Principled Approaches to Automatic Text Summarization

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades Dr. rer. nat.

vorgelegt von
Maxime Peyrard
geboren in Sainte-foy-lès-lyon

Tag der Einreichung:	20. August 2019
Tag der Disputation:	29. January 2019
Referenten:	Prof. Dr. Iryna Gurevych, Darmstadt
	Prof. Dr. Johannes Fürnkranz, Darmstadt
	Prof. Dr. Ani Nenkova, Philadelphia

Darmstadt 2019
D17

Ehrenwörtliche Erklärung[1]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades “Dr. rer. nat.” mit dem Titel “Principled Approaches to Automatic Text Summarization” selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, den 20. August 2019

Maxime Peyrard

[1] Gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

Abstract

Automatic text summarization is a particularly challenging *Natural Language Processing* (NLP) task involving natural language understanding, content selection and natural language generation. In this thesis, we concentrate on the content selection aspect, the inherent problem of summarization which is controlled by the notion of information *Importance*.

We present a simple and intuitive formulation of the summarization task as two components: a summary scoring function θ measuring *how good a text is as a summary of the given sources*, and an optimization technique O extracting a summary with a high score according to θ. This perspective offers interesting insights over previous summarization efforts and allows us to pinpoint promising research directions. In particular, we realize that previous works heavily constrained the summary scoring function in order to solve convenient optimization problems (e.g., Integer Linear Programming). We question this assumption and demonstrate that *General Purpose Optimization* (GPO) techniques like genetic algorithms are practical. These GPOs do not require mathematical properties from the objective function and, thus, the summary scoring function can be relieved from its previously imposed constraints.

Additionally, the summary scoring function can be evaluated on its own based on its ability to correlate with humans. This offers a principled way of examining the inner workings of summarization systems and complements the traditional evaluations of the extracted summaries. In fact, evaluation metrics are also summary scoring functions which should correlate well with humans. Thus, the two main challenges of summarization, the evaluation and the development of summarizers, are unified within the same setup: discovering strong summary scoring functions. Hence, we investigated ways of uncovering such functions.

First, we conducted an empirical study of learning the summary scoring function from data. The results show that an unconstrained summary scoring function is better able to correlate with humans. Furthermore, an unconstrained summary scoring function optimized approximately with GPO extracts better summaries than a constrained summary scoring function optimized exactly with, e.g., ILP. Along the way, we proposed techniques to leverage the small and biased human judgment datasets. Additionally, we released a new evaluation metric explicitly trained to maximize its correlation with humans.

Second, we developed a theoretical formulation of the notion of *Importance*. In a framework rooted in information theory, we defined the quantities: *Redundancy*, *Relevance* and *Informativeness*. *Importance* arises as the notion unifying these concepts. More generally, *Importance* is the measure that guides which choices to make when information must be discarded.

Finally, evaluation remains an open-problem with a massive impact on summarization progress. Thus, we conducted experiments on available human judgment datasets commonly used to compare evaluation metrics. We discovered that these datasets do not cover the high-quality range in which summarization systems and evaluation metrics operate. This motivates efforts to collect human judgments for high-scoring summaries as this would be necessary to settle the debate over which metric to use. This would also be greatly beneficial for improving summarization systems and metrics alike.

Acknowledgments

First, I thank Prof. Dr. Iryna Gurevych for giving me the opportunity to conduct this research, for her continuous support, and for her excellent feedback. Moreover, I would like to thank Dr. Judith Eckle-Kohler for her precious guidance, for her valuable inputs, and for her excellent supervision in recent years

This work has been supported by the German Research Foundation as part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) under grant No. GRK 1994/1.

I'm very thankful to all my colleagues from the AIPHES research group and from UKP Lab for the constructive and helpful feedback that I received during various talks and discussions. In particular, I would like to thank my fellow Ph.D. students (Markus Zopf, Tobias Falke, Teresa Botschen, Avinesh P.V.S, Andreas Hanselowski, Benjamin Heinzerling, Ana Marasovic and Todor Mihaylov) for the deep and insightful conversations during our formal and informal meetings. I'm also grateful to the members of our working group on semantic representation for presenting and discussing interesting ideas.

I thank my student assistants Clément Besnier, Tobias Röding, Maxime Grauer and Patricia Heidt for their valuable contributions.

Last but not least, I would like to express my deepest gratitude to my friends, my family and my girlfriend for their outstanding support and understanding!

Contents

1 Introduction 1

1.1 Contributions . . . 5

1.2 Publication Record . . . 7

1.3 Thesis Organization . . . 8

2 Summarization 11

2.1 Task Specifications . . . 11

2.1.1 Many Tasks, Many Applications . . . 11

2.1.2 Datasets . . . 13

2.2 Evaluation . . . 17

2.2.1 Manual Annotations . . . 18

2.2.2 Automatic Evaluation Metrics . . . 19

2.2.3 Meta-Evaluation . . . 23

2.3 Main Approaches . . . 24

2.3.1 Observed Correlates of Importance . . . 25

2.3.2 Learned Correlates of Importance . . . 28

2.3.3 Extraction and Generation . . . 30

3 A Framework for Optimization-based Summarization 35

3.1 Decomposition into Summary Scoring Function and Optimizer . . . 36

3.1.1 Universality of the Decomposition . . . 36

3.1.2 Interpretation of Previous Works . . . 38

3.1.3 Comparison of Summary Scoring Functions against Human Judgments . . . 42

3.2 General Purpose Optimization for Summarization . . . 47

3.2.1 Adapting GPO for Summarization . . . 48

3.2.2 Is GPO practical? . . . 54

3.3 Approximate Upper-Bound Computation . . . 58

3.3.1 AUB Algorithm . . . 59

3.3.2 Examples: JS and PEAK . . . 60

4 Learning the Summary Scoring Function 65

4.1 The Matrix of Possible Scoring Functions . . . 66

4.1.1 Axis 1: Supervision . . . 67

4.1.2 Axis 2: Learning Constraints on θ . . . 69

4.1.3 Axis 0: Features . . . 72

4.2 An Important Example: Exploiting ROUGE properties . . . 73

4.2.1 Useful Mathematical Properties of ROUGE . . . 74

4.2.2 Optimizing the Approximation . . . 77

4.2.3 Approximation of ROUGE: a Problem Reduction . . . 78

4.3 Using Human Judgments . . . 79

4.3.1 Learning the Metric – Learning the System . . . 79

4.3.2 The Need for Regularization 80
4.4 Comparing Scoring Functions and Induced Summarizers 85
4.4.1 Comparison of the Summary Scoring Functions 85
4.4.2 Comparison of the Summarization Systems 90

5 Theoretical Approach 95
5.1 Semantic Units: Terminology and Assumptions 96
5.1.1 Introduction to Semantic Units 97
5.1.2 Motivation for Semantic Units 97
5.2 Summarization Quantities 99
5.2.1 Redundancy 99
5.2.2 Relevance 100
5.2.3 Informativeness 102
5.2.4 Potential Information 103
5.3 Formal Definition of Importance 104
5.3.1 Importance 104
5.3.2 Examples 107

6 Limitations of Human Judgment Datasets 111
6.1 Data Collection 112
6.2 Correlation Analysis 114
6.2.1 When Metrics Correlate with Each Others 114
6.2.2 When Metrics Do Not Correlate 114

7 Conclusion 121

Appendix 127
A Corpora and Implementation Details 127
B Standard Optimization Algorithms 130
C Proofs . 136

List of Figures 143

List of Tables 145

Bibliography 173

Chapter 1

Introduction

Since the seminal work of Luhn (1958), automatic text summarization has received a lot of attention. Indeed, summarization is directly applicable to many real-world scenarios (Nenkova and McKeown, 2012) and provides an attractive framework for developing NLP techniques such as natural language understanding or generation.

While the task appears rather intuitive, different researchers have proposed different formulations. For instance, Sparck Jones (1999) defined summarization as a “reductive transformation of source texts to summary texts through content reduction by selection and generalization on what is important in the source”. Allahyari et al. (2017) and Radev et al. (2002) also focused the task definition around the identification of *important information elements*.

A slightly different take is introduced by Mani (1999): “text summarization is the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)”. Here, the user and the task specificities are acknowledged and influence the information selection procedure.

Interestingly, Fiori (2014) adopted an empirical position: “The ideal of automatic summarization work is to develop techniques by which a machine can generate summaries that successfully imitate summaries generated by human beings”. This more recent view is inspired by developments in Machine Learning (ML), where summarization systems are *trained* with datasets of human-written summaries.

With the aim of guiding and structuring summarization research, Sparck Jones (1999) proposed a general model of the summarization process as three stages:

- **I** (Interpretation): The input source texts are converted into some *useful* representation, i.e., a mathematical description of the information conveyed by the sources.

- **T** (Transformation): The input representation is mapped to a representation of the desired summary. The critical step of information selection is executed during this stage.

- **G** (Generation): The representation of the desired summary is used to actually produce a human-readable text.

The interpretation step (**I**) is a general problem of *Natural Language Understanding* (NLU): mapping texts to appropriate semantic representations. In practice, summarization approaches came-up with text representations especially designed for the task of summarization.

The transformation step **T** should identify important information elements and decide which ones to keep. Thus, it is the main challenge of summarization and the focus of this thesis. It is heavily influenced by the choices made in the previous step as it operates directly on the representation chosen in step **I**.

The generation step (**G**) aims to produce readable and coherent texts. This is a problem globally shared with the field of *Natural Language Generation* (NLG). Since NLG is a particularly hard challenge on its own, most summarization systems relied on a simplified step **G**: extraction of elements (e.g., sentences) already present in the source documents. Such systems belong to the *Extractive Summarization* (ES) category. ES is naturally formalized as a discrete optimization problem where the text source is considered as a set of sentences and the summary is created by selecting an optimal subset of the sentences under a length constraint (McDonald, 2007; Lin and Bilmes, 2011). In order to focus on the task of identifying important elements, we also follow the ES strategy.

In fact, in this work, we show that summarization is equivalent to the problem of choosing i) an objective function θ for scoring system summaries, and (ii) an optimizer O which search for the subset of sentences maximizing θ. In the ideal case, this objective function would encode all the relevant quality aspects of a summary, such that by maximizing this function we would obtain the best possible summary. Throughout this document, we employ the terms *objective function*, *summary scoring function* or θ interchangeably.

The summary scoring component θ encompasses step **I** and **T**, while the optimizer implements step **G** by extracting the set of sentences with maximal scores according to θ. Remark that θ is doing slightly more than step **T** because it scores any candidate summaries instead of simply describing the most desired one. Figure 1.1 provides a simple illustration of the (θ, O) framework in comparison to the (**I**, **T**, **G**) perspective.

The (θ, O) decomposition is the organizing idea of this work and will be discussed at length in chapter 3. It provides interesting insights into the task of summarization which allow us to pinpoint several issues with previous works and frame new research questions.

First, an analysis of previous works reveals that the summary scoring function and the optimization have been tightly intertwined. In order to efficiently solve the optimization problem, θ is usually constrained to exhibit convenient mathematical properties, e.g., linearity or submodularity (Gillick and Favre, 2009; Lin and Bilmes, 2011).

We hypothesize that this is greatly limiting as realistic summary scoring functions should account for complex (non-linear) interactions between sub-elements like sentences. Furthermore, these restrictions come from computational considerations without conceptual justifications; it is not clear whether the need for efficient optimizers is justified in practice.

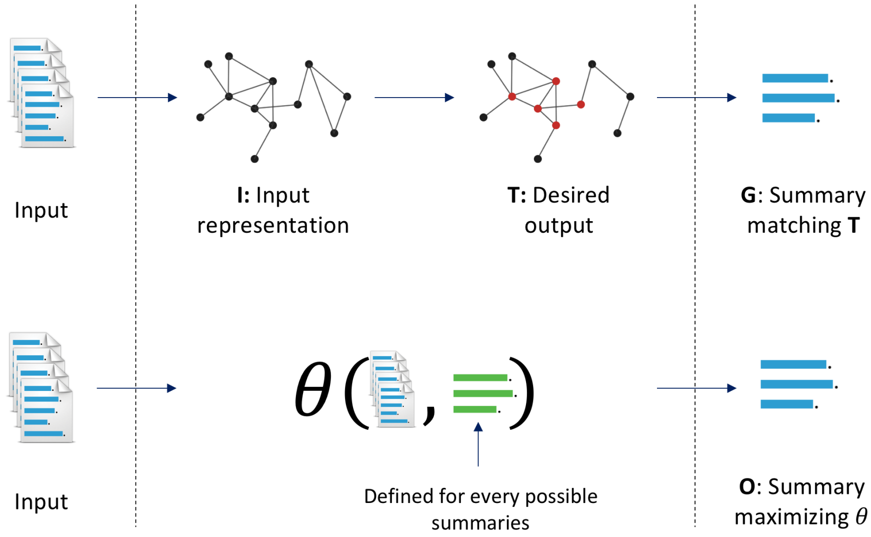

Input
I: Input representation
T: Desired output
G: Summary matching **T**
θ (,)
Input
Defined for every possible summaries
O: Summary maximizing θ

Figure 1.1: Illustration of the (θ, O) framework (second line) in comparison to the (**I**, **T**, **G**) perspective (first line). Here the input representation of step **I** is a graph but, in practice, it could be any kind of mathematical representation.

Indeed, conceptually, θ and O are independent: while the optimization step is an engineering challenge which can be addressed by the field of discrete optimization (Blum and Roli, 2003), the summarization research could focus on crafting and studying strong summary scoring functions. We discuss this idea in chapter 3

Second, the constraint imposed on the summary scoring function are ensured by, first, assigning scores to smaller elements like words (Hong and Nenkova, 2014), n-grams (Gillick and Favre, 2009; Li et al., 2013) or sentences (Conroy and O'leary, 2001; Cao et al., 2015a) and then, defining a combination function for scoring whole summaries (Carbonell and Goldstein, 1998; Ren et al., 2016).

The combined function is then carefully chosen to be linear or submodular with respect to the smaller units. Defining, or learning, scores for smaller units is again limiting the expressiveness power of θ, as interactions between elements are not easily modeled.

In particular, summarization systems struggle with redundancy, which is a complex interaction between all the elements selected in the summary. If the constraint on the summary scoring function is removed, features only computable at the summary-level become available. Such features could easily capture complex phenomena, e.g., redundancy or overall similarity between the summary and the input. We confirm this intuition in chapter 4.

Third, the evaluation of summarization systems remains an open-problem with a major impact. It guides summarization progress by deciding which summaries and systems to promote. The evaluation of summaries is notably difficult due to the vagueness of the task and the lack of true gold standard (Radev et al., 2003).

Ideally, summaries and systems would be evaluated manually by trained human annotators following a set of carefully designed guidelines, e.g., the Pyramid method

(Nenkova et al., 2007). Unfortunately, such manual annotations are expensive to obtain, and not reproducible. Thus, a large body of work has focused on the development of automatic evaluation metrics.

Traditionally, automatic evaluations compare the extracted summaries (system summaries) against a pool of human-written summaries (reference summaries). A prominent example of such automatic evaluation metric is ROUGE (Lin, 2004b). It computes an n-gram overlap between the system and reference summaries. Despite being heavily criticized for its simplistic assumptions, ROUGE has become the standard evaluation metric.

However, given recent advances in the automatic evaluation (Lloret et al., 2018), empirical research could progressively move away from ROUGE towards more meaningful metrics for both evaluating and training systems. We put this idea into practice in chapter 4.

Our (θ, O) framework highlights the importance of the summary scoring function. Yet, there exists no proper way to study these functions independently from their optimization methods. Such analysis could be beneficial to understand the inner workings of summarization systems and guide future work. In Chapter 3, we propose a way to analyze summary scoring functions using available human judgments.

Finally, there is a lack of abstract and theoretical studies on summarization. In particular, the notion of *Importance* is often talked about informally but has barely received a formal treatment. In fact, summarization research has heavily focused on empirical developments, crafting summarization systems to perform well on standard datasets while leaving the formal definition of *Importance* latent (Das and Martins, 2010; Nenkova and McKeown, 2012).

Yet, *Importance* is the key notion guiding the simplification step $\mathbf{T}$. Indeed, summarization is a lossy semantic compression and whenever one compresses with loss of information one must make choices about what to discard. *Importance* can be viewed as the measure that guides these choices. We postulate that establishing formal theories of *Importance* has the potential to advance our understanding of the task and guide future research. In summarization, the lack of efforts to produce abstract theoretical frameworks might impede the progress. We make initial step toward mitigating this issue in chapter 5

These challenges give rise to the following research questions, which we want to approach in the current thesis:

RQ1 Research Question 1: Is it justified to constrain the summary scoring function θ? Implicitly, this questions whether General Purpose Optimization (GPO) techniques, which do not make any assumption about the objective function, are efficient and effective enough to be used in practice.

RQ2 Research Question 2: Is an unconstrained summary scoring function better able to match human judgments? Does this also translate to better summaries when such a function is optimized approximately with a GPO?

RQ3 Research Question 3: How to study the inner component θ of summarization systems? Such an analysis could be useful to understand systems and pinpoint potential areas of improvements.

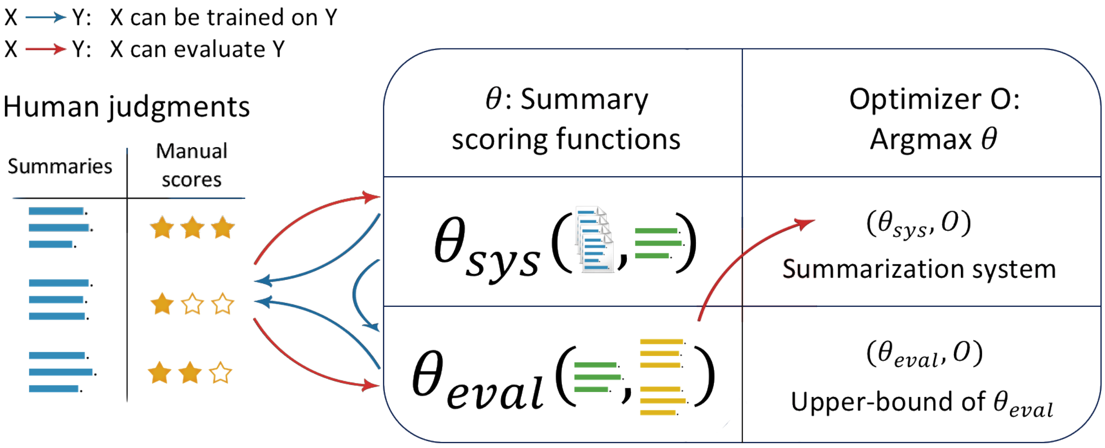


Figure 1.2: Illustration of the (θ, O) framework and some of its consequences. The green summary is the summary being scored. The gold summaries are the human-written reference summaries. Summary scoring functions can be trained and evaluated with human judgments (whether they are evaluation metrics or summarizers' internal scoring functions). Optimizing a θ_{sys} results in extracted summaries. This constitutes a summarizer if θ does not use any evaluation resources. If θ uses evaluation resources, then it is an evaluation metric and optimizing it means computing its upper-bound.

RQ4 Research Question 4: When learning θ from data, what is the impact of the supervision signal? In particular, can we leverage the existing human judgment datasets to improve the training?

RQ5 Research Question 5: What is a formal interpretation of the notion of *Importance*?

1.1 Contributions

The contributions of this thesis can be divided into (i) a description of the (θ, O) framework (illustrated by figure 1.2) used to interpret previous works and demonstrate the practicality of GPO for the summarization use-case , (ii) an empirical quest for discovering summary scoring functions from data resulting in new strong summarization systems and evaluation metrics, and (iii) a theoretical path for defining a summary scoring function via a formal treatment of the notion of *Importance*.

The following lists provide an overview of these contributions:

Contributions associated to RQ1:

- We introduce and describe formally the (θ, O) framework. To further illustrate this decomposition, we interpret, within the framework, several existing summarization systems by identifying their choices of θ and O.
- We adapt various GPO techniques which can optimize any arbitrary function and compare them on summarization datasets. The results show that GPOs

are both efficient and effective enough for the summarization use-case. This frees θ from the previously imposed constraints.

- Several existing summarization systems whose summary scoring functions have been identified are significantly improved by switching from a greedy algorithm to a GPO.

