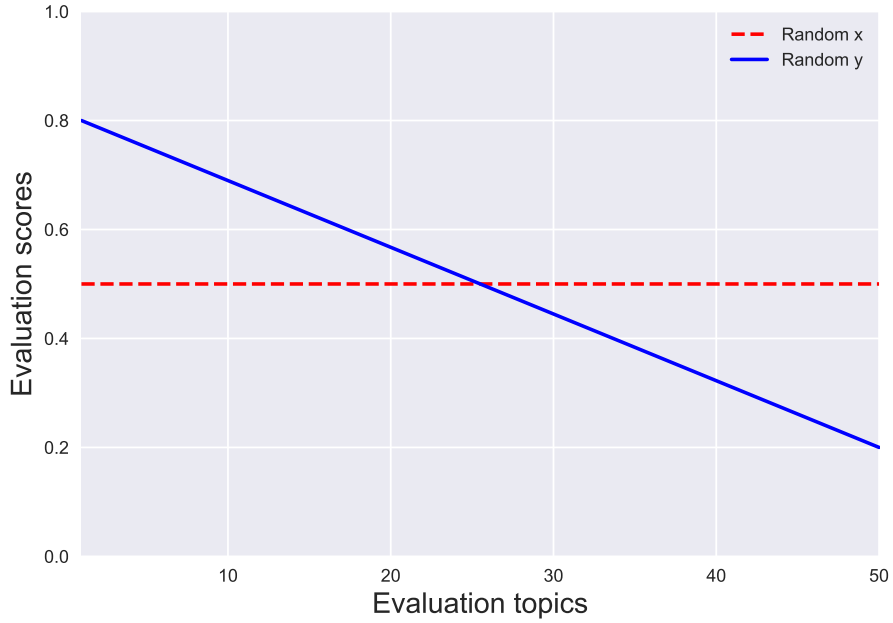



**Figure 4.1:** Visualisation of two possible scenarios for the per-topic distribution of summarisation evaluation scores for the random baseline. The "Random *x*" system illustrates the case where all topics are equal in terms of difficulty. The "Random *y*" system illustrates the case where there are observable "easy" and "hard" topics.

can also analyse how each proposed summarisation algorithm performs on a per-topic basis with respect to a random summary. This can demonstrate whether any improvements in summarisation effectiveness are gained over the more challenging topics.

### 4.1.2 Lead-based Newswire Summarisation Baselines

For the case of single document newswire summarisation, a lead summary is constructed by extracting sentences verbatim from the leading (i.e. first) sentences from a news article, given a desired summary length (e.g. 5 sentences). The lead summarisation baseline is known to perform effectively within the newswire domain (Nenkova, 2005). This is due to journalistic convention[1] of authoring news articles with a high-density of salient information at the beginning of the article. Figure 4.2 shows example lead sentences of newswire documents, published by the Associated Press (AP) from the period between October 16$^{th}$ and November 24$^{th}$ 1998. The 10 documents are denoted by their document identifier, and are drawn from topic "d30001t" of the DUC 2004 dataset. From Figure 4.2, we can observe that the lead sentence often succinctly states important information about the news article. Given the

[1] training.npr.org/digital/leads-are-hard-heres-how-to-write-a-good-one

task is to summarise these 10 documents, it is evident from Figure 4.2 that constructing a summary by extracting (verbatim) such leading sentences may often produce a reasonably effective summary of the document set. Indeed, at the Document Understanding Conference (DUC) summarisation evaluation campaigns (Over et al., 2007), which focused on the summarisation of articles from the newswire domain, lead-based summarisation approaches were used extensively as baselines (e.g. DUC 2001[1], DUC 2002[2], DUC 2003[3], and DUC 2004[4]).

For the case of multi-document newswire summarisation, there are a number of possible variations of the single document methodology (described above) that can be used to derive a lead-based summary given a set of documents to summarise. In Figure 4.3, we illustrate 3 possible lead-based baseline variations. Figure 4.3 presents a hypothetical document set, consisting of 5 documents each containing 3 sentences, and the task is to extract a lead-based summary of 3 sentences in length. For the DUC multi-document newswire summarisation tasks, two methods for deriving lead-based summaries were used as official baselines. The first method is to extract the leading sentences from the most recent document, where the documents are ordered by publication date. This is shown as "Lead 1" in Figure 4.3, where the 3 red sentences (i.e. the extracted summary) come from document 5. The second method is to extract the lead sentence from the first document, then extract the lead sentence from the second document, continuing until the desired summary length is reached. This is shown as "Lead 2" in Figure 4.3, where the extracted summary sentences (shown in red) come from documents 1, 2 and 3 in turn. However, we argue that such multi-document lead-based summarisation baselines can be improved upon, in terms of their summarisation effectiveness.

In particular, we form the hypothesis that applying anti-redundancy filtering to the interleaved DUC lead baseline (i.e. "Lead 2") will result in improved summarisation effectiveness. We describe such anti-redundancy filtering components in Section 4.2.2. We illustrate our proposal in Figure 4.3 as "Lead 3". Similar to "Lead 2", we aim to select lead sentences from each document in turn. The difference is that an anti-redundancy filtering component will reject (i.e. skip) some sentences that exhibit high textual similarity with the sentences that were previously selected for inclusion into the summary. As shown in Figure 4.3, this

[1] duc.nist.gov/past_duc/duc2001/data/eval/baseline_definitions
[2] duc.nist.gov/duc2002/baselines.html
[3] duc.nist.gov/duc2003/baseline_definitions
[4] duc.nist.gov/duc2004/baseline_definitions

1. **APW19981016.0240** – Cambodian leader Hun Sen on Friday rejected opposition parties' demands for talks outside the country, accusing them of trying to "internationalize" the political crisis.
2. **APW19981022.0269** – King Norodom Sihanouk has declined requests to chair a summit of Cambodia's top political leaders, saying the meeting would not bring any progress in deadlocked negotiations to form a government.
3. **APW19981026.0220** – Cambodia's two-party opposition asked the Asian Development Bank Monday to stop providing loans to the incumbent government, which it calls illegal.
4. **APW19981027.0491** – Cambodia's ruling party responded Tuesday to criticisms of its leader in the U.S. Congress with a lengthy defense of strongman Hun Sen's human rights record.
5. **APW19981031.0167** – Cambodia's leading opposition party ruled out sharing the presidency of Parliament with its arch foe Saturday, insisting it alone must occupy the top position in the legislative body.
6. **APW19981113.0251** – Cambodia's bickering political parties broke a three-month deadlock Friday and agreed to a coalition government leaving strongman Hun Sen as sole prime minister, King Norodom Sihanouk announced.
7. **APW19981116.0205** – Cambodian politicians expressed hope Monday that a new partnership between the parties of strongman Hun Sen and his rival, Prince Norodom Ranariddh, in a coalition government would not end in more violence.
8. **APW19981118.0276** – Cambodian leader Hun Sen has guaranteed the safety and political freedom of all politicians, trying to ease the fears of his rivals that they will be arrested or killed if they return to the country.
9. **APW19981120.0274** – Worried that party colleagues still face arrest for their politics, opposition leader Sam Rainsy sought further clarification Friday of security guarantees promised by strongman Hun Sen.
10. **APW19981124.0267** – King Norodom Sihanouk on Tuesday praised agreements by Cambodia's top two political parties previously bitter rivals to form a coalition government led by strongman Hun Sen.

**Figure 4.2:** Lead sentences from the 10 newswire documents of DUC 2004 topic "d30001t".

may result in the "Lead 3" lead-based summary skipping some documents (e.g. document 2), and even progressing onto the 2$^{nd}$ sentence (e.g. document 5). As illustrated in Figure 4.2, skipping some lead sentences due to their redundant nature may permit a summary selection to include such sentences as "APW19981113.0251" (the 6$^{th}$ sentence). This sentence clearly provides important and salient information regarding the event being discussed in the set of news articles. However, this sentence would not be selected under "Lead 1" or "Lead 2".

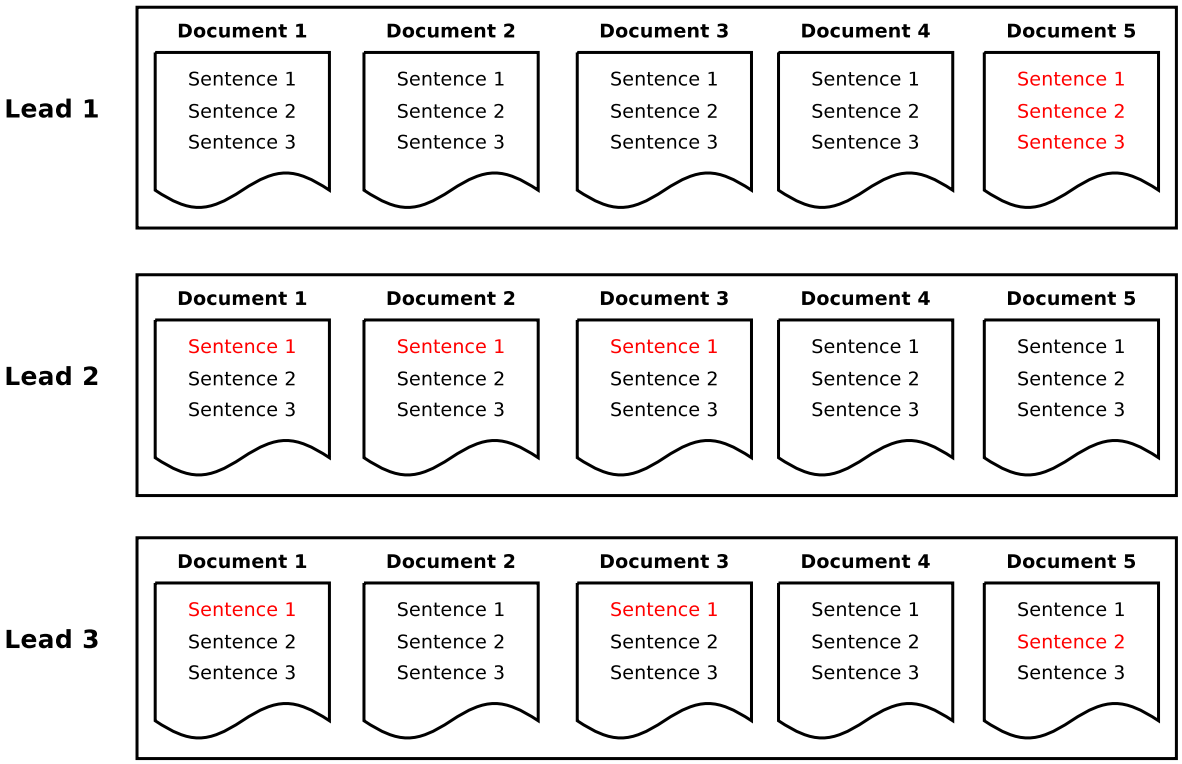



**Figure 4.3:** Three methods for extracting a lead-based multi-document summarisation baseline. Under each approach, sentences annotated in red are selected for the summary (of length 3 sentences). In "Lead 1", lead sentences are extracted from the most recent document. In "Lead 2", the lead sentences are drawn from each document in turn, up to the desired summary length. In "Lead 3", some sentences are skipped due to anti-redundancy filtering (checking for textual similarity). We hypothesise that "Lead 3" is the most effective.

Further, we argue that the lead baseline can also facilitate an analysis of the document sets within a summarisation dataset to identify "easy" and "hard" summarisation topics. Similarly to the random baseline, described in Section 4.1.1, high evaluation scores for the lead baseline indicate an "easy" topic, whereas low evaluation scores indicate a "hard" topic. Specifically, an "easy" topic under the lead baseline is where simply extracting the leading sentences from news articles is sufficient to generate an effective summary. A "hard" topic under the lead baseline suggests that summarisation algorithms must examine additional sentences to identify important and salient content. Referring back to Figure 4.1, under the lead baseline we form similar hypotheses regarding the nature of topics with respect to how difficult they are to summarise. A first hypothesis is that all topics will exhibit similar scores under the lead baseline. A second hypothesis is that there will be observable variability in effectiveness scores over topics under the lead baseline. We test these hypotheses in Section 4.3.

## 4.2 Unsupervised Summarisation Algorithms

In this section, we discuss the two main components of unsupervised summarisation algorithms. Unsupervised summarisation algorithms can be decomposed into "rank" and "filter" components. The ranking component scores sentences by some measure of preference for inclusion into the summary. The filtering component rejects sentences based on some measure of similarity with sentences previously selected for inclusion into the summary. Specifically, we explore various different implementations of methods for: (1) scoring and ranking candidate summary sentences by their summary worthiness; and (2) filtering the ranked list of sentences to select a subset of non-redundant summary sentences.

### 4.2.1 Summary Sentence Scoring Functions

In general, the main task of an extractive summarisation algorithm is to assign scores to sentences (Nenkova and McKeown, 2011). A score for a sentence is a measure of importance, saliance, and more abstractly, summary worthiness. The unique characteristic of each particular summarisation algorithm is the specific criteria used to compute sentence scores. Scoring sentences produces a ranked list of sentences, where the highest-ranking sentences are most suitable for inclusion into the summary. Sentences are selected from this ranking, based on various anti-redundancy filtering components, which are described in Section 4.2.2.

For each of the standard unsupervised baseline algorithms, as enumerated in Section 2.4.1, a number of different algorithm implementation design choices present themselves. In this section, we discuss and explore variations in techniques that can be used to implement the summarisation algorithms in practice. Such variations are evaluated in Section 4.3, to identify strong baselines for future experiments, and to identify discriminative features for training supervised summarisation models. Specifically, we discuss the FreqSum (Nenkova et al., 2006), TsSum (Conroy et al., 2006), Centroid (Radev et al., 2004), LexRank (Erkan and Radev, 2004), and GreedyKL (Haghighi and Vanderwende, 2009) summarisation algorithms, i.e. the set of 5 standard unsupervised summarisation baselines of Hong et al. (2014).

**FreqSum** (Nenkova et al., 2006) – Given a set of input sentences, $S = (s_1, s_2, \ldots, s_n)$, where each sentence contains a number of terms, $s_i = (t_1, t_2, \ldots, t_n)$, a probability is assigned to each term, $t_i \in S$. The probability of each term, $p(t_i) = \frac{n}{N}$, where $n$ is the frequency of $t_i \in S$,

and $N$ is the total number of terms in $S$, is computed such that $\sum_{t_i \in S} p(t_i) = 1$, i.e. a unigram language model, $P(t_1, t_2, \ldots, t_n)$. Candidate summary sentences, $s_i$, are scored by summing the probabilities of the terms that occur in a given sentence, as defined in Equation 4.1:

$$\text{FreqSum}(s_i) = \sum_{t_i \in s_i} p(t_i) \tag{4.1}$$

As evident in Equation 4.1, FreqSum will exhibit a bias towards longer sentences. This is because there is a summation over all terms in each sentence, i.e. longer sentences will obtain higher scores simply by containing more terms. An alternative implementation, discussed by Nenkova et al. (2006), would be to normalise for sentence length as defined in Equation 4.2:

$$\text{FreqSum}(s_i) = \frac{\sum_{t_i \in s_i} p(t_i)}{|s_i|} \tag{4.2}$$

**TsSum** (Conroy et al., 2006) – Given the frequencies of terms computed over a large background corpus, topic words (Lin and Hovy, 2000) are specific terms that occur more often in a set of sentences (i.e. a document) than in the large background corpus. The log-likelihood ratio (LLR) test[1] is applied, comparing the frequency of terms over all the input sentences vs. a background corpus. Given a term's LLR test value, $\lambda$, various threshold parameters can be used to determine topic words from non-topic words. Specifically, as the LLR $\lambda$ follows a $\chi^2$ distribution[2], confidence levels of 5% ($p < 0.05$), 1% ($p < 0.01$), 0.1% ($p < 0.001$), and 0.01% ($p < 0.0001$), provide topic words cutoff parameters of 3.84, 6.63, 10.83, 15.13, respectively. Conroy et al. (2006) used a topic words cutoff parameter value of 10, for example, and words with an LLR test value $> 10$ are considered topic words, i.e. words that discriminately describe the topic of a document. A further design choice of this algorithm is the corpus from which to derive background term frequencies. For example, background term frequencies can be computed over a Wikipedia corpus, or a domain-specific newswire corpus. The TsSum algorithm scores individual sentences, $s_i$, based on the number of topic words in the sentence, $\text{tw} \in s_i$. Specifically, the score for a candidate summary sentence, $s_i$, is the ratio of unique topic words to all unique words, as defined in Equation 4.3:

$$\text{TsSum}(s_i) = \frac{|\text{tw} \in s_i|}{|\text{words} \in s_i|} \tag{4.3}$$

[1] en.wikipedia.org/wiki/Likelihood-ratio_test

[2] ucrel.lancs.ac.uk/llwizard.html

**Centroid** (Radev et al., 2004) – Given a set of input sentences, $S = (s_1, s_2, \ldots, s_n)$, where each sentence contains a number of terms, $s_i = (t_1, t_2, \ldots, t_n)$, term vectors, $v_i = (t_1, t_2, \ldots, t_n)$, are used to represent the sentences. A centroid term vector, $C = (t_1, t_2, \ldots, t_n)$, is computed from the set of sentence term vectors: $C = \frac{(v_1 + v_2 + \ldots + v_n)}{|S|}$. The Centroid algorithm scores candidate summary sentences, $s_i$, by computing the cosine similarity[1] of the sentence term vector, $v_i$, to the centroid term vector, $C$. Cosine similarity is taken as the dot product of two vectors over the product of their Euclidean lengths[2], as defined in Equation 4.4:

$$\text{Centroid}(s_i) = \text{CosSim}(C, v_i) = \frac{C \cdot v_i}{\|C\|_2 \|v_i\|_2} = \frac{\sum_{j=1}^{n} C_j v_{i_j}}{\sqrt{\sum_{j=1}^{n} C_j^2} \sqrt{\sum_{j=1}^{n} v_{i_j}^2}} \tag{4.4}$$

The key design choice of the Centroid algorithm is what term vector weighting scheme is chosen. As stated above, given a sentence, $s_i = (t_1, t_2, \ldots, t_n)$, a term vector for that sentence must be defined, $v_i = (t_1, t_2, \ldots, t_n)$. The values of this vector could be, for example, binary term frequency (i.e. 0 or 1), raw term frequency, logarithmic term frequency, or the product of term frequency and inverse document frequency (i.e. tf.idf). Indeed, there are many such term weighting schemes described in the literature (Croft et al., 2010; Büttcher et al., 2010). We investigate different weighting schemes, denoted *Tf*, *Hy*, *Rt*, and *HyRt* in later experiments. *Tf* is tf.idf, specifically $\log(\text{tf}) * \log(\text{idf})$, where tf is the frequency of a term in a sentence, and $\text{idf} = \frac{N}{N_t}$, the total number of sentences divided by the number of sentences containing term t. *Hy* is a tf $*$ idf variant, where the tf component is computed over all of the input sentences combined, instead of individual sentences. *Rt* and *HyRt* are tf $*$ idf variants where we do not use log smoothing, i.e. raw *tf*.

**LexRank** (Erkan and Radev, 2004) – Given the set of input sentences, $S = (s_1, s_2, \ldots, s_n)$, a graph, $G = (V, E)$, is computed where the sentences are represented in the graph as vertices, $V = (v_1, v_2, \ldots, v_n)$. Undirected weighted edges, $E = (e_1, e_2, \ldots, e_n)$, represent the cosine similarity (c.f. Equation 4.4) between pairs of sentences, $(v_i, v_j)$. Using this graph, sentences (i.e. nodes in the graph) are scored by computing graph-based measures of vertex importance, such as degree centrality or PageRank (Page et al., 1999).

[1] en.wikipedia.org/wiki/Cosine_similarity

[2] en.wikipedia.org/wiki/Euclidean_vector#Length

In the first variation of LexRank, a threshold parameter is applied such that only pairs of sentences that exhibit a cosine similarity above the given threshold are linked in the graph. In particular, sentence pairs exhibiting a cosine similarity below the threshold are not linked, resulting in a graph that is not completely connected. After the graph edges have been established, the edge weights between vertices are not utilised further, i.e. a binary adjacency matrix is formed. In our experiments, we vary the edge linking threshold parameter, $t = [0..1]$, in steps of 0.05. The LexRank algorithm scores a candidate summary sentence, $s_i$, by computing a score for the corresponding vertex, $v_i$. Specifically, sentences are scored using degree centrality, which is the number of edges incident on a vertex, as defined in Equation 4.5:

$$\text{LexRank}(s_i) = \text{Deg. Cent.}(v_i) \tag{4.5}$$

A second variation of LexRank uses the PageRank algorithm to score vertices in the graph. Under this variation, known as continuous LexRank, a threshold parameter is not applied over the graph, and the strength of connection between vertices is directly utilised. Specifically, given a completely connected graph, the edge weights in the graph are used to derive the transition probabilities within the PageRank algorithm. In particular, using continuous LexRank, candidate summary sentences are scored as defined in Equation 4.6:

$$\text{Cont. LexRank}(s_i) = \text{PageRank}(v_i) \tag{4.6}$$

Similarly to Centroid, when using LexRank a term vector weighting scheme is required to represent sentences as vectors. In later experiments we again use the *Tf*, *Hy*, *Rt*, and *HyRt* term vector weighting schemes (described above).

**GreedyKL** (Haghighi and Vanderwende, 2009) – Given a set of input sentences, $S = (s_1, s_2, \ldots, s_n)$, a probability distribution, $P(t_1, t_2, \ldots, t_n)$, is computed over all terms, $t_i \in S$. Further, for each candidate summary sentence, $s_i = (t_1, t_2, \ldots, t_n)$, a probability distribution, $Q_s(t_1, t_2, \ldots, t_n)$, is computed over the terms, $t_i \in s_i \cup E$, where $E \subset S$ (i.e. the summary). Before any sentences have been selected, the extractive summary, $E$, is empty. Iteratively, at each sentence selection step, a candidate summary sentence, $s_i$, is greedily selected for inclusion into the summary $E = (s_1, s_2, \ldots, s_n)$. GreedyKL selects a sentence that minimises the Kullback–Leibler divergence (Kullback and Leibler, 1951), $D_{KL}$, between the probability distribution over all input sentences, $P$, and the probability distribution over the candidate

summary sentence $s_i$ and the current summary, $Q_s$. After a sentence, $s_i$, is selected for inclusion into the summary, $E$, the per-sentence $Q_s$ distributions are re-computed, as the summary text contains more terms (i.e. $Q_s$ represents $t_i \in s_i \cup E$). $D_{KL}$ is defined in Equation 4.7 as:

$$D_{KL}(P\|Q_s) = \sum_i P(i) \log_2 \frac{P(i)}{Q_s(i)} \tag{4.7}$$

An alternative implementation, instead of greedily minimising $D_{KL}(P\|Q_s)$, is to simply score each candidate summary sentence, $s_i$, as the Kullback–Leibler divergence from all input sentences, $S$, and then pass the ranked list through an anti-redundancy filtering component. As such, it is not required to update the probability distribution over the terms in the candidate summary sentence, $Q_s$. This alternative sentence scoring function is defined in Equation 4.8:

$$\text{GreedyKL}(s_i) = D_{KL}(S, s_i) \tag{4.8}$$

In the computation of Kullback–Leibler divergence, $D_{KL}(P\|Q)$, a problem arises when the language models $P$ and $Q$ do not share the same term vocabulary. Given $P$ is a distribution over all input sentences, $S$, and $Q$ is a distribution over an individual sentence, $s_i$, this means that there will be many terms in $P$ that are not in $Q$. When summing over $P$, and taking the $\log_2 \frac{P(i)}{Q(i)}$, if $Q_i$ does not exist for $P_i$, the zero probability for $Q_i$ results in an undefined division by zero. To assign a non-zero probability to terms not occurring in $Q$ that do occur in $P$, it is recommended to smooth the maximum likelihood estimator (MLE) of the $Q$ distribution with a background language model (Zhai and Lafferty, 2004). The choice of smoothing technique is a design choice of this algorithm. In our experiments, we use Jelinek–Mercer smoothing (Jelinek and Mercer, 1980), which introduces a smoothing parameter, $\lambda = [0, 1]$. Further, we use the $P$ distribution as the background language model. Specifically, given the probabilities of two terms, $P_i$ and $Q_i$, smoothing of $Q_i$ with $P_i$ is defined as: $Q_i = (1-\lambda)P_i + \lambda Q_i$.

### 4.2.2 Summary Sentence Anti-redundancy Filtering

As discussed in Section 2.4, summarisation algorithms score sentences, producing ranked lists of candidate summary sentences. Rankings of sentences are passed through an anti-redundancy filtering component. The anti-redundancy filtering component attempts to reduce the probability that the summary will contain repeated information. Each anti-redundancy filtering component takes as input a list of sentences, previously ranked by a summary sentence scoring function. The first, highest-scoring, sentence is selected. Then, iterating down

the list, the next highest-scoring sentence is selected based on the condition that it satisfies a dis-similarity threshold. We experiment with the following anti-redundancy components: "Top-$k$", "CosineSimilarity", "NewWordCount", "NewBigrams", and "NewTopicWords".

**Top-$k$** – The Top-$k$ method serves as a baseline for anti-redundancy filtering components. Given a ranked list of candidate summary sentences, the $k$ highest-ranked sentences are selected for inclusion into the summary. The Top-$k$ method does not consider the redundancy among the sentences selected for the summary. We include a method that does not perform anti-redundancy filtering so we can measure the effectiveness of anti-redundancy methods.

**CosineSimilarity** – The cosine similarity anti-redundancy component is a commonly used technique to reduce redundant information in the summary text (Hong et al., 2014). The thresholding condition states that the next sentence to be included in the summary must not exhibit a specified degree of cosine similarity with all of the sentences previously selected for inclusion into the summary. The specific degree of cosine similarity is a parameter of the filtering component. In our experiments, the value of the cosine similarity threshold ranges from $[0, 1]$ in steps of 0.05. As cosine similarity computations require a vector representation of the sentences, similarly to the Centroid and LexRank algorithms described in Section 4.2.1, we experiment with the *Tf*, *Hy*, *Rt*, and *HyRt* term vector weighting schemes.

**NewWordCount** – Proposed by Allan et al. (2003), the new word count anti-redundancy filtering component selects sentences based on minimising term-overlap between the summary text and candidate summary sentences. Specifically, the thresholding condition states that the next sentence to be added to the summary text (from the ranked list of candidate summary sentences) must contribute $n$ new words to the summary text vocabulary. In our experiments, the value of $n$, the new word count parameter, ranges from $[1, 20]$, in steps of 1.

**NewBigrams** – We propose an anti-redundancy filtering component that is a direct extension of *NewWordCount*. In place of unigrams, bigrams are the unit of measurement. Specifically, the thresholding condition states that the next sentence to be added to the summary text must contribute $n$ new bi-grams to the summary text vocabulary. In our experiments, the value of $n$, the new bi-grams parameter, ranges from $[1, 20]$, in steps of 1.

**NewTopicWords** – We propose a further anti-redundancy filtering component that is a direct extension of *NewWordCount*. In place of unigrams, topic words (Lin and Hovy, 2000) are used as the unit of measurement to assess the textual similarity between the summary text

and candidate summary sentences (drawn from the ranked list). Specifically, the thresholding condition states that the next sentence to be added to the summary text must contribute $n$ new topic words to the summary text vocabulary. In our experiments, the value of $n$, the new topic words parameter, ranges from $[1,20]$, in steps of 1.

We provide a summary of the variations and parameters of summary sentence scoring functions (Section 4.2.1) and anti-redundancy components (Section 4.2.2) in Figure 4.4.

## 4.3 Evaluation

In this section, we conduct an experimental evaluation of unsupervised multi-document newswire summarisation baselines. We begin by stating our research questions, then describe our experimental setup. Results are provided over the DUC 2004 Task 2 dataset, for the task of generic extractive multi-document newswire summarisation.

### 4.3.1 Research Questions

In our Thesis Statement (Section 1.2), we formed Hypothesis 2:

> We hypothesise that the effectiveness of standard multi-document newswire summarisation algorithms can be improved by varying algorithm design choices.

To validate Hypothesis 2, we address the following research questions:

**Research Question 4.1.** What is the minimum expected effectiveness under the extractive summarisation paradigm for the DUC 2004 Task 2 dataset?

**Research Question 4.2.** Can the effectiveness of the DUC lead-based baselines be improved by applying anti-redundancy filtering to an interleaved selection of leading sentences?

**Research Question 4.3.** Can the effectiveness of standard multi-document newswire summarisation algorithms be improved by varying algorithm design choices?

We argue in our Thesis Statement (Section 1.2) that strong baselines are required for experimental validity, and that standard multi-document newswire summarisation algorithms can provide discriminative features for supervised summarisation models. Research questions 4.1, 4.2, and 4.3 address the argument regarding strong baselines, where we establish a

**Summary Sentence Scoring Functions**

- FreqSum
  - Without length normalisation.
  - Normalise by sentence length.
- TsSum
  - LLR $\lambda$ cutoff parameter: $[3.84, 6.63, 10.83, 15.13]$.
  - Genre/domain of background corpus: Wikipedia; Newswire.
- Centroid
  - Term vector weighting scheme: Tf; Hy; Rt; and HyRt.
- LexRank
  - Graph-based vertex importance: Degree centrality; Pagerank.
  - Graph edge threshold parameter: $t = [0..1]$ in steps of 0.05.
  - Term vector weighting scheme: Tf; Hy; Rt; and HyRt.
- GreedyKL
  - Greedy sentence selection.
  - Ranking sentences by KL divergence.
  - Smoothing technique: Jelinek–Mercer (not varied).
  - Smoothing parameter: $\lambda = [0..1]$ in steps of 0.1.

**Anti-redundancy Components**

- Top-$k$
  - Number of sentences to select: $k$.
- CosineSimilarity
  - Term vector weighting scheme: Tf; Hy; Rt; and HyRt.
  - Cosine similarity threshold parameter: cos $= [0..1]$ in steps of 0.05.
- NewWordCount
  - New word count threshold parameter: nwc $= [1..20]$ in steps of 1.
- NewBigrams
  - New bi-grams threshold parameter: nbg $= [1..20]$ in steps of 1.
- NewTopicWords
  - New topic words threshold parameter: ntw $= [1..20]$ in steps of 1.
  - LLR $\lambda$ cutoff parameter: 3.84; 6.63; 10.83; 15.13.
  - Genre/domain of background corpus: Wikipedia; Newswire.

**Figure 4.4:** Variations and parameters of summary sentence scoring functions and anti-redundancy components. For each summarisation algorithm, and each anti-redundancy filtering component, we explore the various techniques used to implement the algorithms in practice. The different variations are evaluated in Section 4.3.

lower-bounds on extractive summarisation effectiveness on DUC 2004, evaluate our proposed improvement for the lead-based baseline over newswire text, and evaluate various implementations of standard summarisation algorithms.

### 4.3.2 Experimental Setup

In the following summarisation experiments, we use newswire documents from the DUC 2004 Task 2 dataset, evaluating for the task of generic extractive multi-document newswire summarisation. The DUC 2004 Task 2 dataset contains 50 topics, with 10 newswire articles per topic, and 4 gold-standard reference summaries per topic. Newswire text is extracted from the <LEADPARA>, <LP>, and <TEXT> document fields. The Stanford CoreNLP toolkit (Manning et al., 2014) is used to split the newswire text into sentences, and tokenise words. Individual tokens are then subjected to the following text processing steps: Unicode normalisation (NFD[1]), case folding, splitting of compound words, removal of punctuation, Porter stemming, and stopword removal (removing the 50 most common English words[2]). To summarise multiple documents for a topic, we combine all sentences from the input documents into a single virtual document. Sentences from each document are interleaved one-by-one in docid order, and this virtual document is provided to the summarisation algorithms.

To evaluate summary texts, we use the ROUGE (Lin, 2004) automatic evaluation metric. Following best practice (Hong et al., 2014), the summaries under evaluation are subject to stemming, stopwords are retained, and we report ROUGE-1, ROUGE-2 and ROUGE-4 recall. Further, for all experiments, summary lengths are truncated to 100 words. For summarisation algorithms with parameters, we learn the parameter settings via a five-fold cross validation procedure, optimising for the ROUGE-2 metric. Statistical significance in ROUGE results is reported using the paired Student's t-test, 95% confidence level. ROUGE results for various summarisation systems are obtained using SumRepo (Hong et al., 2014)[3], which provides the plain-text produced by 5 standard baselines, and 7 state-of-the-art systems, over DUC 2004. Using this resource, we compute ROUGE results over DUC 2004 for the algorithms available within SumRepo, obtaining reference results for use in our experiments.

[1] docs.oracle.com/javase/8/docs/api/java/text/Normalizer.html
[2] en.wikipedia.org/wiki/Most_common_words_in_English
[3] www.seas.upenn.edu/~nlp/corpora/sumrepo.html

**Table 4.1:** ROUGE scores, over DUC 2004 Task 2, for the random baseline and five standard baselines.

| | **R-1** | **R-2** | **R-4** |
|---|---|---|---|
| *Random* | 30.27 | 4.33 | 0.35 |
| **Baselines** | **R-1** | **R-2** | **R-4** |
| *LexRank* | 36.00 | 7.51 | 0.83 |
| *Centroid* | 36.42 | 7.98 | 1.20 |
| *FreqSum* | 35.31 | 8.12 | 1.00 |
| *TsSum* | 35.93 | 8.16 | 1.03 |
| *Greedy–KL* | 38.03 | 8.56 | 1.27 |

### 4.3.3 Experimental Results

#### Research Question 4.1

We begin with research question 4.1, where we seek to establish the minimum expected summarisation effectiveness within the extractive summarisation paradigm, over the DUC 2004 Task 2 dataset. To establish such a lower-bounds, in our experiments we generate 100 random summaries per topic, evaluate the 100 per-topic random samples, then take the mean over the samples as the score for that topic. The per-topic scores are then averaged over all topics to arrive at a final summarisation evaluation score for the random baseline. Table 4.1 presents ROUGE results for the random baseline. Further, for reference purposes, Table 4.1 also provides ROUGE results for the five standard baselines from SumRepo.

In answer to research question 4.1, from Table 4.1, we observe that the random baseline exhibits a ROUGE-1 recall score of 30.27, a ROUGE-2 recall score of 4.33, and a ROUGE-4 recall score of 0.35. As argued in Section 4.1.1, the random baseline provides an accurate estimate of the lower-bounds on the expected effectiveness that should be achieved in the extractive paradigm. If an extractive summarisation algorithm can successfully identify salient sentences, it should outperform a randomly selected summary. Indeed, from Table 4.1, we observe that all of the standard baselines are more effective than the random baseline.

#### Research Question 4.2

We now address research question 4.2, where we investigate if the effectiveness of the lead-based newswire summarisation baselines, as used at the DUC summarisation evaluations,

**Table 4.2:** ROUGE scores, over the DUC 2004 Task 2 dataset, for random and lead, the lead baseline augmented with various anti-redundancy components, and the five standard baselines from SumRepo.

| **Lead (DUC)** | R-1 | **R-2** | R-4 |
|---|---|---|---|
| *Lead (recent-doc)* | 31.46 | 6.13 | 0.62 |
| *Lead (interleaved)* | 34.23† | **7.66**† | 1.18† |
| **Lead (anti-redundancy)** | R-1 | **R-2** | R-4 |
| *CosineSimilarityRt* | 35.67‡ | 7.91 | 1.20 |
| *CosineSimilarityTf* | 36.02‡ | 7.97 | 1.20 |
| *NewWordCount* | 35.54‡ | 8.02 | 1.22 |
| *CosineSimilarityHyRt* | 35.91‡ | 8.08‡ | 1.24 |
| *NewBigrams* | 36.05‡ | 8.11 | 1.18 |
| *CosineSimilarityHy* | 36.38‡ | **8.29**‡ | 1.29 |
| **Baselines (*SumRepo*)** | R-1 | **R-2** | R-4 |
| *LexRank* | 36.00 | 7.51 | 0.83 |
| *Centroid* | 36.42 | 7.98 | 1.20 |
| *FreqSum* | 35.31 | 8.12 | 1.00 |
| *TsSum* | 35.93 | 8.16 | 1.03 |
| *Greedy–KL* | 38.03✔ | **8.56** | 1.27 |

can be improved by applying anti-redundancy filtering. The lead baseline is reported to be particularly effective for the task of newswire summarisation (Nenkova, 2005). This is due to the journalistic convention of authoring news articles where the first sentence(s) are usually very informative. We investigate the method used to derive the lead baseline, and further, the results of augmenting the lead baseline with different anti-redundancy components.

Table 4.2 presents ROUGE results for two variants of the lead baseline used at DUC (recent-doc and interleaved), then the interleaved lead baseline passed through various anti-redundancy components. The lead baselines evaluated correspond to the three different lead-based baselines shown in Figure 4.3. Our hypothesis is that the anti-redundancy filtered interleaved lead baseline is the most effective lead-based newswire summarisation baseline. Further, for reference purposes, Table 4.2 also provides ROUGE results for five standard newswire summarisation baselines computed using SumRepo.

From Table 4.2, we first compare the two DUC lead-based baselines. We observe a statistically significant improvement in ROUGE results (paired Student's t-test, 95% confidence level), as shown using the "†" symbol, for the interleaved lead baseline over the recent-doc lead baseline. This significant increase in ROUGE effectiveness is observed across ROUGE-1, ROUGE-2 and ROUGE-4 recall. Specifically, the recent-doc lead baseline exhibits ROUGE-1, ROUGE-2, and ROUGE-4 recall scores of 31.46, 6.13, and 0.62, whereas the interleaved lead baseline exhibits ROUGE-1, ROUGE-2, and ROUGE-4 recall scores of 34.23, 7.66, and 1.18. From this, we conclude that using multiple lead sentences, from multiple documents, to construct a multi-document lead-based baseline, is more effective than simply using the first $n$ sentences from the most recent document.

Next, we examine the results obtained by augmenting the interleaved lead baseline with different anti-redundancy filtering components. From Table 4.2, we observe several cases where the interleaved lead baseline, when passed through an anti-redundancy component, achieves ROUGE effectiveness scores that exhibit a significant improvement over the non-redundancy filtered interleaved lead baseline. Such cases are indicated in Table 4.2 using the "‡" symbol, which indicates statistically significant improvements as measured with the paired Student's t-test, with a 95% confidence level. In particular, applying anti-redundancy filtering to the interleaved lead baseline results in significant improvements in ROUGE-1 scores for each anti-redundancy component investigated, and significant improvements in ROUGE-2 scores for the CosineSimilarityHyRt and CosineSimilarityHy variations.

Further, from Table 4.2, we observe that the five standard baselines, FreqSum, TsSum, Centroid, LexRank and GreedyKL, do not exhibit significant differences in ROUGE-2 scores compared with the interleaved lead baseline when passed through the CosineSimilarityHy anti-redundancy component. Indeed, only GreedyKL exhibits a ROUGE-1 score (indicated using a "✔" symbol in Table 4.2) that is significantly more effective than the interleaved lead baseline passed through the CosineSimilarityHy anti-redundancy component.

From the observations in Table 4.2, in answer to research question 4.2, we conclude that the lead-based newswire summarisation baselines, as used at the DUC summarisation evaluation campaigns, can be improved by applying anti-redundancy filtering components to the interleaved lead baseline.

**Research Question 4.3**

We now address research question 4.3, where the various techniques that can be used to implement baseline newswire summarisation algorithms are evaluated. Specifically, given the variations of summary sentence scoring functions, described in Section 4.2.1, and variations in summary sentence anti-redundancy components, described in Section 4.2.2, we evaluate the different algorithm design choices to identify strong baselines for our later experiments. We explore the re-implementation of the five standard summarisation baselines from SumRepo, namely: FreqSum, TsSum, Centroid, LexRank and GreedyKL (Hong et al., 2014). Our hypothesis is that the effectiveness of the baselines can be improved, via thoroughly exploring algorithm design choices, when compared to the reference implementations from SumRepo.

Table 4.3 provides ROUGE results over DUC 2004 Task 2 for the standard baselines from SumRepo, and the corresponding re-implementation of each algorithm. For the re-implementations, we note the particular variations of sentence scoring methods and anti-redundancy components. Further, for reference purposes, Table 4.4 provides ROUGE results for the state-of-the-art summarisation systems from SumRepo (most of which are supervised). In Table 4.3, a "✔" indicates a statistically significant improvement over the reference implementation for a given re-implementation. The "†" symbol is used to indicate that there is no statistically significant difference between a given re-implementation and ICSISumm, a state-of-the-art summarisation system (shown in Table 4.4). Statistical significance is based on the paired Student's t-test, with a 95% confidence level.

From Table 4.3, we first observe that the ROUGE results for the re-implementations of each of the five standard baselines always exhibit numerically higher effectiveness scores. Numerically higher scores are observed for all re-implementations, and over each of the three ROUGE metrics. Statistically significant increases in ROUGE scores, for re-implementations over reference implementations, are observed for all re-implementations under ROUGE-1, for the Centroid, LexRank, and GreedyKL algorithms under ROUGE-2, and for the TsSum and LexRank algorithms under ROUGE-4 – as indicated using the "✔" symbol in Table 4.3.

Further, from Table 4.3, we observe several cases where the baseline re-implementations exhibit state-of-the-art effectiveness scores. In particular, all re-implementations under the ROUGE-1 metric, the Centroid, LexRank, and GreedyKL algorithms under the ROUGE-2 metric, and all re-implementations except FreqSum under the ROUGE-4 metric, exhibit no statistically significant difference to ICSISumm (shown in Table 4.4) – as shown using "†".

**Table 4.3:** ROUGE results, over DUC 2004 Task 2, for reference implementations of standard multi-document newswire summarisation baselines from SumRepo, and re-implementations of baseline algorithms.

| **Reference implementation** | | | | **Corresponding re-implementation (c.f. Figure 4.4)** | | | | |
|---|---|---|---|---|---|---|---|---|
| **Algorithm** | **R-1** | **R-2** | **R-4** | **Sentence scoring** | **Anti-redundancy** | **R-1** | **R-2** | **R-4** |
| *FreqSum* | 35.31 | 8.12 | 1.00 | Length normalised | NewWordCount | 37.52✔† | 8.70 | 1.14 |
| *TsSum* | 35.93 | 8.16 | 1.03 | Wikipedia background | CosineSimilarity "Tf" | 37.54✔† | 8.87 | 1.39✔† |
| *Centroid* | 36.42 | 7.98 | 1.20 | "Hy" vectors | NewWordCount | 37.79✔† | 9.37✔† | **1.59**† |
| *LexRank* | 36.00 | 7.51 | 0.83 | Pagerank with priors | CosineSimilarity "Hy" | 38.05✔† | 9.34✔† | 1.44✔† |
| *GreedyKL* | **38.03** | **8.56** | **1.27** | Ranking by KLD | CosineSimilarity "Hy" | **38.44**† | **9.59**✔† | 1.56† |

**Table 4.4:** State-of-the-art systems (reference results).

| **State-of-the-art** | R-1 | **R-2** | R-4 |
|---|---|---|---|
| *CLASSY 04* | 37.71 | 9.02 | 1.53 |
| *CLASSY 11* | 37.21 | 9.21 | 1.48 |
| *Submodular* | 39.23 | 9.37 | 1.39 |
| *DPP* | 39.84 | 9.62 | 1.57 |
| *OCCAMS_V* | 38.50 | 9.75 | 1.33 |
| *RegSum* | 38.60 | 9.78 | 1.62 |
| *ICSISumm*† | 38.44 | **9.81** | 1.74 |

The improvements for the re-implementations (i.e. optimising the standard baselines and closing the gap to the state-of-the-art) are attributed to variations in algorithm design, discussed in detail in Section 4.2 and summarised in Figure 4.4. For example, the most effective standard baseline re-implementation (shown in bold in Table 4.3) is a variation of GreedyKL. Instead of greedily selecting summary sentences that minimise Kullback–Leibler divergence, our variation first scores sentences by their Kullback–Leibler divergence to all other sentences, then passes the ranked list to an anti-redundancy component. Further, varying the term vector weighting scheme, such as using hybrid tf.idf vectors ("Hy"), often leads to effectiveness improvements, as demonstrated by empirical observations in Table 4.3. Furthermore, it is common in the summarisation literature to apply a cosine similarity anti-redundancy component (Hong et al., 2014), however we observe that altering the choice of anti-redundancy component often leads to improvements in effectiveness.

From the results presented in Table 4.3, we can now answer research question 4.3. We conclude that it is possible to optimise the standard baselines, even to the point where they exhibit similar effectiveness to the state-of-the-art over the DUC 2004 Task 2 dataset.

## 4.4 Chapter Summary

In this chapter, we investigated unsupervised summarisation baselines. We provided experimental results to empirically validate Hypothesis 2 from our Thesis Statement (Section 1.2). We validated our claim that the effectiveness of standard multi-document newswire summarisation algorithms can be improved by varying algorithm design choices. By answering Research Question 4.1, we observed the lower-bounds on extractive summarisation effectiveness over the DUC 2004 dataset. By answering Research Question 4.2, we demonstrated the effectiveness of the DUC lead-based baselines be improved by applying anti-redundancy filtering. Such an improved lead baseline is competitive with the standard baselines. By answering Research Question 4.3, we demonstrated that the effectiveness of standard multi-document newswire summarisation algorithms be improved by varying algorithm design choices. Such improved baseline algorithms are competitive with the state-of-the-art baselines.

In conclusion, as the standard lead-based baseline can be improved significantly by using anti-redundancy filtering techniques, this improved lead-based baseline is more appropriate to use in empirical evaluations of summarisation systems. Further, as we have shown that our variations of the standard baselines are effective unsupervised summarisation algorithms, this indicates that such algorithms may be effective for use as features within supervised summarisation models – which we investigate in the next chapter.

# Chapter 5

# On the Effective Training of Supervised Summarisation Models

In this chapter, we address our third challenge, regarding the effective training of supervised machine learned summarisation models. For our later experiments, in Chapter 6 and 7, we investigate the effectiveness of using event-based entity-focused evidence to produce summaries of news-worthy events. Machine learning (Hastie et al., 2009; Witten et al., 2016) provides a principled methodology to evaluate the integration of entity-focused evidence into the multi-document newswire summarisation process, where such entity-focused evidence is expressed as additional features within supervised multi-document newswire summarisation models. However, before we can undertake such experiments, the problems inherent in operationalising a supervised machine learned summarisation framework must be addressed, specifically: labelling training data; defining summarisation features; and model selection.

Mitchell (1997) provides a formal definition of supervised learning: "A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$". Within the context of supervised extractive summarisation (Kupiec et al., 1995; Teufel and Moens, 1997; Aone et al., 1998), the aim is to learn to predict the summary worthiness of individual sentences. As such, we first require high-quality annotated training data, indicating the summary worthiness of candidate summary sentences (i.e. labels). Second, we require a vector-based numerical representation of natural language sentences (i.e. features). Third, we require a machine learning technique (i.e. learner) that is appropriate to the task.

In this chapter, we thoroughly investigate the problem of labelling training data, conducting experimental evaluations of different methods for obtaining such labelled training data. Further, we select and evaluate a specific set of baseline multi-document newswire summarisation algorithms from the literature (Hong et al., 2014), to use as features within supervised summarisation models. Such newswire summarisation features are augmented with entity-focused features in later experiments, to test our hypothesis that using entity evidence results in more effective summaries of events. Furthermore, we experiment with commonly used supervised regression techniques, as such models have been shown to be effective for the task of extractive multi-document newswire summarisation (Ouyang et al., 2011a).

As stated in Hypothesis 3 from our Thesis Statement (Section 1.2), we hypothesise that supervised machine learned summarisation models based on regression techniques, that exhibit state-of-the-art effectiveness, can be trained on discriminative features, derived from standard multi-document newswire summarisation algorithms, using automatically labelled training data, induced from gold-standard summary text(s).

This chapter is based on the following publication: Mackie et al. (2016).

## Chapter Outline

This chapter is organised as follows:

- Section 5.1 formally states the supervised summarisation problem, discussing the regression-based learning techniques that we use in our experiments in this thesis.
- Section 5.2 discusses the mechanics of obtaining training data for supervised summarisation, specifically from gold-standard summaries of text documents.
- Section 5.3 discusses various sentence scoring functions, that label a sentence with a summary worthiness score, with respect to one or more abstractive gold-standard summaries.
- Section 5.4 discusses methods for representing sentences as numerical vectors, using per-sentence scores from baseline multi-document summarisation algorithms as features.
- Section 5.5 examines the effectiveness of supervised summarisation models under various conditions, evaluating different combinations of labels, features, and learners, providing empirical observations over the DUC 2004 Task 2 dataset (generic summarisation).

## 5.1 Learning to Predict Summary Sentences

We begin by introducing the supervised learning problem. Given dependent variables, $\boldsymbol{y}$, and independent variables, $\boldsymbol{X}$, the supervised learning task (Hastie et al., 2009; Witten et al., 2016) can be stated as: $f : \boldsymbol{X} \mapsto \boldsymbol{y}$, i.e. a function mapping $\boldsymbol{X}$ to $\boldsymbol{y}$; or $\boldsymbol{y} = f(\boldsymbol{X})$, i.e. $\boldsymbol{y}$ as a function of $\boldsymbol{X}$; or $P(\boldsymbol{y}|\boldsymbol{X})$, i.e. the probability of $\boldsymbol{y}$ given $\boldsymbol{X}$. Formally, given labels, $\boldsymbol{y}$, for training data, $\boldsymbol{X}$, a model, $\theta$, is learned as a function of the labelled training data, i.e. $\theta = f(\boldsymbol{y}, \boldsymbol{X})$. Predictions, $\hat{\boldsymbol{y}}$, on test data, $\hat{\boldsymbol{X}}$, are a function of the learned model and the un-labelled test data, i.e. $\hat{\boldsymbol{y}} = g(\theta, \hat{\boldsymbol{X}})$, as shown (expressed as function composition) in Equation 5.1:

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix} = g\left(f\left(\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix}\right), \begin{bmatrix} \hat{x}_{1,1} & \hat{x}_{1,2} & \cdots & \hat{x}_{1,n} \\ \hat{x}_{2,1} & \hat{x}_{2,2} & \cdots & \hat{x}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{m,1} & \hat{x}_{m,2} & \cdots & \hat{x}_{m,n} \end{bmatrix}\right) \tag{5.1}$$

Machine learning techniques differ in two key characteristics: how the model, $\theta$, is learned (i.e. fitted to the training data); and how predictions, $\hat{\boldsymbol{y}}$, are computed from the model. Specifically, unique to each learner is the particular implementation of $f()$ and $g()$ from Equation 5.1. We now discuss the supervised extractive multi-document newswire summarisation problem.

As shown in Equation 5.1, each unique item of interest within the problem domain is represented by a row in the matrix $\boldsymbol{X}$ (training) or $\hat{\boldsymbol{X}}$ (test), with a corresponding label in $\boldsymbol{y}$ (known) or $\hat{\boldsymbol{y}}$ (unknown). Within the context of extractive newswire summarisation, such rows (i.e. training and test instances) represent sentences from newswire articles. For each sentence, we require an $n$-dimensional numerical vector-based representation of that sentence, $x_i = (x_{i,1}, x_{i,2} \ldots, x_{i,n})$, and for the training data, a known label, $y_i$, where we aim to predict $\hat{y}_i$.

Regression techniques (Hastie et al., 2009; Witten et al., 2016), based on Support Vector Machines (Vapnik, 1995), have previously been shown to be effective for the task of extractive multi-document newswire summarisation (Ouyang et al., 2011a). The regression task involves inferring numerical predictions for items of interest (i.e. instances), as opposed to the classification task (Aggarwal and Zhai, 2012), where instances are categorised into one or more discrete classes. In this Thesis, we investigate the application of regression techniques for the task of supervised extractive summarisation of newswire, learning to predict the summary worthiness of newswire sentences to extract summaries of news-worthy events.

Formally, where $\hat{y}_i$ is $i^{th}$ numerical prediction in $\hat{\mathbf{y}}$, $n$ is the number of features in $x_i$, $x_{i,j}$ is the $j^{th}$ feature in $x_i$, and $\theta$ are the model parameters (co-efficients), with $\theta_0$ the bias term (controlling the intercept) and $\theta_j$ is the $j^{th}$ feature weight, a linear regression model is defined as the weighted sum (i.e. linear combination) of the features, plus the bias term, defined as: $\hat{y}_i = \theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2} + \cdots + \theta_n x_{i,n}$. Where the loss function solves: $\min_{\theta} ||\boldsymbol{X}\theta - \mathbf{y}||_2{}^2$, i.e. minimising the residual sum of squares, this model is referred to as Ordinary Least Squares.

In our experiments, in Section 5.5, we evaluate a sample of commonly used linear and non-linear regression-based machine learning techniques (i.e. learners). Such machine learning techniques include: Generalised Linear Models[1]; Support Vector Regression[2,3] and Gradient Boosting Regression Trees[4] (Pedregosa et al., 2011; Fan et al., 2008; Chang and Lin, 2011; Chen and Guestrin, 2016). We first address the problem of obtaining high-quality training data (Section 5.2 and 5.3) and then discuss supervised summarisation features (Section 5.4).

## 5.2 Training Data for Supervised Summarisation

In the extractive supervised summarisation setting, the required training data takes the form of per-sentence labels, where each sentence is labelled according to its suitability for inclusion into the summary. However, within the extractive newswire summarisation domain, there is a lack of machine learning training data that is directly annotated in such a per-sentence manner (Nenkova and McKeown, 2011). More commonly, we find human-annotated data in the form of abstractive summaries, produced as part of summarisation evaluation campaigns (Over et al., 2007). Given such abstractive summaries, where there is no direct correspondence between the gold-standard summary sentences and the sentences from the original (summarised) documents, methods that score document sentences with respect to gold-standard summary text(s) have been used to automatically induce per-sentence labels for training supervised summarisation models (e.g. Mani and Bloedorn, 1998; Svore et al., 2007; Toutanova et al., 2007; Ouyang et al., 2011a; Chali and Hasan, 2012; Oliveira et al., 2016).

[1] scikit-learn.org/stable/modules/linear_model.html
[2] www.csie.ntu.edu.tw/~cjlin/libsvm
[3] www.csie.ntu.edu.tw/~cjlin/liblinear
[4] github.com/dmlc/xgboost

Supervised approaches to multi-document newswire summarisation treat extractive summarisation as a sentence scoring problem, where the aim is to rank all sentences extracted from the input news articles based on their suitability for inclusion into a summary. The top ranked sentences are incrementally added to the summary, until the target summary length is reached (e.g. 100 words). However, before each sentence is inserted into the summary, it is common to apply a redundancy removal technique to avoid the inclusion of multiple sentences with the same or similar content – commonly, Maximal Marginal Relevance (Carbonell and Goldstein, 1998) or a cosine similarity filtering mechanism is used (Hong et al., 2014).

When training supervised regression models to score each sentence, there are two prerequisites. First, a series of discriminative features to represent sentences are required (Oliveira et al., 2016). Second, a set of discriminative training instances are needed. These are example sentences, about an event, with associated ground-truth numerical labels quantifying to what extent each sentence is a high-quality candidate summary sentence, indicating how good each sentence is for inclusion into a summary for that event. The per-sentence labels are typically real-valued numerical scores within the range 0 (poor) to 1 (excellent). The goal of the learning process is to effectively combine the features extracted from a sentence, and based on the training instance target label, produce a supervised model that can automatically induce scores for sentences from un-seen events.

The focus of this chapter is how to obtain the ground-truth numerical labels for a sentence. Importantly, unlike in other domains where supervised models are used (e.g. learning-to-rank (Liu, 2009)), the summarisation community has not produced datasets containing human-annotated sentence-level labels to train such models. This is because summaries are evaluated as an atomic unit (since factors such as redundancy, coherence and focus are important (Jones, 1998; Nenkova and McKeown, 2011)), rather than in terms of their individual sentences (cf. learning-to-rank, where search result pages are evaluated in terms of the individual documents ranked). Instead, the ground-truth numerical labels are inferred from gold-standard summaries produced by humans (Kupiec et al., 1995; Mani and Bloedorn, 1998). Examples of human authored summaries include: the introductions to Wikipedia articles[1]; professional summaries of news articles found in the NYT Corpus[2]; and multi-document newswire (i.e. event) summaries produced by assessors at summarisation evaluation campaigns (e.g. the Document Understanding Conference).

[1] en.wikipedia.org/wiki/Wikipedia:Summary_style

[2] catalog.ldc.upenn.edu/LDC2008T19

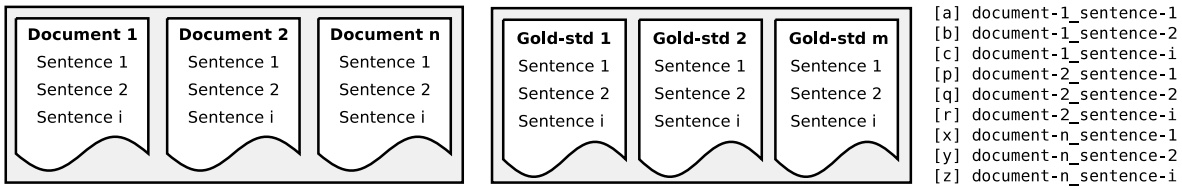



**Figure 5.1:** The per-sentence labels required for extractive supervised summarisation. Given $n$ documents, each containing $i$ sentences, each sentence is scored with respect to a collection of $m$ gold-standard summary text(s).

To leverage these gold-standard summaries, a scoring function is needed that takes as input a sentence (about an event) and one or more gold-standard summaries (about the same event), producing an effectiveness score for that sentence. Based on the implicit assumption that an effective sentence should be textually similar to the gold-standard summaries, prior works have used measures of sentence-to-summary similarity to produce the ground-truth numerical labels. A variety of text similarity measures have been previously used in the literature, such as word overlap (Schilder and Kondadadi, 2008; Ouyang et al., 2011b), cosine similarity (Oliveira et al., 2016) or semantic correspondence (Cheng and Lapata, 2016) with the gold-standard. However, the de-facto standard metric used in the majority of works for calculating the sentence-to-summary similarity is ROUGE recall (Svore et al., 2007; Toutanova et al., 2007; Galanis and Malakasiotis, 2008; Chali et al., 2009; Shen and Li, 2011; Ng et al., 2012; Li et al., 2013, 2015; Cao et al., 2015; Peyrard and Eckle-Kohler, 2016).

This process of labelling summarisation training data is illustrated in Figure 5.1, where we show a collection of $n$ documents, each containing one or more sentences. The desired outcome is that we assign per-sentence training labels (i.e. score sentences) based on the corresponding $m$ gold-standard summary text(s) for this particular set of documents. In order to automatically induce labels for sentences in this manner, it is required that the document set has previously been summarised by human annotators. In this Section, we discuss a range of sentence scoring functions, arguing that such methods for automatically inducing high-quality per-sentence labels from gold-standard abstractive summaries can be used for training supervised summarisation models that exhibit state-of-the-art effectiveness for the task of generic multi-document newswire summarisation.

Formally, given a set of $m$ sentences (from multiple news articles), $S = (s_1, s_2, \ldots, s_m)$, and human-authored (abstractive) gold-standard summaries, $G$, each document sentence, $s_i$, is associated with $n$ gold-standard summary text(s), $g_i = (g_{i,1}, g_{i,2} \ldots, g_{i,n})$. Then, $S$ and $G$ are mapped to per-sentence labels, $\mathbf{y} = (y_1, y_2, \ldots, y_m)$, i.e. $(S, G) \mapsto \mathbf{y}$, as shown in Equation 5.2:

$$\left( \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{pmatrix}, \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,n} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m,1} & g_{m,2} & \cdots & g_{m,n} \end{pmatrix} \right) \longmapsto \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \tag{5.2}$$

Sentence scoring functions, $y_i = f(s_i, (g_{i,1}, g_{i,2} \ldots, g_{i,n}))$, that take as input a sentence, $s_i$, and one or more gold-standard summary text(s), $g_i = (g_{i,1}, g_{i,2} \ldots, g_{i,n})$, and produce a real-valued numerical score (i.e. label), $y_i$, for that sentence, are the subject of this chapter. Specifically, we investigate three general methods for labelling newswire sentences with respect to gold-standard summary text(s). In particular, as described in Section 5.3, we explore scoring (i.e. labelling) sentences using string similarity functions, sentence retrieval models, and ROUGE-$n$ summarisation evaluation metrics. In Section 5.5, we evaluate the effectiveness of learned models, trained on the features defined in Section 5.4, and trained using the labels obtained via the functions we describe in the next section.

## 5.3 Automatically Labelling Training Data

Obtaining more accurate training data will result in more effective supervised summarisation models. We now address the problem of obtaining high-quality training data, for training supervised summarisation models.

### 5.3.1 String Similarity Labels

The first group of per-sentence scoring (i.e. labelling) functions that we investigate are string similarity functions. As demonstrated by Oliveira et al. (2016), labels for newswire sentences can be obtained by scoring each sentence by its cosine similarity to the gold-standard summary text(s). The sentence scoring function is defined as: $y_i = \text{CosSim}(s_i, (g_{i,1}, g_{i,2} \ldots, g_{i,n}))$. Cosine similarity was previously defined in Equation 4.4. The intuition is that a good summary sentence will exhibit high lexical similarity to gold-standard summary sentences.

Further, we propose the use of Kullback-Leibler divergence (Kullback and Leibler, 1951) and Jensen-Shannon divergence (Lin, 1991) as a means to obtain scores for newswire sentences, with respect to gold-standard summary text(s). Such measures of string distance pro-

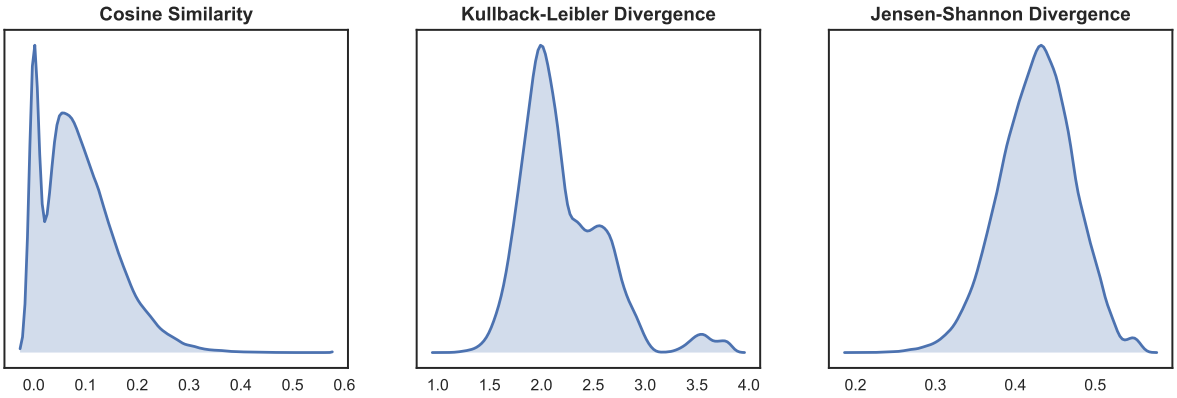
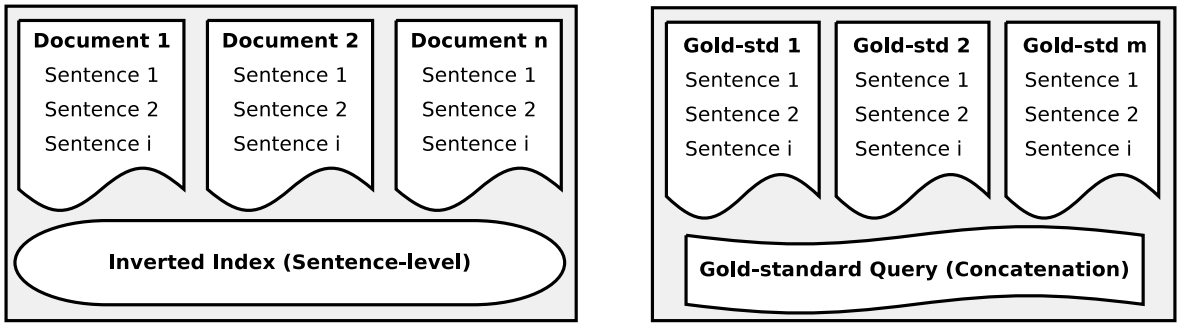



**Figure 5.2:** Kernel density estimation plots over the scores of the string metrics labels, within the training dataset. The $x$-axis is on the scale for each sentence scoring function (illustrating the score range), and the area under the curve illustrates the distribution of scores over the label's range.

vide an information theoretic measure of the divergence of the language model of individual sentences from the language model of the human-authored exemplar summaries. The sentence scoring function is then defined as: $y_i = \text{KLD}(s_i, (g_{i,1}, g_{i,2} \ldots, g_{i,n}))$, for Kullback-Leibler divergence, or defined as: $y_i = \text{JSD}(s_i, (g_{i,1}, g_{i,2} \ldots, g_{i,n}))$, for Jensen-Shannon divergence. Kullback-Leibler divergence was previously defined in Equation 4.7, and Jensen-Shannon divergence was previously defined in Equation 3.5.

Figure 5.2 provides a visualisation of the distribution of scores under this labelling group. For the cosine similarity method, higher scores indicate better summary sentences, whereas lower scores indicate better summary sentences for the divergence methods. Under each labelling function, the desired outcome is a numerical distribution over the sentences in the training set that distinguishes high-quality and low-quality summary sentences. As shown in Figure 5.2, the per-sentences scores from the cosine similarity labelling method range from approximately 0 to 0.6, and from 1 to 4 for the Kullback-Leibler divergence labelling method, and from 0.2 to 0.55 for the Jensen-Shannon divergence labelling method. Cosine similarity and Jensen-Shannon divergence are bounded to the range $[0..1]$, whereas Kullback-Leibler divergence is un-bounded. Figure 5.2 illustrates, for each string similarity labelling method, that there exists an observable threshold (i.e. distinguishing high and low-quality sentences) indicating high-quality summary sentences, which supervised models should learn to predict.




| Topic | Q# | Sentence Identifier | Rank | Score | Model |
|---|---|---|---|---|---|
| d30001t | Q0 | d30001t_APW19981016.0240_2_20 | 0 | 6.79642739097806 | TF_IDF |
| d30001t | Q0 | d30001t_APW19981113.0251_0_5 | 1 | 6.795489345592368 | TF_IDF |
| d30001t | Q0 | d30001t_APW19981124.0267_0_9 | 2 | 6.74501546834585 | TF_IDF |
| d30001t | Q0 | d30001t_APW19981026.0220_1_12 | 3 | 6.713762440336056 | TF_IDF |
| d30001t | Q0 | d30001t_APW19981120.0274_0_8 | 4 | 6.607022879357657 | TF_IDF |
| d30001t | Q0 | d30001t_APW19981016.0240_1_10 | 5 | 6.26640303989501 | TF_IDF |
| d30001t | Q0 | d30001t_APW19981118.0276_12_124 | 6 | 6.096967752135507 | TF_IDF |
| d30001t | Q0 | d30001t_APW19981016.0240_0_0 | 7 | 5.867039438432238 | TF_IDF |
| d30001t | Q0 | d30001t_APW19981118.0276_1_17 | 8 | 5.622857013782313 | TF_IDF |
| d30001t | Q0 | d30001t_APW19981022.0269_11_109 | 9 | 5.478186551401906 | TF_IDF |

**Figure 5.3:** The sentence retrieval method for labelling sentences for supervised summarisation. First, a sentence-level index is created, indexing all sentences from 1 or more documents. Second, a query is derived by concatenating all gold-standard summary sentences together. Third, the gold-standard query is executed on the index, resulting in a ranked list of sentences, where each sentence is scored by a particular retrieval model.

### 5.3.2 Sentence Retrieval Labels

The second group of per-sentence labelling functions that we investigate are ranking models from the Information Retrieval literature (Croft et al., 2010; Büttcher et al., 2010). In the sentence retrieval task, the aim is to retrieve and rank relevant sentences, from within a collection of documents, given a query (Murdock, 2006; Balasubramanian et al., 2007). We propose to label newswire sentences based on their retrieval scores, and retrieval ranks, where the query is taken as the concatenation of all sentences from the gold-standard summary text(s).

We illustrate the proposed approach in Figure 5.3. Labelling sentences using sentence retrieval methods first requires that we construct an inverted index (Croft et al., 2010; Büttcher et al., 2010). Instead of indexing the contents of whole documents, each sentence from each document is indexed individually, so that sentences can be retrieved (i.e. scored and ranked) in response to a query. For constructing the per-sentence inverted index data structure, we use the Terrier Information Retrieval Platform[1] (Macdonald et al., 2012). Next, we construct a query, which is taken as the concatenation of all sentences from the $m$ gold-standard summary text(s). Once the query has been prepared, it is executed on the per-sentence inverted index.

[1] terrier.org

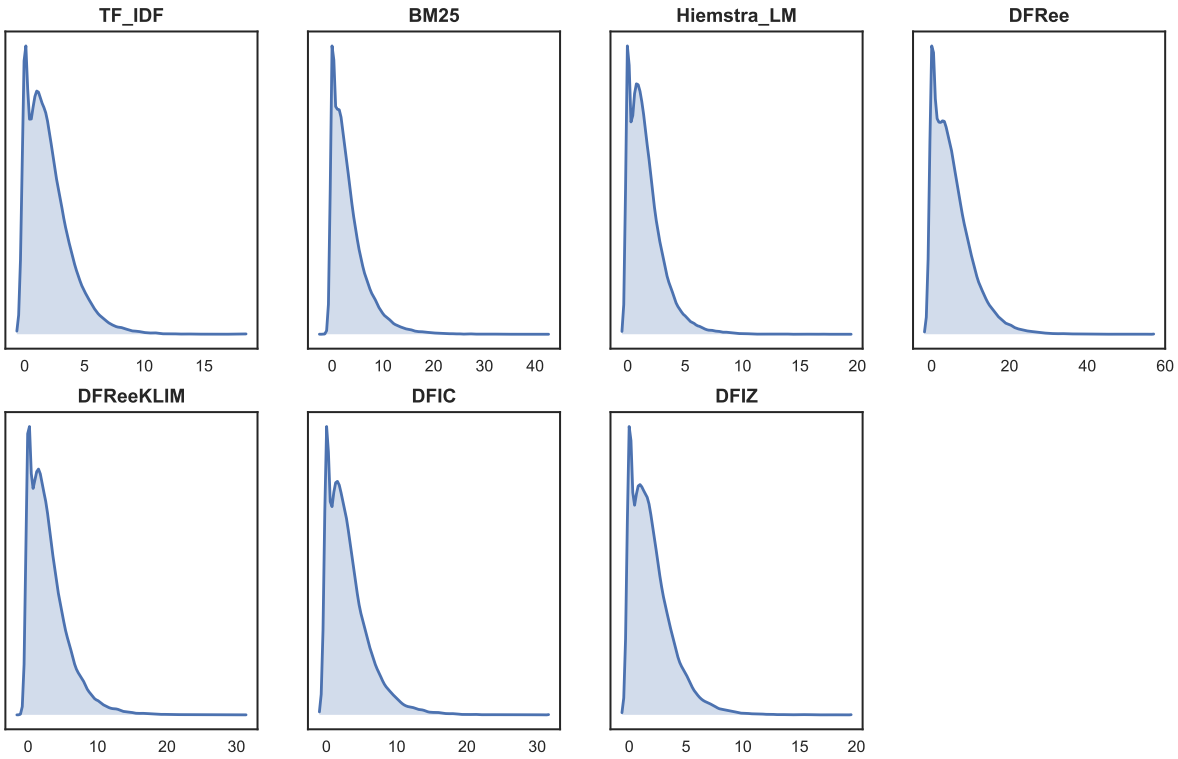



**Figure 5.4:** Kernel density estimation plots over the scores of the sentence retrieval labels, within the training dataset. The $x$-axis is on the scale for each sentence scoring function (illustrating the score range), and the area under the curve illustrates the distribution of scores over the label's range.

As shown in Figure 5.3, executing the gold-standard query on the sentence-level inverted index produces a ranked list of sentences, where each sentence is assigned a retrieval status value (RSV), according to a particular information retrieval model. Specifically, the sentence scoring function is defined as: $y_i = \text{RSV}_{\text{model}}(s_i, (g_{i,1}, g_{i,2} \ldots, g_{i,n}))$, where $\text{RSV}_{\text{model}}$ returns the sentence's score under a particular information retrieval model, with respect to a query – defined as the concatenation of the gold-standard summary, $g_i = (g_{i,1}, g_{i,2} \ldots, g_{i,n})$. For any given sentence, the retrieval score or the rank can be used as a training label. The intuition is that a good summary sentence will exhibit high lexical overlap with the gold-standard query, resulting in a higher ranking compared to low-quality summary sentences.

Figure 5.4 provides a visualisation of the distribution of scores under this labelling group. In our experiments, we use a representative sample of retrieval models[1]: TF_IDF, the standard retrieval model (Croft et al., 2010; Büttcher et al., 2010); BM25 (Robertson et al., 2009), an effective probabilistic retrieval model; Hiemstra_LM (Hiemstra, 2001), from the language

[1] terrier.org/docs/current/javadoc/org/terrier/matching/models/package-summary.html

modelling approach (Ponte and Croft, 1998); DFRee (Amati and van Rijsbergen, 2002) and DFReeKLIM (Amati et al., 2011), from the Divergence from Randomness family of retrieval models (Amati, 2003); and then DFIC and DFIZ, two models based on the divergence from independence (Kocabas et al., 2014). From Figure 5.4, where higher retrieval scores are better, we again note that there exists an observable threshold that distinguishes, under each labelling method, high-quality and low-quality candidate summary sentences.

### 5.3.3 ROUGE-*n* Metrics Labels

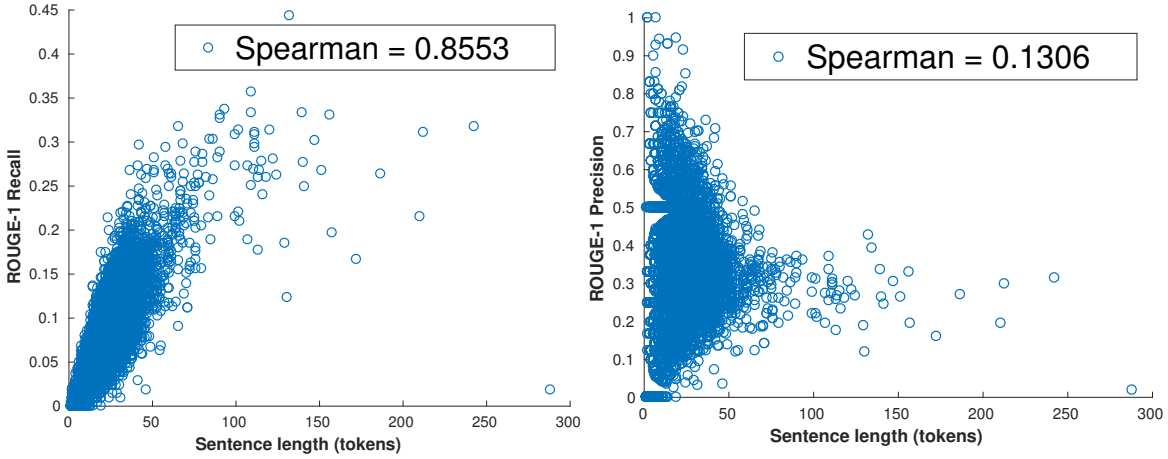
The third group of per-sentence labelling functions that we discuss are ROUGE (Lin, 2004) summarisation evaluation metrics. ROUGE is the standard suite of metrics for evaluating text summarisation, with ROUGE results often reported in the literature (Nenkova and McKeown, 2011). ROUGE is intended to measure the effectiveness of a whole summary, which will typically contain more than one sentence. However, as demonstrated by Svore et al. (2007), to induce per-sentence labels from gold-standard summary text(s), for training supervised summarisation models, the effectiveness of each individual sentence can be evaluated in isolation – i.e. evaluating individual sentences as the summary within a ROUGE-based experiment.