- By leveraging the complementarity of several GPOs, one can compute better upper-bound estimates for evaluation metrics for which it is impossible to compute the exact upper-bound efficiently. In practice, we computed upper-bound estimates for two important example metrics: *Jensen-Shannon divergence evaluation metric* (Lin et al., 2006) and *PEAK* (Yang et al., 2016), an automatic version of *Pyramid*.

Contributions associated to RQ2:

- We trained various summary scoring functions with and without linearity constraints and observed a much better performance for unconstrained functions. In particular, unconstrained functions better correlate with human judgments.

- The unconstrained functions optimized approximately by GPOs extract better summaries than the constrained functions optimized exactly with ILP. This further confirms the hypothesis that removing the constraints on θ is beneficial.

- Evaluation metrics and summarizers' internal scoring functions are both summary scoring functions and can be learned within the same setup. Thus, the two main challenges of summarization, evaluation and crafting summarizers, are unified and framed in the same setup.

- We trained a new evaluation metric **S3** and released it for the community. [1]

Contributions associated to RQ3:

- By analogy with the evaluation of evaluation metrics, we propose to analyze summary scoring functions based on their ability to correlate with human judgments. This results in a principled way of studying the inner workings of summarization systems.

- We observed surprisingly low correlations between existing systems and humans. This suggests that current summarization systems work without modeling human scores.

Contributions associated to RQ4:

- An important special case of θ learning setup is studied: when ROUGE is used as supervision. In such case, based on the mathematical structure of ROUGE, we could derive an almost perfect linear approximation provided scores for sentences are available. Thus, the task of summarization (as evaluated by ROUGE) reduces itself to the task of learning the sentence scores.

[1] `https://github.com/UKPLab/emnlp-ws-2017-s3`

- We observed that learning solely from available human judgments leads to summary scoring functions which are ill-behaved under optimization. We propose a simple regularization strategy to mitigate this issue; the resulting summarizer extracts high-quality summaries.

Contributions associated to RQ5:

- Within an abstract framework rooted in information theory, we formally define several summarization quantities: *Redundancy*, *Relevance* and *Informativeness*. *Importance* arises as the notion unifying these concepts.

- Under simplifying assumptions, the summary scoring function induced by the newly defined notion of *Importance* is shown to correlate well with human judgments. Furthermore, it is capable of discriminating reference summaries from system summaries.

Analysis of Human Judgments:

- Several of our experiments hinted that existing human judgment datasets may have limitations. Thus, we conducted experiments on these datasets commonly used to compare evaluation metrics. We discovered that they do not cover the high-scoring range in which summarization systems and evaluation metrics operate. This casts serious doubts on the trustworthiness of evaluation in the field of summarization in general.

- Our experiments motivate efforts to collect human judgments for high-scoring summaries as this would be necessary to settle the debate over which metric to use. Such data, in combination with the techniques presented in this thesis, would be greatly beneficial for improving summarization systems and metrics alike.

1.2 Publication Record

Several parts of this thesis have been previously published in international peer-reviewed conference and workshop proceedings from major events in natural language processing, e.g., *ACL*, *NAACL* and *COLING*. We list these publications below and indicate the chapters and sections of this thesis which build upon them:

- **Peyrard, Maxime**. (2019a). A Simple Theoretical Model of Importance for Summarization, In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. Florence, Italy. (chapter 1, section 5.1, section 5.2 and section 5.3)

- **Peyrard, Maxime**. (2019b). Studying Summarization Evaluation Metrics in the Appropriate Scoring Range, In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. Florence, Italy. (chapter 1, chapter 6)

- **Peyrard, Maxime** and Gurevych, Iryna. (2018). Objective Function Learning to Match Human Judgements for Optimization-Based Summarization, In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies Association for Computational Linguistics*. pp. 654–660, New Orleans, USA. (chapter 1, section 4.1, section 4.3 and section 4.4)

- **Peyrard, Maxime** and Botschen, Teresa and Gurevych, Iryna. (2017). Learning to Score System Summaries for Better Content Selection Evaluation, In *Proceedings of the EMNLP workshop "New Frontiers in Summarization" Association for Computational Linguistics*. pp 74–84, Copenhagen, Denmark. (chapter 1, section 4.1, section 4.3 and section 4.4)

- **Peyrard, Maxime** and Eckle-Kohler Judith. (2017a). A Principled Framework for Evaluating Summarizers: Comparing Models of Summary Quality against Human Judgments, In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. pp 26–31. Vancouver, Canada. (chapter 1, section 3.1, section 3.2 and section 4.4)

- **Peyrard, Maxime** and Eckle-Kohler Judith. (2017b). Supervised Learning of Automatic Pyramid for Optimization-Based Multi-Document Summarization, In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. pp 1084–1094. Vancouver, Canada. (chapter 1, section 3.3, section 4.1 and section 4.4)

- **Peyrard, Maxime** and Eckle-Kohler Judith. (2016a). A General Optimization Framework for Multi-Document Summarization Using Genetic Algorithms and Swarm Intelligence, In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*. pp 247–257. Osaka, Japan. (chapter 1, section 3.1, section 3.2 and section 4.4)

- **Peyrard, Maxime** and Eckle-Kohler Judith. (2016b). Optimizing an Approximation of ROUGE - a Problem-Reduction Approach to Extractive Multi-Document Summarization, In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. pp 1825–1836. Berlin, Germany. (section 4.2 and section 4.4)

1.3 Thesis Organization

This thesis is structured in seven chapters. We provide a brief overview of the organization of this document and the content of each chapter:

Chapter 2: "*Summarization*":
In order to contextualize the contributions of this thesis, a broad discussion of previous works in summarization is proposed. Applications of summarization, datasets, prominent approaches and evaluation challenges are presented.

Chapter 3 "*A Framework for Optimization-based Summarization*":
This chapter introduces the (θ, O) framework and uses it to interpret previous works. The summary scoring functions from these previous works are compared against human judgments. Furthermore, various GPO techniques are adapted to summarization and their performances are compared on summarization datasets.

Chapter 4 "*Learning the Summary Scoring Function*":
This chapter investigates empirical approaches to discover strong summary scoring functions from data. Several variants of the learning scenario are compared. The results confirm the hypothesis that freeing θ from constraints is beneficial for summarization. Techniques to incorporate human judgments in the training are presented. New summarization systems and evaluation metrics are introduced.

Chapter 5 "*Theoretical Approach*":
This chapter steps back from the empirical approach and adopts a theoretical path to craft summary scoring functions. Several concepts intuitively connected to summarization are formally described: *Redundancy*, *Relevance*, *Informativeness* and *Importance.*

Chapter 6 "*Limitations of Human Judgment Datasets*":
This chapter discusses the limitations of existing human judgment datasets. In particular, they do not cover the high-scoring range in which current systems and metrics operate. Furthermore, existing evaluation metrics do not correlate in this high-scoring range. Improvements cannot be measured reliably because metrics disagree and it is not clear which one to trust. This motivates the collection of human judgments for high-quality summaries

Chapter 7 "*Conclusion*":
Finally, we summarize the main contributions of this thesis. Potential future research directions are proposed from both the empirical and theoretical perspectives.

Chapter 2

Summarization

In this chapter, we discuss relevant previous works in the field of summarization. This means presenting the diversity of summarization tasks, describing existing datasets, introducing prominent approaches and discussing proposed evaluation methodologies. More detailed information about the summarization field in general can be found in Nenkova and McKeown (2011), Torres-Moreno (2014b), Yogan et al. (2016) or Lloret et al. (2018).

2.1 Task Specifications

2.1.1 Many Tasks, Many Applications

According to Nenkova and McKeown (2012) and Saggion and Poibeau (2013) automatic text summarization tasks can be organized along three dimensions:

- **Input type**: single, multi-document, etc.
- **Purpose**: generic, query-based, etc.
- **Output type**: extractive, abstractive

Input Type:
Single document summarization produces a summary for one source (Torres-Moreno, 2014b). In contrast, *multi-document* summarization takes as input a set of related documents.

Different input types offer different challenges and opportunities (Yogan et al., 2016). For example, multi-document summarization relies heavily on redundancy as important information elements are likely to be repeated across documents (Nenkova and McKeown, 2011). Even though redundancy generally correlates with *Importance*, there may be non-redundant yet important information elements. Such elements are difficult to discover without a better modeling of the notion of *Importance* (Zopf et al., 2016a).

The input documents can be drawn from various domains. For example, web pages (Amitay and Paris, 2000; Delort et al., 2003), blogs (Hu et al., 2007; Sharifi et al., 2010), emails (Newman and Blitzer, 2003; Nenkova and Bagga, 2003), scientific articles (Mei and Zhai, 2008; Qazvinian and Radev, 2008), biomedical documents

(Elhadad et al., 2005; Khelif et al., 2007), online live-blogs (P.V.S. et al., 2018), finance articles (Filippova et al., 2009) or streams of articles (Kedzie et al., 2016; Lin et al., 2017). In general, different input domains require domain-specific knowledge to derive more adapted notions of *Importance* (Allahyari et al., 2017).

In practice, most of the research focused on news summarization (Mckeown and Radev, 1995; White et al., 2001). Indeed, news texts exhibit standard language and discuss common topics. Thus, news summarization constitutes a convenient testbed for new approaches.

Purpose:
The majority of existing works operate under the *generic* assumption, where the summary is targeted at a hypothetical average user. In practice, this assumes the summary is intended to a wide audience without further information about individual preferences of the users.

In contrast, *query-based* summarization produces summaries that contain only information relevant to a query submitted by the user. The query can be a set of keywords or a statement in natural language. This task is related to information retrieval (e.g., snippets produced by search engines (Nenkova and McKeown, 2011)).

Similarly, *personalized* summarization tailors summaries to a specific and well-identified user. Personalized and query-based summarization are ways of biasing the summarization task and forcing the system to consider external information (Saggion and Poibeau, 2013).

Finally, *update* summarization addresses another goal of potential interest to end-users. An *update summary* must convey the important development of an event beyond what the user has already seen (Dang and Owczarzak, 2008). Informally, it can be understood as summarization with memory.

Output Type:
Until recently (Yao et al., 2017), the vast majority of research focused on *extractive summarization*, which outputs a selection of important sentences or phrases available in the input sources (Ko and Seo, 2008; Nenkova and McKeown, 2012). By selecting already grammatical elements, extractive summarization reduces to a combinatorial optimization problem (McDonald, 2007). To solve such combinatorial problems, summarization systems have leveraged powerful techniques like *Integer Linear Programming* (ILP) or *submodular* maximization. These approaches are discussed in more details in section 2.3.

In contrast, *abstractive summarization* aims to produce new and original texts (Khan et al., 2016) either from scratch (Rush et al., 2015; Chopra et al., 2016), by fusion of extracted parts (Barzilay and McKeown, 2005; Filippova and Strube, 2008; Filippova, 2010a), or by combining and compressing sentences from the input documents (Knight and Marcu, 2000; Radev et al., 2002). Intuitively, abstractive systems have more degrees of freedom. Indeed, careful word choices, reformulation and generalization should allow condensing more information in the final summary. This should give better abilities to match the desired content expressed by step **T** (the specification of what is wanted in the final summary).

We remind the three stages of summarization laid out by Sparck Jones (1999): **I** is to the input representation, **T** refers to the content selection and **G** is the final generation step. Then, abstractive and extractive summarization seem to differ only in the final step **G**. However, in practice, extractive and abstractive systems tend to use different text representations and different information selection procedures (Yao et al., 2017). For instance, modern abstractive summarization techniques promote end-to-end encoder-decoder approaches, which merge the three steps (**I**, **T** and **G**) into one single trainable model (Rush et al., 2015; Nallapati et al., 2016). Section 2.3 discusses these topics in more detail.

Applications:
The variability of the summarization tasks offers a lot of flexibility to accommodate a wide range of applications. For instance, Mani (1999) discussed headline generation, outlines (notes for students), meeting minutes, movie synopses, reviews (book, CD, movies, etc.), biography, abridgments, bulletins (weather forecasts, stock market reports, news reports), etc.

According to Torres-Moreno (2014a), automatic summarization can reduce the reading time and facilitate information extraction for users. For example, Roussinov and Chen (2001) reported that automatic summarization of results from a search engine reduces search time for users.

Mckeown et al. (2005) observed that writing a report from a set of news articles is faster and easier when automatic summaries are provided as guides. Similarly, in the experiments of Maña López et al. (2004), users could find relevant information from document sets in considerably less time when automatic summaries were available. In particular, Mani et al. (2002) found that "summaries as short as 17% of the full-text length speed up decision making twice, with no significant degradation in accuracy".

Also, Teufel (2001) showed that automatic summaries of scientific articles were almost as helpful as human-written ones for identifying the scientific concepts mentioned in a given paper.

For single document summarization, Sakai and Sparck Jones (2001) observed that indexing automatic summaries instead of full documents was helpful to information retrieval systems. Finally, Burstein and Marcu (2000) discussed the benefits of automatic summarization in automatic essay scoring.

2.1.2 Datasets

The definitions of summarization we have seen previously remained vague to ensure that a wide spectrum of task variations is covered. Ultimately, the task is defined by the datasets and their associated annotations. Indeed, they guide the summarization research by setting concrete targets for summarization systems. Thus, we provide a brief overview of existing datasets and efforts to construct them automatically.

Table 2.1 summarizes the datasets we present in this section and their properties.

Document Understanding Conference (DUC):
Between 2000 and 2007, the *National Institute of Standards and Technology* (NIST) organized the *Document Understanding Conference* (DUC). The goal was to facili-

tate progress in summarization by providing datasets and manual evaluation of systems. This resulted in various datasets and evaluation for single and multi-document summarization of news articles.

In the first major establishment in 2001, [1] a dataset for both single and multi-document summarization consisting of 60 topics of 10 documents was released. In 2002, [2] a similar setup was proposed with abstracts for single document summarization and extracts for multi-document summarization.

The editions of 2003 (Over, 2003) and 2004 [3] built on prior years by adding more topics. For DUC-2005 (Dang, 2005) and 2006 (Dang, 2006), systems had to deliver a brief answer to a complex question concerning a set of 25 to 50 documents. In 2007, this query-focused task was continued together with a pilot update summarization task. In update summarization, the goal is to first generate a summary for a document set A (generic summarization). Then the systems should generate another 100-word summary of a subsequent document set B for the same topic, under the assumption that the reader has already read A.

Text Analysis Conference (TAC):
The summarization track at the *Text Analysis Conference* (TAC) was a direct continuation of the DUC series. In particular, the main tasks of TAC-2008 (Dang and Owczarzak, 2008) and TAC-2009 (Dang and Owczarzak, 2009) were refinements of the pilot update summarization task of DUC 2007. A dataset of 48 topics was released as part of the 2008 edition and 44 new topics were created in 2009. TAC-2008 and TAC-2009 became standard benchmark datasets and we use them throughout the thesis.

TAC-2010 (Owczarzak and Dang, 2010) and TAC-2011 (Owczarzak and Dang, 2011) put emphasis on the guided summarization scenario. The goal is to write a 100-word summary of a set of 10 newswire articles for a given topic from a predefined category. Different categories may have different requirements. For example, biographies and news articles should be summarized differently. This can be understood as an instance of query-focused summarization. In these editions, the update setup was extended to model a scenario where a user is interested in a particular news story and wants to keep track of its development. The user reads some news articles but can't monitor all available newswires. Many of the articles repeat the same information, so she would like a summary of the important points of the articles, that she has not already read.

MultiLing:[4]
During the 2011 (Giannakopoulos et al., 2011) edition of TAC, the MultiLing pilot task was organized to measure the performance of multi-document summarization systems in a multi-lingual setup. In the first edition, 700 documents covering 7 languages (Arabic, Czech, English, French, Hebrew, Hindi, Greek) were clustered in 10 topics to be summarized in about 250 words. In the following 2013 edition (Giannakopoulos, 2013), 5 topics and 3 languages (Chinese, Romanian and Spanish)

[1] `https://www-nlpir.nist.gov/projects/duc/pubs/2001slides/pauls_slides/index.htm`

[2] `https://www-nlpir.nist.gov/projects/duc/pubs/2002slides/overview.02.pdf`

[3] `https://duc.nist.gov/pubs/2004slides/duc2004.intro.pdf`

[4] `http://multiling.iit.demokritos.gr`

were added.

In 2015 (Giannakopoulos et al., 2015) and 2017 (Giannakopoulos et al., 2017), new datasets were proposed to cover new tasks: multilingual single document summarization of *Wikipedia* featured articles in about 40 different languages, Online Forum Summarization (OnForumS) crawled from major online news publishers such as *The Guardian* or *Le Monde*, and Call Centre Conversation Summarization (CCCS) for transcribed speech summarization.

Also for multilingual multi-document summarization, it is worth mentioning that Turchi et al. (2010) released a set of documents related to 4 topics in seven languages (Arabic, Czech, English, French, German, Russian and Spanish). This dataset has the particularity that the relevant sentences of each document are manually annotated.

Large and automatically generated dataset:
The DUC and TAC datasets are small, high-quality and manually created datasets. With the development of successful ML techniques, collecting potentially noisy but large-scale data becomes a valid option. Hence, there have been efforts to automatically collect larger corpora.

For instance, the *Gigaword Corpus* (Napoles et al., 2012) is an archive of nearly 10 million texts from various newswire sources. Even if it does not contain explicit summaries, some works considered headlines as one-sentence summaries (Rush et al., 2015; Chopra et al., 2016).

The *New York Times Annotated Corpus* (Sandhaus, 2008) counts as one of the largest summarization datasets currently available. It contains nearly 1 million carefully selected articles from the *New York Times*, each with summaries written by humans.

Also, the *CNN/Daily Mail* dataset (Hermann et al., 2015) has been decisive in the recent development of neural abstractive summarization (See et al., 2017; Paulus et al., 2017; Cheng and Lapata, 2016). It contains *CNN* and *Daily Mail* articles together with bullet point summaries.

Such large training data made possible end-to-end abstractive summarization with sequence-to-sequence models for single documents. However, Grusky et al. (2018) and Chen et al. (2016) observed that these datasets were biased toward extractive summaries. Therefore, Grusky et al. (2018) proposed the *NewsRoom* dataset crawled from the *Internet Archive*. It contains 1.3 million news articles with the summaries extracted from the HTML metadata. According to their analysis, the summaries combine both abstractive and extractive strategies.

Several dataset construction methods leveraged the vast amount of textual data available in Wikipedia. For example, MultiLing (Giannakopoulos et al., 2015, 2017) proposed to summarize Wikipedia featured articles. Zopf (2018) also viewed the high-quality Wikipedia featured articles as summaries, for which potential sources were automatically searched on the web. This process resulted in a large-scale multi-document summarization dataset. This was the automated version of the *hMDS* corpus (Zopf et al., 2016b) previously created manually. Similarly, (Liu et al., 2018) automatically constructed a multi-document summarization dataset by combining

Wikipedia citations and results of search engines.

Others have taken advantage of the easily accessible micro-blogging services to assemble datasets. For example, the *TGSum* dataset (Cao et al., 2016a) is based on *Twitter*. The authors observed that tweets containing hyperlinks to a document (like a news article) often highlight key points in the corresponding document. Then hashtags are used to cluster documents, which produces a multi-document summarization dataset.

Similarly, Lloret and Sanz (2013) collected 200 news articles with their associated tweets. Here again, the tweet written by the journalist is regarded as a summary of the article. We also mention the *Large Scale Chinese Short Text Summarization* (LCSTS) dataset (Hu et al., 2015) constructed from the Chinese micro-blogging website *SinaWeibo*. This corpus consists of over 2 million Chinese texts from major newswires together with short summaries provided by the journalists.

Examples of other domains/tasks:
As stated previously, there exist many variations of the summarization task. Here, we provide examples of datasets focused on specific tasks or domains.

One may require real-time updates when new information is rapidly created, like during the development of unexpected news events. To this end, the *TREC Temporal Summarization* track ran shared-tasks from 2013 to 2015. Systems should detect useful and new sentence-length updates about an on-going event. In the following two years, this track was merged with the microblog track to become the new *Real-Time Summarization* (RTS) (Lin et al., 2016, 2017). In the same spirit, (P.V.S. et al., 2018) recently crawled the live-blog archives from the *BBC* and *The Guardian* together with some bullet-point summaries reporting the main developments of the event covered.

Recently, Li et al. (2017) proposed the task of reader-aware multi-document summarization in which the readers' comments should be taken into account to generate personalized summaries. To promote this task, they released a dataset together with the paper.

To evaluate their opinion-oriented summarization system, Ganesan et al. (2010) constructed the *Opinosis* dataset. It contains 51 articles discussing the features of commercial products (e.g., iPod's Battery Life).