The per-sentence labels obtained using ROUGE provide a numerical quantification of the effectiveness of each sentence. Specifically, the sentence scoring function is then defined as: $y_i = \text{ROUGE}_{\text{metric}}(s_i, (g_{i,1}, g_{i,2} \ldots, g_{i,n}))$, where $\text{ROUGE}_{\text{metric}}$ is a particular ROUGE metric. The ROUGE metrics we investigate are ROUGE-*n* recall and precision, where $n = [1..4]$. These metrics were previously defined in Section 3.1.1, in Equation 3.1 and Equation 3.2.

The intuition is that per-sentence ROUGE scores should accurately reflect the summarisation effectiveness of individual sentences, based on a sentence independence assumption. Typically, within a ROUGE-based evaluation, several sentences are evaluated as a single unit. Repeated information among the sentences (i.e. redundancy) is penalised in the scoring formulation. Labelling sentences independently does not consider the redundancy among sentences, which may be a limitation of this (and other) per-sentence labelling methods.

We note that, inducing training data for supervised summarisation in this manner is an application of ROUGE for which it was not originally intended. Nevertheless, the ROUGE recall metric is commonly used to label sentences, e.g. using the DUC[1] (Toutanova et al.,

[1] duc.nist.gov




**(a)** ROUGE-1 recall vs. sentence length.

**(b)** ROUGE-1 precision vs. sentence length.

**Figure 5.5:** Spearman's rank correlation coefficient, computed over the sentences of the training dataset, between sentence length and the ROUGE-1 recall metric (Fig 5.5a), and the ROUGE-1 precision metric (Fig 5.5b).

2007; Galanis and Malakasiotis, 2008; Chali et al., 2009; Shen and Li, 2011; Cao et al., 2015) and TAC[1] (Ng et al., 2012; Li et al., 2013, 2015; Peyrard and Eckle-Kohler, 2016) datasets.

The question we address in this Chapter is: which ROUGE metric (i.e. labelling method) produces the most effective supervised summarisation models? Previous research, using ROUGE-based methods to induce per-sentence labels, have trained on ROUGE recall (i.e. ROUGE recall is the de-facto ROUGE-based labelling method). Often, the justification of learning on ROUGE recall is that ROUGE-1 recall is best able to distinguish pairs of systems (Rankel et al., 2013), while ROUGE-2 recall exhibits agreement with manual evaluation (Owczarzak et al., 2012). However, we hypothesise that when inducing per-sentence labels, ROUGE precision is the most effective metric to learn on. This assertion is based on the knowledge that ROUGE recall is sensitive to summary length (Lin, 2004).

Figure 5.5 illustrates this sensitivity. Figure 5.5 shows the Spearman (1904) rank correlation coefficient of sentence length (in words) and the ROUGE-1 metrics, for all sentences within the training dataset (c.f. Section 5.5.2). From Figure 5.5, we observe that ROUGE-1 recall exhibits very strong (Rosenthal, 1996) correlation with sentence length, whereas ROUGE-1 precision is much less correlated with sentence length. This means, if two sentences of unequal length (in words) are evaluated using ROUGE recall, the longer sentences is more likely to obtain higher recall scores simply by containing more *n*-grams.

[1] nist.gov/tac

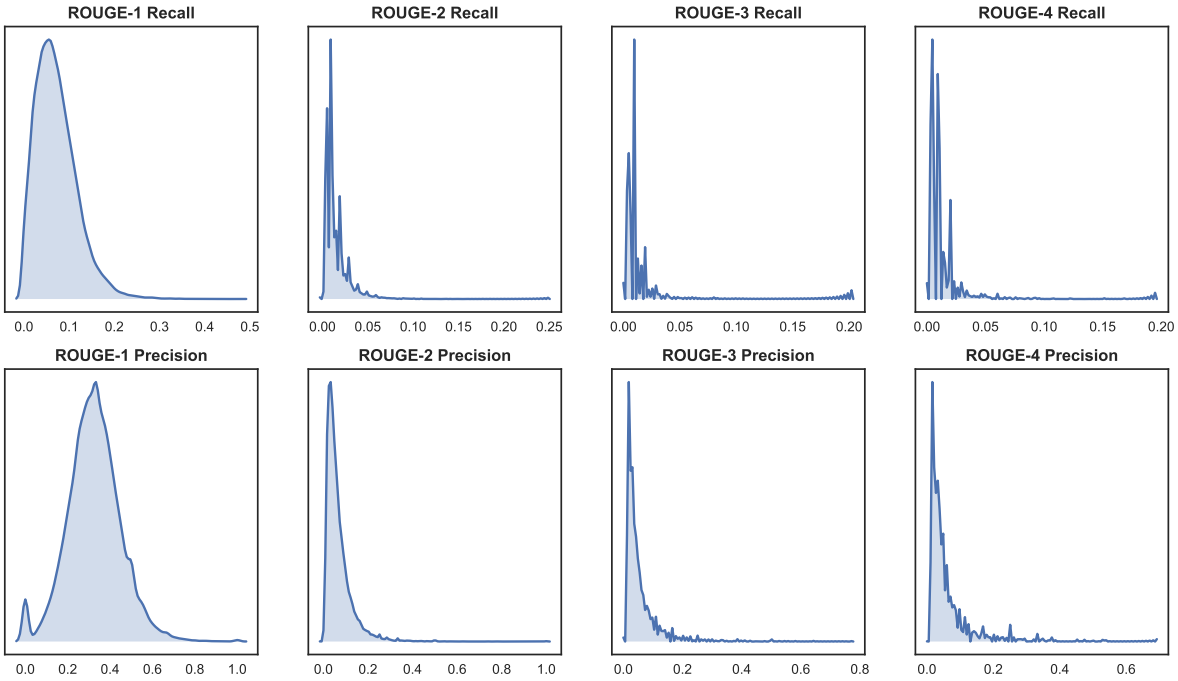



**Figure 5.6:** Kernel density estimation plots over the scores of the ROUGE metrics labels, within the training dataset. The $x$-axis is on the scale for each sentence scoring function (illustrating the score range), and the area under the curve illustrates the distribution of scores over the label's range.

Indeed, it is typical in the experimental setup of summarisation evaluations to truncate the text of summaries under evaluation to equal lengths (Over et al., 2007). When using ROUGE metrics to score each sentence individually (in a training dataset), we are essentially scoring summaries of varying lengths. Therefore, longer sentences will tend to obtain higher ROUGE recall scores than shorter sentences. This may lead to supervised summarisation models, which have been trained on labels induced using ROUGE recall, to exhibit a (possibly less-effective) model bias towards longer summary sentences. We argue that models trained on ROUGE precision will alleviate such model bias towards longer sentences, selecting long and short sentences equally. The assumption (and research question) is that biasing summary sentence selection towards longer sentences produces less effective summarisation models.

Figure 5.6 provides a visualisation of the distribution of scores under this labelling group. From Figure 5.6, where higher ROUGE evaluation scores are better, again we note that there exists an observable threshold that distinguishes, under the precision-based metrics, high-quality and low-quality candidate summary sentences. However, for recall-based metrics, the indication that the labelling method can distinguish between high-quality and low-quality sentence is less obvious, particularly at higher-order ROUGE-$n$ recall metrics (i.e. [3..4]).

## 5.4 Features for Supervised Summarisation

The next problem we address is how to represent newswire sentences in the specific format required for machine learning techniques. In particular, the machine learning techniques we investigate do not operate directly on natural language sentences. Instead, the characteristics of newswire sentences are encoded and represented in a numerical vector-based format, suitable for processing by machine learning techniques (Hastie et al., 2009; Witten et al., 2016). We propose to use the specific set of standard unsupervised summarisation baselines defined by Hong et al. (2014) as summarisation features within supervised summarisation models.

The key desirable property of such per-sentence feature vectors is discriminativeness, i.e. that the numerical encoding accurately reflects the differences in characteristics between sentences. Within the context of supervised summarisation, such characteristics include, for example, importance and salience (Nenkova and McKeown, 2011). Given more effective feature representations, a machine learning technique can more discriminantly model the relationship between sentences and labels, to then better predict labels for previously un-seen sentences, resulting in more effective supervised machine learned summarisation models.

Formally, given a set of $m$ sentences, $S = (s_1, s_2, \ldots, s_m)$, each sentence, $s_i$, is represented by a $n$ dimensional feature vector, $x_i = (x_{i,1}, x_{i,2} \ldots, x_{i,n})$, i.e. the set of $m$ sentences is mapped to an $m * n$ feature matrix, $S \mapsto \boldsymbol{X}_{m,n}$, as shown in Equation 5.3:

$$\begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{pmatrix} \longmapsto \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix} \tag{5.3}$$

As an example, we will discuss the representation of the lead sentence from document "APW19981016.0240", within topic "d30001t", of the DUC 2004 dataset:

> "Cambodian leader Hun Sen on Friday rejected opposition parties demands for talks outside the country, accusing them of trying to internationalize the political crisis."

In the supervised machine learning experimental setup used throughout this Thesis, sentences are subjected to case-folding, stopword removal (removing the 50 most common English words[1]), and Porter (1980) stemming. The sentence is then encoded as a numerical vector.

[1] en.wikipedia.org/wiki/Most_common_words_in_English

**Table 5.1:** Seven summarisation features we use in our experiments, for the sentence: *"Cambodian leader Hun Sen on Friday rejected opposition parties demands for talks outside the country, accusing them of trying to internationalize the political crisis."*

| **Feature** | Position | Length | FreqSum | TsSum | Centroid | LexRank | GreedyKL |
|---|---|---|---|---|---|---|---|
| **Index** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Value | 0 | 18 | 0.0082 | 0.3333 | 0.3524 | 0.0070 | 1.2513 |
| MinMax | 0 | 0.0596 | 0.2136 | 0.1667 | 0.6546 | 0.4053 | 0.3465 |
| *Z*-score | -0.7651 | 0.4580 | 2.0060 | 0.4615 | 1.6066 | 2.0115 | -2.5972 |

### 5.4.1 Baseline Algorithms as Features

The unsupervised summarisation algorithms we propose to use as features are the standard baselines identified by Hong et al. (2014). We argue that such standard baselines can be used as features for training supervised machine learned summarisation models that exhibit state-of-the-art effectiveness, for the task of generic multi-document newswire summarisation, when combined with high-quality training data (Section 5.3), and using linear or non-linear regression techniques (Section 5.1). In particular, we use the FreqSum (Nenkova et al., 2006), TsSum (Conroy et al., 2006), Centroid (Radev et al., 2004), LexRank (Erkan and Radev, 2004), and GreedyKL (Haghighi and Vanderwende, 2009) algorithms.

These baseline algorithms, used to derive per-sentence scores (i.e. features), were previously discussed and defined in Section 4.2. Further, we use two additional lexical features: the position of a sentence within a newswire article (i.e. first, second, etc.); and the length of the sentence (in words). For Position and Length, these features represent hypotheses regarding the summary worthiness of sentences near the beginning of the news article (i.e. Position is a Lead-based feature, c.f. Section 4.1.2), and whether long or short sentences are to be preferred for inclusion into the summary text (where sentence length is measured in words).

As shown in Table 5.1, which provides the numerical encoding for our example sentence, sentences are represented as an $n$-dimensional feature vector, where $n = 7$ (i.e. 7 features). For the FreqSum, TsSum, Centroid, and LexRank features, numerically higher summarisation feature scores are an indication of a higher-quality summary sentence. For the GreedyKL feature, a measure of divergence, numerically lower summarisation feature scores are an indication of a higher-quality summary sentence. For the Position and Length features, the numerical values record in which position a particular sentence occurred (starting from 0),

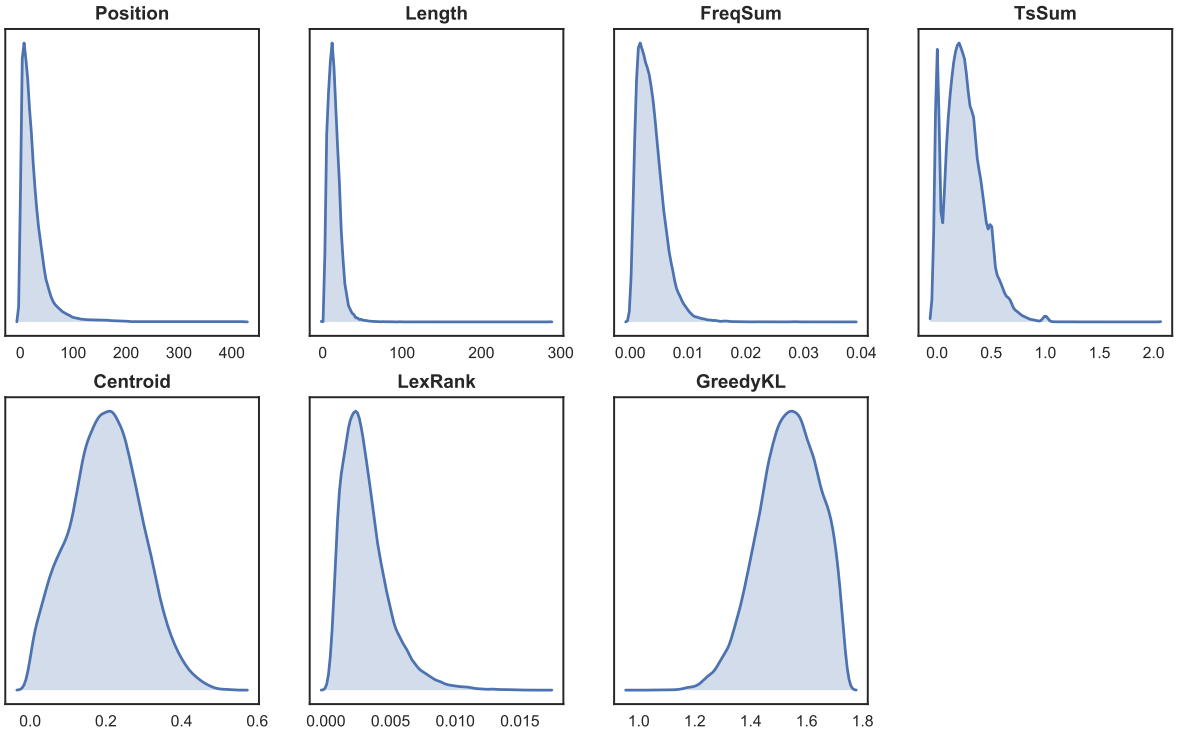



**Figure 5.7:** Kernel density estimation plots over the scores of the summarisation features, within the training dataset. The $x$-axis is on the scale for each sentence scoring function (illustrating the score range), and the area under the curve illustrates the distribution of scores over the feature's range, demonstrating discriminativeness.

and how many words are contained in the given sentence. Further, as shown in Table 5.1, the raw scores computed from the five summarisation algorithms, and the two lexical features (Position and Length), are subjected to a pre-processing step. As per machine learning best-practice guidelines (Müller and Guido, 2017; Géron, 2017), feature normalisation (scaling within the range $[0..1]$) or feature standardisation (to zero mean and unit variance) is applied.

We explore the statistical properties of the features over the training data (c.f. Section 5.5.2) in Figure 5.7. Figure 5.7 provides a kernel density estimation plot (an estimation of the probability density function) of the seven features. For each plot (for each feature), the $x$-axis shows the score range for that feature. For example, the Centroid feature score range is from 0 to 0.6. The area under the curve can be interpreted as a smoothed histogram, illustrating the distribution of scores over the range of the feature. From Figure 5.7, we can hypothesise that the five baseline algorithms (FreqSum, TsSum, Centroid, LexRank, and GreedyKL) will provide a set of discriminative features for machine learning models. When visually interpreting the curves for each feature, the worst-case is that the curve is flat, indicating that the feature does not discriminate between sentences. However, as Figure 5.7 illustrates, scores $> 0.01$

**Table 5.2:** Pearson's $\rho$ correlation coefficients among baseline summarisation features computed over the training dataset. Correlation coefficients are interpreted (Rosenthal, 1996) as: $\rho > .10$ weak; $\rho > .30$ moderate; $\rho > .50$ strong (shown in **bold**); and $\rho > .70$ very strong (shown as **bold** and underline).

| **Pearson's $\rho$** | Position | Length | FreqSum | TsSum | Centroid | LexRank | GreedyKL |
|---|---|---|---|---|---|---|---|
| Position | – | -0.09 | -0.06 | -0.01 | -0.15 | -0.29 | 0.15 |
| Length | | – | 0.02 | 0.09 | **0.51** | 0.28 | **-0.64** |
| FreqSum | | | – | **0.61** | **0.67** | 0.44 | **-0.67** |
| TsSum | | | | – | **0.59** | 0.25 | **-0.53** |
| Centroid | | | | | – | **0.64** | **-0.94** |
| LexRank | | | | | | – | **-0.61** |
| GreedyKL | | | | | | | – |

for FreqSum, scores $> 0.5$ for TsSum, scores $> 0.4$ for Centroid, scores $> 0.005$ for LexRank, and scores $< 1.4$ for GreedyKL numerically quantify that there exists an identifiable subset of high-quality summary sentences under each feature, which supervised machine learned summarisation models should learn to predict.

A further statistical analysis of the features is provided in Table 5.2, showing Pearson's correlation coefficients among the baseline summarisation features and lexical features. In Table 5.2, we annotate the strength of the correlation, qualitatively interpreting correlation coefficients following Rosenthal (1996): $\rho > .10$ weak; $\rho > .30$ moderate; $\rho > .50$ strong (shown in bold); and $\rho > .70$ very strong (underlined bold). From Table 5.2, we observe that the Centroid and GreedyKL features exhibit the highest correlation, $r = -0.94$. Indeed, both of the Centroid and GreedyKL features exhibit (at least) strong correlation with all other features – except Position, which is (at most) weakly correlated with all of the other features.

The effectiveness of the summarisation features we have defined, when applying feature scaling or standardisation, and where features are correlated with each other, are evaluated in Section 5.5.3. Further, we conduct an analysis of each feature's importance in Section 5.5.4.

## 5.5 Evaluation

In this section, we conduct an experimental evaluation of supervised machine learned summarisation models. We begin by stating our research questions, then describe our experi-

mental setup. Results are provided over the DUC 2004 Task 2 dataset, for the task of generic extractive multi-document newswire summarisation. Finally, we discuss and analyse our empirical observations.

### 5.5.1 Research Questions

In our Thesis Statement (Section 1.2), we formed Hypothesis 3:

> We hypothesise that supervised machine learned summarisation models based on regression techniques, that exhibit state-of-the-art effectiveness, can be trained on discriminative features, derived from standard multi-document newswire summarisation algorithms, using automatically labelled training data induced from gold-standard summaries.

To validate Hypothesis 3, we address the following research questions:

**Research Question 5.1.** Can baseline newswire summarisation algorithms be used to provide a set of discriminative features for training effective supervised summarisation models?

**Research Question 5.2.** Which sentence scoring functions can be used to provide high-quality per-sentence labels for training effective supervised summarisation models?

**Research Question 5.3.** Which linear or non-linear regression-based machine learning techniques are effective for learning to predict candidate summary sentence scores?

We claim that supervised machine learned summarisation models, that exhibit state-of-the-art effectiveness for the task of extractive generic multi-document newswire summarisation, can be trained on discriminative features derived from baseline newswire summarisation algorithms, using high-quality labels automatically induced from gold-standard summary text(s) via sentence scoring functions, using regression-based learners.

Our research questions are inter-linked, i.e. within supervised machine learning experiments, to evaluate features (RQ 5.1), we require labels and learners, to evaluate labels (RQ 5.2), we require features and learners, and to evaluate learners (RQ 5.3), we require features and labels. Research Question 5.1 addresses features for supervised summarisation, where we evaluate learned models trained using the proposed set of multi-document newswire summarisation baselines discussed in Section 5.4. Research Question 5.2 addresses training data for

supervised summarisation, where we evaluate learned models trained using the per-sentence labelling methods discussed in Section 5.3. Research Question 5.3 addresses machine learning models, where we evaluate a range of linear and non-linear regression-based techniques.

### 5.5.2 Experimental Setup

In the following supervised summarisation experiments, we summarise newswire articles from the DUC 2001–2004 datasets (generic extractive multi-document newswire summarisation). The DUC 2001 and 2002 datasets are combined for the training set, the DUC 2003 dataset is used as a validation set, with the DUC 2004 dataset reserved as the test set.

We experiment with an Ordinary Least Squares (OLS) linear regression model, using scikit-learn (Pedregosa et al., 2011), a linear Support Vector Machine (SVM) regression model (L2 regularized, L1 loss), using LIBLINEAR (Fan et al., 2008), a non-linear SVM regression model ($\varepsilon$-SVR, Gaussian kernel), using LIBSVM (Chang and Lin, 2011), a Gradient Boosting Regression Tree (GBRT), using XGBoost (Chen and Guestrin, 2016), and a LambdaMART (Wu et al., 2010), a regression-based learning-to-rank (Liu, 2009) model, using QuickRank (Capannini et al., 2016). For the LambdaMART model, the training data labels that we investigate are discretised into graded relevance judgements.

For the OLS model, there are no hyper-parameters to tune. For the linear SVM model, the $C$ hyper-parameter is learned on the validation data. For the non-linear SVM model, we use the validation data to learn the $C$ and $\gamma$ hyper-parameters. For the GBRT model, the learning rate, $\gamma$, and maximum tree depth hyper-parameters are learned on the validation data. For the LambdaMART model, the validation data is used to learn the number of trees, number of leaves, and shrinkage rate. Hyper-parameters of machine learning models are optimised for root mean squared error (RMSE). Further, we treat feature pre-processing (scaling and standardisation) as a hyper-parameter, and learn whether to apply such normalisation (or not) on the validation data, optimising for the ROUGE-2 recall metric. Furthermore, as the output from learned models is a ranking of candidate summary sentences, summary sentences are selected from this ranking using the cosine similarity anti-redundancy filtering component (c.f. Section 4.2.2). The cosine similarity threshold is also learned on the validation data, optimising for the ROUGE-2 recall metric.

**Table 5.3:** Lower-bounds, baseline and state-of-the-art ROUGE results, over DUC 2004, for multi-document newswire summarisation systems. We derive average ROUGE-1 and ROUGE-2 state-of-the-art scores, which provide a means (i.e. a threshold) to classify algorithms as generally exhibiting state-of-the-art effectiveness.

| **Lower-bounds** | **R-1** | **R-2** |
|---|---|---|
| *Random* | 30.27 | 4.33 |
| *Lead* | 31.46 | 6.13 |
| **Baseline Algorithms** | **R-1** | **R-2** |
| *LexRank* | 36.00 | 7.51 |
| *Centroid* | 36.42 | 7.98 |
| *FreqSum* | 35.31 | 8.12 |
| *TsSum* | 35.93 | 8.16 |
| *Greedy–KL* | **38.03** | **8.56** |

| **State-of-the-art** | **R-1** | **R-2** |
|---|---|---|
| *CLASSY 04* | 37.71 | 9.02 |
| *CLASSY 11* | 37.21 | 9.21 |
| *Submodular* | 39.23 | 9.37 |
| *DPP* | **39.84** | 9.62 |
| *OCCAMS_V* | 38.50 | 9.75 |
| *RegSum* | 38.60 | 9.78 |
| *ICSISumm* | 38.44 | **9.81** |
| **Average** | 38.50 | 9.51 |

To evaluate summary texts, we report ROUGE (Lin, 2004) automatic evaluation metrics. Following best practice (Hong et al., 2014), the summaries under evaluation are subject to stemming, stopwords are retained, and we report ROUGE-1 and ROUGE-2 recall, where ROUGE-2 recall is the target metric. Further, for all experiments, summary lengths are truncated to 100 words (Over et al., 2007). ROUGE results for various summarisation systems are obtained using SumRepo (Hong et al., 2014)[1], which provides the plain-text produced by 5 standard baselines, and 7 state-of-the-art systems, over DUC 2004. Using this resource, we compute ROUGE results over DUC 2004 for the algorithms available within SumRepo, obtaining reference (i.e. baseline and state-of-the-art) results for use in our experiments.

To make conclusions about the summarisation effectiveness of supervised summarisation models, we define three measures of success, based on Table 5.3. In Table 5.3, we provide reference ROUGE results over DUC 2004 for the random and lead baselines, 5 standard baseline algorithms, and 7 state-of-the-art systems (Hong et al., 2014). The first measure of success is that learned models outperform the features they are trained on. The second measure of success is that learned models generally exhibit state-of-the-art effectiveness. The third measure of success is that both the fist and second measure of success are met under the target evaluation metric of ROUGE-2 recall.

[1] www.seas.upenn.edu/~nlp/corpora/sumrepo.html

Specifically, as learned models are trained using baseline summarisation algorithms as features, a successful outcome is where a learned model significantly outperforms all of the individual baseline summarisation algorithms. In particular, as shown in Table 5.3, the most effective baseline is GreedyKL, exhibiting a ROUGE-1 score of 38.03, and a ROUGE-2 score of 8.56. Any supervised summarisation model (i.e. combination of features, labels, and learner) that significantly outperforms GreedyKL, under the ROUGE-1 or ROUGE-2 metrics, will be interpreted as producing effective summaries. This first measure of success, for supervised summarisation models, will be indicated in our results using the † symbol.

Further, to classify whether a supervised machine learned summarisation model exhibits state-of-the-art effectiveness, for the task of extractive generic multi-document newswire summarisation, we derive state-of-the-art ROUGE-1 and ROUGE-2 threshold scores. Such threshold scores are based on the average of the ROUGE-1 and ROUGE-2 effectiveness scores of the 7 state-of-the-art systems, as shown in Table 5.3. In our experiments, if a supervised model exhibits a ROUGE-1 score exceeding 38.50, or a ROUGE-2 score exceeding 9.50, this will be interpreted as generally exhibiting state-of-the-art effectiveness under that metric. This second measure of success will be indicated in our results using bold annotation.

The third measure of success is annotated in our results tables using underline, indicating that a particular model has passed our first two measures of success, but doing so under the target ROUGE-2 recall evaluation metric. Specifically, runs triply annotated with †, bold, and underline, shown only under the ROUGE-2 recall metric, are classed as successful outcomes, i.e. state-of-the-art supervised machine learned summarisation models.

### 5.5.3 Experimental Results

#### Research Question 5.1

We begin with Research Question 5.1. We seek to ascertain whether the specific set of unsupervised multi-document newswire summarisation baselines we proposed (in Section 5.4) to use as features within supervised summarisation models are effective (i.e. discriminative). In particular, learned models are trained on features derived from the FreqSum (Nenkova et al., 2006), TsSum (Conroy et al., 2006), Centroid (Radev et al., 2004), LexRank (Erkan and Radev, 2004), and GreedyKL (Haghighi and Vanderwende, 2009) baselines. The summarisation effectiveness of each of these baselines is known (c.f. Table 5.3). If learned models,

trained on such baselines, outperform the effectiveness of each of the individual baselines, and further, exhibit state-of-the-art effectiveness, then we can conclude that the set of baseline features we have defined are effective for training supervised summarisation models.

To answer Research Question 5.1, we refer to the experimental results in Table 5.4. Table 5.4 provides ROUGE-1 and ROUGE-2 recall summarisation effectiveness scores for supervised machine learned summarisation models. We report results for six learners: Ordinary Least Squares; Ridge regression; a linear SVM regression model; a non-linear SVM regression model (with a Gaussian kernel); a GBRT (Gradient Boosted Regression Trees) model; and LambdaMART (a regression-based ranker). Further, results are reported for three groups of labelling methods: string similarity labels; sentence retrieval labels; and ROUGE-*n* metrics labels. The labelling methods are described in Section 5.3. Furthermore, within Table 5.4, the † symbol indicates that a learned model exhibits ROUGE scores that are statistically significantly more effective than all of the individual baseline algorithms. Statistical significance is reported using the paired Student's t-test, with a 95% confidence level. Additionally, ROUGE-1 scores exceeding 38.50, and ROUGE-2 scores exceeding 9.50, are annotated in bold, indicating that a model generally exhibits state-of-the-art effectiveness. Finally, underline annotation indicates that the ROUGE-2 recall score for a learned model is both significantly more effective than the baseline features and exhibits state-of-the-art effectiveness.

From Table 5.4, we answer Research Question 5.1 by examining results over all learners and all labelling methods, with respect to our third measure of success, defined in Section 5.5.2, and shown with †, bold, and underline annotations under the ROUGE-2 recall metric. We first observe that the numerically highest result under the ROUGE-1 recall metric is 39.73, for an SVR model trained using ROUGE-4 precision labels, and the numerically highest result under the ROUGE-2 recall metric is 10.25, for an SVR-RBF model trained using DFIZ labels. With respect to the state-of-the-art summarisation models defined by Hong et al. (2014), shown in Table 5.3, the SVR model trained using ROUGE-4 precision labels is numerically more effective than all state-of-the-art systems, except DPP which has a ROUGE-1 score of 39.84, and the SVR-RBF model trained using DFIZ labels is numerically more effective than all state-of-the-art systems.

Further, from Table 5.4, we observe that there are 44 cases in total where learned models have met our third measure of success. Specifically, there are 44 cases where learned models

**Table 5.4:** ROUGE summarisation effectiveness, over DUC 2004 Task 2, for supervised regression techniques. We report results for three linear regression models: Ordinary Least Squares (OLS); Ridge regression; and Elastic-Net. Further, we report results for two models based on Support Vector Machines (SVM): a linear Support Vector Regression (SVR), and a non-linear SVR (with an RBF kernel). Furthermore, we report results for a tree-based model: Gradient Boosted Regression Trees (GBRT). In the results table, the † symbol is used to indicate a learned model exhibits ROUGE scores that are significantly more effective than all of the individual baseline algorithms (i.e. features) used to train that model (paired Student's t-test, 95% confidence level). Additionally, ROUGE-1 scores exceeding 38.50, and ROUGE-2 scores exceeding 9.50, are annotated in bold – indicating state-of-the-art effectiveness for the task of generic newswire summarisation (Hong et al., 2014). Further, underline annotation indicates that a model has achieved significance over the baselines, and exhibits state-of-the-art scores, but does so under the target evaluation metric of ROUGE-2 recall.

| | **Linear Regression** | | | | **Support Vector Machine** | | | | **Decision Tree** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **OLS** | | **Ridge** | | **Linear SVR** | | **SVR (RBF)** | | **GBRT** | | **λMART** | |
| **String Similarity Labels** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** |
| Cosine Similarity | 38.15 | **9.67**† | 38.15 | **9.67**† | 37.83 | 9.28 | 38.15 | **9.82**† | 37.70 | 8.73 | 37.91 | 9.32 |
| Kullback-Leibler Divergence | **38.74** | 9.48† | **39.16**† | **9.96**† | 38.37 | 9.45† | 38.35 | 8.82 | 38.47 | 9.48† | 38.13 | **9.71**† |
| Jensen-Shannon Divergence | **38.67** | **9.69**† | **39.06**† | **9.90**† | **38.65** | **9.72**† | 37.95 | **9.62**† | 38.31 | 9.40† | 37.76 | **9.55**† |
| **Sentence Retrieval Labels** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** |
| TF_IDF | 37.85 | 9.20 | 37.68 | 9.07 | 37.73 | 9.46† | 38.25 | **9.70**† | 38.23 | 9.27 | 38.42 | 9.41† |
| BM25 | 37.39 | 9.25 | 36.73 | 8.63 | 37.58 | 9.25 | 37.60 | 9.37† | 37.68 | 9.15 | 36.33 | 8.29 |
| Hiemstra_LM | 38.03 | 9.34† | 37.73 | 9.24 | 38.31 | **9.56**† | **39.24**† | **9.70**† | 38.38 | 9.04 | **38.97** | **9.74**† |
| DFRee | 37.15 | 9.00 | 37.15 | 9.00 | 37.61 | 9.22 | 37.97 | 9.38† | 37.65 | 8.98 | 37.98 | **9.57**† |
| DFReeKLIM | 37.81 | 9.13 | 37.60 | 9.01 | 38.20 | 9.35† | **38.53** | **9.74**† | 38.34 | 8.81 | **38.89** | **9.80**† |
| DFIC | 38.14 | 9.34† | 37.70 | 9.11 | 38.44 | **9.56**† | **38.71** | 9.49† | **38.92** | **9.88**† | 37.93 | 9.09 |
| DFIZ | 37.92 | 9.33† | 37.13 | 8.94 | 38.27 | 9.48† | **39.40**† | **10.25**† | 38.41 | 9.40 | **38.50** | **9.82**† |
| **ROUGE-n Metrics Labels** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** | R-1 | **R-2** |
| ROUGE-1 Recall | 32.16 | 6.07 | 32.16 | 6.07 | 32.06 | 5.96 | 33.14 | 6.61 | 33.06 | 6.61 | 34.33 | 7.13 |
| ROUGE-2 Recall | 36.82 | 8.83 | 36.82 | 8.83 | 37.61 | 8.47 | 37.45 | 7.95 | 37.10 | 8.62 | 37.47 | 9.11 |
| ROUGE-3 Recall | 38.09 | **9.57**† | 38.03 | **9.54**† | 38.16 | 8.85 | 35.23 | 7.48 | 37.51 | 8.29 | 37.50 | 8.99 |
| ROUGE-4 Recall | 38.11 | **9.63**† | 38.11 | **9.63**† | 38.13 | 8.82 | 27.15 | 3.12 | 38.30 | 9.05 | 37.05 | 8.72 |
| ROUGE-1 Precision | **39.02**† | **9.62**† | **39.02**† | **9.62**† | **38.77** | **9.57**† | **39.24**† | **9.85**† | **38.85** | **9.60**† | 38.42 | **9.54**† |
| ROUGE-2 Precision | **39.01**† | **9.62**† | **39.01**† | **9.62**† | **38.67** | 9.48† | **38.75** | **9.71**† | **38.88** | **9.85**† | 37.96 | 8.50 |
| ROUGE-3 Precision | **39.37**† | **9.75**† | **38.80** | **9.63**† | **39.12**† | **9.83**† | 33.34 | 5.94 | **39.18**† | **10.08**† | 37.81 | 8.70 |
| ROUGE-4 Precision | **39.46**† | **9.73**† | **39.25**† | **9.86**† | **39.73**† | **10.11**† | 32.86 | 5.73 | **39.33**† | **9.55**† | **38.94** | **9.95**† |

significantly outperform the baseline algorithms they are trained on, and exhibit state-of-the-art ROUGE scores, under the target evaluation metric of ROUGE-2 recall. From the results in Table 5.4, we can now answer Research Question 5.1. We conclude that the specific set of unsupervised multi-document newswire summarisation baselines we proposed (in Section 5.4) to use as features within supervised summarisation models are indeed effective (i.e. discriminative), under various combinations of labels and learners. In our next research questions, we specifically examine the effectiveness of particular labels and learners.

**Research Question 5.2**

We now address Research Question 5.2, where we evaluate the summarisation effectiveness of learned models when trained on different training data labels. Training on higher-quality labels will result in more effective supervised summarisation models, and we seek to determine which labelling methods result in effective supervised summarisation models. The labelling functions under evaluation are defined in Section 5.3, namely: string metrics labels; sentence retrieval labels; and ROUGE-*n* metrics labels. To answer Research Question 5.1, we again refer to the experimental results in Table 5.4. From Table 5.4, we examine results over all learners, for particular labelling methods, with respect to our third measure of success defined in Section 5.5.2.

Specifically, from Table 5.4, for the string similarity labels, we observe that there are 10 cases where learned models have met our third measure of success. The numerically highest ROUGE-2 result is 9.96 for a ridge regression model trained using KL divergence labels. Further, we note that the JSD labels met our third measure of success using 5 different learners. For the sentence retrieval labels, there are 11 cases where learned models have met our third measure of success. The numerically highest ROUGE-2 result is 10.25 for the SVR-RBF model trained on DFIZ labels. Further, we note that sentence retrieval labels are most effective when using the SVR (RBF) and LambdaMART learners. For the ROUGE recall labels, there are 4 cases where learned models have met our third measure of success. The numerically highest ROUGE-2 result is 9.63 for linear regression models (OLS and Ridge) trained on ROUGE-4 recall labels. Further, we note that training on ROUGE recall labels is generally the least effective labelling method, despite its widespread use in the summarisation literature as a training data labelling method. Finally, for the ROGUE precision labels, there are 19 cases where learned models have met our third measure of success. The numerically highest ROUGE-2 result is 10.11 for a linear SVR model trained on ROUGE-4 precision labels. Further, we note that ROUGE precision labels appear to be effective across a range of learners, and observe that training on ROUGE precision labels is more effective than training on ROUGE recall labels. As illustrated in Figure 5.5, we argue that this is due to the high correlation with sentence length exhibited by the ROUGE recall labelling method.

From the results in Table 5.4, we can now answer Research Question 5.2. We conclude that Jensen-Shannon labels and ROUGE precision labels are the most effective labelling tech-

niques. Specifically, such methods are the most consistent labelling function across different types of learner. Further, we conclude that our proposed sentence retrieval labels (Section 5.3.2) can be used to train effective supervised summarisation models, specifically when using the SVR (RBF) and LambdaMART learners.

**Research Question 5.3**

We now address Research Question 5.3, where we seek to identify particular linear or non-linear regression techniques that are effective for training supervised summarisation models. In our experiments, we investigate six machine learning techniques: three linear models; and three non-linear models. The OLS model is arguably the simplest form of linear regression, where ridge regression adds a regularisation hyper-parameter. While a linear SVM is more complex than OLS and Ridge, the linear SVM is less complex than a non-linear SVM (with a Gaussian kernel). GBRT and LambdaMART are further examples of more complex learners. When evaluating machine learning techniques for a particular task within a problem domain, model complexity issues (i.e. training time, risk of overfitting, and interpretability) are balanced with model effectiveness (i.e. predictive ability). Ideally, learned models are both simple and predictive – known as the bias-variance trade-off (Hastie et al., 2009; Witten et al., 2016).

To answer Research Question 5.3, we again refer to the experimental results in Table 5.4, and introduce Table 5.5. Based on our measures of success for supervised summarisation models, defined in Section 5.5.2, we identify 16 learned models in Table 5.4 that significantly outperform the baseline features that the models are trained on (shown using the † symbol), where learned models produce summaries that exhibit state-of-the-art effectiveness (shown using bold annotation), and where such empirical observations are observed for both the ROUGE-1 recall and ROUGE-2 recall evaluation metrics simultaneously. In Table 5.5, we report these 16 (arguably) most effective runs, reporting statistical significance with respect to state-of-the-art summarisation systems (c.f. Table 5.3). In Table 5.5, the state-of-the-art systems are ordered left-to-right by their ROUGE-2 recall effectiveness. Statistical significance test are computed using the paired Student's t-test, with a 95% confidence level. In answering Research Question 5.3, using Table 5.5, we observe model performance with respect to the state-of-the-art to derive conclusions as to which models are most effective.

**Table 5.5:** Statistical significance tests, over DUC 2004 Task 2, for the most effective supervised regression models, with respect to state-of-the-art summarisation systems. We report $p$-values to 2 s.f. using the paired Student's t-test (95% confidence level). All of the most effective regression models shown below exhibit (at least) no significant difference to the state-of-the-art. Further, as indicated using the ✔symbol, some learned models are significantly more effective than certain state-of-the-art systems under certain ROUGE metrics.

| Learner | Labels | CLASSY 04 | | CLASSY 11 | | Submodular | | DPP | | OCCAMS_V | | RegSum | | ICSISumm | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | R-1 | R-2 | R-1 | R-2 | R-1 | R-2 | R-1 | R-2 | R-1 | R-2 | R-1 | R-2 | R-1 | R-2 |
| OLS | R1 Precision | ✔ | 0.19 | ✔ | 0.24 | 0.72 | 0.41 | 0.07 | 0.99 | 0.33 | 0.69 | 0.36 | 0.71 | 0.27 | 0.63 |
| OLS | R2 Precision | ✔ | 0.18 | ✔ | 0.22 | 0.62 | 0.35 | 0.08 | 0.99 | 0.35 | 0.70 | 0.38 | 0.73 | 0.26 | 0.63 |
| OLS | R3 Precision | ✔ | 0.11 | ✔ | 0.11 | 0.69 | 0.19 | 0.34 | 0.73 | 0.08 | 0.96 | 0.11 | 0.96 | 0.06 | 0.85 |
| OLS | R4 Precision | ✔ | 0.12 | ✔ | 0.14 | 0.53 | 0.24 | 0.44 | 0.81 | 0.10 | 0.88 | 0.08 | 0.88 | 0.06 | 0.78 |
| Ridge | KLDiv | ✔ | ✔ | ✔ | ✔ | 0.90 | 0.06 | 0.13 | 0.31 | 0.23 | 0.58 | 0.26 | 0.63 | 0.19 | 0.72 |
| Ridge | JSDiv | ✔ | 0.06 | ✔ | 0.07 | 0.68 | 0.09 | 0.10 | 0.43 | 0.31 | 0.71 | 0.37 | 0.76 | 0.25 | 0.84 |
| Ridge | R1 Precision | ✔ | 0.19 | ✔ | 0.24 | 0.72 | 0.41 | 0.07 | 0.99 | 0.33 | 0.69 | 0.36 | 0.71 | 0.27 | 0.63 |
| Ridge | R2 Precision | ✔ | 0.18 | ✔ | 0.22 | 0.62 | 0.35 | 0.08 | 0.99 | 0.35 | 0.70 | 0.38 | 0.73 | 0.26 | 0.63 |
| Ridge | R4 Precision | ✔ | 0.08 | ✔ | 0.07 | 0.92 | 0.14 | 0.17 | 0.49 | 0.13 | 0.77 | 0.24 | 0.81 | 0.13 | 0.91 |
| SVR | R3 Precision | ✔ | 0.08 | ✔ | 0.11 | 0.83 | 0.15 | 0.13 | 0.54 | 0.29 | 0.85 | 0.38 | 0.88 | 0.25 | 0.98 |
| SVR | R4 Precision | ✔ | ✔ | ✔ | ✔ | 0.16 | ✔ | 0.81 | 0.15 | ✔ | 0.28 | ✔ | 0.40 | ✔ | 0.44 |
| RBF | Hiemstra_LM | ✔ | 0.08 | ✔ | 0.26 | 0.96 | 0.33 | 0.27 | 0.79 | 0.25 | 0.88 | 0.26 | 0.87 | 0.18 | 0.80 |
| RBF | DFIZ | ✔ | ✔ | ✔ | ✔ | 0.69 | ✔ | 0.35 | ✔ | 0.10 | 0.19 | 0.16 | 0.23 | 0.10 | 0.33 |
| RBF | R1 Precision | ✔ | 0.07 | ✔ | 0.11 | 0.96 | 0.15 | 0.13 | 0.50 | 0.14 | 0.83 | 0.17 | 0.85 | 0.14 | 0.96 |
| GBRT | R3 Precision | ✔ | ✔ | ✔ | ✔ | 0.92 | ✔ | 0.15 | 0.22 | 0.25 | 0.39 | 0.26 | 0.43 | 0.22 | 0.51 |
| GBRT | R4 Precision | ✔ | 0.20 | ✔ | 0.46 | 0.82 | 0.63 | 0.27 | 0.77 | 0.16 | 0.54 | 0.14 | 0.48 | 0.13 | 0.47 |

All 16 of our most effective regression models shown in Table 5.5 exhibit (at least) no significant difference to the state-of-the-art summarisation systems under ROUGE-1 and ROGUE-2. Additionally, as shown using the ✔symbol, there are several cases where learned models significantly outperform specific state-of-the-art systems under certain ROUGE metrics. For example, under the ROUGE-1 recall metric, all 16 models significantly outperform CLASSY04 and CLASSY11. Also under the ROUGE-1 recall metric, we observe that our linear SVR model, when trained on ROUGE-4 precision labels, outperforms OCCAMS_V, RegSum, and ICSISumm. Specifically, the SVR (R4 Precision) model outperforms the ROUGE-1 recall effectiveness of the three most effective state-of-the-art summarisation systems (as determined by ROUGE-2 recall scores).

Further, under the target evaluation metric of ROUGE-2 recall, our Ridge (KLDiv), SVR (R4 Precision), RBF (DFIZ), and GBRT (R3 Precision) models significantly outperform both CLASSY04 and CLASSY11. Furthermore, SVR (R4 Precision), RBF (DFIZ), and GBRT

(R3 Precision) significantly outperform Submodular. Moreover, RBF (DFIZ) significantly outperforms DPP, which is the most effective state-of-the-art system that our learned models have significantly outperformed.

The results in Table 5.5 allow us to answer Research Question 5.3. We conclude that the most effective model type is the linear SVR, when trained using ROUGE-4 precision labels, as this model significantly outperforms the effectiveness of more state-of-the-art systems than any other combinations of learner and labels shown in Table 5.5. Having answered our research questions, we now discuss and analyse our empirical results.

### 5.5.4 Discussion & Analysis

#### Feature Importance

Given the set of summarisation features we have defined in Section 5.4, and considering the evidence in Table 5.4 that the combination of such features is effective (i.e. discriminative), we now analyse the features with respect to their usefulness (i.e. importance) in supervised summarisation models. In particular, we seek to understand if any particular single feature is not contributing to the learned model. For example, given we have established that certain features are strongly correlated (c.f. Table 5.2), potentially redundant features could be removed. Further, we seek to understand the role of the lexical features (Position and Length), i.e. if they are important features in comparison to the baseline summarisation features.

Our analysis is conducted using the Gradient Boosting Regression Tree (GBRT) model, using XGBoost (Chen and Guestrin, 2016). In Figure 5.8, we show a GBRT feature importance plot. Feature importance is shown on the $x$-axis, with individual features shown on the $y$-axis – showing the five baseline summarisation features and two lexical features (Position and Length). The model is trained on ROUGE-1 precision labels, using the DUC 2001–2002 training data. The importance score is the frequency of occurrence of that feature over the boosted decision trees within the model, i.e. the number of times that the feature contributes to the branches of the decision trees within the model.

From Figure 5.8, we observe that all features achieve some degree of importance. Specifically, from this analysis, no feature can be interpreted as markedly unimportant, with all features frequently being used in construction of the model. We note that the two lexical features, Position and Length, are contributing to the model learned by the GBRT learner. This

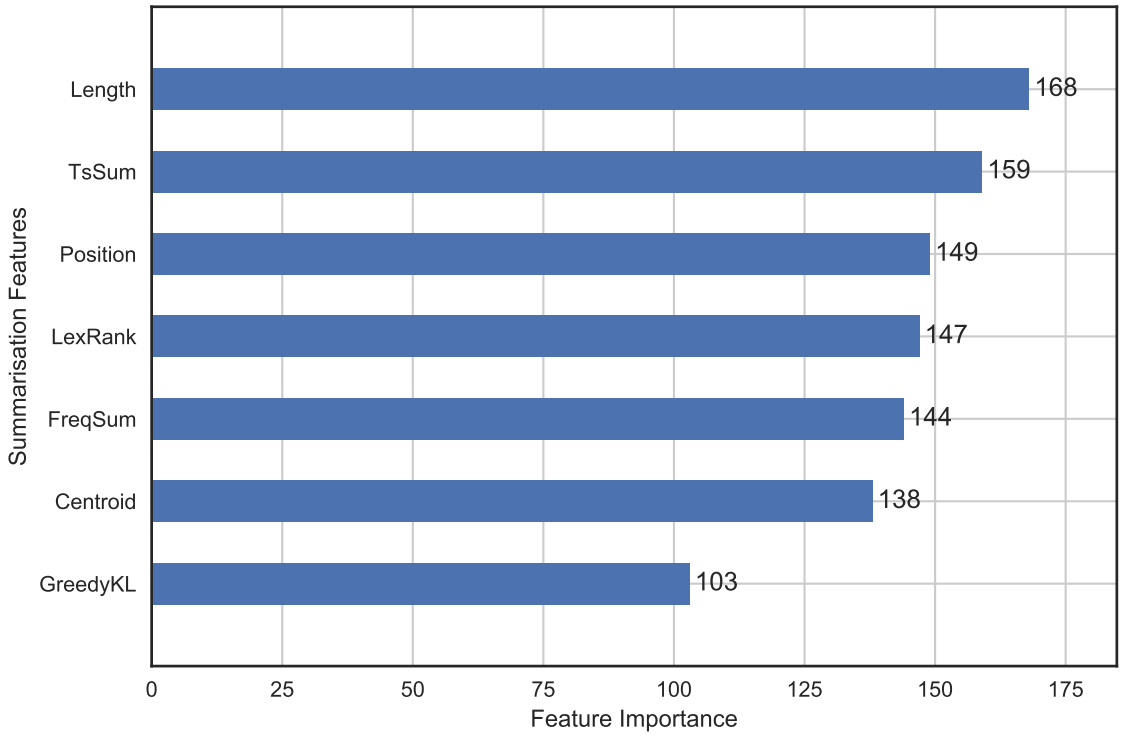



**Figure 5.8:** Feature importance plot, under the Gradient Boosted Regression Trees (GBRT) model, trained on ROUGE-1 precision labels, showing 5 summarisation features, and 2 lexical features (Position and Length).

justifies the inclusion of the two lexical features into the model (in addition to the baseline summarisation features). From Figure 5.8, we conclude that all features that we have defined (in Section 5.4) should be taken forward to our experiments in Chapter 6 and 7.

### Model Fit and ROUGE Effectiveness

We now examine the relationship between model fit and summarisation effectiveness. Specifically, we seek to understand if there are any correlations between RMSE and ROUGE scores, i.e. if there exits a correlation between a supervised machine learned model's prediction error and the summarisation effectiveness of that learned model.

As described in our experimental setup (Section 5.5.2), learned models are trained to perform a task, ranking candidate summary sentences. From this ranking, an anti-redundancy filtering component is then used to select specific summary sentences, with the aim of selecting sentences that are textually dis-similar (i.e. non-redundant). This second stage (the anti-redundancy filtering) is unsupervised. In particular, the learned model is used only in the first stage to generate a ranking of sentences.

**Table 5.6:** Pearson's $\rho$ correlation coefficients between supervised model prediction error (RMSE) and ROUGE summarisation evaluation scores. Negative correlation indicates that as ROUGE scores increase, model error decreases, while positive correlation indicates that as ROUGE scores increase, model error also increases.

| Model | ROUGE-1 | ROUGE-2 |
|---|---|---|
| *OLS* | -0.32 | -0.46 |
| *SVR (RBF)* | 0.52 | 0.44 |

Based on the results in Table 5.4, for various learned models trained on different labels, we can analyse the correlation of ROUGE summarisation effectiveness and model prediction error (i.e. RMSE). Table 5.6, provides such an analysis for the OLS model and the SVR (RBF) model. We compute Pearson's $\rho$ correlation coefficients between model prediction error (RMSE) and ROUGE scores. From Table 5.6, we can observe that the OLS model exhibits correlation between ROUGE performance and RMSE, and further, that the SVR (RBF) model also exhibits correlation between ROUGE performance and RMSE.

Specifically, the OLS model exhibits negative correlation, while the SVR (RBF) model exhibits positive correlation. Negative correlation for the OLS model indicates that when ROUGE scores increase (where higher is better) model error decreases (where lower is better). However, for the SVR (RBF), positive correlation indicates that when ROUGE scores increase, model error also increases. Hence, from the analysis in Table 5.6, we can conclude that a learned model's prediction error (i.e. RMSE) is not necessarily an accurate indication of whether that model will produce effective summaries. We postulate that the anti-redundancy filtering component (which is activated after the application of the learned model) is a confounding variable in our experiments.

### Linear vs. Non-linear Learners

We next consider the characteristics of the interactions between features and labels, which may be linear or non-linear in nature. We first examine the Pearson's $\rho$ correlation coefficients between summarisation features and training data labels. Pearson's correlation is a measure of the strength of the linear relationship between two variables (Rice, 2006). Table 5.7 provides the Pearson's $\rho$ correlation coefficients between the summarisation features and labels we investigate in our experiments. Pearson correlation coefficients are qualita-

**Table 5.7:** Pearson's $\rho$ correlation coefficients between summarisation features and training data labels, computed over the training dataset, providing a quantification of the strength of the linear relationship between features and labels. Pearson correlation coefficients are qualitatively interpreted following Rosenthal (1996): $\rho > .10$ weak; $\rho > .30$ moderate; $\rho > .50$ strong (shown in **bold**); and $\rho > .70$ very strong (**bold** & underline).

| Pearson's $\rho$ | String Metrics Labels | | | Sentence Retrieval Labels | | | | | | | ROUGE-$n$ Metrics Labels | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Cos | KL | JS | TF | BM25 | H–LM | DFR | DFR–K | DFIC | DFIZ | R1R | R2R | R3R | R4R | R1P | R2P | R3P | R4P |
| Position | -0.17 | 0.25 | 0.25 | -0.11 | -0.12 | -0.10 | -0.13 | -0.11 | -0.09 | -0.11 | -0.14 | -0.11 | -0.07 | -0.05 | -0.10 | -0.08 | -0.05 | -0.04 |
| Length | 0.35 | -0.10 | -0.30 | 0.33 | 0.46 | 0.24 | 0.45 | 0.32 | 0.26 | 0.33 | **0.78** | 0.39 | 0.18 | 0.09 | 0.02 | 0.05 | 0.04 | 0.03 |
| FreqSum | 0.35 | -0.24 | **-0.50** | 0.38 | 0.22 | 0.30 | 0.38 | 0.32 | 0.33 | 0.39 | 0.21 | 0.26 | 0.15 | 0.08 | 0.43 | 0.34 | 0.16 | 0.09 |
| TsSum | 0.44 | -0.18 | -0.44 | 0.43 | 0.35 | 0.35 | 0.41 | 0.39 | 0.40 | 0.43 | 0.27 | 0.32 | 0.20 | 0.11 | 0.43 | 0.36 | 0.21 | 0.12 |
| Centroid | **0.62** | -0.30 | **-0.64** | **0.59** | **0.55** | 0.45 | **0.65** | **0.54** | 0.49 | **0.58** | **0.62** | 0.49 | 0.28 | 0.16 | 0.41 | 0.35 | 0.21 | 0.13 |
| LexRank | 0.43 | -0.48 | **-0.58** | 0.33 | 0.30 | 0.27 | 0.37 | 0.30 | 0.27 | 0.31 | 0.35 | 0.30 | 0.18 | 0.12 | 0.24 | 0.22 | 0.14 | 0.10 |
| GreedyKL | **-0.60** | 0.31 | **0.66** | **-0.58** | **-0.56** | -0.44 | **-0.67** | **-0.53** | -0.47 | **-0.57** | **-0.71** | **-0.52** | -0.28 | -0.16 | -0.36 | -0.32 | -0.19 | -0.12 |

tively interpreted following Rosenthal (1996): $\rho > .10$ weak; $\rho > .30$ moderate; $\rho > .50$ strong (shown in **bold**); and $\rho > .70$ very strong (**bold** & underline). From Table 5.7, we observe that, out of the 126 feature and label combinations, there are 2 cases of very strong correlation, 18 cases of strong correlation, 49 cases of moderate correlation, and 57 cases of weak (or less) correlation. In summary, 16% of feature and label combinations exhibit a strong linear relationship, but 84% do not exhibit strong linear relationships.

We continue our analysis of the relationship between features and labels in Figure 5.9, where we visualise the relationship between features and labels using partial dependence plots (Hastie et al., 2009). Partial dependence plots illustrate the nature of the dependence (i.e. linear or non-linear) of the labels and features within the learned function, $\boldsymbol{y} = f(\boldsymbol{X})$. Partial dependence plots are computed from a GBRT model, which we train using ROUGE-1 precision labels, ove the DUC 2001–2002 training data. In Figure 5.9, we generate a separate plot for each of the summarisation features. The $x$-axis shows the score range for that feature, with the deciles of the input variables marked along the range. The $y$-axis shows the partial dependence score. The partial dependence score is computed based on each feature, but also marginalises over all other features by holding values of other features at their mean value. As such, partial dependence plots are not a visualisation of single features in isolation, but visualises the relationship of $\boldsymbol{X}_i$ and $\boldsymbol{y}$ after averaging the effects on $\boldsymbol{y}$ of other features in $\boldsymbol{X}$.

From Figure 5.9, we again observe evidence of both linear and non-linear interactions between features and labels. For example, the Length, FreqSum, TsSum, and GreedyKL features exhibit non-linearity with respect to the ROUGE-1 precision labels. We interpret the

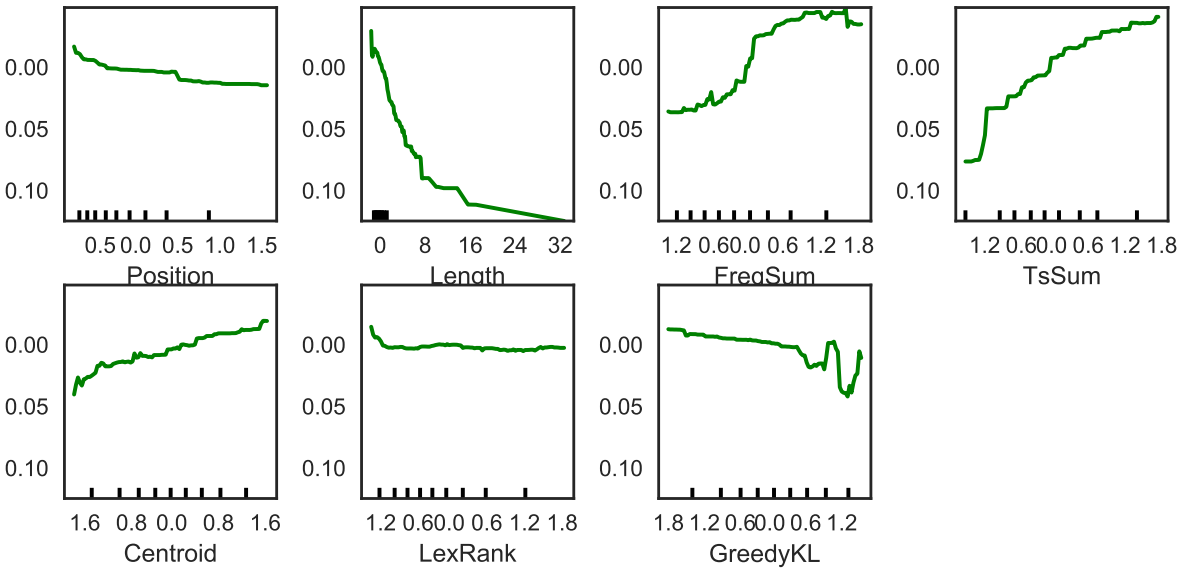



**Figure 5.9:** Partial dependence plots, under the Gradient Boosted Regression Trees (GBRT) model, trained on ROUGE-1 precision, showing the (linear or non-linear) interaction between the features and the training labels.

Position, Centroid, and LexRank features are exhibiting (broadly) linear interactions with the ROUGE-1 precision labels. In summary, From Figure 5.9 we observe that there exist cases of linear and non-linear interactions between features and labels (under the GBRT model).

In conclusion, the evidence from Table 5.7 and Figure 5.9 has demonstrated that there exists both linear and non-linear interactions between features and labels in our training dataset.

## 5.6 Chapter Summary

In this chapter, we investigated supervised machine learned summarisation models. We provided experimental results to empirically validate Hypothesis 3 from our Thesis Statement (Section 1.2). We validated our claim that supervised machine learned summarisation models based on regression techniques, that exhibit state-of-the-art effectiveness, can be trained on discriminative features, derived from standard multi-document newswire summarisation algorithms, using automatically labelled training data, induced from gold-standard summary text(s). By answering Research Question 5.1, we demonstrated that baseline newswire summarisation algorithms be used to provide a set of discriminative features for training effective supervised summarisation models. By answering Research Question 5.2, we demonstrated which sentence scoring functions can be used to provide high-quality per-sentence labels for training effective supervised summarisation models. By answering Research Question 5.3,

we demonstrated that linear and non-linear regression-based machine learning techniques are effective for learning to predict candidate summary sentence scores.

In conclusion, we have identified several combinations of features, labels, and learners, that achieve state-of-the-art effectiveness for the task of multi-document newswire summarisation, over the DUC 2004 dataset. Such learned summarisation models are taken forward to our experiments in the next two chapters, where we investigate the addition of entity-based evidence into the learned models. Further, labelling sentences based on divergence methods (i.e. JSD and KLD) is effective for training regularised linear regression models. Furthermore, labelling sentences based on sentence retrieval methods is effective for training non-linear regression models. Moreover, labelling sentences using ROUGE-*n* precision is effective for training linear and non-linear regression models. Additionally, learning-to-rank techniques are also effective for training supervised machine learning summarisation models.

# Chapter 6

# Retrospective Event Summarisation

In this chapter, we address our fourth challenge, regarding the use of evidence about the named entities (Nadeau and Sekine, 2007) involved in news events to effectively summarise such news events. In particular, given a set of news documents that discuss an event, we investigate how to effectively model such an event using statistics about the named entities mentioned within the news articles. Specifically, proposing and evaluating a series of entity-focused event summarisation features, we define summarisation features that estimate entity importance and entity–entity interaction, which explicitly model the importance of entities and how they interact. The effectiveness of our proposed entity-focused event summarisation features are evaluated within a supervised framework (Hastie et al., 2009; Witten et al., 2016).

In our Thesis Statement (Section 1.2), we hypothesise that by learning a ranking function over newswire sentences, optimising for the importance of entities within the event, and the significance of interactions between entities within the event, the sentences that are available for inclusion into the event summary can be effectively ranked by their summary worthiness, using a supervised summarisation model trained using such entity-focused event summarisation features, augmented with document summarisation features. Hypothesis 4 is investigated in this chapter, for the task of retrospective event summarisation, within the multi-document newswire summarisation scenario. Hypothesis 4 is further examined in Chapter 7, within the context of the temporal summarisation task (Aslam et al., 2013, 2014, 2015), where we additionally introduce temporal entity-focused features, and entity–event relevance features – specifically addressing the temporal and query-biased nature of the task.

### Chapter Outline

This chapter is organised as follows:

- Section 6.1 briefly introduces the named entity recognition and classification task.
- Section 6.2 defines the entity-focused event summarisation features that we investigate in this chapter, specifically: entity-importance, and entity–entity interaction.
- Section 6.3 presents an empirical evaluation of our proposed entity-focused event summarisation features, within a supervised machine learned framework.

## 6.1 Named Entities

In this section, we introduce the named entity recognition and classification task, and state the entity tagging systems that we use in this thesis to identify named entities within news articles. A named entity is by definition the referent of an entity, i.e. the name for a specific real-world object (such as persons, organisations, or locations). The task of named entity recognition and classification has been defined and examined in evaluation workshops such as the Message Understanding Conferences (Grishman and Sundheim, 1996), and the CoNLL 2003 Shared Task (Sang and Meulder, 2003). Indeed, state-of-the-art supervised entity tagging systems are often trained on annotated data from such workshops (Nadeau and Sekine, 2007).

Named entity recognition and classification involves automatically processing natural language text to identify spans of text strings (i.e. surface mentions) of named entities. Further, software tools that perform named entity recognition (e.g. Finkel et al., 2005; Hoffart et al., 2011; Milne and Witten, 2013) typically provide an annotation for each recognised entity. Such annotations can be sparse types, such as: `<PERSON>`; `<ORGANIZATION>`; or `<LOCATION>`. This is the typical output produced by a named entity recognition (NER) system. Alternatively, the annotation can be an identifier to a richer representation for the named entity, such as a Wikipedia[1] article about the entity, or a link to an entry in a knowledge base (Färber et al., 2015), for instance DBPedia[2] (Lehmann et al., 2015) or Wikidata[3] (Vrandečić and Krötzsch, 2014). This is the typical output from a named entity linking (NEL) system. To illustrate the difference, given the named entity "Donald Trump", a NER system may

[1] wikipedia.org
[2] dbpedia.org
[3] wikidata.org

output: "Donald Trump" $\mapsto$ <PERSON>, whereas the output from an NEL system may provide additional contextual information: "Donald Trump" $\mapsto$ wikipedia.org/wiki/Donald_Trump.

The challenges of named entity recognition and linking arise due the ambiguity of natural language text (Nadeau and Sekine, 2007). For example, there may be multiple text string expressions referring to the same entity in any given text. Specifically, the surface mentions "Trump", "The President", and "POTUS", might all refer to the specific named entity "Donald Trump" (i.e. the 45$^{th}$ President of the United States of America). The ambiguity of identifying entity mentions is compounded by unresolved anaphora such as "he" or "him". As such, the accuracy of NER and NEL systems is a concern with respect to the purpose for which they are used. While imperfect, particularly in the genre of social-media (Rizzo et al., 2017), it has been shown that the effectiveness of entity recognition systems over newswire is generally at an acceptable level (Augenstein et al., 2017), i.e. of sufficient accuracy to be utilised in down-stream language processing tasks such as summarisation.

For our experiments in this thesis, we use state-of-the-art entity tagging toolkits that have been demonstrated as being effective within the newswire domain. Specifically, we tag mentions of named entities using the following toolkits: Stanford CoreNLP (Manning et al., 2014) NER (Finkel et al., 2005); Wikipedia Miner NEL (Milne and Witten, 2013); and AIDA NEL (Hoffart et al., 2011). As such, we explore the definition of what an entity could be interpreted as, in terms of computational processes over natural language text. Specifically, using Named Entity Recognition (NER), we obtain a strict interpretation (person, organisation, location). Using Named Entity Linking (NEL), we obtain a loose entity definition (pages within Wikipedia, or entries in a knowledge base).

Within the context of the event summarisation task, we hypothesise that news events are primarily about entities, and to effectively summarise an event we should explicitly account for the named entities involved in the event. To motivate our approach, consider the sample of sentences in Figure 6.1, which is an excerpt from documents within the DUC 2004 dataset. In Figure 6.1, we observe that entities (and concepts generally) are an important characteristic of textual representations of events. Given such prominence of entities within events, we investigate the use of entity-focused event summarisation features to derive effective summaries of events. Our proposed entity-focused event summarisation features are defined in Section 6.2.

[Cuban President] [Fidel Castro] said {Sunday} he disagreed with the {arrest} in [London] of former [Chilean dictator] [Augusto Pinochet], calling it a {case} of "{international} meddling." ... [Pinochet], 82, was placed under {arrest} in [London] {Friday} by [British police] acting on a {warrant} issued by a [Spanish judge] ... The [judge] is probing [Pinochet's] role in the {death} of [Spaniards] in [Chile] under his {rule} in the {1970s} and {80s} ... The [Chilean government] has protested [Pinochet's] {arrest}, insisting that as a {senator} he was travelling on a {diplomatic passport} and had {immunity} from {arrest} ...

**Figure 6.1:** Example sentences from DUC 2004 topic 'd30003t". Named entities (PER, ORG, LOC) are annotated using [brackets], while more general concepts are annotated using {braces}.

## 6.2 Entity-focused Event Summarisation Features

In this section, we describe our proposed entity-focused event summarisation features. Such features are used in a supervised machine learned summarisation framework, to score sentences for inclusion into extractive summaries of news events. In particular, we evaluate two groups of entity-focused event summarisation features, namely: entity importance; and entity–entity interaction. Entity importance is estimated using entity frequency, and entity–entity interaction is estimated via entity co-occurrence. Both features are based on computing statistics over the surface mentions of such entities within the documents being summarised.

Our proposed entity-focused event summarisation features attempt to capture semantic information regarding the nature of an event, i.e. what entities are important, and how entities interact with other entities. Such features are used within an event summarisation algorithm, to score sentences for inclusion into the summary of an event. The intuition is that an effective event summary should provide information about the important entities, and also, provide information about the important interactions between entities. We now describe each feature.

### 6.2.1 Entity Importance

The importance of term frequency is well understood in Information Retrieval (Croft et al., 2010; Büttcher et al., 2010). Term frequency provides an indication of how important a term is within a document. Analogous to this, we hypothesise that the frequency of entities, within a collection of documents about an event, will provide a strong signal indicating what entities are important within the event.

The intuition is that the frequency of mentions of an entity within a set of documents about an event can be used to estimate the importance of such entities within the event – i.e. what entities the event is probably about. For example, if entities $A$, $B$, and $C$ are the most frequently occurring entities within a given set of newswire articles, then we may reasonably infer the event is probably about entities $A$, $B$, and $C$.

The frequency of entities is established via named entity recognition over the input documents, counting surface mentions of each entity. To estimate entity importance, we measure entity collection frequency. Specifically, given a document collection, $C$, of $n$ documents, $C = (d_1, d_2, ..., d_n)$, we establish the set of entities, $E$, that are present in $C$. For each entity, $e_i \in E$, we estimate entity importance using the collection frequency, $cf$, of $e_i$ within $C$, i.e. the total number of times an entity occurs over the input documents being summarised.

Thus, the importance of a given entity, $e_i$, is estimated as:

$$\text{EntityImportance}(e_i) = \sum_{d \in C} \text{cf}(e_i, d) \tag{6.1}$$

### 6.2.2 Entity–entity Interaction

Co-occurrence, as with term frequency, is also known to be a useful feature within formal models of Information Retrieval (Metzler and Croft, 2005). Further, the sequential co-occurrence of single terms (i.e. bi-grams) has been demonstrated to be an effective feature for summarisation (Gillick and Favre, 2009). Similar to this, we hypothesis that the co-occurrence of entities, within documents about an event, will provide a useful indication of the interaction among groups of entities involved in an event.

The intuition is that the frequency of sentence-level surface mentions of pairs of entities, within a set of documents about an event, can be used to estimate the significance of interactions among the entities. For example, if we observe that entity pairs $(A, B)$ and $(B, C)$ frequently co-occur at the sentence-level over the input sentences, we may then infer that the interaction between entity pairs $(A, B)$ and $(B, C)$ is significant within the event.

To estimate entity–entity interaction, we measure sentence-level entity co-occurrence. Specifically, we define a graph, $G = (V, E)$, where the vertices are entities, $V = \{e_1, e_2, ..., e_i\}$, and pairs of entities, $(e_i, e_j)$, make up the set of edges, $E = \{(e_i, e_j), (e_k, e_l), ..., (e_m, e_n)\}$. Graph edges are un-directed, but weighted. Edge weight represents the frequency of sentence-level entity co-occurrence.

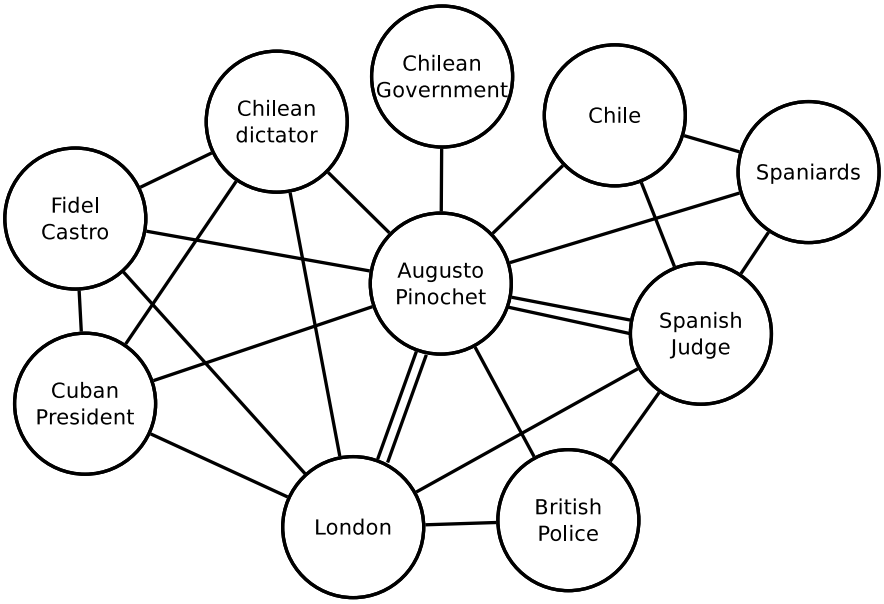



**Figure 6.2:** An example entity–entity interaction graph, based on the sentences in Figure 6.1, used to determine significant interactions among entities. Entities are nodes, and edges represent frequency of co-occurrence.

Thus, the interaction between a given pair of entities, $(e_i, e_j)$, is estimated as the weight on the connecting edge in a graph:

$$\text{EntityInteraction}(e_i, e_j) = G_{\{e_i, e_j\}_{weight}} \tag{6.2}$$

Figure 6.2 provides an illustration of an entity–entity interaction graph. Given the sample of sentences in Figure 6.1, we construct a graph, $G = (V, E)$, where the vertices are entities, $V = \{e_1, e_2, ..., e_i\}$, and pairs of entities, $(e_i, e_j)$, make up the set of edges, $E = \{(e_i, e_j), (e_k, e_l), ..., (e_m, e_n)\}$. As evident from Figure 6.2, we can observe that "Augusto Pinochet" has a co-occurrence frequency of 2 with "London" and "Spanish judge". This is shown using double bars, where single bars indicate co-occurrence frequency of 1. The Chilean dictator was indeed arrested in London at the instigation of a Spanish Judge, and this can be thought of as the initial trigger event in this topic[1]. A summary of an event would ideally contain this information. By using estimates of entity–entity interaction, obtained via a entity co-occurrence graph, we may identify interesting interactions, and importantly, attempt to score sentences for selection into the summary of an event based on this interaction.

[1] wikipedia.org/wiki/Indictment_and_arrest_of_Augusto_Pinochet

### 6.2.3 Sentence Scoring

The entity-focused event summarisation features we have defined, entity importance (Equation 6.1) and entity–entity interaction (Equation 6.2), are used in a supervised machine learned summarisation model. In particular, the learned model is trained to score sentences for potential inclusion into an event summary. Specifically, a sentence ranking is established by scoring sentences via learned model application. Then, as commonly observed in the summarisation literature (Hong et al., 2014), a cosine similarity anti-redundancy component is applied to promote novelty in the set of sentences that are included in the final summary text.

As such, the above event summarisation features, used to estimate entity importance and interaction, are defined as summations over the entities contained in the input sentences. Given a set of candidate summary sentences, $S = (s_1, s_2, ...s_i)$, where each sentence contains a set of entities, $s_i = (e_1, e_2, ..., e_n)$, our per-sentence scoring functions are defined as:

$$\text{score}(s_i) = \sum_{e \in s_i} \text{EntityImportance}(e) \qquad \text{(c.f. Eqn 6.1)} \tag{6.3}$$

$$\text{score}(s_i) = \sum_{(e_i, e_j) \in s_i} \text{EntityInteraction}(e_i, e_j) \qquad \text{(c.f. Eqn 6.2)} \tag{6.4}$$

In our later experiments, for scoring sentences, Equation 6.3 and Equation 6.4 form the two core features under evaluation in this chapter. Further, we also investigate the use of a logarithmic variant of the entity features, a variant that employs sentence length normalisation, and a final logarithmic and length normalised variant. Similar to the $tf$ saturation effect in BM25 (Robertson et al., 2009), the logarithmic variant is intended to minimise the dominance of very frequently occurring entities. The intuition is that we may not want the event summary sentence selection to be highly-biased towards such high-frequency entities. Logarithmic variants are derived by taking the $\log_2$ of entity importance and entity–entity interaction scores. The length normalisation variant is intended to balance the event summary sentence selection among sentences of different lengths. Here, we postulate that effective summaries may not be produced by always selecting long sentences (with many entities). Length normalisation is by the number of tokens in a sentence. Table 6.1 summarises the set of entity-focused event summarisation features we evaluate in our experiments in Section 6.3.

**Table 6.1:** Given a set of candidate summary sentences, $S = (s_1, s_2, ...s_i)$, where each sentence contains a set of entities, $s_i = (e_1, e_2, ..., e_n)$, we define the following per-sentence scoring functions (variants of Eqn 6.3 and 6.4).

| **Variant** | **Entity Importance** (Eimp) | **Entity–entity Interaction** (EEint) |
|---|---|---|
| Raw scores | $\text{score}(s_i) = \sum_{e \in s_i} \text{Eimp}(e)$ | $\text{score}(s_i) = \sum_{(e_i,e_j) \in s_i} \text{EEint}(e_i, e_j)$ |
| $\log_2$ scores | $\text{score}(s_i) = \sum_{e \in s_i} \log_2(\text{Eimp}(e))$ | $\text{score}(s_i) = \sum_{(e_i,e_j) \in s_i} \log_2(\text{EEint}(e_i, e_j))$ |
| Length normalised | $\text{score}(s_i) = \frac{\sum_{e \in s_i} \text{Eimp}(e)}{\text{words} \in s_i}$ | $\text{score}(s_i) = \frac{\sum_{(e_i,e_j) \in s_i} \text{EEint}(e_i,e_j)}{\text{words} \in s_i}$ |
| $\log_2$ and Len. norm. | $\text{score}(s_i) = \frac{\sum_{e \in s_i} \log_2(\text{Eimp}(e))}{\text{words} \in s_i}$ | $\text{score}(s_i) = \frac{\sum_{(e_i,e_j) \in s_i} \log_2(\text{EEint}(e_i,e_j))}{\text{words} \in s_i}$ |

## 6.3 Evaluation

In this section, we conduct an empirical evaluation of supervised machine learned summarisation models, trained on our proposed entity-focused event summarisation features. We evaluate learned models that have been trained using different sets of features, conducting a group-wise feature ablation study. We begin by stating our research question, then describe our experimental setup. Results are provided over the DUC 2004 and TAC 2008 datasets, for the task of generic extractive multi-document newswire summarisation. Finally, we discuss and analyse our empirical observations.