Hasler et al. (2003) collected a dataset of popular science texts, which contains annotations about the importance of each sentence in the sources. It also contains indications about which fragments of a sentence can be removed without affecting the sense.

Interestingly, several works tried to explicitly generate datasets containing heterogeneous sources (Nakano et al., 2010; Benikova et al., 2016; Zopf et al., 2016b). As an example, the *DBS* dataset (Benikova et al., 2016) contains 30 topics about the educational domain in German together with manually created coherent extracts.

Falke and Gurevych (2017) proposed a slight modification of the summarization scenario where the output should be a graph, called a concept map (Villalon and Calvo, 2010; Valerio and Leake, 2006). They released a dataset created with a mix of automatic techniques, crowdsourcing and expert annotations.

Finally, some datasets have also been developed for summarizing: email threads

Dataset	Creation		Input		Purpose	Output		Size	
	Man./Auto.	Type	Genre	Lang		Type	Length	Topics	Doc/Topic
DUC-2001	M	SDS, MDS	News	en	Gen.	Abs.	50-400	60	≈10
DUC-2002	M	SDS, MDS	News	en	Gen.	Abs.	10-400	60	≈10
DUC-2003	M	SDS, MDS	News	en	Gen.	Abs./Ext.	10, 100	30	≈10
DUC-2004	M	SDS, MDS	News	en, ar	Gen.	Abs.	10, 100	50	≈10
DUC-2005	M	MDS	News	en	Query	Abs.	250	50	25-50
DUC-2006	M	MDS	News	en	Query	Abs.	250	50	25
DUC-2007	M	MDS	News	en	Query Upd.	Abs.	250	45	25
TAC-2008	M	MDS	News	en	Gen. Upd. Opi.	Abs.	100	48	10
TAC-2009	M	MDS	News	en	Gen. Upd.	Abs.	100	44	10
TAC-2010	M	MDS	News	en	Query Upd.	Abs.	100	44	10
TAC-2011	M	MDS	News	en	Query Upd.	Abs.	100	44	10
MultiLing-2011	M	MDS	News	7	Gen.	Abs.	240-250	10	10
MultiLing-2013	M	MDS	News	10	Gen.	Abs.	240-250	15	10
MultiLing-2015	A	SDS, MDS	News, Forum, Speech	38	Gen.	Abs.	240-250	15	10
MultiLing-2017	A	SDS, MDS	News, Forum, Speech	41	Gen.	Abs.	240-250	15	10
TREC(2013-2015)	M	Temporal	News, Blogs	en	Upd.	Ext.	-	≈200	Stream
TREC(2016-2017)	A	Temporal	News, Emails	en	Upd.	Ext.	-	≈200	Stream
CL-SciSumm-2016	A	MDS	Sci.	en	Gen.	Abs.	250	30	≈10
CL-SciSumm-2017	A	MDS	Sci.	en	Gen.	Abs.	250	40	≈10
ACL anthology	A	SDS	Sci.	en	Gen.	Abs.	paragraph	≥10K	1
CNN/Daily Mail	A	SDS	News	en	Gen.	Abs.	≈56	≈300K	1
LCSTS	A	SDS	News	zh	Gen.	Abs.	paragraph	≈2M	1
Gigaword	A	SDS	News	en	Headline	Abs.	sentence	≈10M	1
NYT Corpus	A	SDS	News	en	Gen.	Abs.	paragraph	≈1M	1
Newsroom Dataset	A	SDS	News	en	Gen.	Abs. Ext.	Length	1.3M	1
Opinosis	M	MDS	Review	en	Opi.	Abs.	≈20	51	≈100 sentences
(Goldstein et al., 2000)	M	MDS	News	en	Gen.	Ext.	paragraph	25	10
(Ulrich et al., 2008)	M	MDS	Emails	en	Gen.	Abs. Ext.	250	30	≈11
(Zechner, 2002a)	M	MDS	Dialog	en	Gen.	Ext.	paragraph	23	-
(Loupy et al., 2010)	M	MDS	News	fr	Gen.	Abs.	200	20	20
(Carenini et al., 2007)	M	MDS	Emails	en	Gen.	Abs.	30%	20	≥4
(Lloret and Sanz, 2013)	A	SDS	News	en, es	Gen.	Abs.	140 char.	200	1
TGSum	A	MDS	News	en	Gen.	Abs.	140 char.	204	≈20
(P.V.S. et al., 2018)	A	Temporal	Snippets	en	Gen.	Struct. Abs.	≈60	≈2K	≈70
(Li et al., 2017)	M	MDS	News, Comments	en	Pers.	Abs.	100	45	10
(Liu et al., 2018)	A	MDS	Heter.	en	Gen.	Abs.	paragraph	≈2M	1-1K
(Nakano et al., 2010)	M	MDS	Heter.	en	Gen.	Ext.	paragraph	24	352
hMDS	M	MDS	Heter.	en, de	Gen.	Abs.	paragraph	91	≈14
auto-hMDS	A	MDS	Heter.	en, de	Gen.	Abs.	paragraph	≈7K	≈8
(Benikova et al., 2016)	M	MDS	Heter.	de	Gen.	Ext.	≈500	30	4-14
(Falke and Gurevych, 2017)	Mix.	MDS	Heter.	en	Gen.	Struct.	-	30	≈40

Table 2.1: Description of existing datasets

(Ulrich et al., 2008; Carenini et al., 2007), transcribed dialogues (Zechner, 2002a) and meeting recordings (McCowan et al., 2005).

2.2 Evaluation

Evaluating the quality of summaries extracted by systems is a crucial part of summarization research. Unfortunately, the evaluation of summaries is notably difficult due to the vagueness of the task and the lack of true gold standard (Radev et al., 2003).

Ideally, summaries should be assessed by trained human annotators. Hence, several annotation methodologies have been proposed to manually measure various aspects of summary quality. However, there is some degree of subjectivity (Fiori, 2014), reflected by low inter-annotator agreements (Jones and Galliers, 1995).

Furthermore, manual evaluations are expensive and not reproducible. Since the progress in summarization is intertwined with the capability of measuring improvements, a significant body of research was dedicated to the development of automatic metrics. Yet, this remains an open problem (Rankel et al., 2013). Indeed, even if the human-written reference summaries are considered gold, an ideal automatic metric requires perfect semantic similarity capabilities to effectively assess system summaries in comparison to references.

DUC, TAC and later MultiLing were testbeds for new ways of evaluating summaries. Both manual and automatic evaluation methodologies were investigated. In particular, TAC held the AESOP (Automatically Evaluating Summaries of Peers) track between 2009 and 2011. Its purpose was to promote research in the automatic evaluation of summaries.

Not only it is difficult to craft robust and reliable automatic metrics, but there is also no consensus on which metric to use or even which methodology should be adopted to determine the best metric (Graham, 2015). This is the subject of meta-evaluation, the evaluation of evaluation metrics.

In this section, we will focus on the progress made in developing both manual and automatic intrinsic evaluation metrics and briefly discuss issues raised by meta-evaluation. For a general, detailed and recent overview of the evaluation progress in summarization, we recommend Lloret et al. (2018), which also contains a discussion of extrinsic evaluation methodologies.

2.2.1 Manual Annotations

When humans are involved in the evaluation process, we expect the results to be trustworthy. However, reliable manual evaluation with low resource requirements is a challenging problem (Over, 2003). Here, we discuss several annotations strategies developed to mitigate these issues (Lin and Hovy, 2002).

Early developments:
In the early establishments of DUC, the evaluation of candidate summaries was done manually by trained annotators. Evaluators read candidate summaries and then make overall judgments about *content*, *grammaticality*, *cohesion*, and *organization*. In later DUC and TAC editions, other properties like *non-redundancy*, *referential clarity*, *focus* and *coherence* were measured. These quality aspects were measured on a 5-point LIKERT scale (sometimes, a 10-point LIKERT scale was used (Dang and Owczarzak, 2009)).

Apart from these intrinsic qualities, simulated extrinsic manual evaluations methodologies were proposed: *Usefulness* and *Responsiveness*. For example, annotators judged the Responsiveness of each summary by assessing the amount of information in the summary that actually helps to answer the need expressed in the topic statement (Dang, 2005). In these methodologies, candidate summaries were also graded on a LIKERT scale.

Strategies to assess content selection:
More detailed attention was given to assessing the particular aspect of content selection as it appeared to be less subjective (Lin and Hovy, 2002). For example, DUC experimented with *model units*: basic nuggets of information in the reference summaries identified by the annotators. For a given system summary, annotators identified which model units were selected and estimated the strength of the match.

Then, the weighted recall of model units gave the final score of the candidate summary (Lin and Hovy, 2002).

Later, van Halteren and Teufel (2003) used factoids as basic units. Factoids are simple facts contained in a text and were decided by humans. Then, van Halteren and Teufel (2003) proposed to measure the importance of factoids based on their frequencies in the set of reference summaries. A strong candidate summary should exhibit most of the important factoids. This was a precursor to the influential *Pyramid* method.

The Pyramid annotation method:
The Pyramid method (Nenkova and Passonneau, 2004; Nenkova et al., 2007) is also a manual annotation method to assess the content selection of system summaries. The comparison of a system summary to the content of the reference summaries is performed on the basis of *Semantic Content Units* (SCUs) which are semantically motivated, subsentential units, such as phrases or clauses.

The Pyramid method consists of two steps: first, the creation of a *Pyramid set* from reference summaries, and second, the scoring of system summaries based on the Pyramid set.

In the first step, humans annotate phrasal content units in the reference summaries and group them into clusters of semantically equivalent phrases. The resulting clusters are called SCUs and the annotators assign an *SCU label* to each cluster, which is a sentence describing the cluster content in their own words. The final set of SCUs forms the Pyramid set. Each SCU has a weight corresponding to the number of reference summaries in which it appears. Since each SCU must not appear more than once in each reference summary, the maximal weight of an SCU is the total number of reference summaries.

In the second step, humans annotate phrasal content units in a system summary and align them with the corresponding SCUs in the Pyramid set. The Pyramid score of a system summary is then calculated as the sum of the SCU weights for all Pyramid set SCUs being aligned to annotated system summary phrases. The scores can be normalized to ensure that the results are between 0 and 1.

Crowdsourcing:
Lloret et al. (2013) performed a study on the use of crowdsourcing for automatic summarization. Different tasks were proposed for identifying relevant information. The experiments exposed low quality of crowdsourced annotations even with several quality control mechanisms. The analysis performed for determining the reason for these results hinted that "the difficulty of the task itself had more influence than the amount of money paid for each task". This undermines the possibility of using crowdsourcing for evaluation of summarization systems.

2.2.2 Automatic Evaluation Metrics

Even though manual evaluations like Pyramid (Nenkova et al., 2007) are reliable and involve humans, they are expensive and not reproducible. This makes them unsuitable for systematic comparison of summarization approaches. Following the need for cheap and reproducible metrics, a significant body of research was dedicated to crafting automatic evaluation metrics. Most works on automatic evaluation metrics focused on intrinsic content assessment, while linguistic qualities like readability or coherence were rarely tackled.

Early work:
One of the simplest approach employed by early researchers to evaluate their summaries (Edmundson, 1969; Kupiec et al., 1995) was to compare the common sentences between the automatic summary and the references. To perform these comparisons, one can use *recall*, *precision* or *F-measure*. The recall measures the fraction of the sentences selected by the humans that are also identified by the candidate summary. Alternatively, the precision measures the fraction of sentences selected by the automatic summaries that are also in the references. F-measure is the harmonic mean between recall and precision.

However, this evaluation is problematic because sentences not selected in the reference by annotators may still reflect similar information as the ones selected. A system would receive no credit for extracting such sentences, even though the summary would be highly similar to the reference summaries. To alleviate this issue, Jing et al. (1998) proposed to use several reference summaries and therefore collected more examples of sentences selected by humans. Unfortunately, the problem remains if there is some systematic bias in the way humans select sentences. Furthermore, it cannot distinguish between two non-optimal sentences; they are indeed both never selected and receive a score of 0 even if one is more informative than the other.

As a further improvement, Radev and Tam (2003) proposed the concept of *Relative Utility*. Multiple judges rank each sentence in the source documents with a score from 1 to 10. Summaries can then be judged based on the relative utility of their selected sentences. However, assigning relative utility scores to each sentence is a tedious and costly annotation effort.

Counting approaches and ROUGE:
Instead of dealing with sentences, further evaluation metrics avoided the issues mentioned above by moving to smaller units. The intuition is that even two syntactically different sentences can still have several smaller units in common (Hovy et al., 2006). Two syntactically different sentences (or texts) with a significant overlap of small units are then assumed to be similar. The most popular metric implementing this strategy is *ROUGE* (Lin, 2004b) (Recall-Oriented Understudy for Gisting Evaluation).

Actually, there exist several variants of ROUGE which compute slightly different quantities. For instance, *ROUGE-N* computes the n-gram overlap between the candidate summary and the set of reference summaries. Here, the N in ROUGE-N stands for the size of the n-grams (i.e., ROUGE-1 uses unigrams and ROUGE-2 uses bigrams). Instead of just considering n-grams, *ROUGE-SU* computes overlap between skip-grams. Additionally, *ROUGE-L* is the number of words in the longest common subsequence between the references and the evaluated summary divided by the number of words in the references.

Several other approaches based on counting sub-sentence elements are worth mentioning. For instance, Lin et al. (2006) used *Jensen-Shannon* (JS) divergence between n-gram distributions of references and candidate summaries. Compared to ROUGE, they report similar correlations with human judgments in single document summarization, but a better ones in multi-document summarization. In this thesis, we refer to this evaluation metric as JS-Eval-N, where N is also the size of the n-grams. In the following chapters, we will detail the formulation and properties of ROUGE-N and JS-Eval-N.

AutoSummENG (Giannakopoulos et al., 2008; Giannakopoulos and Karkaletsis, 2011) is a method based on n-gram graphs. System and reference summaries are each represented by a graph whose vertices are n-grams and edges contain information about n-gram co-occurrences within some predefined context window. The quality of the summary is estimated based on the similarity of its graph representation to the graph representation of the reference summary. This metric was used in several shared tasks like MultiLing.

Despite subsequent efforts, ROUGE has become a de-facto standard metric because of its simplicity and decent correlation with human judgments at the macro-level (Lin, 2004b). Louis and Nenkova (2008) and Passonneau et al. (2005) reported that ROUGE correlates with both Pyramid and Responsiveness which pushes Lloret et al. (2018) to state: “ROUGE is a low-cost choice for obtaining similar results as manual evaluations”.

Problems with ROUGE:
In general, it is well-known that ROUGE does not capture lexical variations. It cannot detect paraphrasing (same meaning with different lexical units) and can be fooled by summaries using same words with a different purpose. Conroy and Dang (2008) also observe a gap between humans and systems not explained by ROUGE. Furthermore, Sjöbergh (2007) shows that a summarization system which outputs a bag of words can reach state-of-the-art ROUGE-1 scores despite being unreadable.

Moreover, Hong et al. (2014) reported that state-of-the-art systems get similar average ROUGE scores even if they produce different summaries. Similarly, Schluter (2017) observed that, according to ROUGE, “there has been no substantial improvement in performance of summarization systems in the last decade”. This indicates that more sensitive evaluation measures would be required to distinguish systems and guide the summarization research.

Efforts to address the limitations of ROUGE:
In order to account for semantics, Ng and Abrecht (2015) extended ROUGE with word embeddings (*ROUGE-WE*). Indeed, it has been shown that word vectors encapsulate some interesting aspects of semantics (Mikolov et al., 2013b). Instead of hard lexical matching of n-grams, ROUGE-WE uses soft matching based on the cosine similarity of word embeddings.

Hovy et al. (2006) computed recall based on small units called *Basic Elements* defined as triplets of words (head, modifier, argument) and a predefined list of paraphrases for matching semantically equivalent phrases. Tratz and Hovy (2008) refined this idea and automated the paraphrases matching. This resulted in BEwT-E (Basic Elements with Transformations for Evaluation) which was one of the strongest competitors during the automatic evaluation track of TAC-2009: AESOP 2009.

Similarly, Zhou et al. (2006) used the same strategy but obtained paraphrases from the MOSES statistical machine translation toolkit (Koehn et al., 2007).

Steinberger et al. (2009) proposed a metric measuring the amount of shared content between two texts based on Resnik's semantic similarity (Resnik, 1995).

Finally, a line of research aimed at creating strong metrics by automating the Pyramid scoring scheme (Harnly et al., 2005). Yang et al. (2016) proposed *PEAK*, a metric where the components requiring human inputs in the original Pyramid annotation scheme are replaced by state-of-the-art NLP tools. We provide a detailed presentation of PEAK in section 3.3 where we estimate its upper-bound. It is more semantically motivated than ROUGE and approximates correctly the manual Pyramid scores but it is computationally expensive making it difficult to use in practice. Recently, an improvement of PEAK was proposed by Gao et al. (2018).

Learning the metric:
Since evaluation metrics are usually compared based on their ability to correlate well with humans on available human judgment datasets (see section 2.2.3), some works have trained evaluation metrics on these datasets.

For instance, Conroy and Dang (2008) investigated the performances of ROUGE metrics in comparison with human judgments and proposed *ROSE* (ROUGE Optimal Summarization Evaluation), a linear combination of ROUGE variants to maximize correlation with human Responsiveness. It was later refined (Conroy et al., 2010, 2011) as part of the *CLASSY* system to include linguistic features and was well ranked in TAC AESOP 2010 and 2011. Similarly, Rankel et al. (2012) combined content-oriented features like ROUGE with linguistic features to produce a metric that correlates well with human judgments.

Also, Hirao et al. (2007) developed a voting based regression to score summaries with human judgments as the target.

Giannakopoulos and Karkaletsis (2013) introduced *MeMog*, an extension of the previously discussed AutoSummENG (Giannakopoulos and Karkaletsis, 2011) with other standard features and trained with linear regression on human judgments.

Evaluation without references:
Surprisingly, some metrics do not make use of the reference summaries or any other evaluation resources. They compute a score based only on the candidate summary and the source documents.

For example, He et al. (2008) ran the standard ROUGE-N between the source documents and the candidate summaries (as if the candidate summaries were the reference summaries). The resulting scores correlate surprisingly well with methods using human-written summaries. Similarly, Steinberger and Ježek (2012) proposed LSA metrics as proxies for summary quality. These are topic models based on factorizing the document-term matrix co-occurrences with Singular Value Decomposition (SVD). The most frequent topics are expected to appear in good summaries.

Also, Louis and Nenkova (2013) observed that the JS divergence between n-gram distributions of the source documents and the candidate summary strongly correlates with manual Pyramid and Responsiveness scores. Furthermore, Torres-Moreno et al. (2010) and Saggion et al. (2010) used the *FRESA* framework to study whether such divergence based content measures independent of references can be used in various summarization contexts. Louis and Nenkova (2013) also investigated evaluation methodologies based on the consensus of several system summaries.

Such metrics seem to be useful for developing summarization systems. Indeed, they only use information also available to the systems at test time and are able to correlate with humans. In fact, within the (θ, O) framework, such metrics can directly be optimized by a GPO to result in strong summarizers. This also hints at some connections between evaluation metrics and summarization systems via the notion of summary scoring functions. We will extensively discuss this question in the following chapters.

Automatic evaluation of linguistic qualities:
While most metrics aimed at measuring content selection, some works developed ways of assessing quality aspects such as coherence (Nenkova, 2006). For example, interest in the topic of sentence ordering and referential cohesion motivated Lapata and Barzilay (2005) to produce an automatic evaluation of cohesion.

2.2.3 Meta-Evaluation

The comparison of available automatic evaluation metrics is crucial if we wish to trust the evaluation results and develop stronger metrics. It is also important to study and understand them, in order to know their strengths, limitations, and domain of validity. Thus, relevant previous works on meta-evaluation, the evaluation of evaluation metrics, are discussed here.

Lin and Hovy (2003) gave two criteria that any strong evaluation metrics should meet:

- Criterion 1: Automatic evaluations should correlate highly, positively, and consistently with human assessments.

- Criterion 2: The statistical significance of automatic evaluations should be a good predictor of the statistical significance of human assessments.

Later, Owczarzak et al. (2012) re-emphasized the importance of measuring significant difference (criterion 2). They also pointed out that an automatic evaluation should be able to tell apart good automatic systems from bad ones. In particular, the automatic metric should recognize automatic summaries from human-written ones. According to these principles, the AESOP tracks tested evaluation metrics for their capacity to detect statistically significant differences between systems.

Generally, metrics are compared based on their ability to rank systems in agreement with human judgments (criterion 1). Donaway et al. (2000) first observed that content-based measures have strong correlations with humans. However, they also mentioned that the scores can vary significantly depending on which references are used. Similarly, Lin and Hovy (2002) reported that the instability of human-written summaries should be taken into account. Therefore, it has now become standard to use multiple reference summaries (Lin, 2004a).

Radev et al. (2003) also observed that the high-scoring range is the most relevant for comparing evaluation metrics because summarizers aim to extract high-scoring summaries. They compared several summarization metrics based on the summaries produced by 6 summarization systems. Similarly, Saggion et al. (2002) compared evaluation metrics in a multilingual setup, based on summaries generated by several summarization systems.

The meta-evaluation also suffers from discrepancies. Different methodologies to compare evaluation metrics lead to different recommendations. For instance, Owczarzak et al. (2012) used a signed Wilcoxon test to find significant differences between metrics and recommended to use ROUGE-2 recall with stemming and stopwords not removed. Rankel et al. (2013) used accuracy and found ROUGE-4 to perform well. They also hinted that a combination of all ROUGEs can even be better. Furthermore, when only statistically significant improvements in ROUGE are reported, 73% of the time this also corresponds to an improvement according to human judgments. This goes down to 64% when statistical significance is not taken into account. In a wider study, Graham (2015) found ROUGE-2 precision with stemming and stopwords removed to be the best.

2.3 Main Approaches

In this thesis, we are interested in the notion of *Importance*, which is captured by the step **T** described by Sparck Jones (1999) and encoded within the summary scoring function θ. With this goal in mind, we organize the discussion of previous approaches to summarization around the different proxies proposed or discovered for the notion of *Importance*.

In practice, the step **T** is heavily influenced by the choice of the input representation (**I**) because it operates directly on this representation. In fact, we observe that most systems can be understood as computing *Importance* as a general *topical frequency*. Different approaches differ in their definition of *topics* (i.e., the input representation) and in their counting heuristic.

Thus, we start by discussing the different choices made by previous works with respect to what counts as a *topic* and how its *Importance* is estimated. Then, we discuss machine learning approaches which focus on producing high-scoring summaries. There, the notion of *Importance* usually remains latent. This simple classification is summarized in table 2.2.

Whether obtained by supervised or unsupervised techniques, the step **T** was usually constrained to exhibit mathematical properties useful for the extraction step (**G**). Indeed, especially for extractive summarization, the step **G** is a combinatorial optimization problem which can be solved efficiently provided that the scoring function derived in step **T** exhibits some convenient mathematical properties (McDonald, 2007).

	I	T	G
Unsupervised	Define *topics*	*Counts* topics	Generate/Extract frequent topics
Supervised ext.	Define features	score sub-elements	Extract high-scoring sentences
End-to-end abs.	Define features	latent	Maximize scores of Generated summaries

Table 2.2: Rough classification of existing summarization approaches. The unsupervised approaches typically rely on a notion of *topic frequency* which is to be maximized in the final summary. Statistical learning approaches which use extractive summarization usually learn scores for sub-elements like sentences and extract a set of high-scoring sentences. Recently, end-to-end abstractive summarization simply use a loss function on the final summary to maximize its similarity with the reference.

2.3.1 Observed Correlates of Importance

In this part, we briefly present approaches from the first line of table 2.2: the unsupervised approaches. Researchers have proposed several possible representations for input sources and defined various proxies for *Importance* based on these representations. The most effective approaches have then been selected after repeated comparisons on existing summarization datasets. We observe that the notion of *Importance* was mostly modeled by *topical frequency*. It usually involves a definition of topics and a heuristic to compute the associated frequency.

Word frequencies:
Initially, Luhn (1958) introduced the simple but influential idea that sentences containing the most important words are most likely to embody the original document. He further suggested word frequency as a proxy for *Importance*. In this case, the input source is viewed as a bag of words (I). Then, the frequency of each word is computed and the final summary is a set of sentences which contains a lot of frequent words. In practice, he used a threshold to discard stopwords: frequent but unimportant words like *the* or *a*.

Later, Nenkova et al. (2006) provided an experimental justification of this idea. They interpreted the frequency distribution of words in the sources as a probability distribution. In their study, human-written summaries have a higher-likelihood than system summaries, i.e., humans tend to use words appearing frequently in the sources to produce their summaries. Thus, the frequency of words seems to correlate empirically with *Importance*.

Building on these observations, Vanderwende et al. (2007) developed the system *SumBasic*, which scores each sentence by the average probability of its words. A greedy selection ensures that the sentence containing the most probable next words is chosen. After a sentence is selected, the probabilities are adjusted to reduce the chance of words occurring multiple times in the summary. This objective has also been solved with a global optimization algorithm (Yih et al., 2007). This relies on the same assumptions and input representation as Luhn (1958) with the difference that the frequency distributions are updated during the extraction of the summary.

The problem of identifying stopwords originally faced by Luhn (1958) could be addressed by developments in the field of information retrieval. Indeed, Sparck Jones (1972) introduced an effective heuristic to distinguish stopwords from informative content words: TF·IDF.

For a given word w, $TF(w)$ is the frequency of w in the sources. The Inverse Document Frequenc, $IDF(w) = \log(\frac{1}{n})$, is based on the number of documents n in which w appears in a background corpus. Intuitively, stopwords are frequent in all texts, while content words are only frequent in the sources. TF·IDF was a key component in many summarization systems (Erkan and Radev, 2004; Filatova and Hatzivassiloglou, 2004; Fung and Ngai, 2006; Hovy and Lin, 1999).

Based on the same intuition, Dunning (1993) outlined an alternative way of identifying highly descriptive words: the log-likelihood ratio test. It is also comparing the frequency of a word in the source to its frequency in a background corpus. For a given word w, it measures the likelihood that the observed difference between the frequency of w in the sources and in the background corpus is due to chance. The descriptive (or content) words are the ones which are significantly more frequent in the sources than in the background corpus. Words identified with such techniques are usually referred to as *topic signatures* (Lin and Hovy, 2000) and are known to be useful in news summarization (Harabagiu and Lacatusu, 2005).

Finally, the same approaches can be easily generalized to n-grams instead of words. A prominent example is the ICSI system (Gillick and Favre, 2009) which simply aims to extract frequent bigrams. Despite its simple objective function, ICSI has been identified as one of the state-of-the-art models in a study by Hong et al. (2014).

Topic modeling:
Words serve as a proxy to represent the topics discussed in the sources. However, different but similar words may refer to the same topic and should not be counted separately. This observation gave rise to a set of important techniques based on topic models (Allahyari et al., 2017). These approaches can be divided into: topic word approaches (previous paragraph), sentence clustering, Latent Semantic Analysis (LSA) and Bayesian topic models (Nenkova and McKeown, 2012). The previous paragraph already presented topic word approaches, we now discuss the three latter categories.

A simple way to discover topics is to gather similar sentences in the same cluster and view each cluster as one topic of the documents (Radev et al., 2000). In general, this has the disadvantage that topics (clusters) cannot overlap and sentences which discuss different ideas cannot span different clusters. However, many works used this idea and refined the similarity computation or sentence extraction procedure (Ji, 2006; McKeown et al., 1999; Siddharthan et al., 2004; Zhang et al., 2015). Alternatively, one can gather similar words or n-grams instead of full sentences. This is the idea behind lexical chains (Barzilay and Elhadad, 1999) discussed in a following paragraph.

Latent Semantic Analysis (LSA) is an unsupervised technique which yields a representation of texts from the co-occurrence patterns of words (Deerwester et al., 1990). It first constructs a term-sentence matrix, where each row corresponds to a word from the input and each column corresponds to a sentence. The entry of the matrix can be the word frequency or its TF·IDF (Gong and Liu, 2001). Singular Value Decomposition (SVD) is used to project the matrix on a smaller set of rows while preserving the similarity structure. The resulting rows are the latent dimensions, which represent the topics discussed in the sources. Gong and Liu (2001) initially proposed to select sentences covering many of the most frequent topics discovered by LSA.

This idea was refined by several subsequent works (Hachey et al., 2006; Steinberger et al., 2007). For example, Davis et al. (2012) derived word weights via LSA and then selected the set of sentences with maximum weights.

Bayesian topic models are powerful probabilistic models uncovering the latent topics of a text (Allahyari et al., 2017). *Latent Dirichlet Allocation* (LDA) is one of the most prominent example (Blei et al., 2003). The document is represented as a random mixture of latent topics, where each topic is a probability distribution over words. A detailed overview of LDA can be found in Blei (2012).

Several summarization systems have used LDA to uncover the topics discussed in the sources (Daumé III and Marcu, 2006; Wang et al., 2009). A hierarchical extension of LDA, *hLDA*, has been used to organize content in a hierarchy of topic vocabulary (Haghighi and Vanderwende, 2009).

Graph-based approaches:
Approaches like hLDA can exploit repetitions both at the word and at the sentence level (Celikyilmaz and Hakkani-Tur, 2010). Graph-based methods form another powerful class of approaches which combine repetitions at the word and at the sentence level. They were developed to estimate sentence *Importance* based on word and sentence similarities (Mani and Bloedorn, 1997, 1999; Mihalcea and Tarau, 2004).

One of the most prominent examples is LexRank (Erkan and Radev, 2004): A similarity graph $G(V, E)$ is constructed where V is the set of sentences and an edge e_{ij} is drawn between sentences v_i and v_j if and only if the similarity between them is above a given threshold. Then, sentences are scored according to their PageRank score in G.

A significant body of research was dedicated to tweak and improve various components of graph-based approaches. For example, one can investigate different similarity measures (Chali and Joty, 2008). Also, different weighting schemes between sentences have been investigated (Leskovec et al., 2005; Wan and Yang, 2006).

Discourse-based approaches:
Until now, the approaches we saw derived scores for sentences based on some statistical properties like frequency and co-occurrence patterns. In order to provide richer representations of input texts, some approaches have investigated discourse-

based techniques. In this line of work, linguistic knowledge or external information is inputted in the representations (e.g., *WordNet* information (Miller et al., 1990)).

For instance, *lexical chains* model the topic progression in input texts. They are sequences of related words, independent from the grammatical structure. In general, they can provide context for disambiguating terms (Hirst et al., 1998). To find topically related words, the approach usually relies on external resources like WordNet (Miller et al., 1990). In summarization, a chain can be viewed as a topic, where longer chains are more important (Barzilay and Elhadad, 1999). This also fits within the idea of topic frequency. It builds on the intuition that topics are expressed by several related words. Silber and McCoy (2002) and later Galley and McKeown (2003) demonstrated ways of efficiently computing the lexical chains of texts.

Another interesting approach is an analysis of the discourse structure of the input document via *Rhetorical Structure Theory* (RST) (Mann and Thompson, 1988), which represents the text as a tree. The smallest units, called *elementary discourse units* (EDUs), are clauses. Larger units are created by identifying relations between smaller units, which yields the hierarchical tree structure. Discourse units are classified as nuclei or satellites, where nuclei are the central concepts.

Intuitively, a summarization system should identify and extract the nuclei. For example, Ono et al. (1994) proposed to extract summaries by penalizing the extraction of satellite units. In contrast, Marcu (1997) rewards the selection of nuclei. Later, Marcu (1998) also took into account the length of the selected units. More generally, Wolf and Gibson (2004) discussed the idea of building a graph representation of text based on semantic properties instead of similarities. This amounts to counting topics in a semantic space rather than relying on statical properties of the lexical units in the sources.

Abstract Meaning Representation (AMR) is an effort to provide a standardized and rich sentence-level semantic representation. The sentence is represented by a rooted, directed, acyclic graph whose nodes are *concepts* and edges are *relations* (Banarescu et al., 2013). For summarization, Liu et al. (2015a) proposes to first merge the graph representation of each sentence in the sources. They then propose to learn *Importance* scores for concepts and relations in a structured supervised learning setup. The final summary should maximize the *Importance* scores while remaining a valid AMR graph, which can be converted back to texts.

2.3.2 Learned Correlates of Importance

In the previous subsection, the correlates for importance were mostly inferred by the researchers and empirically tested afterward. Thus, in these studies, the notion of *Importance* remained fairly simple as being modeled by *frequent topics*. Topics could be words, clusters of sentences, latent components from LSA / LDA, or a mixture of them with hLDA and graph-based representations. The most *frequent* topics were then targeted for extraction.

As it is common with difficult and vaguely defined tasks, researchers stepped-back from the hope of inferring simple criteria to capture the notion of *Importance* and shifted to Machine Learning (ML). In such approaches, the notion of *Importance* remains latent and hardly interpretable. However, ML offers great flexibility in the input representation **I**. Indeed, a large feature set can be used and later reduced to its most useful components. Moreover, irrelevant features should be discarded automatically by the learning algorithms.

In this part, we are mainly focusing on techniques from the second line of table 2.2. ML approaches have learned scores for simple textual units like words, but also for more complex units like sentences and whole summaries.

Learning scores for sub-elements:
Already in one of the first approaches to summarization, Edmundson (1969) hinted at supervised learning by proposing an automatic search for features correlating with *Importance*. He computed the final score of sentences as a linear combination of several features whose weights are learned from data. As features, he considered: frequency of words, overlap between the title and the sentence, position of the sentence and the number of words matching a list of cue-words indicating summary-worthy content (like "to summarize").

Additionally, the frequency computation of words or n-grams can be replaced with learned weights (Hong and Nenkova, 2014; Li et al., 2013). This usually results in better systems.

More generally, many indicators for sentence importance were proposed (as it can be seen in the previous subsection) and therefore the idea of combining them to develop stronger indicators emerged (Aone et al., 1995). For instance, Kupiec et al. (1995) suggested that statistical analysis of summarization corpora would reveal the best combination of features. In practice, they used a *Naive Bayes* learning algorithm to learn a combination of indicators derived in previous works (Luhn, 1958; Edmundson, 1969).

A variety of works proposed to learn *Importance* scores for sentences (Yin and Pei, 2015; Cao et al., 2015a). This started a huge body of research comparing different learning algorithms, features and training data (Hakkani-Tur and Tur, 2007; Hovy and Lin, 1999; Osborne, 2002; Wong et al., 2008; Zhou and Hovy, 2003).

Usually, the target score of a sentence is given by computing its ROUGE score against reference summaries (Yao et al., 2017). While this is only a proxy for sentence *Importance*, this has been shown to be effective (Cao et al., 2015b, 2016b).

Also, it is worth noting that Fuentes et al. (2007) used Pyramid data to train a system. Alternatively, ranking sentences instead of scoring them has also been investigated (Metzler and Kanungo, 2008; Shen and Li, 2011)

An interesting line of work is based on the assumption that the most important sentences are the ones that permit the best reconstruction of the input document (He et al., 2012). It was refined by a stream of works using distributional similarities (Li et al., 2015; Liu et al., 2015b; Ma et al., 2016).

Learning scores for summaries:
When scoring individual elements (like sentences), one faces the problem of accounting for interactions between the selected elements. For example, two relevant but similar sentences are redundant and should not be extracted together.

To achieve this, instead of predicting sentence scores independently, more complex learning algorithms have been proposed. For instance, sequence labeling going over the list of sentences and indicating whether or not the current sentence should be kept can model dependencies between sentences. Hidden Markov models (Conroy and O'leary, 2001) or Conditional Random Fields (Shen et al., 2007) are simple sequence labelling techniques. With the Markov assumption, the probability of selecting the current sentence depends on whether the previous sentence was selected or not. Nowadays, sequence-to-sequence methods are usually employed (Nallapati et al., 2017).

Additionally, structured output learning permits to score smaller units while providing supervision at the summary level (Li et al., 2009). For example, Sipos et al. (2012) employed structured prediction to train an end-to-end system with a large-margin loss to optimize a convex relaxation of ROUGE. Furthermore, indirect supervision like reinforcement learning (Rioux et al., 2014) and learning to search (Kedzie et al., 2016) have also been studied.

Some works have investigated distributed representations of sources and summaries, with the objective of maximizing the semantic similarity between the extracted summary and the sources (Kobayashi et al., 2015; Kågebäck et al., 2014). While these methods are not explicitly training a correlate for importance, they use pre-trained distributed representations like word embeddings (Mikolov et al., 2013b).

Finally, akin to the development of hLDA and graph-based methods, hierarchical document representations have been learned for the task of summarization. For example, Zhong et al. (2015) used deep Boltzmann machines for hierarchical representations of the input.

2.3.3 Extraction and Generation

In this part, we discuss the last column of table 2.2 which concerns the extraction/-generation step G. We also briefly discuss recent end-to-end abstractive summarization (last lign of table 2.2).

The step **T** outputs the desired properties that the final summary should meet. The optimization step consists in finding a summary which matches these requirements. In extractive summarization, this is a combinatorial optimization problem, where a subset of sentences is chosen (McDonald, 2007). Since the extraction is NP-hard, researchers constrained the scoring function to have simple mathematical properties. Otherwise, greedy optimization techniques have been applied. In this section, we briefly describe these developments.

Even if this thesis does not focus on the production of readable and original text, we briefly describe some techniques involved in abstractive summarization. To get more control over the output summary, a large body of work produced post-processing mechanisms like sentence compression or sentence fusion. Finally, the generation of original texts remains a particularly challenging task, for which recurrent neural networks have recently proved to be helpful (Chopra et al., 2016).

Extraction and combinatorial optimization:
Extractive summarization can be formulated as the problem of selecting a subset of textual units from a document collection such that the overall score of the created summary is maximal under some length constraint (McDonald, 2007). In fact, McDonald (2007) mentioned that most summarization approaches can be seen as optimizing a trade-off between minimum *Redundancy* and maximum *Relevance*.

Prior summarization systems mostly focus on extracting sentences in a greedy fashion and, thus, tend to suffer from redundancy. An early attempt to correct this issue was proposed by Carbonell and Goldstein (1998) with Maximal Marginal Relevance (MMR). At each selection step, the algorithm extracts the sentence that is maximally relevant (to the topic or query) and minimally redundant with sentences already extracted. Subsequent works have proposed various improvements with respect to computation of relevance and redundancy (Radev et al., 2004; Murray et al., 2005; Xie and Liu, 2008).

Maximizing the relevance scores of the selected units is a global inference problem which can be solved using Integer Linear Programming (ILP) (McDonald, 2007). However, the global summary scoring function must be linearly factorizable with respect to the scored elements, which greatly limits the capability of accounting for complex interactions like redundancy.

By using indicator of *Importance* from previous works but replacing the greedy extraction strategy with an exact global inference with ILP, improvements have been observed: traditional indicators like position (Yih et al., 2007), TF·IDF (Filatova and Hatzivassiloglou, 2004) or concept frequency (Ye et al., 2007; Gillick and Favre, 2009; Boudin et al., 2015) used in conjunction with ILP solvers achieved much better results. While finding the exact solution of an ILP remains NP-hard (Filatova and Hatzivassiloglou, 2004), approximate solutions can be found using dynamic programming (McDonald, 2007; Ye et al., 2007; Yih et al., 2007).

Redundancy can also be limited by encouraging summaries to cover a large region of the semantic space (Yogatama et al., 2015). Low redundancy can be enforced by promoting diversity using determinantal point process (Kulesza et al., 2012) or submodular optimization (Lin and Bilmes, 2011).

Submodularity is a natural framework for summarization because summaries try to maximize the coverage of relevant units and coverage functions are submodular (Lin and Bilmes, 2011). Furthermore, when the summary scoring function is submodular, the greedy optimization algorithm displays mathematical guarantees that the extracted summaries will be near-optimal (Nemhauser and Wolsey, 1978). Then, various works have studied potential coverage functions for summarization (Lin and Bilmes, 2011; Kågebäck et al., 2014; Yin and Pei, 2015).

Post-processing of extractive summaries:
A general problem of extractive summarization arises when long relevant sentences are chosen. Such sentences may also contain irrelevant information which includes noise in the summary. In DUC 2005 (Dang, 2005), "more than half of the summaries were perceived as not having good referential clarity, focus, structure and coherence".

Humans tend to perform sentence compression during summarization (Jing and McKeown, 1999). Hence, rule-based systems using syntactic and discourse knowledge have been proposed to automatically compress the extracted sentences (Jing, 2000; Zechner, 2002b; Zajic et al., 2007). Compression rules can also be learned from statistical analysis of the data (Knight and Marcu, 2002; Turner and Charniak, 2005; Galley and McKeown, 2007). This introduced a large body of work on sentence compression (Nenkova and McKeown, 2011).