### 6.3.1 Research Questions

In our Thesis Statement (Section 1.2), we formed Hypothesis 4:

> By learning a ranking function over newswire sentences, optimising for the importance of entities within the event, the significance of interactions between entities within the event, and the topical relevance of entities to the event, we hypothesise that the sentences that are available for inclusion into the event summary can be effectively ranked by their summary worthiness, using a supervised summarisation model trained using such entity-focused event summarisation features, augmented with document summarisation features.

To validate Hypothesis 4, we address the following research question:

**Research Question 6.1.** Within a supervised summarisation framework, does augmenting document summarisation features with entity-focused event summarisation features lead to an increase in supervised summarisation effectiveness?

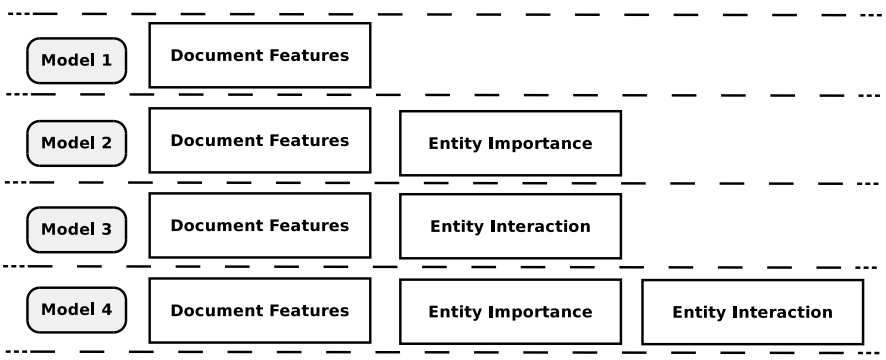



**Figure 6.3:** An illustration of the experimental setup for our feature-group ablation study.

We claim that, given supervised machine learned summarisation models, which have been trained on standard document summarisation features, the summarisation effectiveness of such learned models can be improved by adding entity-focused event summarisation features. In our experiments, we train supervised machine learned summarisation models using regression-based learners. Supervised summarisation provides a principled means to evaluate the combination of newswire summarisation features with entity importance and entity–entity interaction features. Specifically, we seek to ascertain if augmenting document summarisation features, derived from standard multi-document summarisation baselines, with our proposed entity-focused event summarisation features results in an increase in supervised summarisation effectiveness. To answer our research question, we train supervised summarisation models using different sets of features, and perform a ROUGE Lin (2004) evaluation over the DUC 2004 and TAC 2008 newswire summarisation datasets. We validate Hypothesis 4 if any learned models that have been trained using entity-based features outperform any learned models trained using document summarisation features alone.

### 6.3.2 Experimental Setup

Our experimental design is illustrated in Figure 6.3. In particular, supervised machine learned summarisation models are trained using a set of baseline summarisation features. This is shown in Figure 6.3 as "model 1". Then, we train further supervised summarisation models where the baseline features group is augmented with: entity importance features (shown as "model 2" in Figure 6.3); entity–entity interaction features ("model 3" in Figure 6.3); and a combination of importance and interaction features ("model 4" in Figure 6.3). We validate our claim, that entity-focused event summarisation features can be used to derive effective sum-

maries of news events, if any of the models that have been augmented with entity-focused features exhibit summarisation effectiveness that exceeds the learned models trained using only baseline newswire summarisation features. In the remainder of this section, we describe the newswire summarisation datasets used in our experiments, provide details of the specific named entity recognition and classification (NERC) toolkits used, the ROUGE-based summary evaluation process, and the configuration of the supervised machine learned models.

### Processing of the Summarisation Datasets

In our experiments, we summarise news events using newswire documents from the Document Understanding Conference (DUC 2004[1]). Additionally, we report results over the 2008 Text Analysis Conference dataset[2]. Each dataset consists of approx. 50 topics, where a topic contains approx. 10 news articles to be summarised. Each topic is associated with a set of gold-standard reference summaries, authored by human assessors. System-produced summaries are compared to these gold-standard summaries, to evaluate summarisation effectiveness. For each topic within the TAC 2008 dataset, we use the 10 newswire articles from document set A, and the 4 reference summaries for document set A. The update part of the task (set B), and the topic statements, are not used in our experiments, i.e. we use TAC 2008 for non-update generic summarisation.

To determine the entities within the newswire documents, we perform Named Entity Recognition (NER) using the Stanford CoreNLP (Manning et al., 2014) NER toolkit (Finkel et al., 2005), using the 3-class model (tagging Person, Organisation, and Location entity mentions). Named Entity Linking (NEL) is performed using the Wikipedia Miner toolkit (Milne and Witten, 2013), which was trained on the January 2015 dump of the English Wikipedia. The NER and NEL processes are run on the plain text of the newswire documents.

Further, a text processing pipeline is applied to the newswire documents. Specifically, the CoreNLP toolkit is used to split the newswire text into sentences, tokens are normalised (NFD[3]), down-cased, compound words are split, punctuation is removed, and Porter (1980) stemming applied. Further, we perform stopword[4] removal. Sentences from the input doc-

[1] duc.nist.gov/duc2004

[2] tac.nist.gov/2008/summarization

[3] docs.oracle.com/javase/8/docs/api/java/text/Normalizer.html

[4] en.wikipedia.org/wiki/Most_common_words_in_English

uments, for a given topic, are combined into a single virtual document. We use a sentence interleaving technique, constructing the virtual document by taking one sentence at a time from each document in turn. The virtual document (i.e. the interleaved sentences), and the NER/NEL annotations, are provided as the input to the summarisation process.

### Summarisation Evaluation Procedure

We use ROUGE (Lin, 2004) to assess effectiveness of our proposed entity-focused event summarisation features. ROUGE[1] measures the *n*-gram overlap between summaries under evaluation and human authored gold-standard reference summaries. We report ROUGE-1 (uni-gram overlap) and ROUGE-2 (bi-gram overlap) recall, with stopwords retained, stemming applied, and truncating summary texts to 100 words. ROUGE-2 is the target metric, due to the reported agreement of ROUGE-2 with manual evaluation (Owczarzak et al., 2012).

### Supervised Summarisation Configuration

We produce extractive summaries of news events using supervised regression techniques, training learned models using the features and labels previously described in Chapter 5. Specifically, Support Vector Regression (Chang and Lin, 2011) (SVR), and Multiple Additive Regression Trees (Friedman, 2001) (MART). For SVR, we experiment with Linear and RBF kernels. We report results over DUC 2004 and TAC 2008 (the test data), training supervised models on DUC 2002, for a clear train/test separation. Further, we split DUC 2002 into training and validation sets, with a 60/40 train/validation ratio. We label the training data with ROUGE-N partials, which is the score for each sentence (training instance) computed using ROUGE-1 and ROUGE-2 recall and precision, as fully described in Chapter 5. Learned models are trained using five standard document summarisation baselines (Hong et al., 2014) as features, namely: LexRank; Centroid; FreqSum; TsSum; and Greedy–KL; plus sentence position (in the document) and sentence length (7 features in total).

Learned models, trained on entity-focused features are used to score sentences for inclusion into the event summary. This produces a ranking of sentences, with the highest-ranked sentences preferred for inclusion into the summary. The summary is built by selecting the top-*k* sentences in the list, where *k* is the desired summary length. However, simply select-

[1] www.berouge.com

ing the $k$ highest-scoring sentences can lead to redundant summaries, so a cosine similarity threshold is applied. For each candidate summary sentence, iterating down the ranked list of sentences, it is compared to all sentences previously selected for the summary. Only sentences that exhibit sufficient cosine dis-similarity are selected for the summary, based on an anti-redundancy threshold. Sentence selection continues for the desired summary length.

On the DUC 2002 validation split, we learn various experimental parameter settings. Specifically, we learn the cosine similarity threshold value ($[0..1]$), used in the anti-redundancy filtering component. Further, over the DUC 2002 validation data, we learn hyper-parameters for machine learned models, specifically the "C" parameter of SVM-based models. Furthermore, the validation data is also used to learn which particular ROUGE-N partial labels should be used for each model. In particular, the most effective labelling method, used to train the different learned models, is learned on the validation data. In experiments in this chapter, we validate for ROUGE-1 and ROUGE-2 recall and precision labels. Parameter settings learned on the DUC 2002 validation data are then applied to the DUC 2004 and TAC 2008 test data.

### 6.3.3 Experimental Results

#### Research Question 6.1

We now address our research question. Supervised summarisation models are trained using varying feature sets. Baseline supervised summarisation models are trained on features derived from standard document summarisation algorithms. Further supervised summarisation models are trained by combining of such features with our proposed entity-focused event summarisation features (described in Section 6.2). This allows us to investigate whether the combination of standard document summarisation features with entity-focused event summarisation features leads to more effective summaries of news events.

Table 6.2 presents the results of our supervised learning experiments. Table 6.2 reports the ROUGE-1 and ROUGE-2 recall effectiveness, over DUC 2004 and TAC 2008, for learned models trained using various combinations of baseline features and entity-focused features. Entity features are derived using either named entity recognition (NER) or named entity linking (NEL). Per model, the baseline effectiveness is reported first. Then, we augment baseline features with entity importance, entity–entity interaction, and finally a combination of entity importance and entity–entity interaction. Statistical significance (two tailed paired-sample

**Table 6.2:** ROUGE-1 and ROUGE-2 recall effectiveness of entity-focused event summarisation features, evaluated over DUC 2004 and TAC 2008. We report results for supervised machine learned models, trained on ROUGE-N labels. Row-wise (vs. baseline) statistically significant (two tailed paired-sample t-test, 95% significance level) increases/decreases are indicated with ✔✗ (⊖ no significant difference).

| DUC 2004 | Baseline | | Named Entity Recognition | | | | | | Named Entity Linking | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Importance | | Interaction | | Combination | | Importance | | Interaction | | Combination | |
| | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** |
| *SVR Linear* | 37.79 | 9.48 | 38.74⊖ | 10.05⊖ | 38.82✔ | 9.94⊖ | 38.54⊖ | 9.93⊖ | 38.69⊖ | 10.08⊖ | 38.95✔ | 9.93⊖ | 39.07✔ | 10.16✔ |
| *SVR RBF* | 35.05 | 7.34 | 38.81✔ | 9.49✔ | 34.24⊖ | 6.71⊖ | 37.28✔ | 8.94✔ | 34.19⊖ | 7.16⊖ | 37.39✔ | 9.23✔ | 37.96✔ | 9.57✔ |
| *MART* | 36.53 | 8.65 | 37.88✔ | 8.93⊖ | 36.53⊖ | 8.65⊖ | 36.41⊖ | 8.63⊖ | 37.27⊖ | 8.91⊖ | 36.53⊖ | 8.65⊖ | 36.32⊖ | 8.47⊖ |

| TAC 2008 | Baseline | | Named Entity Recognition | | | | | | Named Entity Linking | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Importance | | Interaction | | Combination | | Importance | | Interaction | | Combination | |
| | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** |
| *SVR Linear* | 35.81 | 8.87 | 36.89⊖ | 9.47⊖ | 37.10✔ | 9.76✔ | 36.57⊖ | 9.31⊖ | 36.98⊖ | 9.50⊖ | 36.86⊖ | 9.29⊖ | 37.20✔ | 9.72⊖ |
| *SVR RBF* | 33.47 | 7.34 | 37.80✔ | 10.25✔ | 32.38⊖ | 6.50⊖ | 35.41✔ | 8.94✔ | 32.81⊖ | 7.46⊖ | 37.06✔ | 9.59✔ | 36.42✔ | 9.46✔ |
| *MART* | 33.89 | 8.22 | 36.20✔ | 8.86✔ | 33.89⊖ | 8.22⊖ | 34.83✔ | 8.63⊖ | 35.59✔ | 8.77✔ | 33.89⊖ | 8.22⊖ | 34.42✔ | 8.26⊖ |

t-test, 95% significance) is reported row-wise, for each entity-focused run vs. the baseline.

From Table 6.2, we first observe several cases where the augmentation of entity information (to standard document summarisation features) has led to significant improvements in summarisation effectiveness. Specifically, over the DUC 2004 dataset, there are 5 cases where the addition of entity-focused features has led to significant improvements in the ROUGE-2 recall target metric. Further, over the TAC 2008 dataset, we observe 7 cases where augmenting baseline summarisation models with entity focused features has led to significant improvements in summarisation effectiveness under the ROUGE-2 recall target metric.

The numerically highest ROUGE-2 effectiveness score over DUC 2004 is for a linear SVR model trained using a combination of NEL-based entity-focused features (R2 = 10.16). The numerically highest ROUGE-2 effectiveness score over TAC 2008 is for a non-linear SVR model trained using NER-based entity importance features (R2 = 10.25). Additionally, while there are cases where using entity information has not led to significant improvements, we observe that there has not been a significant degradation in effectiveness in such cases. However, from Table 6.2, we cannot conclude that using NER or NEL (i.e. strict or loose entity definitions) is more effective for deriving entity-focused event summarisation features. Specifically, when entity-focused features are used in combination with standard baseline features in supervised summarisation models, there is no clear pattern that either NER or NEL is more suitable for computing statistics over entities.

**Table 6.3:** ROUGE-1 and ROUGE-2 effectiveness for state-of-the-art summarisation systems over DUC 2004.

| State-of-the-art | R-1 | R-2 |
|---|---|---|
| *CLASSY04* | 37.71 | 9.02 |
| *CLASSY11* | 37.21 | 9.21 |
| *Submodular* | 39.23 | 9.37 |
| *DPP* | **39.84** | 9.62 |
| *OCCAMS_V* | 38.50 | 9.75 |
| *RegSum* | 38.60 | 9.78 |
| *ICSISumm*‡ | 38.44 | **9.81** |

In terms of learned model performance, we note that the linear SVR model is the most stable learner across different conditions. In particular, the linear SVR produces effective summaries when using NER or NEL, and when using entity importance features, entity-entity interaction features, or the combination of entity features. Additionally, when comparing against the state-of-the-art, in Table 6.3, we observe that the linear SVR model consistently achieves state-of-the-art effectiveness scores under the ROUGE-2 recall target metric. Specifically, the linear SVR model exhibits ROUGE-2 recall scores of 10.05, 9.94, and 9.93 under NER, and ROUGE-2 recall scores of 10.08, 9.93, and 10.16 under NEL. As shown in Table 6.3, such ROUGE-2 effectiveness scores are comparable with the state-of-the-art.

The experimental results presented in Table 6.2 allow us to answer our research question. From Table 6.2, we conclude that entity-focused event summarisation features provide value in the event summarisation task, when combined with standard document summarisation features. Specifically, augmenting learned models that are trained using standard summarisation features with entity-focused event summarisation features leads to significant improvements in summarisation effectiveness. Further, such learned models, using entity-focused event summarisation features, exhibit state-of-the-art effectiveness for the task of generic extractive multi-document newswire summarisation.

### 6.3.4 Discussion & Analysis

We now examine the effectiveness of our proposed entity-focused features individually, within an unsupervised sentence scoring framework. In particular, we evaluate the different variants

**Table 6.4:** ROUGE-1 and ROUGE-2 recall effectiveness of entity-focused event summarisation features, evaluated over the DUC 2004 Task 2 newswire dataset. We report results for top-$k$ selection models, and reference results computed using SumRepo. Statistical significance is reported using the two-tailed paired-sample t-test, with a 95% significance level. Statistical significance between pairs of corresponding NER and NEL features is indicated using ▲▼, with ⊖ indicating no significant difference observed. Further, a ✔ indicates no statistically significant difference to the *Centroid*† summarisation baseline.

Entity-focused Event Summarisation

| NER | Raw Feature | | Log2 | | Len. Norm. | | Log2 & Len. Norm. | |
|---|---|---|---|---|---|---|---|---|
| | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** |
| *Entity Importance* | 33.74 | 6.36 | 33.92 | 6.42 | 32.04 | 5.40 | 32.27 | 5.33 |
| *Entity Interaction* | 32.51 | 6.01 | 33.09 | 6.17 | 32.05 | 5.75 | 32.97 | 5.92 |

| NEL | Raw Feature | | Log2 | | Len. Norm. | | Log2 & Len. Norm. | |
|---|---|---|---|---|---|---|---|---|
| | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** | **R-1** | **R-2** |
| *Entity Importance* | 35.78▲✔ | 7.13⊖ | 36.46▲✔ | 7.93▲✔ | 33.69▲ | 5.99⊖ | 35.17▲✔ | 6.48▲ |
| *Entity Interaction* | 34.78▲ | 7.17▲✔ | 35.06▲✔ | 7.48▲✔ | 34.84▲ | 7.18▲✔ | 35.55▲✔ | 7.55▲✔ |

Reference Results

| **Baselines** | **R-1** | **R-2** |
|---|---|---|
| *Random* | 30.27 | 4.33 |
| *Lead* | 31.46 | 6.13 |
| *LexRank* | 36.00 | 7.51 |
| *Centroid*† | 36.42 | 7.98 |
| *FreqSum* | 35.31 | 8.12 |
| *TsSum* | 35.93 | 8.16 |
| *GreedyKL* | 38.03 | 8.56 |

of entity-focused features shown in Table 6.1, using the traditional top-$k$ rank-then-select approach to summarisation (c.f. Chapter 4). Table 6.4 presents the results of this analysis. We evaluate entity importance and entity–entity interaction features, computed using Named Entity Recognition (NER) and Named Entity Linking (NEL). We report ROUGE-1 and ROUGE-2 recall effectiveness, over DUC 2004, and additionally provide results for several standard document summarisation baselines.

From Table 6.4, we first observe that the NEL runs are significantly more effective than the NER runs, for both entity importance and entity–entity interaction. Further, we observe that the NEL runs are not significantly different from the effectiveness of Centroid, a standard document summarisation baseline. From the results in Table 6.4, we conclude that taking a loose entity definition is more effective, within the top-$k$ summarisation approach. This is in contrast to when using such features in a supervised model, as indicated in Table 6.2, where there is no clear distinction between NER and NEL runs.

Results presented in Table 6.2 demonstrate that learned models trained using entity-focused event summarisation features exhibit state-of-the-art effectiveness for the task of generic extractive multi-document newswire summarisation. Further, results presented in Table 6.4 demonstrate that the NEL-based entity-focused features are at least as effective as standard summarisation baselines. This is despite the fact that the entity-based features are sparse, in

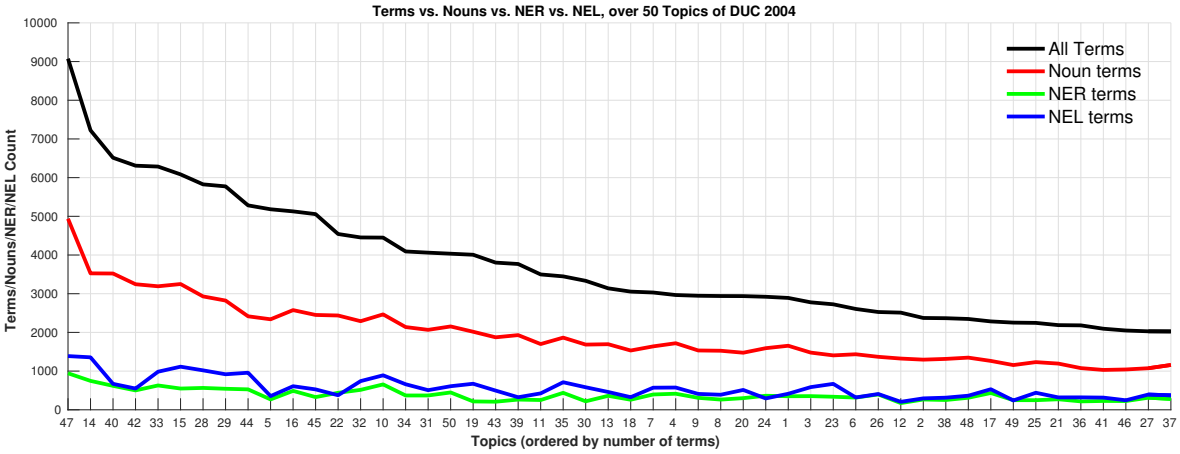



**Figure 6.4:** Counts of terms vs. nouns vs. entities (for both NER and NEL), across the 50 topics of DUC 2004. This quantifies the amount of evidence available to baselines (using terms), and the amount of evidence available to our proposed entity-focused event summarisation approaches (using only NER or NEL entities).

comparison to term-based features. To illustrate this point, we plot the distribution of terms vs. entities over DUC 2004 in Figure 6.4. From Figure 6.4, we note that there is a marked drop in the volume of evidence used by the entity-focused approaches, with respect to the evidence used by the standard baselines. Specifically, entity-focused runs use only entities, whereas the standard baselines (shown in Table 6.4) use all available terms. The difference can be measured in the thousands for a range of topics. From the results in Table 6.4, and the quantification of evidence used in Figure 6.4, we conclude that the entity-focused features we have proposed are effective features for the task of event summarisation, and entities are an important feature to consider when summarising events from newswire documents.

## 6.4 Chapter Summary

In this chapter, we investigated the use of entity-based evidence to improve learned summarisation models that are trained on document summarisation features. We provided experimental results to empirically validate Hypothesis 4 from our Thesis Statement (Section 1.2). We validated our claim that learning a ranking function over newswire sentences, optimising for the importance of entities within the event, the significance of interactions between entities within the event, and the topical relevance of entities to the event, the sentences that are available for inclusion into the event summary can be effectively ranked by their summary worthiness. By answering Research Question 6.1, we demonstrated that within a super-

vised summarisation framework, augmenting document summarisation features with entity-focused event summarisation features results in increased summarisation effectiveness.

In conclusion, within a supervised summarisation framework, by utilising entity-focused event summarisation features, in combination with document summarisation features, summaries can be produced that are comparable in effectiveness to the state-of-the-art. Further, we conclude that the importance of entities, and the interaction between entities, can be used to derive effective features for summarising news events. Furthermore, we conclude that using named entity linking (NEL) is an effective method for computing entity-focused features.

# Chapter 7

# Temporal Event Summarisation

In this chapter, we address our fifth challenge, relating to the effective summarisation of evolving news events. Building on our work in the previous chapters, we move from the retrospective generic summarisation task to the TREC Temporal Summarisation (TREC-TS) task (Aslam et al., 2013, 2014, 2015). Continuing our investigation of supervised machine learned models for extractive summarisation, and given the task we now address is a query-biased summarisation task, we introduce a series of query-biased summarisation features. Specifically, we now summarise documents where a short text query has been given, and the system-produced summary should reflect the information need expressed in the query. Further, we introduce our proposed entity-event relevance features, to measure the relevance of specific entities to the event being summarised. Furthermore, the set of entity-focused event summarisation features, proposed and evaluated in the previous chapter, are now extended to address the temporal nature of the TREC-TS task. Moreover, again based on entity-evidence, we propose and evaluate methods for controlling the number of sentences emitted over time to form a temporal summary of an evolving event (i.e. varying summary length using entities).

Specifically, we continue to examine Hypothesis 4, and further, in this chapter, we now investigate Hypothesis 5 from our Thesis Statement (Section 1.2). In particular, we investigate our claim that adding entity-based evidence to supervised machine learned summarisation models, that have been trained on standard document summarisation features, will result in improvements in summarisation effectiveness. Additionally, we investigate our claim that entity-focused evidence can be used as a means to control the volume of sentences emitted over time to form a temporal summary of an evolving news event.

This chapter is based on the following publications: McCreadie et al. (2013, 2015).

### Chapter Outline

This chapter is organised as follows:

- Section 7.1 discusses temporal summarisation systems that were developed by participants in the 2013–2015 TREC Temporal Summarisation track.
- Section 7.2 defines the summarisation features we use in our experiments in this chapter, including query-biased features, temporal variants of our proposed entity-focused event summarisation features, and entity-focused methods for anti-redundancy filtering.
- Section 7.3 presents our experiments conducted within the context of the query-biased temporal summarisation task (i.e. TREC-TS), providing a thorough empirical evaluation of the event summarisation features proposed in this chapter.

## 7.1 Temporal Summarisation Systems

In this thesis, we investigate the summarisation of evolving news events (Guo et al., 2013), conducting experiments within the multi-document newswire summarisation task. Such news events may be expected, with stories about that event appearing before and after the event, e.g. political elections or severe weather events. Further, news events may be unexpected, with stories appearing only after the onset of the event, e.g. terrorist bombings or public transportation accidents. Within the context of Topic Detection and Tracking (Allan, 2002), an event is described as "some unique thing that happens at some point in time". Within the context of the TREC Temporal Summarisation track (Aslam et al., 2013, 2014, 2015), an event is formalised as a series of discrete sub-events, represented by time-stamped informational nuggets (c.f. Section 2.3.3). This definition accounts for the notion of event granularity, specifically that an event is a composite artefact (made up of inter-related sub-events).

We note, the use of "event" in this thesis is similar in name, but distinct from, the notion of events investigated within the context of Natural Language Processing tasks, such as the Automatic Content Extraction (ACE) research programme (Doddington et al., 2004). Our use of the word "event" is to be interpreted within the context of event detection and tracking (c.f.

Allan, 2002; Petrovic et al., 2010; McMinn et al., 2013; Osborne et al., 2014). Our work is a natural continuation of event detection and tracking research, i.e. once an event has been detected, it is important to develop systems that summarise such events (e.g. Allan et al., 2001; Afantenos et al., 2005). Recently, research concerning the temporal summarisation of evolving events (Guo et al., 2013) has begun to examine large-scale event summarisation, developing standardised corpora and evaluation metrics specific to temporal summarisation.

As discussed in Section 2.3.3, the Text Retrieval Conference[1] (TREC) introduced the 2013–2015 Temporal Summarisation Track[2]. The stated aims of the TREC Temporal Summarisation (TREC-TS) evaluation campaign are to promote research examining automatic summarisation systems that extract sentences from high-volume textual streams of news and blog data, to form summaries of large-scale evolving news events (Aslam et al., 2013, 2014, 2015). The TREC-TS task is related to, but distinct from, the TAC Update Summarisation task (Dang and Owczarzak, 2008). While the TAC Update Summarisation task involves summarising changes in news events over time, the experimental setup operated under the assumption that all input documents were relevant to the event being summarised, and such documents were professionally authored newswire articles obtained directly from press agencies and newspaper publishers (similarly to the DUC 2004 dataset used in previous chapters). Most importantly, the TAC Update Summarisation task was limited to a single batch update (i.e. summarising from one single batch of documents to another).

In contrast, the TREC-TS task does not assume that the input document stream is on-topic. As such, systems must be able to identify relevant sentences from a much larger collection of non-relevant sentences. Specifically, the ratio of relevant sentences to non-relevant sentences is 2,309,416 to 18,755 in the TREC-TS dataset we use in our experiments. Further, the input documents that are to be summarised in the TREC-TS task are obtained by crawling publicly accessible web-pages (i.e. not commercial newswire). As such, systems must be able to adapt to the text processing errors arising from a high-volume of automatically extracted sentences from news-related web-pages. Furthermore, the TREC-TS task involves the summarisation of evolving news events over a larger time-period (typically numbered in days). These three factors, a mixture of relevant and non-relevant input sentences, a collection of sentences that contain text processing errors, and the requirement to summarise events over longer time periods, ensure that the TREC-TS task is a realistic and important research challenge.

[1] trec.nist.gov

[2] trec-ts.org

The TREC-TS task, as an extractive summarisation problem, was initially tackled by participants (i.e. teams) within the unsupervised rank-then-select paradigm (c.f. Section 2.4). In the TREC-TS 2013 (Aslam et al., 2013) track, the best run was from the Johns Hopkins University (HLTCOE) team (Xu et al., 2013). This team did not index the corpus, but processed the document stream in temporal order, one document at a time. A document pre-processing step was implemented, filtering by event time, event profile keywords, and cosine similarity to the event tracking query. This team employed Wikipedia-based query-expansion to enrich the topic representation, expanding with terms from pages similar to the event, e.g. earthquakes. For selecting sentences, features included the cosine similarity of the expanded query to the document title and description, existence of named entities within the document and words commonly associated with news events (e.g. "killed", or "injured"). Other teams indexed the corpus in hourly batches, and used Information Retrieval (Croft et al., 2010; Büttcher et al., 2010) techniques such as BM25 (Robertson et al., 2009) to filter the corpus for relevant documents based on the event query. The University of Waterloo team (Baruah et al., 2013) also performed query-expansion via Wikipedia. The Beijing University of Technology (BJUT) team (Yang et al., 2013) selected sentences by first clustering the documents and taking the sentence most similar to the centroid of such clusters. The Chinese Academy of Sciences (ICTNET) team (Liu et al., 2013) filtered for relevant documents by searching on document titles only. Similarly to the HLTCOE run, a set of trigger words (i.e. cue words) was used as a feature to extract a set of important sentences, from which non-redundant summary sentences were determined by the SimHash algorithm. The Beijing University of Post and Telecommunications (PRIS) team (Zhang et al., 2013) extracted sentences by LDA topic modelling, scoring sentences based on how well they matched an event's topic model.

We now discuss the unsupervised entity-focused temporal summarisation system that we developed (McCreadie et al., 2015) for participation in the 2015 (Aslam et al., 2015) TREC Temporal Summarisation track. In particular, we formed the hypothesis that events are primarily about entities, and effective summaries of evolving news events can be produced using summarisation features that are derived from the entities involved in the events. The features we investigated were entity importance and entity–entity interaction (c.f. Chapter 6), which attempt to capture the salient entities and their connection with other entities. Further, we also investigated two distinct methods of processing the corpus, summarising the content of each

**Table 7.1:** TREC-TS 2015 results for the "2015RelOnly" corpus (Task 3).

| TeamID | RunID | nE(Gain) | Comp. | E(Latency) | HM(nE(LG),Lat.Comp.) |
|---|---|---|---|---|---|
| WaterlooClarke | UWCTSRun4 | **0.1840** | 0.1710 | 0.3983 | **0.0853** |
| BJUT | DMSL2N2 | 0.0645 | 0.6557 | 0.5606 | 0.0649 |
| **uogTr** | uogTrhEQR2 | 0.0667 | 0.5459 | 0.5335 | 0.0639 |
| **uogTr** | uogTrhEEQR4 | 0.0714 | 0.5342 | 0.5249 | 0.0632 |
| BJUT | DMSL2A1 | 0.0600 | 0.6777 | 0.5787 | 0.0622 |
| **uogTr** | uogTrdEQR1 | 0.0402 | 0.6590 | 0.6741 | 0.0508 |
| **uogTr** | uogTrdEEQR3 | 0.0418 | 0.6096 | 0.6401 | 0.0505 |
| TREC Median | – | 0.0595 | 0.5627 | 0.5524 | 0.0472 |
| UvA.ILPS | COS | 0.0428 | 0.5708 | 0.5951 | 0.0471 |
| UvA.ILPS | COSSIM | 0.0281 | 0.7325 | 0.6952 | 0.0372 |
| udel fang | WikiOnly2 | 0.0446 | 0.5522 | 0.5008 | 0.0353 |
| UvA.ILPS | LexRank | 0.0224 | 0.7490 | 0.6836 | 0.0299 |
| ISCASIR | runvec2 | 0.0190 | **0.7881** | **0.7210** | 0.0250 |
| UvA.ILPS | LDAv2 | 0.0202 | 0.7423 | 0.6338 | 0.0241 |
| ISCASIR | runvec1 | 0.0174 | 0.7852 | 0.6458 | 0.0215 |

event either document-by-document, or in hour-by-hour batches. In the case of hour-by-hour, all sentences from documents within that hour are combined into a single virtual document. Summarising each document as it arrives simulates a real-time scenario, whereas batching the documents in hourly chunks represents a near real-time task. We submitted runs to TREC-TS 2015 Task 3, "Summarisation Only", which used the "RelOnly" corpus[1], where the input documents presented to the event summarisation algorithm are a reasonably topically cohesive set of documents about an event (i.e. pre-filtered). In our experiments in Section 7.3, we also use a "RelOnly" corpus, deriving a version that covers all years (2013–2015) of the track, as described in our experimental setup in Section 7.3.2.

Table 7.1, reproduced from Aslam et al. (2015), presents the results of Task 3, reporting the TREC-TS metrics discussed in Section 2.3.3. The most effective system under the *H* metric, the harmonic mean of normalised expected latency gain and latency comprehensiveness, was an unsupervised run from the University of Waterloo (Raza et al., 2015). Near real-

[1] dcs.gla.ac.uk/~richardm/TREC-TS-2015RelOnly.aws.list

time indexing (five minute batches) was deployed, querying the frequently updated indices using the topic query (with query expansion) to filter for relevant documents. Once relevant documents were identified, a simple lead-based algorithm (i.e. selecting the first sentence) was used to derive candidate summary sentence updates. Such candidate updates were only emitted if they passed an anti-redundancy filter.

Further, from Table 7.1, under the $H$ metric, we observe that our submitted runs performed above the track average. We also observe that processing the corpus hour-by-hour is more effective than processing document-by-document. More specifically, we observe that the document-by-document method is more effective under comprehensiveness metrics, while the hour-by-hour method is more effective under gain metrics. Furthermore, from Table 7.1, examining the entity-focused features, entity importance ($E$) and entity–entity interaction ($EE$), we observe that both features exhibit very similar effectiveness under the harmonic mean metric. More specifically, entity–entity interaction is more effective than entity importance, for both document-by-document and hour-by-hour, under normalised expected gain, although not when latency is taken into account. Additionally, entity importance is more effective than entity–entity interaction under comprehensiveness metrics for the document-by-document method. From the results in Table 7.1, we conclude that using entities to derive temporal event summarisation features can lead to effective summaries of evolving events. We also conclude that, as we found that processing the corpus in hourly batches results in more effective event summary sentence selection decisions, in our later experiments in Section 2.3, we should continue to process the TREC-TS corpus in hourly batches.

Similar to the work in this thesis, a number of temporal summarisation systems have been proposed out-with the context of the TREC-TS evaluation campaign. Such systems re-use the TREC-TS summarisation dataset, and the TREC-TS summarisation evaluation metrics. Further, after three consecutive years of the track, where sentence-level summarisation evaluation judgements were accumulated, supervised machine learned summarisation approaches became feasible. For instance, learning-to-rank (Liu, 2009) techniques (McCreadie et al., 2014), Gaussian process regression (Kedzie et al., 2015), and sequential decision making (Kedzie et al., 2016). In this thesis, we also conduct experiments using supervised machine learned summarisation models, also reusing the TREC-TS experimental setup (i.e. data and metrics). In the next section, we discuss the features we use in such supervised models.

## 7.2 Temporal Summarisation Features

Within the experimental setup of the TREC Temporal Summarisation (TREC-TS) task (Aslam et al., 2013, 2014, 2015), 45 summarisation events are defined, $E = (e_1, e_2, \ldots, e_{45})$. Each event, $e_i \in E$, is an evolving news event that is of significant interest to the general public, such that a Wikipedia article about the event exists (i.e. a news-worthy event). Each event is represented by a short text query, $q_i$, and spans a particular time period (numbered in days). For example, topic number 14 is defined as follows: "boston marathon bombing"; 15th April 2013 through 20th April 2013; <wikipedia.org/wiki/Boston_Marathon_bombings>.

We segment such event time periods into $n$ discrete hourly batches, $(T_{e_i} = t_1, t_2, \ldots, t_n)$. Further, a corpus of $m$ documents exists, $C = (d_1, d_2, \ldots, d_m)$, which spans the time period from December 2011 through May 2013. The documents, $d_i \in C$, discuss the events in $E$. Given an evolving news event, $e_i$, that spans the time period, $T_{e_i}$, the corpus of documents is time-filtered such that event-specific subsets of the corpus are created, $C_{e_i} \subset C$, for $d_i$ within $T_{e_i}$. For each time period, $t_i$ in the event, $e_i$, the documents are segmented into discrete sentences, as we are addressing a sentence-level extractive summarisation task.

As described in Chapter 5, for the purposes of conducting supervised summarisation experiments, the natural language text of the sentences is mapped to numerical feature vectors (i.e. summarisation features), with sentences labelled according to their summary worthiness. In our experiments in this chapter, we label TREC-TS sentences using ROUGE-$n$ precision labels, with respect to the gold-standard nuggets for each event. This process produces several time-stamped batches of per-sentence features, $\boldsymbol{X}$, and corresponding time-stamped batches of per-sentence labels, $\boldsymbol{y}$. As such, the resulting training data for supervised summarisation experiments within the context of the TREC Temporal Summarisation task is defined as:

$$\underset{(t_1)\text{ first hour}}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix}} \quad \underset{(t_2)\text{ second hour}}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix}} \quad \underset{(t_i)\text{ subsequent hours}}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix}} \tag{7.1}$$

In the remainder of this section, we define the summarisation features, $\boldsymbol{X}$, that we use in our experiments in Section 7.3. We first define generic summarisation features, then define the

query-biased and query-context summarisation features that specifically address the query-biased nature of the task. Then, we define our entity-focused event summarisation features, extending the features previously defined in Chapter 6 to address the temporal and query-biased nature of the task at hand. In total, our experiments include the following 44 features.