It is known that human summarizers not only compress sentences, they also merge different sentences (Jing and McKeown, 2000). In clustering-based summarization, Barzilay and McKeown (2005) generated one sentence representing the whole cluster instead of selecting one sentence from each cluster. Later, Marsi and Krahmer (2005) and Filippova and Strube (2008) explored the more general idea of finding the union of two sentences.

Finally, some works were interested in the problem of sentence ordering after the extraction procedure (Barzilay and Elhadad, 2002; Barzilay and Lapata, 2008). This helped to improve coherence and readability of the extracted summaries.

Generation and abstractive summarization:
We already mentioned that generating new texts offers the possibility to compress more information than simply reusing available sentences. This is supported by Cheung and Penn (2013) who analyzed human-written and system summaries. They found that human-written summaries are more abstractive and use more information fusion.

Woodsend and Lapata (2012) developed an abstractive generation procedure relying on a quasi-synchronous tree substitution grammar (QTSG) to induce paraphrases. Then, an ILP is solved to produce texts covering content selection, surface realization, paraphrases and stylistic conventions. The idea of solving ILP to select units based on a syntactic parse tree has been investigated by subsequent works (Banerjee et al., 2015; Filippova, 2010b). For instance, Bing et al. (2015) generates new candidate sentences by merging noun-phrases and verb-phrases.

Summarization based on AMR graphs (Liu et al., 2015a) can also induce abstractive summaries provided a language generation component can convert AMR graphs back to texts.

Kintsch and Van Dijk (1978) proposed a theory of text comprehension and production based on a model of human memory. Later, Fang and Teufel (2016) implemented this idea into an abstractive summarizer, where the textual units are propositions.

Recently end-to-end training based on the encoder-decoder framework with LSTMs (Sutskever et al., 2014) has achieved huge success in sequence transduction tasks like machine translation. For abstractive summarization, large single-document summarization datasets rendered possible the application of such techniques.

For instance, Rush et al. (2015) introduced a sequence-to-sequence model for sentence simplification. Later, Chopra et al. (2016) and Nallapati et al. (2016) extended this work with attention mechanisms. Since words from the summary are often retained from the original source, copy mechanisms (Gu et al., 2016; Gulcehre et al., 2016) have been investigated (Nallapati et al., 2016; See et al., 2017).

Furthermore, direct optimization of ROUGE for abstractive summarization has been attempted with reinforcement learning (Ayana et al., 2016; Paulus et al., 2017).

Interestingly, Kikuchi et al. (2016) studied ways of controlling the output length of the LSTM decoder. This is particularly relevant for summarization because of the length constraint.

Chapter Summary

- Summarization approaches can be categorized depending on their **Input type**, **Purpose** and **Output type**. This covers many applications for which automatic summarization is known to be beneficial.

- The datasets ultimately guide the summarization progress by defining what are *good* summaries.

- Collecting gold standard datasets with trained human annotators is expensive and results in a few data points. Thus, many works tried to automatically produce large-scale datasets. Indeed, modern ML techniques can benefit from noisy but large-scale data.

- Evaluation is an open problem. Despite being criticized, ROUGE remains the default evaluation metric. New and more semantically aware metrics arise as promising candidates to replace ROUGE.

- Most approaches, whether supervised or unsupervised, used a simple notion of *Importance* as topical frequency. In modern end-to-end approaches, the notion of *Importance* remains latent.

- Approaches to extractive summarization heavily constrained the summary scoring function in order to use convenient optimization procedures. To ensure such properties, when systems are trained on data, only smaller elements are scored (e.g., words or sentences).

Chapter 3

A Framework for Optimization-based Summarization

The organizing idea of this work is to decompose summarization into two components: a summary scoring function θ indicating how good a text is as a summary of the given sources, and an optimization technique O extracting a summary with a high score according to θ.

In section 3.1, after describing formally the (θ, O) framework, we identify the summary scoring functions of several existing summarizers.

Furthermore, we notice that θ can be studied independently from O based on its ability to correlate with human judgments. This gives a principled way to examine the inner workings of summarization systems. Such an analysis informs us that current systems do not model the human scores but employ different strategy. One possible approach to summarization could be to enforce summarization systems to mimick human scores. This is explored in the next chapter.

While the (θ, O) decomposition is a simple and intuitive formulation of the summarization process, it holds interesting consequences and can shape our perspective on summarization research. In particular, developing an optimization technique O is an engineering problem which is not summarization specific.

In contrast, the discovery of summary scoring functions is the central problem of automatic summarization. By realizing this and by providing a methodology to evaluate θ on its own, we can focus the summarization research on discovering new and better summary scoring functions.

Hence, unlike previous works, we argue for a study of the summary scoring functions independently from the optimization techniques. Consequently, no particular constraint should be imposed on θ. However, the optimization of such unconstrained functions is an NP-hard optimization problem (McDonald, 2007).

To extract a summary, we have to rely on existing General Purpose Optimization (GPO) techniques which do not make any assumption about θ. Section 3.2 describes such GPO algorithms and demonstrates that they are both effective and efficient enough for maximizing complex summary scoring functions. This renders the search of unconstrained scoring function possible.

Another interesting use-case of GPO is the possibility to approximately compute upper-bounds of evaluation metrics for which no exact solution can be found efficiently. Indeed, evaluation metrics are also complex summary scoring functions which may be optimized via GPO techniques. This is discussed in section 3.3.

More generally, this framework opens-up many research directions in evaluation, system design and optimization. Thus, we study some relevant consequences and discuss important examples along the way. This chapter provides answers to **RQ1** and **RQ3**. The following chapters build upon the groundwork established here.

3.1 Decomposition into Summary Scoring Function and Optimizer

The task of extractive summarization (ES) can naturally be cast as a discrete optimization problem where the text source is considered as a set of sentences and the summary is created by selecting an optimal subset of sentences under a length constraint (McDonald, 2007; Lin and Bilmes, 2011).

This view entails defining an objective function (θ) which is maximized by the particular optimization technique being used, e.g., Integer Linear Programming (ILP) when the objective function is linear (McDonald, 2007). In the ideal case, this objective function would encode all the relevant quality aspects of a summary, such that by maximizing them we would obtain the best possible summary.

In this work, we go one step further and prove that ES is equivalent to the problem of choosing (i) an objective function θ for scoring summaries, and (ii) an optimizer O. We use (θ, O) to denote the resulting *decomposition* of an extractive summarizer. To illustrate the framework, we interpret several previous works by identifying their choices of θ.

Furthermore, the decomposition enables a principled analysis of summarizers based on their summary scoring functions. Indeed, while the task consists in extracting high-quality summaries, it is often beneficial to evaluate the inner workings of systems for diagnosing problems and guiding progress.

3.1.1 Universality of the Decomposition

Let $\mathcal{D}$ be the set of all possible input sources. We note $D \in \mathcal{D}$ one input document collection. In the context of extractive summarization, D is a document to summarize and is viewed as a set of elements (e.g., sentences): $D = \{s_i\}$.

An extractive summary S is then a subset of elements from D, that is to say S is an element of $\mathcal{P}(D)$, the power set of D. Finally, we introduce $\mathcal{S}$, the set of all possible summaries:

$$\mathcal{S} = \bigcup_{D} \mathcal{P}(D) \tag{3.1}$$

This simply states that an extractive summary is a set of sentences taken from the input document D. Now, we define the notions of *objective function*, *optimizer* and *extractive summarizer* necessary for stating the (θ, O) decomposition theorem.

Objective function:
An *objective function* is a function which takes as input a document collection $D \in \mathcal{D}$, and a possible summary $S \in \mathcal{P}(D)$ from D, and outputs a score:

$$\begin{array}{rcrcl} \theta & : & \mathcal{D} \times \mathcal{P}(D) & \rightarrow & \mathbb{R} \\ & & (D, S) & \mapsto & \theta(D, S) \end{array} \tag{3.2}$$

When it is not ambiguous, we can drop the mention of the input considered $\theta(D, S)$ and simply note $\theta(S) \in \mathbb{R}$. Intuitively, this score represents the quality of the summary, i.e., how good the text S is as a summary of the sources D. In the ideal case, this objective function would encode all relevant quality aspects of a summary, such that by maximizing them we would obtain the best possible summary. We also note Θ the set of all possible objective functions. In this thesis, we refer to *objective function*, *summary scoring function* or θ interchangeably.

Optimizer:
Once an objective function θ has been chosen, an extractive summarizer ought to select the set of sentences S^* with maximal score $\theta(S^*)$ under a length constraint:

$$S^* = \underset{S \in \mathcal{P}(D)}{\operatorname{argmax}}\, \theta(S) \tag{3.3}$$

$$\text{such that, } len(S) = \sum_{s \in S} len(s) \leq L \tag{3.4}$$

Here, L is the length constraint imposed on the final summary. Intuitively, the objective function θ measures what constitutes a *good* summary and the optimizer searches the set of all possible summaries to select the best one. In general, the search space is way too large to perform an exhaustive search and approximate optimization techniques have to be employed.

We define an *optimizer* O as the technique which solves this optimization problem either exactly or approximately. Formally, O is an operator which outputs a high-scoring summary $S^* = \{s_j\}$ from an input document $D \in \mathcal{D}$ and a previously chosen objective function $\theta \in \Theta$:

$$\begin{array}{rcrcl} O & : & \Theta \times \mathcal{D} & \rightarrow & \mathcal{S} \\ & & (\theta, D) & \mapsto & S^* \end{array} \tag{3.5}$$

Here, the optimizer is assumed to be deterministic because each input is associated to one single summary S^*. For a stochastic optimization strategy, we understand the optimizer O to be the tuple of both the strategy and the random seed. Indeed, for fixed random seeds, stochastic optimization strategies become deterministic.

Extractive Summarizer:
Conceptually, an *extractive summarizer* is a simple mapping between input sources to summaries. Therefore, an extractive summarizer σ is represented by a set function

mapping a document collection $D \in \mathcal{D}$ to a summary $\sigma(D) = S_{D,\sigma} \in \mathcal{P}(D)$:

$$\begin{array}{rcccc} \sigma & : & \mathcal{D} & \rightarrow & \mathcal{S} \\ & & D & \mapsto & \sigma(D) = S_{D,\sigma} \end{array} \tag{3.6}$$

Intuitively, an extractive summarizer is simply a function which maps any input document D to an extractive summary, i.e., a set of sentences from D.

Again, this defines only a deterministic summarizer: one summary is associated to each source D by the mapping σ. In practice, some summarizers may include randomness, in particular when stochastic optimization is employed. The deterministic assumption remains applicable by including the random seed as part of the specification of σ. Thus, two summarizers generating different outputs for the same inputs are considered different.

Decomposition Theorem:
With the previous definitions, it is clear that every tuple (θ, O) uniquely defines a summarizer (noted $\sigma_{\theta,O}$) because $O(\theta, \cdot)$ produces one summary for any document collection D:

$$\begin{array}{rcccc} \sigma_{\theta,O} & : & \mathcal{D} & \rightarrow & \mathcal{S} \\ & & D & \mapsto & O(\theta, D) = \underset{S \in \mathcal{P}(D)}{\operatorname{argmax}}\, \theta(S) \end{array} \tag{3.7}$$

This says that, once we have chosen an objective function (θ) and a way to optimize it (O), we can extract a set of sentences from any document and thus construct an extractive summarizer.

In fact, this decomposition is *universal*, i.e., for any extractive summarizer σ, there exists at least one tuple (θ, O) which describes perfectly the summarizer:

Theorem 1.

$$\forall \sigma,\ \exists (\theta, O)\ \textit{such that:} \tag{3.8}$$

$$\forall D \in \mathcal{D},\ \sigma(D) = O(\theta, D) \tag{3.9}$$

Theorem 1 is quite intuitive but implies that ES is equivalent to the problem of choosing a tuple (θ, O). In particular, even a summarizer that was not crafted with a scoring function and an optimization method in mind has an implicit definition of θ. In Appendix C.1, we provide a rigorous proof of this theorem. The consequence is that we do not loose generality by viewing summarization as the components θ and O.

3.1.2 Interpretation of Previous Works

Theorem 1 states that every summarizer can be thought of as a tuple (θ, O). In order to illustrate this theorem, we analyze a range of different summarizers regarding their (potentially implicit) θ.

Edmundson: (Edmundson, 1969)
Edmundson (1969) presented a heuristic which scores sentences according to 4 different features:

- **Cue-phrases**: It is based on the hypothesis that the probable relevance of a sentence is affected by the presence of certain cue words such as 'significant' or 'important'. Bonus words have positive weights, stigma words have negative weights and all the others have no weight. The final score of the sentence is the sum of the weights of its words.
- **Key**: High-frequency content words are believed to be positively correlated with relevance (Luhn, 1958). Each word receives a weight based on its frequency in the document if it is not a stopword. The score of the sentence is also the sum of the weights of its words.
- **Title**: It measures the overlap between the sentence and the title.
- **Location**: It relies on the assumption that sentences appearing early or late in the source documents are more relevant.

By combining these scores with a linear combination, we can recognize the objective function:

$$\theta_{Edm.}(S) = \sum_{s \in S} \alpha_1 \cdot C(s) + \alpha_2 \cdot K(s) + \alpha_3 \cdot T(s) + \alpha_4 \cdot L(s) \tag{3.10}$$

The sum runs over sentences and C, K, T and L output the sentence scores for each method (Cue, Key, Title and Location). The optimizer is greedy.

TF·IDF: (Luhn, 1958; Sparck Jones, 1972)
A simple idea inspired by Luhn (1958) is that high-frequency content words are important. A useful frequency score for an n-gram can be its TF·IDF (Sparck Jones, 1972), where Term-Frequency (TF) is computed on the source document and the Inverse Document Frequency (IDF) is estimated from a background corpus. Then, the scoring function is given by:

$$\theta_{TF \cdot IDF}(S) = \sum_{g \in S} TF(g) \cdot IDF(g) \tag{3.11}$$

The sum runs over n-grams or skip-grams (g) selected in the summary. In the original paper, Luhn (1958) did not use TF·IDF which was introduced later (Sparck Jones, 1972). The optimizer is also greedy.

ICSI: (Gillick and Favre, 2009)
A global linear optimization that extracts a summary by solving a maximum coverage problem of the most frequent bigrams in the source documents. ICSI has been among the best systems in a classical ROUGE evaluation (Hong et al., 2014). Here, the identification of θ is trivial because it was originally formulated as an optimization task. If c_i is the i-th bigram selected in the summary and w_i is its weight computed from D, then:

$$\theta_{ICSI}(S) = \sum_{c_i \in S} c_i \cdot w_i \tag{3.12}$$

The extraction strategy is an Integer Linear Program. Note that Li et al. (2013) have later refined this approach by learning the weights w_i instead of using the document frequency of the bigram.

LexRank: (Erkan and Radev, 2004)
This is a well-known graph-based approach. A similarity graph $G(V, E)$ is constructed where V is the set of sentences and an edge e_{ij} is drawn between sentences v_i and v_j if and only if the cosine similarity between them is above a given threshold. Sentences are scored according to their PageRank score in G. Thus, $\theta_{LexRank}$ is given by:

$$\theta_{LexRank}(S) = \sum_{s \in S} PR_G(s) \tag{3.13}$$

Here, PR is the PageRank score of sentence s. The optimizer O is greedy.

KL-Greedy: (Haghighi and Vanderwende, 2009)
In this approach, the summary should minimize the Kullback-Leibler (KL) divergence between the word distribution of the summary S and the word distribution of the documents D (i.e., $\theta_{KL} = -KL$):

$$\theta_{KL}(S) = -KL(S||D) = -\sum_{g \in S} \mathbb{P}_S(g) \log \frac{\mathbb{P}_S(g)}{\mathbb{P}_D(g)} \tag{3.14}$$

$\mathbb{P}_X(w)$ represents the frequency of the word (or n-gram) w in the text X. The minus sign indicates that KL should be lower for better summaries. Indeed, we expect a good system summary to exhibit a similar probability distribution of n-grams as the sources.

Alternatively, the Jensen-Shannon (JS) divergence can be used instead of KL. Let M be the average word frequency distribution of the candidate summary S and the source documents D distribution:

$$\forall g \in S,\ \mathbb{P}_M(g) = \frac{1}{2}(\mathbb{P}_S(g) + \mathbb{P}_D(g)) \tag{3.15}$$

Then, the formula for JS is given by:

$$\theta_{JS}(S) = -JS(S||D) = \frac{1}{2}\left(KL(S||M) + KL(D||M)\right) \tag{3.16}$$

JS and KL are examples of summary scoring functions that are neither linear nor submodular (Louis and Nenkova, 2013). In section 3.2, we show an example of optimizing KL and JS divergence with GPO strategies.

LSA: (Steinberger and Jezek, 2004)
Latent Semantic Analysis is an approach involving a dimensionality reduction of the term-document matrix via Singular Value Decomposition (SVD). It belongs to the class of topic models where the goal is to uncover latent topics. The sentences extracted should cover the most important latent topics. The importance of the latent topic t is estimated by its associated singular value λ_t. This gives the overall summary scoring function:

$$\theta_{LSA} = \sum_{t \in S} \lambda_t \tag{3.17}$$

Here, t is a latent topic and λ_t the associated singular value given by SVD. The sum runs over the topics that have been extracted in the summary. The optimization is also greedy.

Semantic Similarity: (Kobayashi et al., 2015)
The assumption is that a summary should be similar to the input documents, with the similarity measured with distributional representations (e.g., word embeddings). The summary and the input documents are both considered to be the average of their constituting word embeddings. The score of a summary is given by its *semantic similarity* with the document, measured by the cosine similarity:

$$\theta_{DS}(S) = \frac{\mathbf{v}_S \cdot \mathbf{v}_D}{||\mathbf{v}_S|| \cdot ||\mathbf{v}_D||} \tag{3.18}$$

Here, $\mathbf{v}_X = \sum\limits_{w \in X} \mathbf{w}$ is the vector representing the average of the word vectors from text X ($\mathbf{w}$ is the vector representing the word w). The authors, concerned by efficient optimization, proposed a submodular approximation of this function. Thus, the optimization was a greedy algorithm.

In fact, this approach can be generalized: let $E(X)$ be an embedding model which maps a text X to a metric space equipped with a distance d. Then, we can define a general scoring function:

$$\theta_{E,d}(S) = -d(E(S), E(D)) \tag{3.19}$$

The goal is to minimize the difference between the representation of the summary and the representation of the sources. For example, Kobayashi et al. (2015) used the average word embeddings for E and the cosine distance for d. Ma et al. (2016) used a Bag of Words (BoW) model for E and the Euclidian distance for d.

ROUGE: (Lin, 2004b)
ROUGE is an evaluation metric, which takes the reference summaries into account. Even though it cannot form a summarizer when combined with an optimizer, it is a summary scoring function.

For simplicity, we assume there is only one reference summary noted R^*. Let R_N denote the number of n-gram tokens in R^*. R_N is a function of the summary length in words, in particular, R_1 is the size of the reference summary R^* in words. Finally, let $F_S(g)$ denote the number of times the n-gram type g occurs in S. For a single reference summary, ROUGE-N is computed as follows:

$$\theta_R(S) = \frac{1}{R_N} \sum_{g \in S^*} min(F_S(g), F_{S^*}(g)) \tag{3.20}$$

Lin and Bilmes (2011) showed that ROUGE is a submodular function. This fact was exploited by many summarization systems, e.g., (Lin and Bilmes, 2011; Sipos et al., 2012). We will study in more depth the properties of θ_R in section 4.2.

JS-Eval: (Lin et al., 2006)
Like ROUGE, JS-Eval is an evaluation metric which uses the reference summary R^*. Let M be the average word frequency distribution of the candidate summary S and the reference R^* distribution:

$$\forall g \in S, \mathbb{P}_M(g) = \frac{1}{2}(\mathbb{P}_S(g) + \mathbb{P}_{R^*}(g)) \tag{3.21}$$

Here, $\mathbb{P}_X(g)$ is the frequency of the n-gram g in the text X.

Then, the formula for JS-Eval is given by:

$$\theta_{JS-Eval}(S) = -JS(S||R^*) = \frac{1}{2}\left(KL(S||M) + KL(R^*||M)\right) \tag{3.22}$$

This is a similar formula as the θ_JS introduced above, but this time the summary is compared to the references. There is also a negative sign because we expect a good system summary to exhibit a similar probability distribution of n-grams as the reference.

We will study this function in section 3.3 and compute an approximation of its upper-bound. Indeed, unlike ROUGE, it does not possess mathematical properties convenient for optimization.

Pyramid: (Nenkova et al., 2007)
Another important evaluation we consider is the manual Pyramid method. The comparison of system summary content to reference summary content is performed on the basis of SCUs which correspond to semantically motivated, subsentential units, such as phrases or clauses (see chapter 2).

Each SCU has a weight corresponding to the number of reference summaries in which the SCU appears. The Pyramid score of a system summary is then calculated as the sum of the SCU weights for all SCUs in the Pyramid set appearing in the system summary:

$$\theta_{Pyr}(S) = \frac{1}{M} \sum_{scu \in S} w(scu) \tag{3.23}$$

There, scu represents an SCU identified in the candidate summary S and $w(scu)$ is its weight from the Pyramid set. M denotes the maximal Pyramid score possible for this pyramid set, such that $\theta_{Pyr}(S)$ is a score between 0 and 1. In section 3.3, we will study an automatic approximation of this method: PEAK (Yang et al., 2016) and estimate its upper-bound.

3.1.3 Comparison of Summary Scoring Functions against Human Judgments

According to theorem 1, every summarizer σ induces a summary scoring function θ_σ either explicitly or implicitly. Furthermore, since the optimization is not summarization-specific, all the assumptions about the summarization task are encoded in θ_σ. Hence, we propose to compare summarizers based on the quality of their assumptions encoded by their respective summary scoring functions.

More precisely, we remark that comparing the summary scoring functions of summarizers can be done with the exact same procedure used for comparing automatic evaluation metrics. Indeed, an evaluation metric is a summary scoring function which just has access to more resources (e.g., reference summaries).

Generally, high-quality summary scoring functions, whether using evaluation resources or not, should mimic humans as well as possible (Lin and Hovy, 2003). Therefore, we propose to analyze summary scoring functions based on their ability to correlate with human judgments. While correlation analyses on human judgment data have been performed in the context of validating automatic evaluation metrics (Lin, 2004b; Nenkova et al., 2007; Owczarzak et al., 2012; Louis and Nenkova, 2013), there is no prior work which uses such data for a principled comparison of summary scoring functions. This analysis would reveal whether the strategies employed by systems follow the average summarization strategy of humans.

This analysis has the advantage of being applied directly to human judgment datasets (with summaries scored by humans). It does not aim to replace the *standard final* evaluation of the extracted summary (e.g., manually with Pyramid or automatically with ROUGE), but may be a valuable complement, especially for diagnosing issues and guiding future research efforts. It also puts the focus on the central question of summarization: discovering strong summary scoring functions. While it is possible to imagine summarization systems which use different strategies than humans, it might be a good guide to follow human judgments when crafting summary scoring functions.

Comparing θ against human judgments:
Let σ be a summarizer induced by the scoring function θ_σ. We measure the quality of θ_σ by measuring its correlation with human scores.

Suppose we have a dataset of human judgments consisting of m topics: $\{\mathcal{T}_1, \dots, \mathcal{T}_m\}$, where each topic $\mathcal{T}_i$ contains n scored summaries:

$$\mathcal{T}_i = \{(S_{i,1}, h_{i,1}), \dots, (S_{i,n}, h_{i,n})\} \tag{3.24}$$

We note $\mathbf{h_i} = [h_{i,1}, \dots, h_{i,n}]$ the vector of human scores for the i-th topic. Similarly, $\theta_\sigma(\mathbf{S_i}) = [\theta_\sigma(S_{i,1}), \dots, \theta_\sigma(S_{i,n})]$ is the vector of scores given by θ_σ to the summaries of the i-th topic. This is illustrated by figure 3.1.

Let c be a correlation measure between two lists of scored elements. Then, the average correlation between θ_σ and human judgments is given by the following formula:

$$C(\theta_\sigma) = \frac{1}{m} \sum_{i=1}^{m} c(\mathbf{h_i}, \theta_\sigma(\mathbf{S_i})) \tag{3.25}$$

In practice, there are 3 common choices for the correlation metric c:

- *Pearson's r*: It is a value correlation metric which depicts linear relationships

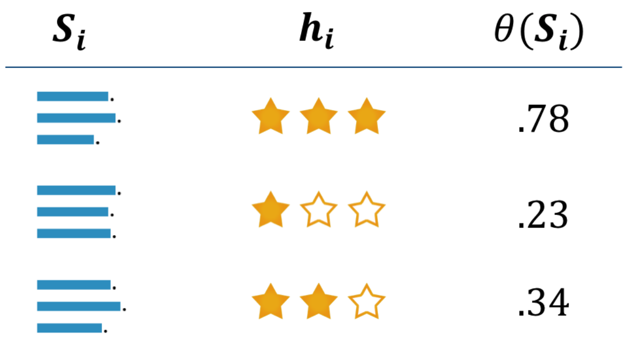


Figure 3.1: Illustration of evaluation setup for one topic $\mathcal{T}_i$ for $n = 3$. The scores produced by humans and by the summary scoring function are compared with a correlation measure c. These correlations are averaged over all topics.

between the score progression in the two lists. Formally, the correlation between two ranked lists $\mathbf{x}$ and $\mathbf{y}$ is given by:

$$r(\mathbf{x}, \mathbf{y}) = \frac{\sum(x_i - \hat{x})(y_i - \hat{y})}{\sqrt{\sum(x_i - \hat{x})^2}\sqrt{\sum(y_i - \hat{y})^2}} \tag{3.26}$$

The range of Pearson's r is from -1 to 1, where -1 is total negative linear correlation, 1 total positive linear correlation and 0 is no linear correlation.

- *Kendall's* τ: It is a rank correlation metric which compares the orders induced by both scored lists. Intuitively, it is high when both lists exhibit a similar ordering of items. It is proportional to the number of concordant pairs minus the number of discordant pairs. A concordant pair is a set of items ordered in the same way by both lists. The formula is:

$$\tau(\mathbf{x}, \mathbf{y}) = \frac{2}{n(n-1)} \sum_{i<j} sgn(x_i - x_j) sgn(y_i - y_j) \tag{3.27}$$

  The range of Kendall's tau is also from -1 to 1 with 0 indicating no correlation and 1 indicating perfect correlation.

- nDCG: It is a metric that compares ranked lists and puts more emphasis on the top elements by a logarithmic decay weighting. It uses the scores of one list to estimate the *relevance* of items. Then, scores produced by the other list are compared against the relevance scores from the first one:

$$DCG(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \frac{y_i}{\log(i+1)} \tag{3.28}$$

$$nDCG(\mathbf{x}, \mathbf{y}) = \frac{DCG(\mathbf{x}, \mathbf{y})}{IDCG(\mathbf{x}, \mathbf{y})} \tag{3.29}$$

  With $IDCG$ the ideal (maximal) DCG score possible obtain for a perfect ordering. Thus, $nDCG$ is always between 0 and 1.

Example: θ evaluation of baselines:
We applied this evaluation procedure to several summary scoring functions described previously:

$$\theta_{Edm.}, \theta_{TF \cdot IDF}, \theta_{ICSI}, \theta_{LexRank}, \theta_{JS} \text{ and } \theta_{KL}.$$

For θ_{JS} and θ_{KL}, the divergences are based on unigram distributions.

We used the same procedure to compute the correlations of two automatic evaluation metrics: ROUGE-N and JS-Eval-N for $N = 1$ and $N = 2$:

$$\theta_{ROUGE-1}, \theta_{ROUGE-2}, \theta_{JS-Eval-1}, \theta_{JS-Eval-2}$$

For reference, we also report the results ofs a random scoring function.

In these experiments, we use the human judgments available in two multi-document summarization datasets from the Text Analysis Conference (TAC) shared task: TAC-2008 and TAC-2009.[1] TAC-2008 and TAC-2009 contain $m = 48$ and $m = 44$ topics, respectively. Each topic consists of 10 news articles to be summarized in a maximum of 100 words. At the time of the shared tasks, 57 systems were submitted to TAC-2008 and 55 to TAC-2009. We use the manual Pyramid scores as human scores.

Correlations with human judgments are measured using the 3 metrics described above: Pearson's r, Kendall's τ and Normalized Discounted Cumulative Gain (nDCG). The results are reported in table 3.1 for TAC-2008 and TAC-2009.

Example: end-to-end evaluation of baselines:
Furthermore, for comparison, we computed the standard evaluation of the same summarizers on these datasets. By standard evaluation, we mean the evaluation of the extracted summaries in comparison against the pool of reference summaries $\{R_i\}$ using an automatic evaluation metric (e.g., ROUGE or JS-Eval).

For the standard evaluation, the dataset still consists of m topics: $\{\mathcal{T}_1, \ldots, \mathcal{T}_m\}$, but a topic consists of pairs of inputs and reference summaries: $\mathcal{T}_i = (D_i, R_i)$. Here D_i is the input documents from topic i and R_i are the associated human-written reference summaries. We note $\sigma(D_i)$ the summary extracted by σ for the input D_i and its *standard* evaluation is given by:

$$R(\sigma) = \frac{1}{m} \sum_{i=1}^{m} \theta_{eval}(\sigma(D_i), R_i) \tag{3.30}$$

Here, $\theta_{eval}(\sigma(D_i), R_i)$ is the score of the summary extracted by σ for the topic i measured by the evaluation metric θ_{eval}. Thus, $R(\sigma)$ is the average score of σ over all topics in the datasets.

We run the summarizers described in the previous paragraph and evaluated them with several evaluation metrics: ROUGE-1 (R-1), ROUGE-2 (R-2), JS-Eval-1 (JS-1) and JS-Eval-2 (JS-2). The results are reported in table 3.2 for TAC-2008 and TAC-2009. For reference, we optimized the evaluation metrics in order to get their upper-bound. For ROUGE-N, an ILP can be solved to compute the exact upper-bound (Takamura and Okumura, 2010). Unfortunately, for JS-Eval there is no efficient way to compute the exact upper-bound. Therefore, we report the results of a greedy optimization. In section 3.3, we compute better estimates of the JS-Eval upper-bound.

[1] `http://tac.nist.gov/2009/Summarization/`, `http://tac.nist.gov/2008/Summarization/`

	TAC-2008			TAC-2009		
θ	r	τ	nDCG	r	τ	nDCG
Random	.036	.015	.779	.020	.013	.735
ICSI	.193	.129	.785	.188	.136	.772
Edmunds.	.190	.155	.792	.385	**.276**	.804
LexRank	.259	.152	**.826**	.390	.250	.816
TF·IDF	**.267**	.184	.824	**.429**	.271	**.830**
KL	.202	.192	.781	.242	.216	.764
JS	.280	**.230**	.790	.262	.220	.760
ROUGE-1	.748	.488	**.961**	.806	.547	**.965**
ROUGE-2	.718	.490	.960	.803	.550	.963
JS-Eval-1	**.751**	**.495**	.960	**.820**	**.572**	**.965**
JS-Eval-2	.714	.474	.953	.775	.543	.952

Table 3.1: Correlation of θ functions with human judgments across various systems on TAC-2008 and TAC-2009.

	TAC-2008				TAC-2009			
Summarizer	R-1↑	R-2↑	JS-1↓	JS-2↓	R-1↑	R-2↑	JS-1↓	JS-2↓
Random	.289	.045	.543	.658	.291	.051	.551	.658
ICSI	**.365**	**.101**	.460	**.619**	**.364**	**.104**	.477	**.618**
Edmunds.	.325	.075	.492	.637	.337	.079	.474	.629
LexRank	.347	.082	.470	.630	.355	.084	.469	.629
TF·IDF	.328	.067	.507	.645	.333	.066	.513	.646
(KL,Greedy)	.353	.088	.457	.624	.357	.091	**.461**	.622
(JS,Greedy)	.356	.090	**.456**	.623	.358	.088	.469	.625
(R-1,ILP)	**.454**	.147	**.388**	.587	**.474**	.160	**.378**	.577
(R-2,ILP)	.451	**.194**	.387	**.552**	.472	**.208**	.381	**.543**
(JS-1,Greedy)	.418	.142	.399	.584	.430	.152	.392	.573
(JS-2,Greedy)	.417	.168	.407	.562	.427	.173	.407	.557

Table 3.2: Evaluation summaries extracted by $O(\sigma, \cdot)$ for several systems TAC-2008 and TAC-2009.

Analysis:
In table 3.1, we first observe relatively low correlations (< 0.3 Kendall's τ) between existing summary scoring functions and human judgments. Even though the correlations are better than random, they remain far from a perfect correlation of 1. Interestingly, summarization systems do not model the human scores. Some nDCG scores are higher but the scale of nDCG is different as it can be noticed by looking at the performance of the random baseline. In contrast to the others, nDCG goes from 0 to 1.

In the standard evaluation reported in table 3.2, we see that systems are ranked differently. In fact, systems with high end-to-end ROUGE scores do not necessarily have a good model of summary quality. Indeed, the best performing θ functions are not extracting the best summaries according to standard evaluation metrics. For example, ICSI is the best system according to ROUGE in table 3.2, but its summary scoring function does not model human judgments well. LexRank, TF·IDF and Edmundson have better correlations with human judgments. This shows that systems follow a summarization strategy different from the one humans use. In the next chapter, we propose to develop summarization systems which explicitly aim to follow the human strategy.

Overall, it seems that JS also exhibits good correlations with humans, which would support the findings of Louis and Nenkova (2008). We notice that summary scoring functions like JS or Edmundson correlate well with humans, but are optimized greedily which may explain the lower scores of their extracted summaries. In section 3.2, we show that just replacing the greedy optimization with more powerful techniques yields significant improvements.

In table 3.1, evaluation metrics have much higher correlations because they can use reference summaries. The 4 evaluation metrics considered have similar performances, with JS-Eval-1 slightly stronger. This confirms the findings of Lin et al. (2006).

It is worth noting that systems perform differently on TAC-2009 and TAC-2008. There are several differences between the two datasets like redundancy level or guidelines for annotations. This confirms that, to ensure robustness, claims of improvements over baselines should be verified on several datasets.

3.2 General Purpose Optimization for Summarization

In chapter 2, we realized that previous approaches solved the optimization problem using ILP (Schrijver, 1986) or submodular function maximization (Krause and Golovin, 2014) with constrained summary scoring functions (Gillick and Favre, 2009; Lin and Bilmes, 2011). When θ did not exhibit convenient mathematical properties, it was typically optimized with a greedy algorithm (Haghighi and Vanderwende, 2009).

It is problematic because constraining θ limits its expressiveness. In fact, we postulate that realistic summary scoring functions are unlikely to be linear (confirmed in chapter 4). However, using a greedy algorithm for maximizing unconstrained summary scoring functions often lead to poor results (Gutin et al., 2002).

In order to solve the NP-hard discrete optimization problem (McDonald, 2007) in the general case where the objective function does not have specific properties, we must rely on search heuristics (Blum and Roli, 2003). Fortunately, the well-studied field of optimization proposes a range of techniques to tackle difficult combinatorial problems (Schrijver, 2003).

In this section, we present examples of techniques capable of extracting summaries from non-linear, non-submodular objective functions adapted to summarization. We refer to such optimization as General Purpose Optimization (GPO). The pseudo-code of the standard algorithms are described in Appendix B.

Here, we briefly outline the intuitions behind them and present the adaptions required to apply them to summarization. Finally, we show that they are suited for the summarization use-case.

More generally, GPO techniques are especially appealing because they enable a decoupling of θ and O which allows the investigation of complex θ functions. The search for unconstrained θ is likely to result in better approximations of human judgments, which we demonstrate in chapter 4.

3.2.1 Adapting GPO for Summarization

When the objective function does not exhibit any particular mathematical properties, several heuristics can be applied to still approximately optimize it (Blum and Roli, 2003). Hence, we present several examples of such techniques adapted to summarization. For each algorithm, the general pseudo-code is available in Appendix B.

After briefly describing simple heuristics such as the greedy algorithm and beam-search, we move on to discussing stochastic search algorithms, often referred to as *meta-heuristics* (Bianchi et al., 2009). In optimization where the search space is discrete and large (such as summarization), the use of meta-heuristics is often helpful (Blum and Roli, 2003).

Meta-heuristics are stochastically searching the solution space guided by a heuristic. The heuristic aims to find near-optimal solutions and avoid local optima under a time or computation budget.

Common meta-heuristics use the neighboorhood of a current candidate solution as a starting point to further explore the solution space. Although they tend to prefer better neighbors (with higher fitness score), they can also accept worse neighbors to escape local optima and explore larger spans of the solution space.

In general, meta-heuristics cannot provide guarantees about the solutions found because it is always possible to imagine an adversarial optimization problem for any specific search strategy (Blum and Roli, 2003). However, some heuristics, usually *inspired by nature*, tend to work well on real-world problems (Bianchi et al., 2009).

In this section, we discuss several important examples: *Simulated Annealing*, *Genetic Algorithm* and *Swarm Intelligence*. They mainly differ by the heuristics they employ to guide the search in solution space.

Greedy:
The class of greedy algorithms forms a well studied algorithmic paradigm following the strategy of making locally optimal choices at each stage (Cormen et al., 2009). In general, this does not produce an optimal solution but outputs reasonably good solutions efficiently.

Nevertheless, making locally optimal but globally bad decisions at early stages can constrain the search to a poor area of the fitness landscape. Greedy algorithms can even produce the worst solution if the search space exhibits some adversarial properties (Gutin et al., 2002).

A natural question is to ask for which problems the greedy algorithms work well. For example, it is known to produce a globally optimal solution whenever the optimization problem has a *Matroid* structure (Papadimitriou and Steiglitz, 1982). [2]

Another well-known property is that maximization of submodular functions under constraints can be done greedily with a guarantee that the solution is close to the optimal solution (Nemhauser and Wolsey, 1978). If we note S the summary extracted by the greedy algorithm and S^* the globally optimal summary, then $\theta(S) \geq (1 - \frac{1}{e}) \cdot \theta(S^*) \approx 0.632 \cdot \theta(S^*)$. It is particularly relevant for summarization as ROUGE-N is known to be submodular (Lin and Bilmes, 2011). In general, coverage functions are submodular.

The simple greedy algorithm selects the sentence with the best score at each step. We refer to this algorithm as *Greedy* in the following sections. However, we also use a slightly better greedy algorithm in practice denoted *Greedy-M*. Greedy-M selects the sentences with the best marginal gains. At each stage, it selects the sentence which yields the best improvements, and its pseudo-code is described in Appendix B by algorithm 8. Greedy-M is the algorithm used for maximization of submodular functions (Krause and Golovin, 2014).

Beam Search:
The naive greedy algorithm can easily be stuck in a poor area of the solution space by taking bad decisions in the early steps. The beam search algorithm proposes an improvement by allowing some level of backtracking. It keeps track of the top k best candidates at any time instead of just one.

It is based on the breadth-first search where the decisions are organized in a tree (Cormen et al., 2009). Beam-search produces a solution at least as good as the Greedy algorithm. The pseudo-code is described in Appendix B by algorithm 9.

Random Search:
The simplest stochastic search is random search which randomly samples summaries and measures their fitness scores with the objective function. The pseudo-code for sampling summaries is detailed in the algorithm 1. This is basically the naive greedy algorithm applied to random (uniform) sentence scores. The function $RandomChoice(C)$ randomly selects one element from the list C.

Simulated Annealing:
We now present a meta-heuristics approach called Simulated Annealing (SA) where the search is guided by a stochastic sampling method inspired by the metropolis-hasting algorithm (Kirkpatrick et al., 1983). In contrast to population-based heuristics like Genetic Algorithm or Artificial Bee Colonies described later, it only considers one candidate solution at a time.

[2] A Matroid is a structure that generalizes the notion of linear independence in vector spaces. For reference see (Papadimitriou and Steiglitz, 1982).

Algorithm 1: Sampling Summaries for Extractive Summarization

```
Input  : $D = \{s_1, \ldots, s_n\}$: document as a set of sentences
         $L$: length constraint
Output: $S = \{s_j\}$: summary as a set of sentences
Function SampleCandidate $(D, L)$:
    $S := \{\}$
    while 1 do
        $C := \{s \in D \mid s \notin S,\ len(S \cup \{s\}) \leq L\}$
        if $C = \emptyset$ then
            return $S$
        end
        $S := S \cup RandomChoice(C)$
    end
```

The heuristic employed by simulated annealing can be understood by the analogy it draws with thermodynamic properties of physical systems. The system state is the candidate solution, the energy is the cost function (minus the fitness function), a state change corresponds to a modification of the candidate solution and the temperature is a control parameter. The final state of the system reaching thermodynamic equilibrium represents the final solution.