### 7.2.1 Generic Features

In our experiments in Section 7.3, we use 12 generic summarisation features:

$$\textbf{generic} = \begin{pmatrix} f_1(\text{Sentences}) & f_2(\text{Tokens}) & f_3(\text{Position}) & f_4(\text{Length}) & \\ f_5(\text{FreqSum}) & f_6(\text{TsSum}) & f_7(\text{Centroid}) & f_8(\text{GreedyKL}) & f_9(\text{LexRank}) \end{pmatrix} \tag{7.2}$$

Features 3 through 9 are standard summarisation baseline features, previously defined and evaluated in Chapter 4, Chapter 5, and Chapter 6. Feature 1 and feature 2 are new to experiments in this chapter. Feature 1, "Sentences", is the number of sentences contained within an hourly batch of documents. Feature 2, "Tokens", is the total number of words contained within an hourly batch of documents. Both of these additional features quantify the volume of content (i.e. sentences and terms) within an hourly batch of documents. In particular, over each hour-by-hour batch, the number of sentences and terms varies, which raises numerical comparability issues with respect to the scores obtained from baseline algorithms over batches. This problem is only observed when training supervised summarisation models over multiple time batches on the TREC-TS dataset, and is not of concern in previous experiments over the DUC 2004 dataset.

For example, scores under the FreqSum algorithm are not directly comparable across batches, as the computation of within-batch per-sentence FreqSum scores is based on the frequency of terms within each batch. Specifically, higher or lower FreqSum scores are a function of the number of terms in any given hourly batch. As such, we hypothesise that, within supervised machine learned models, feature 1 and feature 2 may act to quantify the other features, with respect to differences in scores of baseline algorithms over hourly batches. We return to this point later, regarding the comparability of feature scores over hourly batches, in our discussion and analysis (Section 7.3.4).

### 7.2.2 Query-biased Features

In our experiments in Section 7.3, we use 4 query-biased summarisation features:

$$\textbf{query-biased} = \begin{pmatrix} f_{10}(\text{DFRee.qe}) & f_{11}(\text{DFIZ.qe}) & f_{12}(\text{DirichletLM.qe}) & f_{13}(\text{BM25.qe}) \end{pmatrix} \tag{7.3}$$

As the TREC-TS task is a query-biased summarisation task, we introduce a series of query-biased summarisation features. Similarly to our work in Chapter 5, we employ sentence retrieval techniques (Murdock, 2006; Balasubramanian et al., 2007) from the Information Retrieval literature (Croft et al., 2010; Büttcher et al., 2010). Specifically, as previously defined in Section 5.3.2, a series of information retrieval models are used to score sentences with respect to a query. In Chapter 5, we demonstrated the effectiveness of such scores as labels ($\boldsymbol{y}$), whereas in this chapter, we examine the use of such scores as features ($\boldsymbol{X}$). In this case, the topic query from the TREC-TS task is used, e.g. "boston marathon bombing".

Sentence retrieval experiments are again conducted using the Terrier Information Retrieval Platform[1] (Macdonald et al., 2012). Sentences are scored according to the DFRee (Amati and van Rijsbergen, 2002), DFIZ (Kocabas et al., 2014), language modelling (Ponte and Croft, 1998), and BM25 (Robertson et al., 2009) retrieval models. Additionally, the four retrieval models utilise query expansion, denoted using ".qe", which expands the TREC-TS topic query with the $m$ most informative terms, obtained via the $n$ highest-ranked documents given the original query. Specifically, the top-10 terms from the top-3 ranked documents are added to the original query, where term informativeness is computed via the Bo1 (Bose-Einstein 1) model from the Divergence from Randomness family of retrieval models (Amati, 2003). In our experiments in Section 7.3, the absolute ranks of the sentences returned from this expanded query is used as the feature, where an alternative option would be to use the retrieval model scores. We now discuss our query-context features.

### 7.2.3 Query-context Features

In our experiments in Section 7.3, we use 8 query-context summarisation features:

$$\textbf{query-context} = \begin{pmatrix} f_{14}(\text{DFRee.qe} - \text{prev}) & f_{16}(\text{DFIZ.qe} - \text{prev}) & f_{18}(\text{DirichletLM.qe} - \text{prev}) & f_{20}(\text{BM25.qe} - \text{prev}) \\ f_{15}(\text{DFRee.qe} - \text{next}) & f_{17}(\text{DFIZ.qe} - \text{next}) & f_{19}(\text{DirichletLM.qe} - \text{next}) & f_{21}(\text{BM25.qe} - \text{next}) \end{pmatrix} \tag{7.4}$$

[1] terrier.org

Further to the query-biased features defined above (Section 7.2.2), we also experiment with a series of query-context features. In particular, we score sentences with respect to their context within documents, forming the hypothesis that a sentence does not exist in isolation. Specifically, a sentence is scored by considering sentence-level context windows of sizes: plus one; and minus one – i.e. the previous sentence and the next sentence. Given a sentence, under a particular information retrieval weighting model, we take the previous sentence score, and subsequent sentence score, as query-context features for that given sentence. In our experiments, we use the same set of retrieval models (and query expansion settings) as used in the query-biased feature group above. We have separated out the query-context features from the query-biased features to support the evaluation of query-context features in isolation.

### 7.2.4 Entity-batch Features

In our experiments in Section 7.3, we use 11 entity-batch summarisation features:

$$\textbf{entity-batch} = \begin{pmatrix} f_{22}(\text{TotalEntities}) & f_{23}(\text{EntitiesInSentence}) & & \\ f_{24}(\text{Eimp}) & f_{25}(\text{EimpLog2}) & f_{26}(\text{EimpNorm}) & \\ f_{27}(\text{EEint}) & f_{28}(\text{EEintLog2}) & f_{29}(\text{EEintNorm}) & f_{30}(\text{EEintLLR}) \\ f_{31}(\text{QueryEntities}) & f_{32}(\text{WikipEntities}) & & \end{pmatrix} \tag{7.5}$$

We now discuss our proposed entity-focused event summarisation features. The entity importance features, 24, 25, and 26, have previously been defined in Section 6.2.1. The entity-entity interaction features, 27, 28, and 29, have previously been defined in Section 6.2.2, and we additionally define a new entity-entity interaction feature, EEintLLR (feature 30). The EEintLLR feature computes an estimate of the interaction between pairs of entities. In particular, using the pair independence hypothesis likelihood ratio (Dunning, 1993; Jones et al., 2006), the significance of the dependence of one entity on another can be computed, based on observed sentence-level co-occurrence of surface mentions of those entities.

Additionally, similarly to the generic feature group, we introduce two quantification features: TotalEntities (feature 22); and EntitiesInSentence (feature 23). The TotalEntities feature quantifies the number of surface mentions of entities within all of the sentences within a particular hourly batch. The EntitiesInSentence features quantifies the number of entities in each sentence. We hypothesise that entity quantification features will be useful as we are training learners using data that contains variable cross-batch scores.

Further, we also introduce our entity-event relevance features: QueryEntities (feature 31); and WikipEntities (feature 32). The QueryEntities feature quantifies the number of "query entities" in a given sentence. Given entities that occur in sentences that (boolean) matched the topic query terms, we promote such entities (to "query entities") as they are related to the topic query terms. As such, although the text of the surface mention of an entity may not match the query terms, we can still capture the query-relevant nature of such entities via their association with sentences that do match the query terms.

The WikipEntities feature requires external evidence, specifically, an index of Wikipedia, which is created using the Terrier (Macdonald et al., 2012) Information Retrieval Platform. The version of Wikipedia indexed pre-dates the events in the TREC-TS corpus. The topic query is executed on the Wikipedia index, returning a ranked list of Wikipedia pages. As we are using the AIDA (Hoffart et al., 2011) named entity linking tool to perform entity recognition, we can link the surface mentions of entities within sentences of the TREC-TS corpus to articles (i.e. linked entities) returned via the Wikipedia index. The set of entities returned by querying the Wikipedia index are assumed to be relevant to the event in question. The feature is computed as the count of the number of such "wikipedia entities" in a sentence.

### 7.2.5 Entity-temporal Features

In our experiments in Section 7.3, we use 12 entity-temporal summarisation features:

$$\textbf{entity-temporal} = \begin{pmatrix} f_{33}(\text{EventBatches}) & f_{34}(\text{EventSentences}) & f_{35}(\text{TotalEntities}) & \\ f_{36}(\text{Eimp}) & f_{37}(\text{EimpLog2}) & f_{38}(\text{EimpNorm}) & \\ f_{39}(\text{EEint}) & f_{40}(\text{EEintLog2}) & f_{41}(\text{EEintNorm}) & f_{42}(\text{EEintLLR}) \\ f_{43}(\text{QueryEntities}) & f_{44}(\text{WikipEntities}) & & \end{pmatrix} \tag{7.6}$$

Having defined our entity-batch feature group, we now discuss our proposed entity-temporal feature group. We have previously defined features 35 through 44, and we now introduce EventBatches (feature 33), and EventSentences (feature 34). Both features again quantify the other features, where EventBatches is the number of hours in a given topic (i.e. the number of batches), and EventSentences is the number of sentences in a given topic. These quantification features differ from previous quantification features, as they operate at the whole-topic level, as opposed to the batch level.

The important difference in the other 10 features, compared to the entity-batch feature group, is the method used to compute the feature scores over time. Specifically, the feature scores in the entity-batch group are computed anew at each hourly batch, i.e. there is no continuation from batch-to-batch. In contrast, the entity-temporal features are cumulative features, that maintain their state over batches as the events evolve over time. In particular, we define the entity-temporal features as summations over the event timeline, $(T_{e_i} = t_1, t_2, \ldots, t_n)$.

For example, taking entity importance (feature 36), we previously defined this feature as:

$$\text{EntityImportance}(e_i) = \sum_{d \in C} \text{cf}(e_i, d) \tag{7.7}$$

We now define this feature (and all others in this feature group) as a summation over the time period of the event being summarised (up to a specific point , $t_i$, on the event timeline):

$$\text{EntityImportance}(e_i) = \sum_{t_i \in T} \mathit{EntityImportance}(e_i) \tag{7.8}$$

To compute such cumlative feature scores, we maintain entity-focused statistics over hourly batches. For example, referring back to the discussion of our entity–entity interaction feature, we defined an entity co-occurrence graph that was used to compute the feature score. For the entity-temporal variant of this feature, this graph structure is now evolved over the time period of the event. In particular, as new entities are observed, new nodes in the graph are created, as these new entities are observed to co-occur at the sentence-level with existing entities, new edges are connected, and as previously seen entities co-occur again, edge weights are increased. We hypothesise that the entity-temporal features will outperform the entity-batch features, in terms of the summarisation effectiveness of models trained on such features.

### 7.2.6 Entity-focused Sentence Selection

We previously discussed how the entity-based features we proposed in Chapter 6 could be extended to the temporal summarisation setting. These features are used to rank candidate summary sentences for inclusion into the summary. By improving this ranking, intuitively, we can produce better summaries. However, there is another alternative approach for improving the summary. Over time, the summarisation system will encounter repeated (i.e. redundant) sentences with respect to what has been previously returned to the user in the temporal summary output over time. Ideally, we do not want to show the user multiple sentences with the

same information. Hence, one way to improve summarisation effectiveness would be to reduce the number of redundant sentences returned. In this section, we describe three different approaches to tackle redundancy in the temporal summaries that we produce.

In particular, we propose one method that uses classical textual similarity to identify redundant sentences, and propose two methods that use entities to remove redundant sentences. Textual redundancy in summaries is a common problem that has affected summarisation systems since early works on multi-document summarisation (c.f. Section 2). One common method for removing redundancy in a summary is to apply a variant of Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998). MMR incrementally selects sentences in a greedy manner, where, in each iteration, the sentence that is most textually dissimilar to those previously selected is chosen. Another common method (Hong et al., 2014) for calculating textual (dis)similarity is cosine similarity between sentences. A fixed similarity threshold is used to determine if a pair of sentences are too similar. If a given candidate summary sentence is highly similar to an previously selected summary sentence, it is discarded (i.e. not selected for the summary). In our later experiments, we report summarisation performance both with and without this type of summary redundancy removal.

While focusing on the reduction of textual redundancy is a common approach, we hypothesise that we might be able to more effectively reduce redundancy by instead focusing on the redundancy of entities within each sentence. The reason for this is that due to the nature of the English language, and how news article sentences are written, sentences can appear textually similar but cover different information. For instance, consider the following sentences:

> The cruise ship Costa Concordia crash has so far resulted in over 100 deaths.
>
> The Costa Concordia is a cruise ship crashed into the Isola del Giglio, resulting in over 200 injuries.

As we can see from these two sentences, they share many terms but contain different information. However, a textual similarity comparison would rate these as very similar, as almost half of the terms in the two sentences overlap. Tracking event entities might help distinguish these sentences, as while both sentences contain the entity “Costa Concordia”, the second sentence also contains a new unseen entity “Isola del Giglio”.

Hence, we introduce two entity-focused anti-redundancy filtering components:

**OneNewEntity** – In this case, for each sentence that would normally be added to the summary, we first identify all of the entities that it contains. If the sentence contains no entities, we discard it, on the assumption that it does not contain useful information. Otherwise, we iteratively check to see if the current sentence contains new entities, i.e. does it contain entities that have not been covered by previous sentences returned to the user. If the current sentence contains at least one new (previously unseen) entity, we add that sentence to the temporal summary output (shown to the user), otherwise we discard it as redundant.

**NewOrHotEntities** – As an event evolves over time, new information related to an entity may appear. A potential issue with the OneNewEntity approach is that sentences containing this new information may be discarded, as updated information may not always correspond with the inclusion of new entities. To tackle this, we also evaluate a more relaxed version of the OneNewEntity approach that we refer to as NewOrHotEntities. In this case, we introduce a secondary criterion that allows for more sentences to be selected, even if they do not contain new entities. We do so by incorporating the popularity of entities over time. More precisely, in addition to tracking the entities already seen, we also track their frequency across hourly batches over time. We first apply the OneNewEntity test to see if the sentence contains any new entities, however if the sentence is to be discarded, we then check to see if it contains any currently popular (i.e. high-frequency) entities. If the sentence contains one or more high-frequency entities, we add it to the summary instead of discarding it. In this way, if an entity is currently important to the event (has a high frequency), we are able to return multiple sentences containing that entity, while still limiting the number of sentences that contain no new entities. In our later experiments, we consider the currently popular entities to be the three entities with the highest frequency at that time point.

## 7.3 Evaluation

In this section, we conduct an empirical evaluation of supervised machine learned summarisation models, trained on temporal variants of our proposed entity-focused event summarisation features. We evaluate learned models that have been trained using different sets of features, conducting a group-wise feature ablation study. We begin by stating our research question,

then describe our experimental setup. Empirical observations are reported for the task of query-biased temporal summarisation, within the context of the TREC Temporal Summarisation Track (Aslam et al., 2013, 2014, 2015). Finally, we discuss and analyse our empirical observations.

### 7.3.1 Research Questions

In our Thesis Statement (Section 1.2), we formed Hypothesis 4:

> By learning a ranking function over newswire sentences, optimising for the importance of entities within the event, the significance of interactions between entities within the event, and the topical relevance of entities to the event, we hypothesise that the sentences that are available for inclusion into the event summary can be effectively ranked by their summary worthiness, using a supervised summarisation model trained using such entity-focused event summarisation features, augmented with document summarisation features.

Further, in our Thesis Statement (Section 1.2), we formed Hypothesis 5:

> As real-world news events exhibit temporal patterns of activity and inactivity, reflecting ongoing developments in the evolution of the event over time, we argue that selecting a fixed number of summary sentences at pre-determined periodic time-intervals is non-optimal, and we hypothesise that entity-focused event summarisation features can be used to derive effective anti-redundancy methods, and that an effective temporal summary of an evolving event consists of a variable number of sentences selected at event-determined periodic time-intervals, mirroring event evolution over time.

To validate Hypothesis 4 and Hypothesis 5, we address the following research questions:

**Research Question 7.1.** For addressing the TREC Temporal Summarisation (TREC-TS) task, can a classifier be trained to reduce the number of input sentences to be summarised?

**Research Question 7.2.** When addressing the query-biased nature of the TREC-TS task, are query-biased summarisation features derived from sentence retrieval scores effective?

**Research Question 7.3.** When addressing the temporal aspects of the TREC-TS task, are the temporal variants of our proposed entity-focused event summarisation features effective?

**Research Question 7.4.** Within a supervised summarisation framework, does augmenting document summarisation features with entity-focused event summarisation features lead to an increase in supervised summarisation effectiveness for the TREC-TS task?

**Research Question 7.5.** For the TREC-TS task, does varying the number of sentences emitted over time, using entity-based evidence, lead to more effective temporal summaries?

### 7.3.2 Experimental Setup

For our experiments in this chapter, we summarise documents from the TREC Temporal Summarisation track dataset. However, due to the size of the original TREC-TS 2013 dataset (approx. $\frac{1}{2}$ billion documents), and the engineering challenges of processing such very-high volumes of documents, various document sampling methods have been proposed within the TREC-TS track. In particular, starting with the 2014 TREC-TS track (Aslam et al., 2014), pre-filtered[1] versions of the TREC-TS corpus were made available to track participants. Specifically, the TREC-TS-2013F, TREC-TS-2014F, and TREC-TS-2015F datasets were derived by the track organisers. Such datasets were created by manually authoring event-related queries, and issuing such queries on an indexed version of the full corpus, producing a ranking (i.e. pre-filtered set) of assumed-relevant documents for each event. Further, the resulting corpora was subjected to additional (manual) filtering for Task 3 of the 2015 edition of the TREC-TS track, producing the dataset referred to as TREC-TS-2015RelOnly (Aslam et al., 2015). The RelOnly dataset, however, was only made available for TREC-TS 2015 (covering 20 topics).

Hence, in the experiments in this chapter, we derive a new TREC-TS dataset similar to the RelOnly corpus used in TREC-TS 2015, but covering all topics (i.e. TREC-TS 2013–2015). We refer to this new dataset as the "TRECTS-RelOnly" dataset. Instead of manually identifying relevant documents, we use the 2013–2015 TREC-TS track relevance judgements to derive a corpus of 11,383 documents, containing 2,328,171 sentences, out of which there are 11,902 known to be relevant (from the TREC-TS track evaluation judgements). Specifically, for every document identifier, for a relevant sentence, in the 2013–2015 TREC-TS track

[1] trec-ts.org/home/corpus-filtering-details

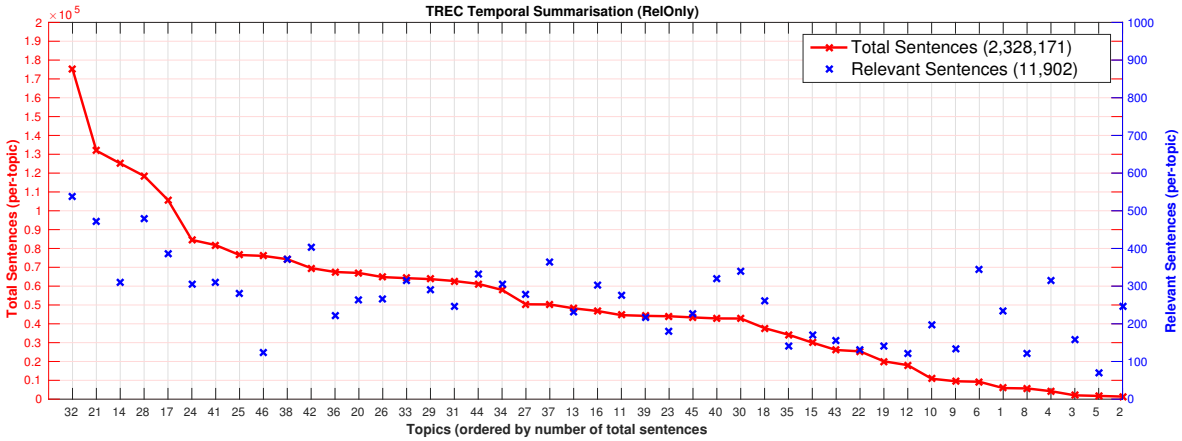



**Figure 7.1:** TREC-TS "RelOnly" corpus – containing only documents where the docid appears in the qrels, where approx. $\frac{1}{2}$ of topics contain more than 50,000 sentences.

**Table 7.2:** Breakdown of the 2013–2015 TREC-TS topics into a 5-fold cross-validation set.

| Fold | 2013-2015 TREC-TS Topics | | | | | | | | | Train | Valid | Test |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| f1 | 01 | 06 | 12 | 17 | 22 | 27 | 32 | 37 | 42 | (f1,f2,f3) | f4 | f5 |
| f2 | 02 | 08 | 13 | 18 | 23 | 28 | 33 | 38 | 43 | (f2,f3,f4) | f5 | f1 |
| f3 | 03 | 09 | 14 | 19 | 24 | 29 | 34 | 39 | 44 | (f3,f4,f5) | f1 | f2 |
| f4 | 04 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | (f4,f5,f1) | f2 | f3 |
| f5 | 05 | 11 | 16 | 21 | 26 | 31 | 36 | 41 | 46 | (f5,f1,f2) | f3 | f4 |

relevance assessments, we include that whole document into our TRECTS-RelOnly dataset. Figure 7.1 provides collection statistics illustrating the characteristics of the dataset. As can be seen from Figure 7.1, over half of the topics still have over 50,000 sentences to be taken as input to the summarisation process. When processing the documents in the TRECTS-Relonly corpus, to identify named entities, we use the AIDA NEL (Hoffart et al., 2011) toolkit, trained on a version of Wikipedia dated prior to the on-set of all events in the TREC-TS corpus.

For our investigations over the TRECTS-RelOnly corpus, we conduct experiments within a supervised summarisation framework (c.f. Chapter 5). We train linear Support Vector Machine regression (SVR) models, specifically, L2-regularised, L1-loss SVR (Fan et al., 2008). In this chapter, we fix the "C" value of the SVM learner to 1.0. Learned models are trained on ROUGE-2 precision labels. As per machine learning best-practice guidelines (Müller and Guido, 2017; Géron, 2017), feature normalisation (scaling within the range $[0..1]$) is applied. Further, we split the TRECTS-RelOnly dataset into a 5-fold cross validation train, validation, and test set. The specific topic split we use in our experiments is provided in Table 7.2.

The summarisation effectiveness of learned models is evaluated using the TREC-TS evaluation framework (c.f. Section 2.3.3), and based on concatenating the track summarisation evaluation judgements for all years, i.e. we evaluate over all 45 topics of the TREC-TS dataset. In the following experiments, for Research Questions 7.2, 7.3, and 7.4, where we evaluate for fixed length summaries, we report the TREC-TS metrics of: normalised expected latency gain, denoted "nE(Gain)", comprehensiveness, denoted "Comp.", the harmonic mean of normalised expected latency gain and latency comprehensiveness, "HM(nE(LG),Lat.Comp.)", and also report expected latency, denoted "E(Latency)". The TREC-TS evaluation metrics are defined in Section 2.3.3. For Research Question 7.5, where we evaluate for varying-length summaries, we report ROUGE-1 precision (Lin, 2004), in addition to the TREC-TS metrics. For Research Question 7.1, where we examine supervised classifier performance, we report classification confusion matrices (Witten et al., 2016).

### 7.3.3 Experimental Results

#### Research Question 7.1

We begin with Research Question 7.1, where we seek to determine if a classifier can be trained to pre-filter TRECTS-RelOnly sentences. We would wish to reduce the input to the automatic text summarisation process, as computing various summarisation features for training supervised summarisation models can be computationally expensive. In particular, the dataset we use in our experiments contains 2,328,171 input sentences. In an online streaming scenario, where we are summarising events in real-time, reducing the number of candidate summary sentences could be beneficial for commercial applications of summarisation systems.

We present the results of our classification experiment in Table 7.3, where we report classification matrices showing the effectiveness of a Naive Bayes classifier (Aggarwal and Zhai, 2012). The classifier is trained on labels obtained from the TREC-TS track relevance assessments, where we have positive and negative labels (i.e. binary classification) indicating whether each sentence is relevant ("RelSent"), or non-relevant ("NonRel"). We train the classifer over a 5-fold cross validation (c.f. Table 7.2). The features used in the text classifier are tf.idf vectors, and we train a multinomial Naive Bayes model[1]. As the TRECTS-Relonly

[1] scikit-learn.org/stable/modules/naive_bayes.html

**Table 7.3:** Results of training a Naive Bayes classifier to predict TREC-TS summary sentences (RelSent). We report a classification matrix for three approaches, where we first demonstrate the poor performance of classification over the imbalanced TREC-TS dataset, then report results when over-sampling the minority class, and when under-sampling the majority class. Results indicate that the TRECTS-RelOnly dataset can be effectively pre-filtered using an under-sampling technique, reducing the input sentence set by 1,467,520 sentences, from 2,328,171 sentences to 860,651 sentences, a reduction factor of over 60%, at a cost of 1,536 relevant sentences.

| **Imbalanced** | | | | **Over-sample** | | | | **Under-sample** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | NonRel | RelSent | All | | NonRel | RelSent | All | | NonRel | RelSent | All |
| NonRel | 2,309,128 | 288 | 2,309,416 | NonRel | 2,058,397 | 251,019 | 2,309,416 | NonRel | 1,465,984 | 843,432 | 2,309,416 |
| RelSent | 18,753 | 2 | 18,755 | RelSent | 11,337 | 7,418 | 18,755 | RelSent | 1,536 | 17,219 | 18,755 |
| All | 2,327,881 | 290 | 2,328,171 | All | 2,069,734 | 258,437 | 232,8171 | All | 1,467,520 | 860,651 | 2,328,171 |

training dataset exhibits a class imbalance, 18,755 positive examples to 2,309,416 negative examples, we experiment with over- and under-sampling techniques (Lemaître et al., 2017).

From the experimental results in Table 7.3, we first observe that training a classifier on the imbalanced dataset is not effective. The classifier simply learns to predict the majority class, failing at the task we wish to achieve. Next, we observe results for randomly over-sampling the minority class. From the confusion matrix, we can see that the classifier has correctly identified 7,418 relevant sentences, and the total summarisation input has been reduced to 258,437 sentences. However, this comes at a cost of discarding (i.e. classifying as non-relevant) 11,337 relevant sentences. Finally, we observe the results of randomly under-sampling the majority class. The classifier has correctly identified 17,219 out of 18,755 relevant sentences. If we take the positively classified sentences as input to the summarisation process, we have now reduced the total input size (i.e. number of sentences to summarise) by 1,467,520 sentences, from 2,328,171 sentences to 860,651 sentences, a reduction factor of over 60%. We note, there is still a cost, specifically: 1,536 relevant sentences. Depending on the requirements of any given real-time streaming summarisation system, and the computational complexity of computing a particular set of summarisation features, such a cost/benefit ratio may be desirable. The results in Table 7.3 allow us to answer our first research question. We conclude that it is effective to train a classifier to predict TREC-TS summary sentences, when under-sampling the majority class, and using a multinomial Naive Bayes classifier.

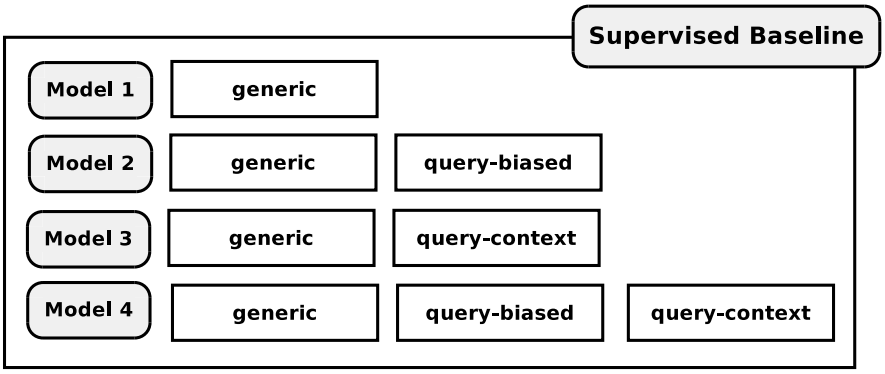



**Figure 7.2:** An illustration of the experimental setup for Research Question 7.2. Four different learned models are trained. One model consists of only generic summarisation features, another model consists of generic plus query-biased features, a further model consists of generic plus query-context features, and a final model consists of generic plus both query-biased and query-context features. We claim that our proposed query-biased summarisation features are effective, and seek to determine which models to take forward to our later experiments.

### Research Question 7.2

We now investigate Research Question 7.2, where we seek to determine if our proposed query-biased and query-context features are effective. We illustrate the experimental setup for Research Question 7.2 in Figure 7.2. Learned models are trained using four different feature sets: generic features; generic plus query-biased; generic plus query-context; and generic plus both query-biased and query-context. These feature groups were defined in Section 7.2. The expectation is that adding query-based features to a set of generic summarisation features will lead to improvements in supervised summarisation effectiveness. We first seek to validate that our proposed query-based features are effective, and further, we also seek to identify the most effective supervised baselines to take forward to future experiments.

We present the results of our experiments in Table 7.4, which reports the summarisation effectiveness of temporal summaries under the top-$k$ selection method, where $k = [1, 3, 5, 10]$ (i.e. fixed-length summaries per-hour). We report TREC-TS evaluation metrics: gain; comprehensiveness; and the mean of gain and comprehensiveness. Latency is also reported, but not directly discussed in our results. Within each top-$k$ group, we annotate the most effective scores in bold. Over all top-$k$ groups, we annotate the most effective scores using ✔. Further, statistically significant (paired Student's t-test, 95% confidence level) increases in summarisation effectiveness w.r.t the model trained on only generic features are indicated using ▲.

**Table 7.4:** Research Question 7.2 – TREC Temporal Summarisation (TREC-TS) results for non-entity supervised summarisation models. We report effectiveness scores for an SVM regression model (SVR), trained on ROUGE-2 precision labels, using 4 different feature sets: generic, generic plus query-biased, generic plus query-context, and generic plus query-biased and query-context. Further, we report the effectiveness of temporal summaries under the top-$k$ selection method, where $k = [1, 3, 5, 10]$ (i.e. fixed-length summaries per-hour). Within each top-$k$ group, we annotate the most effective scores in bold, and show the most effective scores across all top-$k$ groups as using the ✔symbol. Statistically significant (paired Student's t-test, 95% confidence level) increases in summarisation effectiveness w.r.t the model trained on only generic features are indicated using ▲.

| Learned Model | Top-$k$ | nE(Gain) | Comp. | HM(nE(LG),Lat.Comp.) | E(Latency) |
|---|---|---|---|---|---|
| Generic (baseline) | 1 | 0.1824 | 0.0771 | 0.0646 | 0.5378 |
| Generic+QueryBiased | 1 | 0.1967 | 0.1563▲ | **0.1372▲** | **0.8975** |
| Generic+QueryContext | 1 | **0.2391✔** | 0.1276▲ | 0.1117▲ | 0.8325 |
| Generic+QueryBiased+QueryContext | 1 | 0.1896 | **0.1684▲** | 0.1369▲ | 0.8367 |
| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
| Generic (baseline) | 3 | 0.1694 | 0.1584 | 0.0887 | 0.6936 |
| Generic+QueryBiased | 3 | 0.1541 | 0.2551▲ | 0.1514▲ | 1.0000 |
| Generic+QueryContext | 3 | **0.1807** | 0.2472▲ | **0.1607▲** | 1.0020 |
| Generic+QueryBiased+QueryContext | 3 | 0.1443 | **0.2836▲** | 0.1595▲ | **1.0084** |
| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
| Generic (baseline) | 5 | 0.1513 | 0.2041 | 0.1103 | 0.7630 |
| Generic+QueryBiased | 5 | 0.1427 | 0.3369▲ | **0.1833▲** | **1.1238** |
| Generic+QueryContext | 5 | **0.1545** | 0.3033▲ | 0.1818▲ | 1.0636 |
| Generic+QueryBiased+QueryContext | 5 | 0.1306 | **0.3393▲** | 0.1653▲ | 1.1067 |
| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
| Generic (baseline) | 10 | **0.1429** | 0.2812 | 0.1350 | 0.9277 |
| Generic+QueryBiased | 10 | 0.1073 | 0.4150▲ | 0.1716▲ | **1.1854✔** |
| Generic+QueryContext | 10 | 0.1224 | 0.3968▲ | **0.1940▲✔** | 1.1519 |
| Generic+QueryBiased+QueryContext | 10 | 0.1075 | **0.4188▲✔** | 0.1621 | 1.1578 |

From Table 7.4, we first observe that the most effective runs over all top-$k$ conditions, as indicated using the ✔symbol, are models that use query-based features. For example, Generic + QueryContext features at top-1 are the most effective under the gain metric, Generic + QueryBiased + QueryContext features at top-10 are the most effective under the comprehensiveness metric, and Generic + QueryContext features at top-10 are the most effective under the harmonic mean metric. Further, from Table 7.4, we observe that at top-$k$ conditions of 1, 3, and 5, query-based features are always most effective under gain, comprehensiveness, and harmonic mean (shown using bold annotation). At top-10, query-based features are most effective under comprehensiveness and mean, but the generic features are most effective un-

der the gain metric (the single case generic features outperformed query-based features). We note, as shown using bold annotation, the Generic + QueryBiased + QueryContext feature group always exhibits the most effective comprehensiveness scores. We further note, again shown using bold annotation, the Generic + QueryContext feature group is most effective under the gain metric at top-1, top-2, and top-5. Furthermore, from Table 7.4, we observe that using query-based features results in statistically significant increases in comprehensiveness and mean scores, under top-$k$ conditions of 1, 3, 5, and 10 – except for one case, Generic + QueryBiased + QueryContext features at top-10 under the mean metric, which however does exhibit a marked numerical increase (0.1350 to 0.1621).

The results in Table 7.4 allow us to answer our second research question. We conclude that our proposed query-biased and query-context features are effective, and that augmenting the generic feature group with such query-based features results in more effective temporal summaries of evolving news events. Based on the results in Table 7.4, we select the most effective query-based feature groups, shown in bold under the harmonic mean metric, to take forward to later experiments. Specifically, we use the Generic + QueryBiased features at top-1, the Generic + QueryContext features at top-3, the Generic + QueryBiased features at top-5, and the Generic + QueryContext features at top-10, in our later experiments.

### Research Question 7.3

We now investigate Research Question 7.3. Given the set of entity-focused event summarisation features that were proposed in Section 7.2, we seek to determine if our proposed temporal variants (Section 7.2.5) of the entity-based features are more effective than the batch variants (Section 7.2.4). We illustrate the experimental setup for Research Question 7.3 in Figure 7.3. Learned models are trained using different features sets, and the summarisation effectiveness of such learned models is examined. The expectation is that the temporal variants will be more effective than batch variants, for addressing the TREC Temporal Summarisation task.

Table 7.5 presents the results of our experiments. In Table 7.5, we report the summarisation effectiveness for learned models trained on entity-batch features, entity-temporal features, and a combination of entity-batch plus entity-temporal features. The effectiveness of temporal summaries is assessed under the top-$k$ selection method, where $k = [1,3,5,10]$ (i.e. fixed-length summaries per-hour). Within each top-$k$ group, the most effective scores are

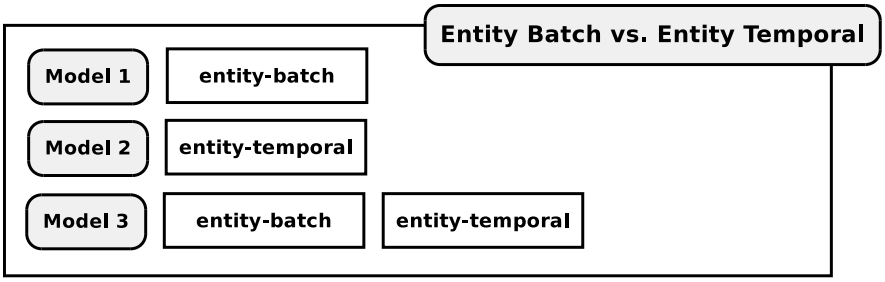



**Figure 7.3:** An illustration of the experimental setup for Research Question 7.3. Three different learned models are trained. One model consists of entity-batch features, another model consists of entity-temporal features, and a further model consists of entity-batch plus entity-temporal features. We claim that our proposed temporal entity-focused summarisation features are more effective for the TREC-TS task, compared to batch features.

shown using bold annotation. Further, the most effective scores across all top-$k$ groups is shown using the ✔ symbol. Furthermore, statistically significant (paired Student's t-test, 95% confidence level) increases in summarisation effectiveness w.r.t the entity-batch model are shown using the ▲ symbol.

From the results in Table 7.5, we first observe the most effective runs across all top-$k$ groups (shown using ✔). Under the gain metric, the entity-batch baseline is the most effective model (gain of 0.1507). This is the single case where entity-batch features alone are more effective than when using entity-temporal features. Indeed, under the comprehensiveness and harmonic mean metrics, we observe that the temporal variants are most effective across top-$k$ groups. Specifically, under comprehensiveness, the entity-batch + entity-temporal group at top-10 is the most effective model (comp. of 0.4895). and under the mean metric, the entity-temporal feature group at top-10 is the most effective model (mean of 0.1699).

From Table 7.5, we now observe the summarisation effectiveness of learned models within top-$k$ groups (shown using bold annotation). At top-1, temporal variants are more effective under comprehensiveness and mean, where (as previously noted) the batch variant is more effective under the gain metric at top-1. At top-3, top-5, and top-10, however, the temporal variants of our proposed entity-focused event summarisation features are always more effective than the batch variants, under the gain, comprehensiveness and mean metrics. We also note that, the combination of entity-batch and entity-temporal features is always most effective under the comprehensiveness metric, and that the entity-temporal feature group is most effective under the harmonic mean metric at top-3, top-5, and top-10.