At each step, SA considers a neighbor n of the current state s and probabilistically decides between moving to this neighbor or staying in place. The acceptance probability $P(n, s, T)$ of moving to the neighbor depends on the energy (or fitness score) of the state and a parameter T: the temperature.
The temperature controls the tendency to move to a worse neighbor. In fact, if $P(n, s, T)$ is 0 whenever the neighbor is worse than the current state, then SA becomes the greedy algorithm. These probabilities ultimately lead the system to move to states of lower energy. However, in practice, a budget is given to the algorithm either in time or in the number of evaluations. For more details, we refer to Blum and Roli (2003).

The adaptation to summarization is straight-forward:

- **Physical State** The candidate summary $S = \{s_i\}$ considered as a set of sentences, a subset of the sentences in the documents.

- **Energy** The internal energy of the physical state (candidate summaries) is the function we wish to minimize. Since we wish to maximize θ, we choose $E = -\theta$.

- **Move** Moving from one summary to a neighboring one is done by randomly removing one of its sentences and adding a new one that does not violate the length constraint. We detail the pseudo-code of this procedure in algorithm 2. It the same as a mutation from the Genetic Algorithm perspective.

- **Acceptance Probability** Suppose we have a state s with a score $\theta(s)$ and a neighbor n with a score $\theta(n)$. We move to the neighbor according to the following probability:

$$P(n, s, T) = \begin{cases} e^{\frac{\theta(n)-\theta(s)}{T}} & \text{if } \theta(n) < \theta(s) \\ 1 & \text{otherwise} \end{cases} \tag{3.31}$$

  Thus, we automatically move if the neighbor is better and randomly move if it is worse. The temperature is a hyper-parameter controlling how much we allow to move to worse neighbors, i.e., the higher T the more often we move to worse neighbors.

Algorithm 2: Mutation Operator for Extractive Summarization

```
Input  : $S = \{s_j\}$: summary as a set of sentences
         $D = \{s_1, \dots, s_n\}$: document as a set of sentences
         $L$: length constraint
Output: $N = \{s_j\}$: mutated summary as a set of sentences
Function Mutate $(S, D, L)$:
    $C := \{s \in D \mid s \notin S,\ len(S \cup \{s\}) \leq L\}$
    $S := S \setminus RandomChoice(S)$
    $S := S \cup RandomChoice(C)$
```

Genetic Algorithm:
The Genetic Algorithm (GA) is a stochastic search method using mechanisms inspired by biological evolution, such as reproduction, mutation and selection (Wright, 1932; Goldberg, 1989).

The candidate solutions are represented as individuals in a population. The fitness function is the function to optimize and determines the quality of an individual. Evolution of the population takes place after multiple iterations where biological operators are applied: reproduction and mutation search the space of solutions by creating new candidate solutions, while the selection operator ensures that better candidate solutions survive to the next generation more often than worse ones.

In order to produce a GA for summarization, we use a simple analogy to the problem of extractive summarization:

- **Population** The individuals of the population are the candidate solutions which are valid extractive summaries. Valid means that the summary meets the length constraint. The size of the population is a hyper-parameter of the algorithm. A summary is simply a binary vector indicating which sentences it contains.

- **Fitness Function** The fitness function which evaluates the individuals (i.e., summaries) is the function we wish to maximize: θ. The population is scored and sorted according to the fitness function, a threshold indicates which summaries will survive to the next generation. The survival rate is another hyper-parameter.

- **Mutation** The mutation of a summary is done by randomly removing one of its sentences and adding a new one that does not violate the length constraint. It is described by algorithm 2. The mutations affect individuals of a population randomly, and the mutation rate is a hyper-parameter.
- **Reproduction** The reproduction is done by randomly selecting parents among the survivors of the previous generation. Then, the union set of the sentences of the parents is considered. The child is a random valid summary extracted from these sentences.This is described by the algorithm 3. There is also a reproduction rate which controls the number of children in each generation.
- **Initial Population** The initial population is created by randomly building valid summaries as described by algorithm 1. We observe a convergence speed-up by including good summaries in the initial population (e.g., summaries produced by baseline algorithms).

Algorithm 3: Reproduction Operator for Extractive Summarization

```
Input  : $P = [S_i] = [\{s_j\}]$: A list of summaries or parents (each a set of
         sentences)
         $L$: length constraint
Output: $N = \{s_j\}$: child summary as a set of sentences
Function Reproduction $(P, L)$:
    $pool := \bigcup_i S_i$
    $N := SampleCandidate(pool, L)$
```

Artificial Bee Colony:
Swarm Intelligence (SI) refers to the collective behavior of decentralized, self-organized artificial systems. The agents in such systems follow very simple rules, and although there is no centralized control structure, local and random interactions between agents lead to the emergence of *intelligent* global behavior, unknown to the individual agents themselves (Beni and Wang, 1993; Bonabeau et al., 1999; Parsopoulos and Vrahatis, 2002).

While the population of the GAs consists of candidate solutions, the swarm population is made up of agents which search the solution space and interact locally with the environment. The candidate solutions are points in the space investigated by the agents. The agent interactions with the solution space usually consist in evaluating a given area.

To apply an SI algorithm to an optimization problem, one must define a space where candidate solutions live. A point in the space of candidate solutions corresponds to one solution. Simple communication channels allow agents to exchange information about promising areas. Examples of such natural systems are *ant colonies*, *fireflies glowing*, *fish schooling* or *bird flocking*.

One successful model we decided to follow in this work is the model of *honey bees* searching for nectar in a field, also known as Artificial Bee Colony (ABC) (Karaboga and Basturk, 2007; Karaboga et al., 2014). In ABC, there are three groups of bees: employed, onlookers and scouts bees.

There is a population of employed bees who each investigate one food source at a time (the number of employed bees in the colony is equal to the number of food sources investigated in parallel). Employed bees collect food from their food source and dance in this area after evaluating the quantity of food in the direct neighborhood. They measure the quantity of food with the so-called *nectar function* which is the analogy to the fitness function in the GA. The dance indicates the amount of food in the area identified by the employed bee.

When the food source is abandoned, the employed bee becomes a scout and starts to search for a new source elsewhere. Onlookers watch the dances of employed bees and choose food sources which are especially promising. The overall behavior allows the swarm to find areas which contain a lot of nectar.

In order to adapt this algorithm to summarization, we also use a simple analogy:

- **Food location** The locations in the field are the candidate solutions which are the valid extractive summaries. The number of food locations considered in parallel is a hyper-parameter equivalent to the size of the population in the Genetic Summarizer.

- **Location coordinates** The summaries are points in the space searched by the bees. The coordinates are given by the binary vector indicating which sentences the summary contains.

- **Nectar function** The nectar function which evaluates the food locations (i.e., summaries) is the function to maximize. It corresponds to the fitness function.

- **Employed bees local search** At each iteration, employed bees evaluate the direct neighborhood of their assigned food location. To move to a neighbor, a sentence is randomly removed from the current summary (i.e., food location) and a new sentence that does not violate the length constraint is added. This is the analogy to a mutation and is also described by algorithm 2.

- **Employed bees dance and onlooker bees** When each employed bee has evaluated a summary, all the summaries are scored and sorted. Onlooker bees observe the resulting distribution of scores and randomly choose one location to join and help with the neighbor search. This choice is based on the following probability:

$$P_i = \frac{score_i}{\sum_k score_k} \tag{3.32}$$

  As a result, onlookers choose high scoring locations (i.e., summaries) more often. Thus, onlooker bees intensify local search in promising areas.

- **Scouting bees** An employed bee stays in place if the neighbor it evaluates is not better than its current location. After several iterations at the same place, the employed bee abandons it and becomes a scouting bee. To move to another place, the scouting bee selects a random valid summary. This is equivalent to generating a random individual from the initial population in the Genetic Summarizer. The number of iterations before becoming a scouting bee is the second hyper-parameter.

Comparison of stochastic search algorithms:
In the Genetic Summarizer, the reproduction produces significant changes in the summaries, because, on average, half of the genotype of the child is different from its parents. But at the same time, it stays in a reasonable distance range from its parents because it also keeps half of the genotype from each parent (on average).

In this sense, we say that the Genetic Summarizer has efficient mid-range search capabilities (Goldberg, 1989). The local search is much reduced because it is done via mutations happening randomly and (potentially) rarely in the population.

The long-range search is done via the insertion of random individuals into the population whenever the population becomes too small. A new completely random individual is quite likely to have a low fitness score and to die in the next generation with few opportunities to reproduce or mutate.

ABC presents complementary search capabilities. The employed bees perform an intensive local search around a specific location and the onlooker bees help them around the locations of interest (Karaboga et al., 2014). For long-range search, the scout bees regularly look for new locations and investigate each new area for at least t rounds (where t is the hyper-parameter controlling the number of attempts before becoming a scout bee).

However, in ABC, the mid-range search is limited because it is achieved only by, either successfully applying several local movements, or by randomly scouting the mid-range areas, both of which are unlikely.

Similarly, SA is solely based on local search. However, it is possible to move to a neighbor even if it has lower fitness than the current position. The annealing scenario remains dependent on initialization because it is difficult to move far away from bad local optima. This approach can benefit from random restarts for long-range searching capabilities (Bianchi et al., 2009).

Even if we did not conduct an extensive hyper-parameters optimization, we observe that ABC has fewer hyper-parameters than GA, which makes it simpler to optimize. The Simulated Annealing has only one hyper-parameter.

3.2.2 Is GPO practical?

We have described several solutions to optimize any objective function without requiring particular mathematical properties. Now, we test whether these solutions are applicable to the summarization use-case.

For this, we implemented the optimization strategies and test their behavior for different summary scoring functions. These experiments reveal that GPOs are practical for summarization as they are capable of yielding high-scoring summaries. This opens up several possibilities for future research as the summary scoring functions can be unconstrained and searched independently from the optimization strategy.

Improvements over Greedy:

In this experiment, we test the previous algorithms by measuring the fitness of the summary they extract when optimizing two important summary scoring functions:

- **ROUGE-1** This function is submodular and therefore Greedy-M is expected to perform well (Lin and Bilmes, 2011). Indeed, Greedy-M is the algorithm prescribed when the objective function is submodular. We report results using a submodular function in order to also estimate the performances of GPOs on this special case. The experiments reveal that, when the function is submodular, it may still be preferable to employ the Greedy-M optimization.
- **Jensen-Shannon** This is the $\theta_{JS} = -JS(D, S)$ described in the previous section. This function is also important for summarization but does not have nice exploitable mathematical properties that can be leveraged by optimization techniques. In particular, it is neither linearly factorizable nor submodular. In this case, we observe that GPOs give a large improvement over greedy algorithms.

We report a comparison of optimization algorithms for these two functions in figure 3.2 for one representative topic of TAC-2008.

For θ_{JS}, the GA and other meta-heuristics are capable of identifying better summaries in a decent amount of time. This already suggests that meta-heuristics are usable in practice for complex fitness functions.

However, this effect is weaker when for the submodular function ROUGE. In this case, Greedy-M is an efficient and effective optimization strategy. GPOs can find better solutions than Greedy-M only if they run for a longer time (about 25 seconds).

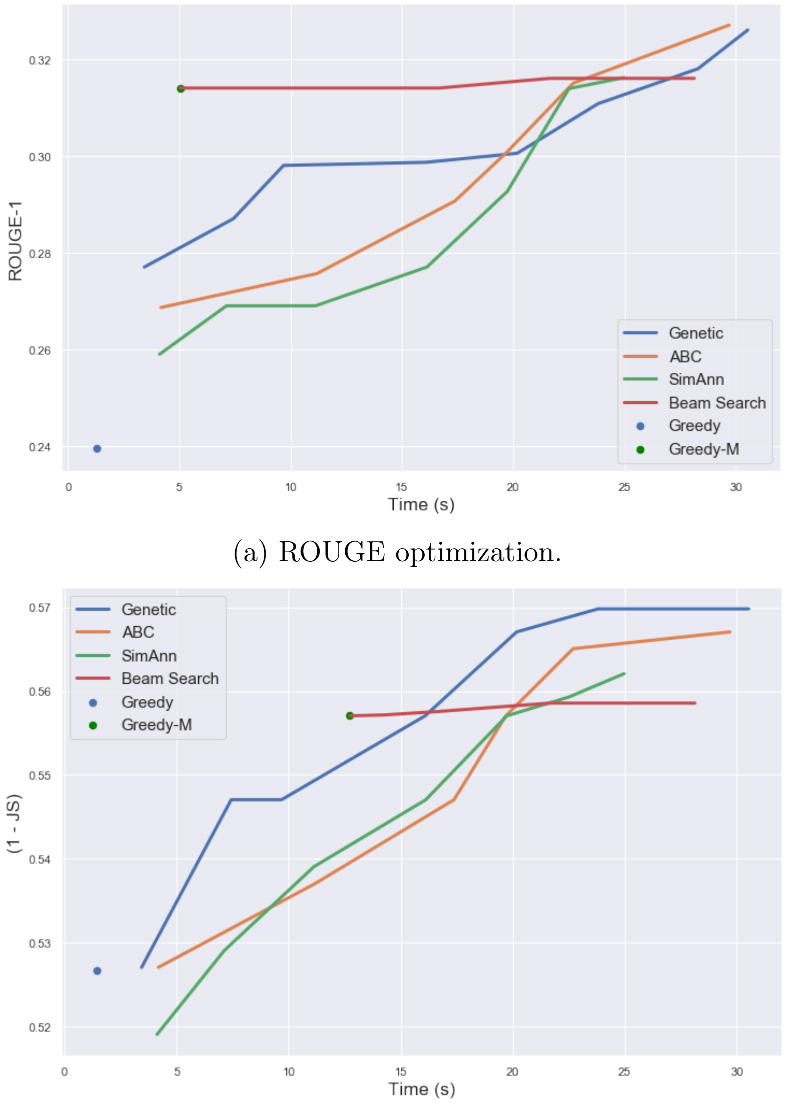
Better summarizers simply via better optimizers:

In the previous section, we noticed that most summary scoring functions from existing systems were optimized by a greedy algorithm. Thus, we ran a simple experiment: we optimized these summary scoring functions with the more powerful optimization techniques described above.

Each of the summary scoring function from table 3.2 (Edmund., TF·IDF, LexRank, KL and JS) is now optimized with every optimization strategy presented above. Only ICSI is left out because it is already optimized exactly by an ILP (so no improvement is possible).

We used both benchmark datasets: TAC-2008 and TAC-2009 and table 3.3 contains the average improvement (over both TAC-2008 and TAC-2009) obtained by switching from greedy to another optimization strategy. To test the practicality of these optimization strategies, they receive a time budget of 30 seconds per topic.

Random Search is not capable of extracting better summaries and often performs worse than Greedy. Therefore, we do not consider it anymore. Indeed, most of the randomly selected summaries are poor and have low fitness.



(a) ROUGE optimization.

(b) JS optimization.

Figure 3.2: The figure 3.2a is the fitness vs. time comparison of the various optimization procedure when ROUGE is used as the fitness function; figure 3.2b is the same comparison when JS is used as the fitness function.

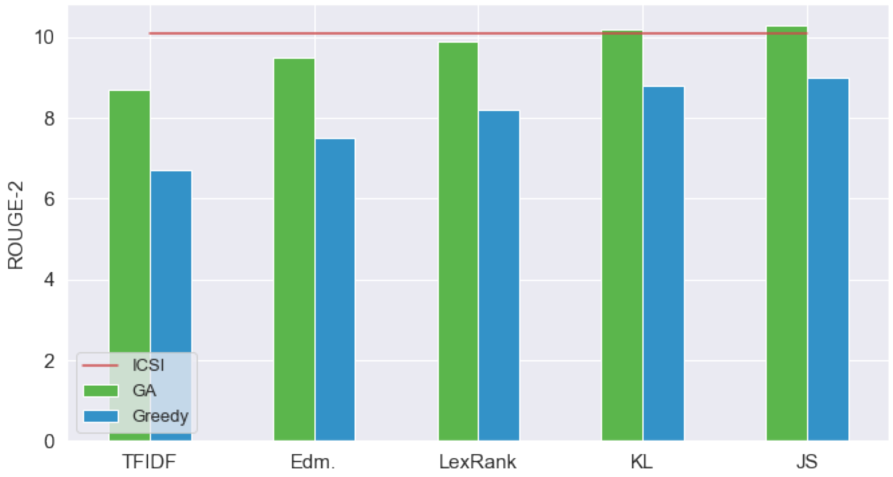


Figure 3.3: Performance improvements when the genetic algorithm is used instead of greedy for TAC-2008 (measured with ROUGE-2).

However, the other optimization strategies all show consistent improvements over Greedy. In particular, SA, GA and ABC perform significantly better [3] than Greedy without significant difference between each other. This is particularly interesting because we could significantly improve summarizers without modifying their core assumptions encoded by the summary scoring functions.

Not only this encourages the use of GPO for subsequent summarization systems, but it also motivates the investigation of θ independently from the chosen optimization strategy (since any θ can be optimized by a GPO).

Furthermore, figure 3.3 shows the improvements obtained for each summary scoring function when the genetic algorithm is used instead of the greedy algorithm. It is particularly interesting to observe the simple and old from Edmundson (1969) performing close to the strong ICSI summarizer (Gillick and Favre, 2009). In fact, Kedzie et al. (2018) also recently observed that even modern end-to-end abstractive summarizers mostly leverage features like sentence position, which is the **Location** method from Edmundson (1969).

Remark that some of these scoring functions, like $\theta_{TF \cdot IDF}, \theta_{Edm.}$ or $\theta_{LexRank}$ are linear and could be optimized directly with an ILP which could give them another performance boost. However, θ_{KL} and θ_{JS} have to be optimized by a GPOs.

Time efficiency:
Different summarization applications may come with different runtime limitations. In these experiments, we arbitrarily fixed 30 seconds as the time limit. During manual annotations reported in chapter 4, our annotators took an average of 30 minutes to read a document set. Thus, a system summarizing a similar document set in 30 seconds is 60 times faster.

Furthermore, the ILP used for computing upper-bound in table 3.2 has a simi-

[3] at 0.01 with significance testing done with t-test to compare two means

Optimizer	R-1↑	R-2↑	JS-1↓	JS-2↓
Greedy-M	+1.1	+0.3	-1.7	-1.1
Beam Search	+1.1	+0.4	-1.8	-1.1
Random Search	+0.2	-0.1	+0.2	-0.1
SA	+1.5	+1.0	-2.2	-1.5
GA	**+1.6**	+1.1	**-2.3**	**-1.9**
ABC	+1.4	**+1.2**	-2.1	-1.3

Table 3.3: The average improvement in extracted summaries observed when switching form Greedy to another optimization algorithm (reported in pp). The average is taken over every θ's and both datasets (TAC-2008 and TAC-2009).

lar runtime of 25 seconds per topic on average. Even in the submodular case, the Greedy-M algorithm takes an average of about 10 seconds per topic. It is 3 times faster than the limit we fixed for the GPOs, but it remains the same order.

One could greatly improve the speed by optimizing the code and removing the plotting and debugging check-points. In fact, population-based optimization techniques like GA and ABC can be easily parallelized (Bianchi et al., 2009) which would further improve their efficiency and scalability.

In general, if a strict time budget T is given, one can combine greedy approaches and GPOs to get the best of both worlds. First, the greedy algorithm runs and generate its solution $S^{(0)}$. After the greedy algorithm has terminated, if the time budget is over, $S^{(0)}$ is the final result. Otherwise, $S^{(0)}$ can be used as starting point for GPOs: as the initial position for Simulated Annealing or inserted in the initial population of population-based GPOs (Genetic Algorithm and Artificial Bee Colony). Then, the GPO can search for improvements over $S^{(0)}$ until the time budget is over. This results in a simple algorithm exploiting the allocated time budget.

3.3 Approximate Upper-Bound Computation

In the previous sections, we motivated the use of GPOs as extraction techniques in order to free the summary scoring functions from previously imposed constraints.

In this section, we discuss another direct benefit stemming from such algorithms: one can compute better upper-bound estimates of evaluation metrics for which the upper-bound cannot be found efficiently. Indeed, optimizing an evaluation metric means finding its upper-bound.