**Table 7.5:** Research Question 7.3 – TREC Temporal Summarisation (TREC-TS) results for entity-focused supervised summarisation models. We report effectiveness scores for an SVM regression model (SVR), trained on ROUGE-2 precision labels, using 3 different feature sets: entity-batch, entity-temporal, and entity-batch plus entity-temporal. Further, we report the effectiveness of temporal summaries under the top-$k$ selection method, where $k = [1,3,5,10]$ (i.e. fixed-length summaries per-hour). Within each top-$k$ group, we show the most effective scores in bold, and the most effective scores across top-$k$ groups as ✔. Statistically significant (paired Student's t-test, 95% confidence level) increases in summarisation effectiveness w.r.t the entity-batch model are shown using the ▲ symbol.

| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
|---|---|---|---|---|---|
| EntityBatch (baseline) | 1 | **0.1507✔** | 0.2008 | 0.1272 | 0.8509 |
| EntityTemporal | 1 | 0.1463 | 0.1987 | 0.1399 | 0.9589 |
| EntityBatch+EntityTemporal | 1 | 0.1338 | **0.2616▲** | **0.1529▲** | **1.0349** |
| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
| EntityBatch (baseline) | 3 | 0.1207 | 0.3032 | 0.1462 | 0.9983 |
| EntityTemporal | 3 | **0.1270** | 0.3009 | **0.1624** | 1.0883 |
| EntityBatch+EntityTemporal | 3 | 0.1016 | **0.3774▲** | 0.1593 | **1.1744** |
| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
| EntityBatch (baseline) | 5 | 0.1041 | 0.3514 | 0.1427 | 1.0602 |
| EntityTemporal | 5 | **0.1220** | 0.3376 | **0.1675** | 1.1734 |
| EntityBatch+EntityTemporal | 5 | 0.0945 | **0.4297▲** | 0.1556 | **1.1921** |
| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
| EntityBatch (baseline) | 10 | 0.0989 | 0.4291 | 0.1533 | 1.1582 |
| EntityTemporal | 10 | **0.1031** | 0.4111 | **0.1699✔** | 1.2331 |
| EntityBatch+EntityTemporal | 10 | 0.0801 | **0.4895▲✔** | 0.1502 | **1.2610✔** |

Considering the statistical significance tests (shown using ▲), we observe that the entity-batch + entity-temporal feature combination is always (i.e. over all top-$k$ groups) significantly more effective than entity-batch alone under the comprehensiveness metric. Further, the entity-batch + entity-temporal feature combination is significantly more effective than entity-batch features alone under the harmonic mean metric at top-1. From the results in Table 7.5, we can now answer our third research question. We conclude that the temporal variants of our proposed entity-focused event summarisation features are more effective than the batch variants, when producing temporal summaries of evolving news events. Further, we conclude that the combination of batch and temporal entity-focused features is particularly effective at producing more comprehensive event summaries.

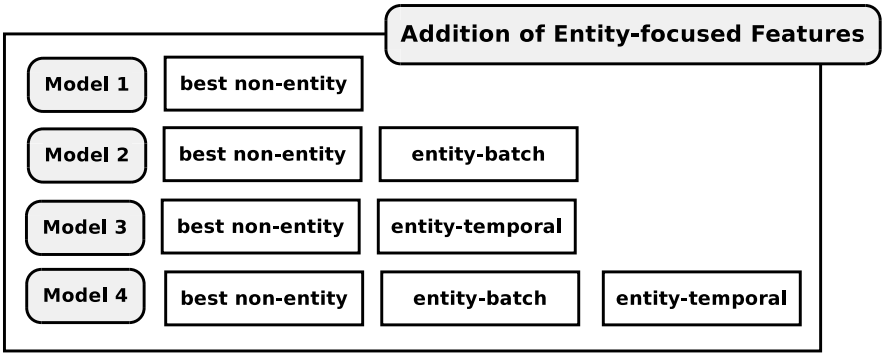



**Figure 7.4:** An illustration of the experimental setup for Research Question 7.4. Four different learned models are trained. One model consists of the best non-entity features, another model consists of the best non-entity features plus entity-batch features, a further model consists of the best non-entity features plus entity-temporal features, and a final model combines the best non-entity features plus both entity-batch and entity-temporal features. We claim that our proposed entity-focused event summarisation features are effective for the TREC-TS task, when used to augment standard document summarisation features.

### Research Question 7.4

We now address Research Question 7.4, where we seek to validate our claim (Hypothesis 4) that augmenting document summarisation features with entity-focused event summarisation features will result in more effective temporal summaries of evolving events. We illustrate the experimental setup for Research Question 7.4 in Figure 7.4. Learned models are trained with varying feature groups. We take the most effective non-entity learned models from Table 7.4, and train further supervised summarisation models with the addition of entity-focused event summarisation features. The expectation is that, by adding entity-focused event summarisation features, to standard document summarisation features, we will observe increases in summarisation effectiveness.

Table 7.6 presents the results of this experiment, which reports the summarisation effectiveness of temporal summaries under the top-$k$ selection method, where $k = [1, 3, 5, 10]$ (i.e. fixed-length summaries per-hour). We report the following TREC-TS evaluation metrics: gain; comprehensiveness; and the mean of gain and comprehensiveness. Within each top-$k$ group, we annotate the most effective scores in bold. Over all top-$k$ groups, we annotate the most effective scores using ✔. Further, statistically significant (paired Student's t-test, 95% confidence level) increases in summarisation effectiveness w.r.t the most effective supervised model from previous experiments are indicated using ▲.

**Table 7.6:** Research Question 7.4 – TREC Temporal Summarisation (TREC-TS) results for entity-focused supervised summarisation models. We report effectiveness scores for SVM regression models (SVR), trained on ROUGE-2 precision labels. The supervised baselines are the best non-entity runs from Table 7.4, which are compared to three variants of entity-based learned models (varying feature groups). Further, we report the effectiveness of temporal summaries under the top-$k$ selection method, where $k = [1,3,5,10]$ (i.e. fixed-length summaries per-hour). Within each top-$k$ group, we annotate the most effective scores in bold, and show the most effecive scores across all top-$k$ groups as using the ✔symbol. Statistically significant (paired Student's t-test, 95% confidence level) increases in summarisation effectiveness w.r.t the baseline model (trained on non-entity features) are shown using the ▲ symbol.

| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
|---|---|---|---|---|---|
| Generic+QueryBiased (baseline) | 1 | **0.1967✔** | 0.1563 | 0.1372 | 0.8975 |
| +EntityBatch | 1 | 0.1698 | 0.1821 | 0.1146 | 0.7847 |
| +EntityTemporal | 1 | 0.1743 | **0.1975** | **0.1690** | **1.0527** |
| +EntityBatch+EntityTemporal | 1 | 0.1873 | 0.0936 | 0.0802 | 0.8383 |
| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
| Generic+QueryContext (baseline) | 3 | **0.1807** | 0.2472 | 0.1607 | 1.0020 |
| +EntityBatch | 3 | 0.1723 | 0.2682 | 0.1768 | 0.9654 |
| +EntityTemporal | 3 | 0.1641 | 0.2156 | 0.1394 | 0.9572 |
| +EntityBatch+EntityTemporal | 3 | 0.1508 | **0.3184▲** | **0.1955▲✔** | **1.1438** |
| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
| Generic+QueryBiased (baseline) | 5 | 0.1427 | 0.3369 | 0.1833 | 1.1238 |
| +EntityBatch | 5 | 0.1233 | 0.3253 | 0.1412 | 1.0010 |
| +EntityTemporal | 5 | 0.1154 | **0.3949▲** | **0.1914** | **1.1976** |
| +EntityBatch+EntityTemporal | 5 | **0.1459** | 0.2274 | 0.1606 | 1.1041 |
| **Learned Model** | **Top-$k$** | **nE(Gain)** | **Comp.** | **HM(nE(LG),Lat.Comp.)** | **E(Latency)** |
| Generic+QueryContext (baseline) | 10 | 0.1224 | 0.3968 | **0.1940** | 1.1519 |
| +EntityBatch | 10 | **0.1293** | 0.3845 | 0.1769 | 1.1477 |
| +EntityTemporal | 10 | 0.1211 | 0.3586 | 0.1554 | 1.0791 |
| +EntityBatch+EntityTemporal | 10 | 0.0952 | **0.4534▲✔** | 0.1667 | **1.2107✔** |

From Table 7.6, we first note the most effective runs over all top-$k$ groups (shown as ✔). Under the gain metric, we observe that the baseline (i.e. non-entity model) at top-1 is the most effective (gain 0.1967). Under the comprehensiveness metric, wee see that the entity-batch + entity-temporal combination at top-10 is the most effective (comp. 0.4534). Further, under the harmonic mean metric, we observe that the entity-batch + entity-temporal combination at top-3 is the most effective (mean 0.1955). When examining the effectiveness at different top-$k$ conditions, from Table 7.6, we observe that entity-based runs are most effective: under the gain metric at top-5 and top-10; under the comprehensiveness metric at all top-$k$ conditions;

and under the harmonic mean metric at top-1, top-3, and top-5. Considering the statistical significance tests (shown using ▲), we observe that entity-based features significantly outperform non-entity models at top-3 under the harmonic mean metric. Further, entity-based models significantly outperform non-entity models under the comprehensiveness metric at the top-3, top-5, and top-10 conditions.

The results in Table 7.6 allow us to answer our fourth research question, and validate Hypothesis 4 from our Thesis Statement (Section 1.2). Specifically, we can conclude that augmenting document summarisation features, with entity-focused event summarisation features, has led to marked and significant improvements in summarisation effectiveness. In particular, entity-based features allow us to produce more comprehensive summaries of evolving news events, when compared to using only document summarisation features.

### Research Question 7.5

We now address Research Question 7.5, where we seek to validate our claim (Hypothesis 5) that entity-based evidence can be used to control the volume of sentences emitted over time, to form a temporal summary of an evolving news event, and that utilising such entity-focused anti-redundancy techniques will result in more effective summaries. The experimental setup for Research Question 7.5 is in contrast to previous supervised summarisation experiments. In particular, we now move from fixed-length summary selection, to a varying-length summary selection. Specifically, the number of sentences emitted by the system in each hour will now vary over the event timelline. As such, we now additionally introduce the ROUGE-1 precision metric, for evaluating summaries of varying lengths (Lin, 2004). The entity-focused anti-redundancy techniques investigated in this research question are defined in Section 7.2.6.

Table 7.7 presents the results from this experiment. We report the effectiveness of temporal summaries under the top-$k$ selection method, where $k = [1, 3, 5, 10]$. However, in this experiment, the top-$k$ sentences are a sample, which is passed through an anti-redundancy component. The fixed-length baselines reported in Table 7.7 are the most effective non-redundancy filtered entity-focused runs from Table 7.6, which are now subjected to entity-based anti-redundancy filtering methods. Additionally, an oracle method is reported, that returns all relevant sentences (observed via the TREC-TS relevance assessments) at each time period (i.e. worst-case redundancy). We report TREC-TS evaluation metrics: gain; compre-

**Table 7.7:** Research Question 7.5 – TREC Temporal Summarisation (TREC-TS) results for entity-focused supervised summarisation models. We report effectiveness scores for SVM regression models (SVR), trained on ROUGE-2 precision labels. The fixed-length baselines are the most effective non-redundancy filtered entity-focused runs from Table 7.6, which are now subjected to entity-based anti-redundancy filtering methods. Additionally, an oracle method is reported, that returns all relevant sentences (observed via the qrels) at each time period (i.e. worst-case redundancy). Further, we report the effectiveness of temporal summaries under the top-$k$ selection method, where $k = [1, 3, 5, 10]$, but the summary length varies per-hour due to anti-redundancy filtering (unlike previous experiments). Within each top-$k$ group, we annotate the most effective scores in bold. For the TREC-TS metrics, statistically significant (paired Student's t-test, 95% confidence level) increases in summarisation effectiveness w.r.t the baseline (non-redundancy filtered) model are shown using the ▲ symbol. For the ROUGE-1 precision (R1P) metric, we use the △ symbol to indicate that the 95% confidence interval of a given run is not overlapping with the 95% confidence interval of the baseline.

| Approach | Top-$k$ | nE(Gain) | Comp. | HM(nE(LG),Lat.Comp.) | E(Latency) | R1P | 95% conf. int. |
|---|---|---|---|---|---|---|---|
| Oracle (baseline) | – | 0.0527 | **0.7035** | 0.1114 | **1.4566** | 0.1365 | 0.0875 – 0.1928 |
| OneNewEntity | – | 0.1176▲ | 0.5338 | 0.2267▲ | 1.3886 | 0.2742△ | 0.2132 – 0.3427 |
| OneNewEntity (+Cosine) | – | **0.1185**▲ | 0.5307 | **0.2283**▲ | 1.3887 | **0.2768**△ | 0.2158 – 0.3448 |
| NewOrHotEntities | – | 0.0646▲ | 0.6535 | 0.1287▲ | 1.4138 | 0.1635 | 0.1091 – 0.2252 |
| NewOrHotEntities (+Cosine) | – | 0.0742▲ | 0.6476 | 0.1502▲ | 1.4173 | 0.1814 | 0.1261 – 0.2421 |

| Approach | Top-$k$ | nE(Gain) | Comp. | HM(nE(LG),Lat.Comp.) | E(Latency) | R1P | 95% conf. int. |
|---|---|---|---|---|---|---|---|
| Fixed-length (baseline) | 1 | 0.1743 | **0.1975** | **0.1690** | **1.0527** | 0.2497 | 0.1976 – 0.3060 |
| OneNewEntity | 1 | **0.2327**✔ | 0.1229 | 0.1482 | 1.0524 | 0.3619 | 0.3049 – 0.4219 |
| OneNewEntity (+Cosine) | 1 | **0.2327**✔ | 0.1229 | 0.1483 | 1.0524 | **0.3620** | 0.3050 – 0.4219 |
| NewOrHotEntities | 1 | 0.1836 | 0.1694 | 0.1598 | 1.0463 | 0.3125 | 0.2557 – 0.3742 |
| NewOrHotEntities (+Cosine) | 1 | 0.1891 | 0.1672 | 0.1607 | 0.9926 | 0.3252 | 0.2667 – 0.3856 |

| Approach | Top-$k$ | nE(Gain) | Comp. | HM(nE(LG),Lat.Comp.) | E(Latency) | R1P | 95% conf. int. |
|---|---|---|---|---|---|---|---|
| Fixed-length (baseline) | 3 | 0.1508 | **0.3184** | **0.1955**✔ | **1.1438** | 0.1441 | 0.1041 – 0.1866 |
| OneNewEntity | 3 | **0.1917** | 0.1322 | 0.1319 | 1.0581 | 0.2782△ | 0.2278 – 0.3328 |
| OneNewEntity (+Cosine) | 3 | **0.1917** | 0.1322 | 0.1319 | 1.0581 | **0.2793**△ | 0.2289 – 0.3351 |
| NewOrHotEntities | 3 | 0.1559 | 0.2539 | 0.1673 | 1.0829 | 0.2130 | 0.1628 – 0.2633 |
| NewOrHotEntities (+Cosine) | 3 | 0.1652 | 0.2489 | 0.1719 | 1.0734 | 0.2240 | 0.1741 – 0.2735 |

| Approach | Top-$k$ | nE(Gain) | Comp. | HM(nE(LG),Lat.Comp.) | E(Latency) | R1P | 95% conf. int. |
|---|---|---|---|---|---|---|---|
| Fixed-length (baseline) | 5 | 0.1154 | **0.3949**✔ | 0.1914 | **1.1976**✔ | 0.0937 | 0.0636 – 0.1283 |
| OneNewEntity | 5 | 0.1391 | 0.1794 | 0.1538 | 1.1678 | 0.2168△ | 0.1698 – 0.2671 |
| OneNewEntity (+Cosine) | 5 | **0.1393** | 0.1778 | 0.1538 | 1.1687 | **0.2175**△ | 0.1707 – 0.2678 |
| NewOrHotEntities | 5 | 0.1242 | 0.3241 | 0.1855 | 1.1765 | 0.1490 | 0.1052 – 0.1944 |
| NewOrHotEntities (+Cosine) | 5 | 0.1335▲ | 0.3132 | **0.1940** | 1.1638 | 0.1621 | 0.1165 – 0.2082 |

| Approach | Top-$k$ | nE(Gain) | Comp. | HM(nE(LG),Lat.Comp.) | E(Latency) | R1P | 95% conf. int. |
|---|---|---|---|---|---|---|---|
| Fixed-length (baseline) | 10 | 0.1293 | **0.3845** | **0.1769** | **1.1477** | 0.0713 | 0.0457 – 0.1024 |
| OneNewEntity | 10 | 0.1601 | 0.1574 | 0.1300 | 1.0767 | 0.1835△ | 0.1402 – 0.2286 |
| OneNewEntity (+Cosine) | 10 | **0.1670**▲ | 0.1616 | 0.1313 | 1.0646 | **0.1849**△ | 0.1415 – 0.2296 |
| NewOrHotEntities | 10 | 0.1450 | 0.3137 | 0.1669 | 1.0989 | 0.1311 | 0.0916 – 0.1738 |
| NewOrHotEntities (+Cosine) | 10 | 0.1529▲ | 0.3011 | 0.1746 | 1.0879 | 0.1411 | 0.1006 – 0.1849 |

hensiveness; and the mean of gain and comprehensiveness. Further, we now report ROUGE-1 precision, with confidence intervals. Within each top-$k$ group, we annotate the most effective scores in bold. Over all top-$k$ groups, we annotate the most effective scores using ✔. Further, statistically significant (paired Student's t-test, 95% confidence level) increases in summarisation effectiveness w.r.t the non-redundancy filtered baseline is indicated using ▲.

We first examine the oracle run. From Table 7.7, we observe that the non-redundancy filtered oracle baseline run achieves very high comprehensiveness scores (i.e. recall) but very low scores under the gain metric (i.e. precision). This demonstrates the trade-off between recall and precision, when returning every relevant sentence in the TRECTS-RelOnly corpus. While such an (unrealistic) approach offers very comprehensive summaries, there is far too much content for a user to consume, hence the poor scores under the gain metric. When we apply our proposed entity-focused anti-redundancy techniques to the oracle baseline, we reduce (i.e. filter) the volume of sentences that are emitted over time to form the temporal summary. From Table 7.7, we can observe the behaviour of summarisation evaluation metrics when this anti-redundancy condition is applied. Specifically, we see statistically significant improvements in the gain metric. We further observe statistically significant improvements in the harmonic mean metric. Further, as shown by ROUGE-1 precision, we observe improvements in scores where there are non-overlapping confidence intervals (shown using △). From the results over the oracle run, we can conclude that our proposed entity-focused anti-redundancy filtering techniques enable us to produce more effective summaries of evolving news events, where we specifically increase the precision of the summaries, i.e. reduce the burden on the user reading the summaries.

We now examine the application of our proposed entity-focused anti-redundancy techniques to the most effective non-redundancy filtered entity-focused runs from Table 7.6. From Table 7.7, we observe that the TREC-TS gain metric and the ROUGE-1 precision metric show marked numerical increases when applying entity-focused anti-redundancy filtering to non-redundancy filtered baselines. Considering the approaches we have evaluated, at top-1 and top-3, OneNewEntity appears to be more effective under the gain metric, but at top-5 and top-10, the difference between the two approaches is less obvious. Under the ROUGE-1 precision metric, however, there is clear indication that the OneNewEntity approach is more effective.

Our claim is that by applying the proposed entity-focused anti-redundancy techniques, we can produce more effective summaries of evolving events. The results in Table 7.7 allow us to validate this claim. In particular, we observe marked numerical increases in gain and precision metrics, when applying the entity-based filtering techniques. Specifically, we observe statistically significant improvements under the gain metric for OneNewEntity (+Cosine) at top-5, for OneNewEntity (+Cosine) at top-10, and for NewOrHotEntities (+Cosine) at top-10. Further, from Table 7.7, we observe that the OneNewEntity approach improves the ROUGE-1 precision score vs. the non-filtered baselines at the top-3, top-5, and top-10 conditions, such that the improvements in precision scores exhibit non-overlapping confidence intervals with the baseline run. As such, we conclude that our proposed entity-focused anti-redundancy techniques can be used to produce more effective summaries of evolving events, where we specifically improve the precision of the summaries.

### 7.3.4 Discussion & Analysis

#### On the Use of Quantifying Features for Supervised Temporal Summarisation

In Section 7.2, we defined the summarisation features evaluated within this chapter. In several cases, we proposed features that, we argued, should act as quantifying features useful for training supervised summarisation models within the TREC-TS task. In particular, when training supervised models for the temporal summarisation task, the cross-batch (hour-by-hour) scores provided to the learner may not be directly comparable. Specifically, per-sentence scores for various summarisation features (baselines and entity-focused features) will be a function of the number of sentences and terms in any given batch. We now return to this discussion, and seek to identify if such quantification features are indeed important within supervised models.

Our analysis is conducted using the GBRT (Chen and Guestrin, 2016) (tree-based) model, due to the interpretability of such machine learning techniques (Hastie et al., 2009). In Figure 7.5 and Figure 7.6, we show GBRT feature importance plots. Figure 7.5 shows the generic features group (c.f. Section 7.2.1), and Figure 7.6 shows the entity-temporal feature group (c.f. Section 7.2.5). Feature importance is shown on the $x$-axis, with individual features shown on the $y$-axis. The model is trained on ROUGE-2 precision labels, using the TRECTS-RelOnly training data. The feature importance score is the frequency of occurrence of that feature over the boosted decision trees within the model, i.e. the number of times that the feature

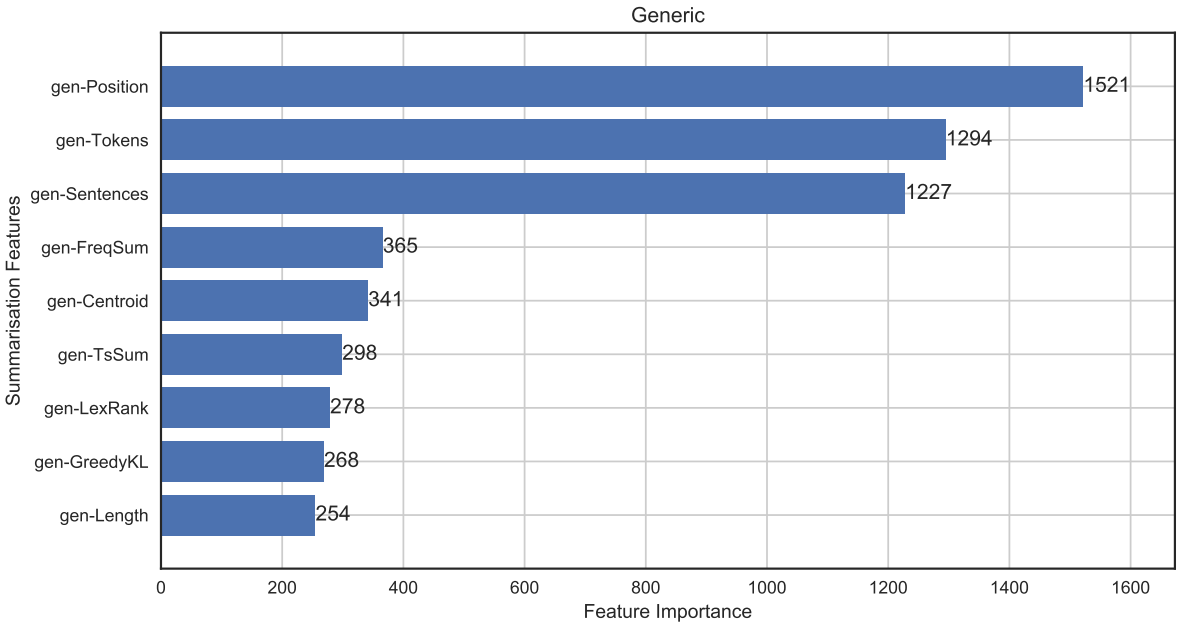



**Figure 7.5:** Feature importance plots, under the Gradient Boosted Regression Trees (GBRT) model, trained on ROUGE-2 precision labels, and generic features (c.f. Section 7.2.1). We examine the effectiveness of quantification features such as "gen-Tokens" and "gen-Sentences".

contributes to the branches of the decision trees within the model.

From Figure 7.5, we first observe that the most important generic summarisation feature is "gen-Position", which is a lead-based feature (c.f. Section 4.1.2). As discussed in Section 7.1, a lead-based feature was used in the most effective temporal summarisation system (Raza et al., 2015) at the TREC-TS 2015 track Aslam et al. (2015). Further, from Figure 7.5, we observe that the "gen-Tokens" and "gen-Sentences" quantification features are the next most important features under the GBRT learned model. Furthermore, from Figure 7.6, we observe that the "et-EventBatches", "et-EventSentences", and "et-TotalEntities" quantification features are the three most important features within the entity-temporal feature group, under the GBRT model. From these observations, we can infer that such quantification features are indeed useful when training supervised summarisation models over the TREC-TS task. In particular, such features provide additional information to the learner that allows for the numerical quantification of cross-batch features scores of summarisation algorithms.

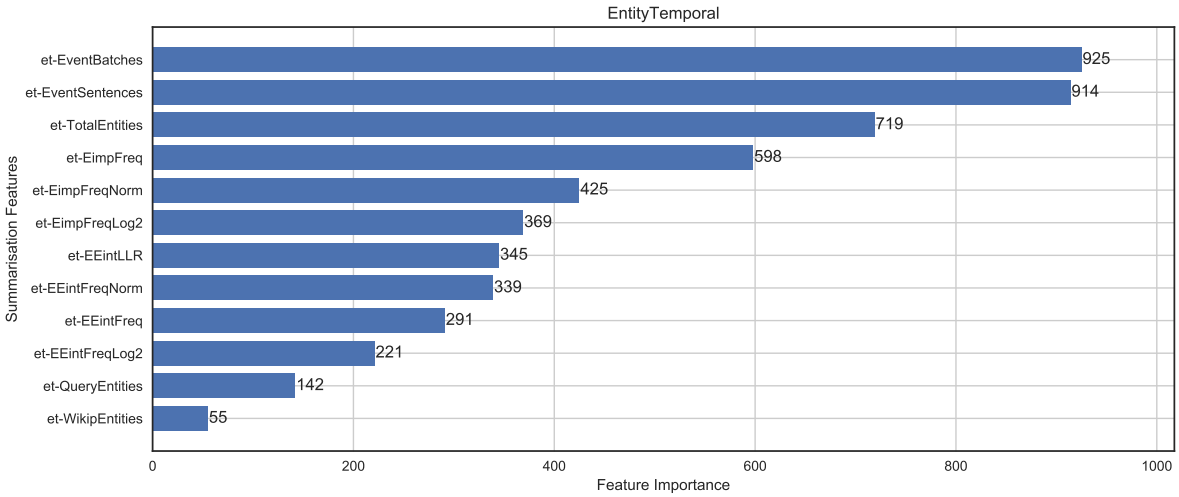



**Figure 7.6:** Feature importance plots, under the Gradient Boosted Regression Trees (GBRT) model, trained on ROUGE-2 precision labels and entity-temporal features (c.f. Section 7.2.5). We examine the effectiveness of quantification features, such as "et-EventBatches", "et-EventSentences", and "et-TotalEntities".

## 7.4 Chapter Summary

In this chapter, we investigated the use of entity-based evidence to improve learned temporal summarisation models that are trained on document summarisation features. We provided experimental results to empirically validate Hypothesis 4 and Hypothesis 5 from our Thesis Statement (Section 1.2). We validated our claim that adding entity-based evidence to learned models trained on baseline document summarisation features leads to increases in temporal summarisation effectiveness. Further, we validated our claim that entity-based evidence can be used as a means to control the number of sentences emitted into a temporal summary of en evolving news event. By answering Research Question 7.1, we demonstrated that a classifier can be trained to reduce the number of input sentences to be summarised. By answering Research Question 7.2, we demonstrated that query-biased summarisation features derived from sentence retrieval scores are effective. By answering Research Question 7.3, we demonstrated that temporal variants of our proposed entity-focused event summarisation features were effective. By answering Research Question 7.4, we demonstrated that augmenting document summarisation features with entity-focused event summarisation features leads to an increase in supervised summarisation effectiveness. By answering Research Question 7.5, we demonstrated that varying the number of sentences emitted over time, using entity-based evidence, can lead to more effective temporal summaries.

In conclusion, entity-focused event summarisation features are effective for the task of temporal summarisation. In particular, entity-importance, entity–entity interaction, and entity-event relevance features are effective for use in supervised machine learned summarisation models, which are used for summarising evolving news events. Further, we conclude that using a classifier to pre-filter temporal summarisation document streams is an effective method to reduce the number of summarisation sentences that must be processed by temporal summarisation systems. We also conclude that it is important to use query-biased summarisation features for the TREC Temporal Summarisation task, and that query-biased features derived from sentence retrieval methods are effective. Furthermore, using entity-based anti-redundancy techniques can result in more precise summaries of evolving news events.

# Chapter 8

# Conclusions

More effective event summaries, better assisting people with their news-based information access requirements, can help to reduce information overload in today's 24-hour news culture. This thesis demonstrated that evidence about named entities (i.e. people, places, and organisations) involved in news-worthy events can be used to effectively summarise such news events. In particular, within a supervised machine learning framework, we proposed a series of effective entity-focused event summarisation features. Such entity-focused features estimate: the importance of entities within events; the significance of interactions between entities within events; and the topical relevance of entities to events. By augmenting supervised summarisation models, trained on discriminative multi-document newswire summarisation features, with evidence about the named entities involved in the events, we produced more effective summaries of news-worthy events. The proposed entity-focused event summarisation features were evaluated over two multi-document newswire summarisation scenarios, the retrospective event summarisation task, and the temporal event summarisation task.

The contributions of this thesis are two-fold. First, this thesis demonstrated the effectiveness of entity-focused event evidence for identifying important and salient event summary sentences, and as a means to control the volume of content emitted as a summary of an evolving event. Second, this thesis demonstrated the validity of automatic summarisation evaluation metrics, the effectiveness of standard summarisation baselines, and the effective training of supervised machine learned summarisation models.

## 8.1 Summary of Contributions

The contributions of this thesis are as follows:

**Chapter 3**

- We conducted a crowd-sourced user-study confirming and quantifying the validity of automatic summarisation evaluation metrics.
- We showed that automatic summarisation evaluation metrics exhibit strong rank correlation with non-expert crowd-sourced manual judgements for the linguistic quality of a summary text.

**Chapter 4**

- We showed that the commonly used lead-based baseline can be significantly improved via the addition of anti-redundancy filtering.
- We re-implemented several standard baselines, demonstrating that the effectiveness of such baselines can be markedly improved by thoroughly exploring algorithm design choices.

**Chapter 5**

- We investigated a set of standard summarisation baselines for use as effective features within supervised summarisation models.
- We investigated labelling supervised summarisation training data using Kullback-Leibler divergence, Jensen-Shannon divergence, sentence retrieval scores, and ROUGE-*n* precision scores.
- We investigated a range of linear and non-linear regression-based learners for the newswire summarisation task.
- We provided evidence that several combinations of such features, labels, and learners achieve state-of-the-art effectiveness for the task of generic extractive multi-document newswire summarisation.

**Chapter 6**

- We demonstrated that entity-based features are effective for training supervised summarisation models, when combined with document summarisation features.
- We also demonstrate that named entity linking is an effective method for deriving such entity-focused event summarisation features.

**Chapter 7**

- We showed that a classifier can be trained to effectively pre-filter (i.e. reduce) the number of sentences taken as input to the temporal summarisation task.
- We demonstrated that a range of query-biased summarisation features derived from sentence retrieval techniques are effective.
- We demonstrated that entity-based features are effective for training supervised temporal summarisation models, when combined with document summarisation features.
- We investigated temporal variants of entity-based features, demonstrating the such temporal variants are effective for the temporal summarisation task.
- We also demonstrated the utility of entity-focused anti-redundancy techniques, for controlling the number of sentences emitted as a summary of an evolving event.

## 8.2 Summary of Conclusions

The main conclusions of this thesis are as follows.

Automatic summarisation evaluation methods, despite their apparent bluntness and common criticisms, are reasonably aligned with user expectations of summary quality. As such, automatic summarisation evaluation metrics therefore remain useful proxies for manual evaluation for measuring summarisation effectiveness, particularly during system research and development stages. Further, to provide stronger baselines for the empirical evaluation of newswire summarisation systems, it is advisable to explore algorithm design choices of such baselines. Furthermore, applying anti-redundancy filtering to the standard lead-based newswire summarisation baseline results in a significantly stronger baseline.

State-of-the-art supervised extractive multi-document newswire summarisation models can be trained on standard baseline features, using linear and non-linear regression-based learners, when obtaining high-quality training data – in particular via divergence based methods, sentence retrieval techniques, or ROUGE-*n* precision scores. Within a supervised machine learning framework, entity-based evidence is effective for summarising news events. In particular, augmenting standard document summarisation baselines with entity-focused event summarisation features leads to improvements in summarisation effectiveness over the retrospective summarisation task, and the temporal summarisation task. Moreover, applying entity-based anti-redundancy techniques results in improvements in the precision of temporal summaries of evolving news events.

## 8.3 Directions for Future Work

In this section, we outline three research directions for possible future work.

### Considering Cross-stream Entity Statistics

In this thesis, we examined summarisation within the context newswire streams from multiple providers. However, in the 24-hour news environment, social media is increasingly playing a prominent role in news consumption. Indeed, the widespread adoption of mobile devices in conjunction with always-on internet access now enables the general public to report news as it happens from on the ground via social media platforms. Further, social media allows traditional media outlets to rapidly disseminate news content to consumers. In the experiments in this thesis, we did not investigate the summarisation of content from such social media sources, such as microblogs (Mackie et al., 2014a). However, when summarising evolving news events from newswire sources only, we may miss aspects of the event that are only reported on social media platforms. Indeed, with respect to the entity-focused event summarisation approaches we have proposed in this thesis, by additionally including evidence from social media platforms, we may observe new entities and their interactions. Moreover, given the large volume of posts about events on social media, this could be a valuable resource to better estimate the important entities at any given point in time. Hence, a direction for future work would be to integrate entity-based evidence from social media platforms and

also examine how to normalise entity evidence from different stream types.

## Real-world Knowledge via Priors over Entities

In the experiments in this thesis, the estimation of entity-focused event summarisation features is based only on the observed statistics within the stream of summarisation documents. For example, given a set of 100 documents, we compute entity importance for each entity as if it did not exist prior to those documents being summarised. However, this assumption does not hold. For many entities, it may be possible to derive a prior background statistic, that reflects the expectation of how important a given entity is in the world. From an entity-focused event summarisation perspective, it seems intuitive that summarisation algorithms should incorporate this prior knowledge about the expected influence and importance of entities when selecting sentences that contain those entities for inclusion into a summary. Such background knowledge, or priors over entity importance and entity–entity interaction, could be computed from language resources such as Wikipedia or knowledge bases such as Wikidata or DBPedia. This may allow us to, for example, more accurately identify surprising (i.e. unexpected) interactions between entities, if such entity-entity interactions are significantly different from the prior expectations, allowing us to promote the selection of novel sentences, discussing those interesting entities, that would otherwise be ignored. Therefore, a direction for future work would be to examine the integration of knowledge base entity evidence into the sentence scoring component of temporal summarisation systems.

## Tracking Event statistics over Time

A common phenomenon when reporting on news events is to include important numerical statistics, such as the number of people injured, or the monetary amount of damage in particular areas. However, as an event evolves, these values change over time as the event develops and new information becomes available. Current summarisation systems do not include components that track how these values change. Hence, this can cause problems when a value changes significantly, for example when a tropical storm makes landfall, the number of people injured, or damage to property will rise. However, as the summarisation system will have observed similar sentences in the past, regarding specific people and locations (i.e. entities), textual changes in numerical values only may not be classed as sufficiently novel

to warrant being included in the next temporal summary update. On the other hand, such small changes in the text of sentences representing values can represent a much larger societal impact that user would wish to be informed about in a summary of that event. Future work in this area might involve the investigation of methodologies to identify key values to be tracked, matching those values across multiple updates, and verifying ambiguous values when multiple sources disagree.

## 8.4 Closing Remarks

Given the volume of online coverage about news events, and the general public's intense interest in such events, automatic summarisation systems that effectively summarise events are becoming increasingly important and consumer-relevant. The work in this thesis has made a significant and interesting contribution to the supervised extractive summarisation of news events. We highlighted the importance of deriving supervised summarisation features that are specific to the domain of documents being summarised, i.e. as events are about entities, entity-focused event summarisation features are effective. We also demonstrated that label engineering is every bit as important as feature engineering when performing supervised machine learned summarisation.

# Bibliography


Afantenos, S., V. Karkaletsis, and P. Stamatopoulos (2005). Summarization from medical documents: a survey. *Artificial Intelligence in Medicine 33*(2), 157 – 177. Information Extraction and Summarization from Medical Documents. [2.2.3]

Afantenos, S. D., K. Liontou, M. Salapata, and V. Karkaletsis (2005). An Introduction to the Summarization of Evolving Events: Linear and Non-linear Evolution. In *Proceedings of the Workshop on Natural Language Understanding and Cognitive Science*, NLUCS '05. [7.1]

Aggarwal, C. C. and C. Zhai (2012). A Survey of Text Classification Algorithms. In *Mining text data*. Springer. [5.1, 7.3.3]

Allan, J. (2002). Introduction to Topic Detection and Tracking. In *Topic Detection and Tracking: Event-based Information Organization*. Springer. [2.1, 2.3.2, 2.3.3, 7.1]

Allan, J., R. Gupta, and V. Khandelwal (2001). Temporal Summaries of News Topics. In *Proceedings of the Conference on Research and Development in Information Retrieval*, SIGIR '01. [7.1]

Allan, J., C. Wade, and A. Bolivar (2003). Retrieval and Novelty Detection at the Sentence Level. In *Proceedings of the Conference on Research and Development in Informaion Retrieval*, SIGIR '03. [4.2.2]

Amati, G. (2003). *Probability Models for Information Retrieval Based on Divergence From Randomness*. Ph. D. thesis, University of Glasgow. [5.3.2, 7.2.2]

Amati, G. et al. (2011). FUB, IASI-CNR, UNIVAQ at Microblogging Track of TREC 2011. In *Proceedings of the Text Retrieval Conference*, TREC '11. [5.3.2]

Amati, G. and C. J. van Rijsbergen (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *Transactions on Information Systems 20*(4). [5.3.2, 7.2.2]

Aone, C., M. E. Okurowski, and J. Gorlinsky (1998). Trainable, Scalable Summarization using Robust NLP and Machine Learning. In *Proc. of the Conference on Computational Linguistics*, COLING '98. [2.4, 5]

Artstein, R. and M. Poesio (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics 34*(4). [3.3.3]

Aslam, J., F. Diaz, M. Ekstrand-Abueg, R. McCreadie, V. Pavlu, and T. Sakai (2014). TREC 2014 Temporal Summarization Track Overview. In *Proceedings of the Text Retrieval Conference*, TREC '14. [2.1, 2.2.1, 2.3.3, 6, 7, 7.1, 7.2, 7.3, 7.3.2]

Aslam, J., F. Diaz, M. Ekstrand-Abueg, R. McCreadie, V. Pavlu, and T. Sakai (2015). TREC 2015 Temporal Summarization Track Overview. In *Proceedings of the Text Retrieval Conference*, TREC '15. [2.1, 2.2.1, 2.3.3, 6, 7, 7.1, 7.1, 7.2, 7.3, 7.3.2, 7.3.4]

Aslam, J., F. Diaz, M. Ekstrand-Abueg, V. Pavlu, and T. Sakai (2013). TREC 2013 Temporal Summarization. In *Proceedings of the Text Retrieval Conference*, TREC '13. [1, 2.1, 2.2.1, 2.3.3, 6, 7, 7.1, 7.2, 7.3]

Augenstein, I., L. Derczynski, and K. Bontcheva (2017). Generalisation in Named Entity Recognition: a Quantitative Analysis. *Computer Speech & Language 44*. [6.1]

Balasubramanian, N., J. Allan, and W. B. Croft (2007). A Comparison of Sentence Retrieval Techniques. In *Proceedings of the Conference on Research and Development in Information Retrieval*, SIGIR '07. [5.3.2, 7.2.2]

Banko, M., V. O. Mittal, and M. J. Witbrock (2000). Headline generation based on statistical translation. In *38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, October 1-8, 2000*. ACL. [2.2.3]

Baruah, G., R. Guttikonda, A. Roegiest, and O. Vechtomova (2013). University of waterloo at the TREC 2013 temporal summarization track. In *Proceedings of the Text Retrieval Conference*, TREC '13. [7.1]

Barzilay, R. and K. McKeown (2005). Sentence Fusion for Multidocument News Summarization. *Computational Linguistics 31*(3). [2.1]

Büttcher, S., C. L. A. Clarke, and G. V. Cormack (2010). *Information Retrieval - Implementing and Evaluating Search Engines*. MIT Press. [2, 2.4.1, 3.1.1, 4.2.1, 5.3.2, 5.3.2, 5.3.2, 6.2.1, 7.1, 7.2.2]

Cao, Z., F. Wei, L. Dong, S. Li, and M. Zhou (2015). Ranking with Recursive Neural Networks and Its Application to Multi-Document Summarization. In *Proc. of the AAAI Conference on Artificial Intelligence*, AAAI '15. [5.2, 5.3.3]

Capannini, G., C. Lucchese, F. M. Nardini, S. Orlando, R. Perego, and N. Tonellotto (2016). Quality versus Efficiency in Document Scoring with Learning-To-Rank Models. *Information Processing & Management 52*(6). [5.5.2]

Carbonell, J. G. and J. Goldstein (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the Conference on Research and Development in Information Retrieval*, SIGIR '98. [5.2, 7.2.6]

Carver, L. and M. Turoff (2007). Human-computer Interaction: The Human and Computer As a Team in Emergency Management Information Systems. *Communications of the ACM 50*(3). [1, 2.2.2]

Chali, Y. and S. A. Hasan (2012). Query-focused Multi-document Summarization: Automatic Data Annotations and Supervised Learning Approaches. *Natural Language Engineering 18*(1). [5.2]

Chali, Y., S. A. Hasan, and S. R. Joty (2009). Do Automatic Annotation Techniques Have Any Impact on Supervised Complex Question Answering? In *Proc. of the Joint conference of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing*, ACL-IJCNLP '09. [5.2, 5.3.3]

Chang, C.-C. and C.-J. Lin (2011). LIBSVM: A Library for Support Vector Machines. *Transactions on Intelligent Systems and Technology 2*. [5.1, 5.5.2, 6.3.2]

Chen, T. and C. Guestrin (2016). XGBoost: a Scalable Tree Boosting System. In *Proceedings of the Conference on Knowledge Discovery and Data Mining*, KDD '16. [5.1, 5.5.2, 5.5.4, 7.3.4]

Cheng, J. and M. Lapata (2016). Neural Summarization by Extracting Sentences and Words. In *Proceedings of the Association for Computational Linguistics*, ACL '16. [5.2]

Clarke, J. and M. Lapata (2007). Modelling Compression with Discourse Constraints. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07. [2.1]

Conroy, J. M. and H. T. Dang (2008). Mind the Gap: Dangers of Divorcing Evaluations of Summary Content from Linguistic Quality. In *Proceedings of the Conference on Computational Linguistics*, COLING '08. [2.3.1]

Conroy, J. M., J. D. Schlesinger, J. Goldstein, and D. P. OâĂŹleary (2004). Left-brain / Right-brain Multi-document Summarization. In *Proceedings of the Document Understanding Conference*, DUC '04. [2.2, 2.4]

Conroy, J. M., J. D. Schlesinger, J. Kubina, P. A. Rankel, and D. P. O'Leary (2011). CLASSY 2011 at TAC: guided and multi-lingual summaries and evaluation metrics. In *Proceedings of the Text Analysis Conference*, TAC '11. [2.2, 2.4]

Conroy, J. M., J. D. Schlesinger, and D. P. O'Leary (2006). Topic-focused Multi-document Summarization Using an Approximate Oracle Score. In *Proceedings of the Conference on Computational Linguistics and Association for Computational Linguistics*, COLING-ACL '06. [2.2, 2.4.1, 4.2.1, 4.2.1, 5.4.1, 5.5.3]

Croft, W. B., D. Metzler, and T. Strohman (2010). *Search Engines - Information Retrieval in Practice*. Pearson. [2, 2.4.1, 3.1.1, 4.2.1, 5.3.2, 5.3.2, 5.3.2, 6.2.1, 7.1, 7.2.2]

Dang, H. and K. Owczarzak (2008). Overview of the TAC 2008 Update Summarization Task. In *Proceedings of the Text Analysis Conference*, TAC '08. [2.1, 2.3.1, 7.1]

Dang, H. T. (2005). Overview of duc 2005. In *Proceedings of the Document Understanding Conference*, Volume 2005. [2.3.1]

Davis, S. T., J. M. Conroy, and J. D. Schlesinger (2012). OCCAMS – An Optimal Combinatorial Covering Algorithm for Multi-document Summarization. In *Proceedings of the International Conference on Data Mining*, ICDM '12. [2.2, 2.4]

Demner-Fushman, D., W. W. Chapman, and C. J. McDonald (2009). What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics 42*(5), 760 – 772. Biomedical Natural Language Processing. [2.2.3]

Doddington, G. R., A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. Strassel, and R. M. Weischedel (2004). The Automatic Content Extraction (ACE) Program - Tasks, Data, and Evaluation. In *Proceedings of Language Resources and Evaluation*, LREC '04. [7.1]

Dorr, B., D. Zajic, and R. Schwartz (2003). Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text Summarization Workshop - Volume 5*, HLT-NAACL-DUC '03, Stroudsburg, PA, USA, pp. 1–8. Association for Computational Linguistics. [2.2.3]

Dunning, T. (1993). Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics 19*(1). [7.2.4]

Edmunds, A. and A. Morris (2000). The Problem of Information Overload in Business Organisations: a Review of the Literature. *International Journal of Information Management 20*(1). [1]

Edmundson, H. P. (1969). New Methods in Automatic Extracting. *Journal of the ACM 16*(2). [2.1, 2.1]

Ekstrand-Abueg, M., R. McCreadie, V. Pavlu, and F. Diaz (2016). A Study of Realtime Summarization Metrics. In *Proceedings of the Conference on Information and Knowledge Management*, CIKM '16. [2.3.1, 2.3.3]

Ellouze, S., M. Jaoua, and L. H. Belguith (2016). Automatic Evaluation of a Summary's Linguistic Quality. In *Proceedings of Applications of Natural Language to Information Systems*, NLDB '16. [2.2.3, 2.3.1, 3.3.4]

Erkan, G. and D. R. Radev (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research 22*(1). [2.2.1, 2.4, 2.2, 2.4.1, 4.2.1, 4.2.1, 5.4.1, 5.5.3]

Fan, R., K. Chang, C. Hsieh, X. Wang, and C. Lin (2008). LIBLINEAR: a Library for Large Linear Classification. *Journal of Machine Learning Research 9*. [5.1, 5.5.2, 7.3.2]

Färber, M., B. Ell, C. Menne, and A. Rettinger (2015). A Comparative Survey of DBPedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web Journal*. [6.1]

Filippova, K. and M. Strube (2008). Sentence fusion via dependency graph compression. In *Proceedings of Empirical Methods in Natural Language Processing*, EMNLP '08. [2.1]

Finkel, J. R., T. Grenager, and C. D. Manning (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the Association for Computational Linguistics*, ACL '05. [6.1, 6.3.2]

Fisher, R. A. (1921). On the Probable Error of a Coefficient of Correlation Deduced From a Small Sample. *Metron 1*. [3.3.2, 3.3.3]

Friedman, J. H. (2001). Greedy Function Approximation: a Gradient Boosting Machine. *Annals of Statistics*. [6.3.2]

Galanis, D. and P. Malakasiotis (2008). AUEB at TAC 2008. In *Proc. the Text Analysis Conference*. [5.2, 5.3.3]

Galliers, K. S. J. &. J. R. (1997). Evaluating natural language processing systems: An analysis and review. *Machine Translation 12*(4). [2.3.1, 2.3.1]

Gatt, A. and E. Krahmer (2017). Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation. *CoRR abs/1703.09902*. [2.1]

Genest, P. and G. Lapalme (2012). Fully Abstractive Approach to Guided Summarization. In *Proceedings of the Association for Computational Linguistics*, ALC '12. [2.1]

Géron, A. (2017). *Hands on Machine Learning with scikit-learn and Tensorflow*. O'Reilly Media. [5.4.1, 7.3.2]

Gillick, D. and B. Favre (2009). A Scalable Global Model for Summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ACL ILP-NLP '09. [2.4, 2.2, 2.4, 6.2.2]

Gillick, D. and Y. Liu (2010). Non-expert Evaluation of Summarization Systems is Risky. In *Proceedings of the Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, NAACL HLT CSLDAMT '10. [2.3.1, 3.3.4]

Goldberg, Y. (2016). A Primer on Neural Network Models for Natural Language Processing. *Journal of Artificial Intelligence Research 57*. [2.1]

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press. [2.1]

Google (2016). Putting a Spotlight on Local News Sources. blog.google/topics/journalism-news/putting-spotlight-on-local-news-sources/. Google News Blog. [1]

Graham, Y. (2015). Re-evaluating Automatic Summarization with BLEU and 192 Shades of ROUGE. In *Proceedings of Empirical Methods in Natural Language Processing*, EMNLP '15. [2.3.1, 3, 3.3.4]

Grishman, R. and B. Sundheim (1996). Message understanding conference 6: A brief history. In *Proceedings of the Conference on Computational Linguistics*, COLING '96. [6.1]

Guo, Q., F. Diaz, and E. Yom-Tov (2013). Updating Users about Time Critical Events. In *Proceedings of the European Conference on Information Retrieval*, ECIR '13. [7.1]

Haghighi, A. and L. Vanderwende (2009). Exploring Content Models for Multi-document Summarization. In *Proceedings of the North American Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '09. [2.2, 2.4.1, 4.2.1, 4.2.1, 5.4.1, 5.5.3]

Harman, D. and P. Over (2004). The Effects of Human Variation in DUC Summarization Evaluation. In *Proceedings of the Association for Computational Linguistics*, ACL '04 Workshop: Text Summarization Branches Out. [2.2.1, 2.3.1]

Hastie, T., R. Tibshirani, and J. H. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer. [1, 1.1, 2.4, 5, 5.1, 5.1, 5.4, 5.5.3, 5.5.4, 6, 7.3.4]

Hiemstra, D. (2001). *Using Language Models for Information Retrieval*. Ph. D. thesis, University of Twente, Enschede, Netherlands. [5.3.2]

Hoffart, J., M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum (2011). Robust Disambiguation of Named Entities in Text. In *Proceedings of Empirical Methods in Natural Language Processing*, EMNLP '11. [6.1, 7.2.4, 7.3.2]

Holton, A. E. and H. I. Chyi (2012). News and the Overloaded Consumer: Factors Influencing Information Overload Among News Consumers. *Cyberpsychology, Behavior, and Social Networking 15*(11). [1]

Hong, K., J. Conroy, B. Favre, A. Kulesza, H. Lin, and A. Nenkova (2014). A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization. In *Proceedings of the Conference on Language Resources and Evaluation*, LREC '14. [1, 1.1, 2.1, 2.2.1, 2.2.3, 2.3.1, 2.4, 2.2, 2.4, 2.4.1, 3.1.1, 3.3.3, 3.3.3, 4, 4.2.1, 4.2.2, 4.3.2, 4.3.3, 4.3.3, 5, 5.2, 5.4, 5.4.1, 5.5.2, 5.5.2, 5.5.3, 5.4, 6.2.3, 6.3.2, 7.2.6]

Hong, K. and A. Nenkova (2014). Improving the Estimation of Word Importance for News Multi-Document Summarization. In *Proceedings of the European Association for Computational Linguistics*, EACL '14. [2.2, 2.4]

Hovy, E. and C.-Y. Lin (1998). Automated Text Summarization and the SUMMARIST System. In *Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, TIPSTER '98. [2.1]

Hovy, E., C.-Y. Lin, L. Zhou, and J. Fukumoto (2006). Automated Summarization Evaluation with Basic Elements. In *Proceedings of Language Resources and Evaluation*, LREC 2006. [2.3.1]

Jelinek, F. and R. Mercer (1980). Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proceedings of the Workshop on Pattern Recognition in Practice*. [4.2.1]

Jones, K. S. (1998). Automatic Summarizing: Factors and Directions. *Advances in Automatic Text Summarization*. [1, 2, 2.2, 2.2.2, 5.2]

Jones, K. S. (2007). Automatic Summarising: the State of the Art. *Information Processing & Management 43*(6). [1, 2, 2.4]

Jones, R., B. Rey, O. Madani, and W. Greiner (2006). Generating Query Substitutions. In *Proceedings of the Conference on World Wide Web*, WWW '06. [7.2.4]

Jurafsky, D. and J. H. Martin (2009). *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall. [2, 2.3.1]

Kedzie, C., F. Diaz, and K. McKeown (2016). Real-Time Web Scale Event Summarization Using Sequential Decision Making. In *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI '16. [7.1]

Kedzie, C., K. McKeown, and F. Diaz (2015). Predicting Salient Updates for Disaster Summarization. In *Proceedings of the Association for Computational Linguistics*, ACL '15. [7.1]

Kendall, M. G. (1938). A New Measure of Rank Correlation. *Biometrika 30*(1/2). [3.3.2]

Khuller, S., A. Moss, and J. Naor (1999). The Budgeted Maximum Coverage Problem. *Information Processing Letters 70*(1). [2.4]

Kittur, A., E. H. Chi, and B. Suh (2008). Crowdsourcing User Studies with Mechanical Turk. In *Proceedings of the Conference on Human Factors in Computing Systems*, CHI '08. [3.2.2]

Kocabas, I., B. T. Dinçer, and B. Karaoglan (2014). A Nonparametric Term Weighting Method for Information Retrieval Based on Measuring the Divergence from Independence. *Information Retrieval 17*(2). [5.3.2, 7.2.2]

Kulesza, A. and B. Taskar (2012). Determinantal Point Processes for Machine Learning. *Foundations and Trends in Machine Learning 5*(2-3). [2.2, 2.4]

Kullback, S. and R. A. Leibler (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics 22*(1). [3.1.3, 4.2.1, 5.3.1]

Kupiec, J., J. Pedersen, and F. Chen (1995). A Trainable Document Summarizer. In *Proceedings of the Conference on Research and Development in Information Retrieval*, SIGIR '95. [1, 2.4, 5, 5.2]

Lapata, M. and R. Barzilay (2005). Automatic evaluation of text coherence: Models and representations. In L. P. Kaelbling and A. Saffiotti (Eds.), *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pp. 1085–1090. Professional Book Center. [2.3.1]

Lehmann, J., R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer (2015). DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web 6*(2). [6.1]

Lemaître, G., F. Nogueira, and C. K. Aridas (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research 18*(17), 1–5. [7.3.3]

Li, C., Y. Liu, and L. Zhao (2015). Improving Update Summarization via Supervised ILP and Sentence Reranking. [5.2, 5.3.3]

Li, C., X. Qian, and Y. Liu (2013). Using Supervised Bigram-based ILP for Extractive Summarization. In *Proceedings of Association for Computational Linguistics*, ACL '13. [5.2, 5.3.3]

Li, P., W. Lam, L. Bing, and Z. Wang (2017). Deep Recurrent Generative Decoder for Abstractive Text Summarization. In *Proceedings of Empirical Methods in Natural Language Processing*, EMNLP '17. [2.1]

Lin, C.-Y. (2004). ROUGE: a Package for Automatic Evaluation of Summaries. In *Proceedings of the Association for Computational Linguistics*, ACL '04. [1.1, 2.2.1, 2.3.1, 2.3.1, 2.3.2, 2.2, 3, 3.1, 3.1.1, 3.3.4, 4.3.2, 5.3.3, 5.5.2, 6.3.1, 6.3.2, 7.3.2, 7.3.3]

Lin, C.-Y. and E. Hovy (2000). The Automated Acquisition of Topic Signatures for Text Summarization. In *Proceedings of the Conference on Computational Linguistics*, COLING '00. [2.4.1, 4.2.1, 4.2.2]

Lin, C.-Y. and E. Hovy (2002). Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4*, AS '02, Stroudsburg, PA, USA, pp. 45–51. Association for Computational Linguistics. [2.2.1, 2.3.1]

Lin, H. and J. A. Bilmes (2012). Learning Mixtures of Submodular Shells with Application to Document Summarization. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, AUAI '12. [2.2, 2.4]

Lin, J. (1991). Divergence Measures Based on the Shannon Entropy. *Transactions on Information Theory 37*(1). [3.1.3, 5.3.1]

Liu, Q., Y. Liu, D. Wu, and X. Cheng (2013). ICTNET at temporal summarization track TREC 2013. In *Proceedings of the Text Retrieval Conference*, TREC '13. [7.1]

Liu, T.-Y. (2009). Learning-to-rank for Information Retrieval. *Foundations and Trends in Information Retrieval 3*(3). [1.1, 5.2, 5.5.2, 7.1]

Lloret, E. and M. Palomar (2012). Text Summarisation in Progress: A Literature Review. *Artificial Intelligence Review 37*(1). [1, 2, 2.4]

Lloret, E., L. Plaza, and A. Aker (2013). Analyzing the capabilities of crowdsourcing services for text summarization. *Language Resources and Evaluation 47*(2), 337–369. [2.3.1]

Lloret, E., L. Plaza, and A. Aker (2017). The Challenging Task of Summary Evaluation: an Overview. *Language Resources and Evaluation*. [2.2.1, 2.3.1, 2.3.1, 2.3.1, 2.3.1]

Louis, A. and A. Nenkova (2013). Automatically Assessing Machine Summary Content Without a Gold Standard. *Computational Linguistics 39*(2). [2.3.1, 3, 3.3.4]

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development 2*(2). [2, 2.1, 2.1]

Macdonald, C., R. McCreadie, R. L. Santos, and I. Ounis (2012). From puppy to maturity: Experiences in developing terrier. [5.3.2, 7.2.2, 7.2.4]

Mackie, S., R. McCreadie, C. Macdonald, and I. Ounis (2014a). Comparing Algorithms for Microblog Summarisation. In *Proceedings of the Conference and Labs of the Evaluation Forum*, CLEF '14. [1.4, 4, 8.3]

Mackie, S., R. McCreadie, C. Macdonald, and I. Ounis (2014b). On Choosing an Effective Automatic Evaluation Metric for Microblog Summarisation. In *Proceedings of the Information Interaction in Context Symposium*, IIiX '14. [1.4, 3]

Mackie, S., R. McCreadie, C. Macdonald, and I. Ounis (2016). Experiments in Newswire Summarisation. In *Proceedings of the European Conference on Information Retrieval*, ECIR '16. [1.4, 3, 4, 5]

Mani, I. and E. Bloedorn (1998). Machine Learning of Generic and User-focused Summarization. In *Proceedings of the American Association for Artificial Intelligence*, AAAI '98. [5.2]

Mani, I., B. Gates, and E. Bloedorn (1999). Improving Summaries by Revising Them. In *Proceedings of the Association for Computational Linguistics*, ACL '99. [2.1]

Mani, I., D. House, G. Klein, L. Hirschman, T. Firmin, and B. Sundheim (1999). The Tipster AUMMAC Text Summarization Evaluation. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, EACL '99. [2.3.1]

Mani, I., G. Klein, D. House, L. Hirschman, T. Firmin, and B. Sundheim (2002). SUMMAC: a Text Summarization Evaluation. *Natural Language Engineering 8*(01). [1, 2.3.1]

Manning, C. D. and H. Schütze (2001). *Foundations of Statistical Natural Language Processing*. MIT Press. [2, 2.3.1]

Manning, C. D., M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the Association for Computational Linguistics*, ACL '14. [4.3.2, 6.1, 6.3.2]

McCreadie, R., M.-D. Albakour, S. Mackie, N. Limosopathan, C. Macdonald, I. Ounis, and B. T. Dinçer (2013). University of Glasgow at TREC 2013: Experiments with Terrier in Contextual Suggestion, Temporal Summarisation and Web Tracks. In *Proceedings of the Text Retrieval Conference*, TREC '13. [1.4, 2.2.1, 7]

McCreadie, R., C. Macdonald, and I. Ounis (2014). Incremental Update Summarization: Adaptive Sentence Selection based on Prevalence and Novelty. In *Proceedings of the Conference on Information and Knowledge Management*, CIKM '14. [7.1]

McCreadie, R., S. Vargas, C. MacDonald, I. Ounis, S. Mackie, J. Manotumruksa, and G. McDonald (2015). University of Glasgow at TREC 2015: Experiments with Terrier in Contex-

tual Suggestion, Temporal Summarisation and Dynamic Domain Tracks. In *Proceedings of the Text Retrieval Conference*, TREC '15. [1.4, 2.2.1, 7, 7.1]

McDonald, R. T. (2007). A Study of Global Inference Algorithms in Multi-document Summarization. In *Proceedings of the European Conference on Information Retrieval*, ECIR '07. [2.4]

McKeown, K., R. J. Passonneau, D. K. Elson, A. Nenkova, and J. Hirschberg (2005). Do summaries help? a task-based evaluation of multi-document summarization. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, New York, NY, USA, pp. 210–217. ACM. [2.3.1]

McKeown, K. and D. R. Radev (1995). Generating summaries of multiple news articles. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, New York, NY, USA, pp. 74–82. ACM. [2.1]

McLellan, P., A. Tombros, J. M. Jose, I. Ounis, and M. Whitehead (2001). Evaluating Summarisation Technologies: A Task Oriented Approach. In *New Developments in Digital Libraries*, NDDL '01. [1, 2.3.1]

McMinn, A. J., Y. Moshfeghi, and J. M. Jose (2013). Building a large-scale corpus for evaluating event detection on twitter. In Q. He, A. Iyengar, W. Nejdl, J. Pei, and R. Rastogi (Eds.), *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pp. 409–418. ACM. [7.1]

Metzler, D. and W. B. Croft (2005). A markov random field model for term dependencies. In R. A. Baeza-Yates, N. Ziviani, G. Marchionini, A. Moffat, and J. Tait (Eds.), *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, pp. 472–479. ACM. [6.2.2]

Metzler, D. and T. Kanungo (2008). Machine Learned Sentence Selection Strategies for Query-Biased Summarization. In *Proceedings of the Conference on Research and Development in Information Retrieval*, SIGIR '08 Learning-to-rank Workshop. [2.2.2]

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of Neural Information Processing Systems*, NIPS '13. [3.1.2]

Milne, D. and I. H. Witten (2013). An Open-source Toolkit for Mining Wikipedia. *Artificial Intelligence 194*. [6.1, 6.3.2]

Mishra, A. and K. Berberich (2017). How Do Order and Proximity Impact the Readability of Event Summaries? In *Proceedings of the European Conference on Information Retrieval*, ECIR '17. [3.2.1]

Mitchell, T. (1997). *Machine Learning*. McGraw Hill. [5]

Müller, A. C. and S. Guido (2017). *Introduction to Machine Learning with Python: a Guide for Data Scientists*. O'Reilly Media. [5.4.1, 7.3.2]

Murdock, V. G. (2006). *Aspects of Sentence Retrieval*. Ph. D. thesis, University of Massachusetts Amherst. [5.3.2, 7.2.2]

Nadeau, D. and S. Sekine (2007). A Survey of Named Entity Recognition and Classification. *Lingvisticae Investigationes 30*(1). [6, 6.1]

Nenkova, A. (2005). Automatic Text Summarization of Newswire: Lessons Learned from the Document Understanding Conference. In *Proceedings of the Conference on Artificial Intelligence*, AAAI '05. [2.2.1, 2.3.2, 4.1.2, 4.3.3]

Nenkova, A. (2008). Entity-driven Rewrite for Multi-document Summarization. In *Proceedings of the Joint Conference on Natural Language Processing*, IJCNLP '08. [2.1]

Nenkova, A. and K. McKeown (2011). Automatic Summarization. *Foundations and Trends in Information Retrieval 5*(2-3). [1, 2, 2.1, 2.3.1, 2.3.1, 2.3.1, 2.4, 3, 3.1.1, 4.2.1, 5.2, 5.3.3, 5.4]

Nenkova, A. and R. J. Passonneau (2004). Evaluating Content Selection in Summarization: The Pyramid Method. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technology*, HLT-NAACL '04. [2.3.1]

Nenkova, A., L. Vanderwende, and K. McKeown (2006). A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors That Influence Summarization. In *Proceedings of the Conference on Research and Development in Information Retrieval*, SIGIR '06. [2.2.1, 2.4, 2.2, 2.4.1, 4.2.1, 4.2.1, 5.4.1, 5.5.3]

Ng, J. and V. Abrecht (2015). Better Summarization Evaluation with Word Embeddings for ROUGE. In *Proceedings of Empirical Methods in Natural Language Processing*, EMNLP '15. [3.1, 3.1.2, 3.3.4]

Ng, J.-P., P. Bysani, Z. Lin, M.-Y. Kan, and C.-L. Tan (2012). Exploiting Category-Specific Information for Multi-Document Summarization. [5.2, 5.3.3]

Ofcom (2015). News Consumption in the UK. www.ofcom.org.uk/research-and-data/tv-radio-and-on-demand/news-media/news-consumption. Office of Communications. [1]

Oliveira, H., R. Ferreira, R. Lima, R. D. Lins, F. Freitas, M. Riss, and S. J. Simske (2016). Assessing Shallow Sentence Scoring Techniques and Combinations for Single and Multi-document Summarization. *Expert Systems with Applications 65*. [1, 5.2, 5.3.1]

Osborne, M., S. Moran, R. McCreadie, A. von Lünen, M. D. Sykora, A. E. Cano, N. Ireson, C. Macdonald, I. Ounis, Y. He, T. Jackson, F. Ciravegna, and A. O'Brien (2014). Real-time detection, tracking, and monitoring of automatically discovered events in social media. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pp. 37–42. The Association for Computer Linguistics. [7.1]

Ouyang, Y., W. Li, S. Li, and Q. Lu (2011a). Applying Regression Models to Query-focused Multi-document Summarization. *Information Processing & Management 47*(2). [1, 5, 5.1, 5.2]

Ouyang, Y., W. Li, S. Li, and Q. Lu (2011b). Applying Regression Models to Query-focused Multi-document Summarization. *Information Processing & Management 47*(2). [5.2]

Over, P., H. Dang, and D. Harman (2007). DUC in Context. *Information Processing & Management 43*(6). [1.1, 2.1, 2.1, 2.2.1, 2.2.3, 2.3.1, 2.3.2, 3.1.1, 3.2.1, 3.2.2, 4.1.2, 5.2, 5.3.3, 5.5.2]

Over, P. and J. Yen (2004). An Introduction to DUC 2004: Intrinsic Evaluation of Generic Nesw Text Summarization Systems. [2.3.2]

Owczarzak, K., J. M. Conroy, H. T. Dang, and A. Nenkova (2012). An Assessment of the Accuracy of Automatic Evaluation in Summarization. In *Proceedings of the Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, NAACL-HLT WEAS '12. [1.1, 2.4, 3, 3.1.1, 3.3.2, 3.3.4, 5.3.3, 6.3.2]

Page, L., S. Brin, R. Motwani, and T. Winograd (1999). The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford InfoLab. [2.4.1, 4.2.1]

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12*. [5.1, 5.5.2]

Petrovic, S., M. Osborne, and V. Lavrenko (2010). Streaming first story detection with application to twitter. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pp. 181–189. The Association for Computational Linguistics. [7.1]

Pew Research (2016). The Modern News Consumer. www.journalism.org/2016/07/07/the-modern-news-consumer/. Pew Research Center. [1]

Peyrard, M. and J. Eckle-Kohler (2016). Optimizing an Approximation of ROUGE – a Problem-Reduction Approach to Extractive Multi-Document Summarization. In *Proc. of the Association for Computational Linguistics*, ACL '16. [5.2, 5.3.3]

Pitler, E., A. Louis, and A. Nenkova (2010). Automatic Evaluation of Linguistic Quality in Multi-Document Summarization. In *Proceedings of the Association for Computational Linguistics*, ACL '10. [2.2.3, 2.3.1, 3.3.4]

Ponte, J. M. and W. B. Croft (1998). A Language Modeling Approach to Information Retrieval. In *Proceedings of the Conference on Research and Development in Information Retrieval*, SIGIR '98. [5.3.2, 7.2.2]

Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program 14*(3). [5.4, 6.3.2]

Radev, D. R., H. Jing, M. Styś, and D. Tam (2004). Centroid-based Summarization of Multiple Documents. *Information Processing & Management 40*(6). [2.2.1, 2.2, 2.4.1, 4.2.1, 4.2.1, 5.4.1, 5.5.3]

Radev, D. R. and D. Tam (2003). Summarization Evaluation using Relative Utility. In *Proceedings of the Conference on Information and Knowledge Management*, CIKM '03. [4.1.1]

Rankel, P. A., J. M. Conroy, H. T. Dang, and A. Nenkova (2013). A Decade of Automatic Content Evaluation of News Summaries: Reassessing the State of the Art. In *Proceedings of the Association for Computational Linguistics*, ACL '13. [1.1, 3, 5.3.3]

Rath, G., A. Resnick, and T. Savage (1961). The formation of abstracts by the selection of sentences. part i. sentence selection by men and machines. *Journal of the Association for Information Science and Technology 12*(2), 139–141. [2.2.1, 2.3.1]

Raza, A., D. M. Rotondo, and C. L. A. Clarke (2015). Waterlooclarke: TREC 2015 temporal summarization track. In *Proceedings of the Text Retrieval Conference*, TREC '15. [7.1, 7.3.4]

Reuters Institute (2017). Digital News Report. www.digitalnewsreport.org. Reuters Institute for the Study of Journalism. [1]

Rice, J. A. (2006). *Mathematical Statistics and Data Analysis*. Cengage Learning. [5.5.4]

Rizzo, G., B. Pereira, A. Varga, M. van Erp, and A. E. C. Basave (2017). Lessons learnt from the Named Entity rEcognition and Linking (NEEL) Challenge Series. *Semantic Web 8*(5). [6.1]

Robertson, S., H. Zaragoza, et al. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval 3*(4). [5.3.2, 6.2.3, 7.1, 7.2.2]

Rosenthal, J. A. (1996). Qualitative Descriptors of Strength of Association and Effect Size. *Journal of Social Service Research 21*(4). [3.3.2, 5.3.3, 5.2, 5.4.1, 5.7, 5.5.4]

Saggion, H. and T. Poibeau (2013). Automatic Text Summarization: Past, Present and Future. In *Multi-source, Multilingual Information Extraction and Summarization*, pp. 3–21. Springer. [1, 2, 2.4]

Saggion, H., J. Torres-Moreno, I. da Cunha, E. SanJuan, and P. Velázquez-Morales (2010). Multilingual Summarization Evaluation without Human Models. In *Proceedings of the Conference on Computational Linguistics*, COLING '10. [2.3.1, 2.3.1, 3.1, 3.1.3, 3.3.4]

Sang, E. F. T. K. and F. D. Meulder (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Conference on Natural Language Learning*, CoNLL '03. [6.1]

Savenkov, D., P. Braslavski, and M. Lebedev (2011). Search Snippet Evaluation at Yandex: Lessons Learned and Future Directions. In *Proceedings of the Conference of the Cross-Language Evaluation Forum*, CLEF '11. [2.2.2, 2.3.1]

Schilder, F. and R. Kondadadi (2008). FastSum: Fast and Accurate Query-based Multi-document Summarization. In *Proceedings of the Association for Computational Linguistics: Human Language Technologies*, ACL-HLT '08. [5.2]

Schluter, N. (2017). The Limits of Automatic Summarisation According to ROUGE. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, EACL '17. [2.3.1]

See, A., P. J. Liu, and C. D. Manning (2017). Get to the point: Summarization with pointer-generator networks. In R. Barzilay and M. Kan (Eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1073–1083. Association for Computational Linguistics. [2.1]

Sharifi, B. P., D. I. Inouye, and J. K. Kalita (2013). Summarization of Twitter Microblogs. *The Computer Journal 57*(3). [2.2.1]

Shen, C. and T. Li (2011). Learning to rank for query-focused multi-document summarization. In *Proc. of International Conference on Data Mining*, ICDM '11. [5.2, 5.3.3]

Sjöbergh, J. (2007). Older Versions of the ROUGEeval Summarization Evaluation System were Easier to Fool. *Information Processing & Management 43*(6). [1.1, 2.3.1, 3, 3.2.1]

Spearman, C. (1904). The Proof and Measurement of Association Between Two Things. *The American Journal of Psychology 15*(1). [3.3.2, 5.3.3]

Svore, K. M., L. Vanderwende, and C. J. Burges (2007). Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources. In *Proc. of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07. [5.2, 5.3.3]

Teufel, S. (2001). Task-Based Evaluation of Summary Quality: Describing Relationships between Scientific Papers. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, NAACL '01. [2.3.1]

Teufel, S. and M. Moens (1997). Sentence Extraction as a Classification Task. In *Proceedings of the Association for Computational Linguistics*, ACL '97. [2.4, 5]

Teufel, S. and M. Moens (2002). Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics 28*(4). [2.2.1]

Teufel, S. and H. van Halteren (2004). Evaluating Information Content by Factoid Analysis: Human Annotation and Stability. In *Proceedings of Empirical Methods in Natural Language Processing*, EMNLP '04. [2.3.1]

Tombros, A. and M. Sanderson (1998). Advantages of Query Biased Summaries in Information Retrieval. In *Proceedings of the Conference on Research and Development in Information Retrieval*, SIGIR '98. [2.1, 2.2.1, 2.2.2, 2.3.1]

Torres-Moreno, J.-M. (2014). *Automatic Text Summarization*. John Wiley & Sons. [1, 2, 2.1, 2.4]

Toutanova, K., C. Brockett, M. Gamon, J. Jagarlamudi, H. Suzuki, and L. Vanderwende (2007). The PYTHY Summarization System: Microsoft Research at DUC 2007. In *Proc. of Document Understanding Conference*, DUC '07. [5.2, 5.3.3]

van Halteren, H. and S. Teufel (2003). Examining the consensus between human summaries: Initial experiments with factoid analysis. In *Proceedings of the HLT-NAACL 03 on Text Summarization Workshop - Volume 5*, HLT-NAACL-DUC '03, Stroudsburg, PA, USA, pp. 57–64. Association for Computational Linguistics. [2.2.1, 2.3.1]

Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. *Data Mining and Knowledge Discovery*. [5.1]

Viégas, F. B. and M. Wattenberg (2008, July). Timelines: Tag clouds and the case for vernacular visualization. *interactions 15*(4), 49–52. [2.2.3]

Vrandečić, D. and M. Krötzsch (2014). Wikidata: A Free Collaborative Knowledgebase. *Communication of the ACM 57*(10). [6.1]

Wan, S. and K. McKeown (2004). Generating Overview Summaries of Ongoing Email Thread Discussions. In *Proceedings of the Conference on Computational Linguistics*, COLING '04. [2.2.1]

Wang, C., F. Jing, L. Zhang, and H. Zhang (2007). Learning query-biased web page summarization. In *Proceedings of the Conference on Information and Knowledge Management*, CIKM '07. [2.2.2]

Williams, E. J. (1959). *Regression Analysis*, Volume 14. Wiley New York. [3.3.2, 3.3.3]

Witbrock, M. J. and V. O. Mittal (1999). Ultra-summarization: A statistical approach to generating highly condensed non-extractive summaries (poster abstract). In F. C. Gey, M. A. Hearst, and R. M. Tong (Eds.), *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pp. 315–316. ACM. [2.2.3]

Witten, I. H., E. Frank, M. A. Hall, and C. J. Pal (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann. [1, 1.1, 2.4, 5, 5.1, 5.1, 5.4, 5.5.3, 6, 7.3.2]

Wu, Q., C. J. C. Burges, K. M. Svore, and J. Gao (2010). Adapting Boosting for Information Retrieval Measures. *Information Retrieval Journal 13*(3). [5.5.2]

Xu, T., D. W. Oard, and P. McNamee (2013). HLTCOE at TREC 2013: Temporal summarization. In *Proceedings of the Text Retrieval Conference*, TREC '13. [7.1]

Yang, Z., F. Yao, H. Sun, Y. Zhao, Y. Lai, and K. Fan (2013). BJUT at TREC 2013 temporal summarization track. In *Proceedings of the Text Retrieval Conference*, TREC '13. [7.1]

Yeung, C. A. and A. Jatowt (2011). Studying How the Past is Remembered: Towards Computational History Through Large Scale Text Mining. In *Proceedings of the Conference on Information and Knowledge Management*, CIKM '11. [1]

Zajic, D. M., B. J. Dorr, J. J. Lin, and R. M. Schwartz (2007). Multi-candidate Reduction: Sentence Compression as a Tool for Document Summarization Tasks. *Information Processing and Management 43*(6). [2.1]

Zhai, C. and J. Lafferty (2004). A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Transactions on Information Systems 22*(2). [4.2.1]

Zhang, C., W. Xu, F. Meng, H. Li, T. Wu, and L. Xu (2013). The information extraction systems of PRIS at temporal summarization track. In *Proceedings of the Text Retrieval Conference*, TREC '13. [7.1]