When this optimization can be done exactly, the true upper-bound is found. This is the case for ROUGE because its optimization can be framed as an ILP and solved exactly (Takamura and Okumura, 2010).

Unfortunately, this is not possible for many realistic evaluation metrics. In particular, we used GPOs to compute upper-bound estimates for two important evaluation metrics introduced previously: JS-Eval (Lin et al., 2006) and Automatic

Pyramid (PEAK) (Yang et al., 2016). These are two summary scoring functions for which no ILP can be used to find the exact upper-bound.

3.3.1 AUB Algorithm

Let $\tilde{\theta}$ be an evaluation metric for which we want to estimate the upper-bound: the set of sentences maximizing $\tilde{\theta}$. Formally, we have to solve the following optimization problem:

$$S^* = \underset{S}{\operatorname{argmax}}\, \tilde{\theta}(S, R^*) \tag{3.33}$$

Here R^* indicates the reference summary. This is the same optimization objective used by a summarizer extracting a summary as defined in section 3.1. However, $\tilde{\theta}$ uses of reference summaries.

We describe a simple procedure in algorithm 4 (AUB) which takes advantage of the previously defined optimization techniques to compute a strong estimate of the upper-bound for any function. Each algorithm optimizes the function and returns a solution. The highest scoring solution is kept since the upper-bound is at least as high-scoring as the best one. Using the max over several different optimization techniques results in better approximations for the whole dataset because some techniques may return better solutions for some topics but not all.

The AUB algorithm is itself a GPO which could optimize a summary scoring function as part of a summarizer. In fact, it will result in better optimizations than any individual algorithm presented before. However, this comes at the cost of more computation time. The upper-bound of an evaluation metric has to be estimated only once and, thus, we can afford a higher runtime. In our examples, we let AUB ran for around 5 minutes per topics. For comparison, in our experiments with summarization systems of chapter 4, we let the genetic algorithm ran for only 30 seconds.

Algorithm 4: Approximate Upper-Bound Computation Algorithm

```
Input  : $D = \{s_1, \dots, s_n\}$: document as a set of sentences
         $\theta$: objective function
         $L$: length constaint
         $A$: list of GPOs
Output: Approximate upper-bound as a set of sentences
Function $AUB(A, D, \theta, L)$:
    $solutions := []$
    for $a \in A$ do
        $solutions \leftarrow a(D, \theta, L)$
    end
    return $\underset{s \in solutions}{\operatorname{argmax}}\ \theta(s)$
```

3.3.2 Examples: JS and PEAK

We now present two important examples of approximate upper-bound computations. First, JS-Eval (Lin et al., 2006), despite being a simple function, does not exhibit any convenient mathematical properties useful for optimization. Second, the automatic version of Pyramid PEAK (Yang et al., 2016) is a complex and computationally expensive metric which also requires an approximate upper-bound computation. PEAK is a strong contestant for replacing ROUGE because its scoring process is more semantically aware.

JS approximate upper-bound:
We saw in section 3.1 that JS divergence between the n-gram distributions of the candidate summary and the reference summaries is a valuable automatic metric (Lin et al., 2006). It provides an interesting alternative for ROUGE. Here, we briefly remind of its formulation:

$$\theta_{JS-Eval}(S) = -JS(S||R^*) = \frac{1}{2}(KL(S||M) + KL(R^*||M)) \tag{3.34}$$

Where S is the candidate summary, R^* is the reference summary and M is the average n-gram distribution $M = \frac{1}{2}(\mathbb{P}(S) + \mathbb{P}(R^*))$.

JS cannot be linearly decomposed into sentence scores (Louis and Nenkova, 2013). Thus, no ILP formulation can be employed to optimize it. Furthermore, JS is not submodular (Louis and Nenkova, 2013), and there is no guarantee about the quality of the solution extracted by the greedy algorithm.

Nevertheless, it is useful to have an idea about the upper-bound when JS-Eval is used as the evaluation metric because it contextualizes the scores obtained by summarization systems. We use AUB to compute the approximate upper-bound of JS-Eval for both TAC-2008 and TAC-2009. We note it (JS-Eval-N, AUB). The results are reported in table 3.4 for the unigram version and in table 3.5 for the bigram version. For comparison, we also report the scores obtained by the ROUGE-2 upper-bound when evaluated with JS-Eval-1 in table 3.4 and by JS-Eval-2 in table 3.5. We also estimated the upper-bound of JS-Eval with a greedy algorithm as it was done in table 3.1 (JS-Eval-N, Greedy).

	TAC-2008↓	TAC-2009↓
R-UB	.387	.381
(JS-Eval-1, Greedy)	.399	.392
(JS-Eval-1, AUB)	**.362**	**.353**

Table 3.4: Comparison of three methods to compute the JS-Eval-1 upper-bound on TAC-2008 and TAC-2009: a) computing the ROUGE-2 upper-bound, b) JS-Eval-1 optimized with the Greedy algorithm and c) JS-Eval-1 optimized with AUB described in algorithm 4. (All approaches are scored with JS-Eval-1)

Remember that divergence scores should be as low as possible. Then, it is clear that AUB yields much better upper-bound estimates than the greedy algorithm. Interestingly, the upper-bound for ROUGE-2 gives better upper-bound estimates for

	TAC-2008↓	TAC-2009↓
R-UB	.552	.543
(JS-Eval-2, Greedy)	.562	.557
(JS-Eval-2, AUB)	**.528**	**.498**

Table 3.5: Comparison of three methods to compute the JS-Eval-2 upper-bound on TAC-2008 and TAC-2009: a) computing the ROUGE-2 upper-bound, b) JS-Eval-2 optimized with the Greedy algorithm and c) JS-Eval-2 optimized with AUB described in algorithm 4. (All approaches are scored with JS-Eval-2)

JS-Eval-N than the greedy optimization (This could already be seen in table 3.1). This provides a new understanding of the results described in table 3.2. In particular, we realize that systems are farther away from the upper-bound than previously expected. This leaves a large room for improvements, for summarization evaluated with JS-Eval.

PEAK: automatic Pyramid:
Previously, in section 2.2, we introduced PEAK, an automated version of the manual Pyramid annotations. PEAK (Yang et al., 2016) uses *clauses* as the content expressing units and represents them as propositions in the open IE paradigm. An open IE proposition is a triple of subject, predicate and object phrases. PEAK uses the state-of-the-art system clausIE (Del Corro and Gemulla, 2013) for proposition extraction.

While PEAK includes the automatic creation of Pyramid sets from reference summaries, as well as automatic Pyramid scoring of system summaries, in this work, we use PEAK for automatic scoring only. As for the Pyramid sets, we can assume that these have already been created, either via PEAK or by humans (e.g., using the TAC 2009 data[4]). We remind that the Pyramid set is made of Semantic Content Units (SCUs) each associated with a weight indicating its importance.

Since automatic scoring with PEAK requires the Pyramid sets to consist of open IE propositions as the automated counterparts of the SCUs, we first converted the manually constructed SCUs into open IE propositions by applying clausIE on the *SCU labels* (a sentence describing an SCU). As a result, each Pyramid set is represented as a list of propositions $\{p_j\}$ with a weight taken from the underlying SCU.

For scoring, PEAK processes a system summary with clausIE, converting it from a list of sentences to a list of propositions $\{p_i^{(S)}\}$. A bipartite graph G is constructed, where the two sets of nodes are the summary propositions $p^{sum} = \{p_i^{(S)}\}$ and the pyramid propositions $p^{pyr} = \{p_j\}$. An edge is drawn between $p_i^{(S)}$ and p_j if the similarity is above a given threshold noted t. PEAK computes the similarity with the ADW system (*Align, Disambiguate and Walk*), a system for computing text similarity based on WordNet, which reaches state-of-the-art performance but is slow to compute (Pilehvar et al., 2013).

Since each system summary unit can be aligned to at most one SCU, the align-

[4] http://tac.nist.gov/2009/Summarization

ment of the summary propositions $\{p_i^{(S)}\}$ and the Pyramid propositions $\{p_j\}$ is equivalent to finding a maximum weight matching, which PEAK solves using the Munkres-Kuhn bipartite graph algorithm. The final Pyramid score is computed from the matched Pyramid propositions $\{p_j\}$.

PEAK approximate upper-bound:
Let $D = \{s_i\}$ be a document collection considered as a set of sentences. Again, a summary S is simply a subset of D. We use p^{pyr} to denote the set of propositions in the Pyramid sets extracted from the SCU labels using clausIE.

Intuitively, the task is to extract the set of sentences which contain the propositions matching most of the highest-weighted SCUs, thus resulting in the best matching of propositions. Unfortunately, it cannot be solved directly via ILP because of the Munkres-Kuhn bipartite graph algorithm within PEAK.

While Munkres-Kuhn is itself an ILP, we solve a different problem. In our optimization problem, Munkres-Kuhn would act as a constraint because we are looking for the best matching among all valid matchings. Munkres-Kuhn only yields the valid matching for one particular set of sentences.

In theory, one global ILP can be written down by enumerating all possible matchings in the constraints but it will have a completely unrealistic runtime. Instead, we have to rely on search-based optimization techniques and the AUB from algorithm 4.[5]

In order to run AUB with PEAK as the fitness function, we need to evaluate many summaries. Every time a summary is evaluated with PEAK, we need to compute the similarity between the propositions of the candidate summary with the propositions in the Pyramid set. Then, the matching algorithm (Munkres-Kuhn) has to be run.

The runtime might become an issue, because the similarity computation between propositions via ADW is fairly slow. However, all the necessary information is available in the similarity matrix A defined by:

$$A_{ij} = ADW(p_i^D, p_j^{pyr}) \tag{3.35}$$

Here A_{ij} is the semantic similarity between the proposition p_i^D from the source document i and the proposition p_j^{pyr} from the Pyramid set j. A has dimensions $m \times n$ if m is the number of propositions in the document collection and n is the number of propositions in the Pyramid set. We keep the AUB runtime low by pre-computing the similarity matrix A for each topic in TAC-2009. However, pre-computing A for the whole dataset took 2 weeks on a compute server with 10 CPUs.

In table 3.6, we report the PEAK scores of summaries extracted by the AUB noted (PEAK, AUB). For comparison, we also report the scores of summaries extracted by the greedy algorithm (PEAK, Greedy) and the ROUGE-2 upper-bound (R-UB). During the computation of the upper-bound with AUB, we set the similarity threshold parameter to $t = .6$. For evaluating the extracted summaries, we

[5] In practice, for this scenario, the genetic algorithm always provided the best summary.

report PEAK for both $t = .6$ and $t = .7$.

	PEAK-60	PEAK-70
R-UB	0.509	0.307
(PEAK, Greedy)	0.518	0.309
(PEAK, AUB)	**0.579**	**0.379**

Table 3.6: Comparison of three methods to compute the PEAK upper-bound on TAC-2009: a) computing the ROUGE-2 upper-bound, b) PEAK optimized with the Greedy algorithm and c) PEAK optimized with AUB described in algorithm 4. (All approaches are scored with PEAK using $t = .6$ (PEAK-60) and $t = .7$ (PEAK-70))

We observe large gaps between the scores produced by R-UB and (PEAK, AUB). This observation empirically confirms that the two metrics measure different properties of system summaries. Unlike for JS-Eval, even the greedy optimization already finds better estimates of the upper-bound than the summaries extracted for the ROUGE-2 upper-bound.

Chapter Summary

- Without loss of generality, summarization can be viewed as two components: a **summary scoring function** and an **optimization technique**. This decomposition is noted (θ, O).
- Every summarizer defines implicitly or explicitly a summary scoring function θ. We can measure whether this scoring function correlates with human judgments and thus measure whether systems follow a strategy similar to humans.
- The θ of existing summarizers present surprisingly low correlations with human judgments.
- GPO techniques, which can optimize any arbitrary function can be adapted to summarization. They are shown to be both **efficient and effective** enough for the summarization use-case.
- Since GPOs are usable, there is **no need to constrain** θ to exhibit particular mathematical properties. This augments the expressive power of θ. This new advantage is leveraged by the following chapter.
- Existing summarizers for which θ has been identified can be significantly improved by using a GPO (like the Genetic Algorithm) instead of a greedy optimization.
- With GPO, the upper-bound of evaluation metrics without convenient mathematical properties can be estimated. (Important examples: JS-Eval and PEAK upper-bound have been estimated)

Chapter 4

Learning the Summary Scoring Function

So far, we have argued for the separation of the summarization task into its 2 defining components: θ and O. The design of O is mostly an engineering problem which can be informed by specialized works in the field of combinatorial optimization. In section 3.2, we introduced and compared such techniques applied to the summarization task. These experiments demonstrated that GPOs are usable which opens up the possibility to search for an unconstrained summary scoring function.

The focus of summarization research can now be put on the discovery and study of summary scoring functions. Indeed, these functions should encode all relevant quality aspects of a summary (that we wish to model), such that by maximizing them we would obtain the best possible summaries.

Additionally, chapter 3 discussed ways to analyze the summary scoring function θ independently from O directly with human judgments. The summary scoring functions of existing systems display low correlations with human scores. Here, we propose to explicitly develop summary scoring functions having high correlations with humans.

In the next chapter (5), we will propose a formulation of θ derived within an abstract theoretical framework rooted in information theory. However, in this chapter, we tackle the problem of discovering θ automatically from data.

In particular, we aim to infer it from observed scored summaries for the specific aspect of content selection as evaluated by humans. Indeed, content selection is the main problem of summarization and lossy semantic compression in general. Other aspects such as readability or grammaticality are general problems of NLG that we do not consider here.

For complex or vaguely defined tasks, it is common to employ Machine Learning (ML) tools to automatically discover statistical regularities in available data (Bishop, 2006). The ML approach requires minimal prior specification and allows to search θ from the available datasets of human judgments. However, important design choices remain: supervision signal, the learning constraints or the feature space. These dimensions of variations are discussed in section 4.1.

The vast majority of previous work focused on scoring smaller textual units (e.g., sentences) with ROUGE as supervision (Yao et al., 2017). This is a restricted scenario and we investigate a much wider spectrum of possibilities. In particular, we propose to learn the summary scoring function at the summary-level instead of manually specifying a combination of the scores of the sub-elements. This gives access to much more powerful features, especially ones capturing redundancy.

Additionally, we discuss how to incorporate the available human judgments in the learning setup. This constitutes a more meaningful supervision signal than simply relying on automatic evaluation metrics like ROUGE and contributes to **RQ4**.

Finally, the different summary scoring functions are compared based on their ability to correlate with humans and to extract high-quality summaries after optimization. The results confirm the superiority of the unconstrained scoring functions and answer **RQ3**.

4.1 The Matrix of Possible Scoring Functions

In chapter 3, we saw that θ has rarely been learned directly at the summary level. Instead, smaller components such as words, n-grams or sentences have been scored either by trained models or unsupervised heuristics. In such cases, a summary-level scoring function has to be defined either implicitly or explicitly by combining the scores from the smaller units (Carbonell and Goldstein, 1998; McDonald, 2007).

We hypothesize that constraining θ greatly limits the expressive power of the resulting scoring functions.

In particular, it appears difficult to model sentence interactions and redundancy simply by combining sub-elements scores (Carbonell and Goldstein, 1998). Hopefully, we showed previously that GPOs are practical, thus θ can be freed from its previously imposed constraints.

In fact, this perspective generalizes previous works because combinations of scores from smaller units are a special case of the unconstrained scenario. For instance, linear (McDonald, 2007) or submodular (Lin and Bilmes, 2011) scoring functions become special cases of this more general setup where θ is learned without particular constraints.

Additionally, we discuss another degree of freedom when learning θ by considering various supervision signals instead of solely ROUGE scores. The dimensions of variations of the θ learning setup can be summarized in a matrix. We briefly introduce these axes here and detail them in the following section. The two main axes we consider (supervision and constraints on θ) are summarized in table 4.1.

- **Axis 1: Supervision**: Ideally, the supervision would come directly from humans but manual annotations are expensive to obtain. Thus, automatic evaluation metrics like ROUGE usually provided supervision (Cao et al., 2015b). However, other metrics could also be considered.

	ROUGE-2	JS-Eval-2	PEAK	h
linear constraint	θ^{lin}_{R2}	θ^{lin}_{JS2}	θ^{lin}_{Peak}	θ^{lin}_{h}
No constraint	θ_{R2}	θ_{JS2}	θ_{Peak}	θ_{h}

Table 4.1: Summary of the two main axes of variations: i) the supervision signal which may come from various approximation of humans up until actually using human judgment datasets as target scores. ii) the constraints imposed on θ: either linear or no constraint. The top left corner is where most previous works lie. For the second row, non-linear features become available.

- **Axis** 2**: Constraints on** θ: This axis corresponds to the learning algorithms. Indeed, each learning algorithm specifies a hypothesis space restraining the choice of θ. These can be categorized into broader categories, such as algorithms that impose linearity constraints, submodularity constraints or no strong constraint (except for continuity and smoothness).
- **Axis** 0**: Features**: Conceptually, the features could be part of the learning algorithm as they contribute to the specification of the hypothesis space. However, most learning algorithms work independently from the specific feature choice. We refer to the features as axis 0 because we do not investigate them in great detail. Instead, we fixed a simple and *standard* feature set to focus the comparison on the other two axes. We describe them in the next section.

4.1.1 Axis 1: Supervision

The source of supervision is an important dimension of variation when learning a summary scoring function. It defines which summaries are considered as *good* and *bad*. If we wish our system to mimic humans, the supervision should ideally come directly from humans. Because obtaining such manual annotations is expensive, a large body of work employed surrogate functions such as ROUGE as a proxy for summary quality.

However, we already presented several promising alternative evaluation metrics in chapter 3. Furthermore, even if they are scarce, we may want to leverage the human judgments made available during the DUC/TAC shared tasks. In section 4.3, we propose a simple method to incorporate existing human judgments into the training.

Here, we briefly described these various supervision possibilities.

ROUGE:
System summaries are commonly evaluated using ROUGE (Lin, 2004b), a recall-oriented metric that measures the n-gram overlap between a system summary and a set of human-written reference summaries. Since it is the standard evaluation metric, it seems reasonable to adopt it for supervision as well. We study the properties of this metric in great detail in section 4.2.

In fact, previous works used ROUGE variants extensively as target scores for textual units like n-gram (Li et al., 2013) or sentences (Cao et al., 2015a). Alternatively, structured output learning can receive ROUGE feedback at the summary-level and propagates it to the smaller units (Nishikawa et al., 2014; Takamura and Okumura, 2010; Sipos et al., 2012).

ROUGE is a simple and natural choice for training systems. It is easy to use and fast to compute, giving the possibility to generate many scored summaries. This is a cheap way to construct a training dataset for summarization systems.

Other Automatic Metrics:
While ROUGE seems to be a decent supervision signal, it also has several problems (see section 2.2). ROUGE has been widely criticized for being too simplistic and not suitable for capturing important quality aspects (Lloret et al., 2018). In particular, ROUGE cannot detect sentences which are semantically equivalent but expressed with different words (Nenkova et al., 2007).

ROUGE is just one possible proxy for summary quality – there are other automatic metrics to evaluate system summaries, which also correlate well with human judgments (Louis and Nenkova, 2013).

In chapter 3, we already introduced two potential alternatives: Automatic Pyramid (PEAK) and the Jensen-Shannon divergence between n-gram distributions of the candidate summary and the reference summaries (JS-Eval-N).

Given the recent advances in the automatic evaluation of summaries regarding content selection, we believe that empirical research in summarization should progressively move away from ROUGE towards more meaningful metrics for both training and evaluating systems. Therefore, we conduct a systematic comparison of the systems trained with the metrics described above.

Humans:
Ultimately, summarization systems are expected to mimic humans, meaning that their internal scoring functions should ideally correspond to human scores.

We note h as the function explaining the observed human judgment data, which is an approximation of the *ideal* human scoring function H. The ideal H refers to the true process by which humans summarize texts including all biological phenomena happening in the brain. Such H is not accessible and might not even be well-defined for an *average human summarizer*.

In particular, saying that $h \approx H$ even if H is not observable means we recognize that the existing human judgments h are not necessarily a perfect signal. They could be improved. For example, improving annotations guidelines to ensure high inter-annotator agreement could improve the resulting human judgment datasets h. However, for the purpose of this thesis, we assume h is given and consider it as a meaningful approximation of this ideal H.

Then, we can learn the objective function θ from a pool of manually annotated summaries (h) to ensure the extraction of summaries considered *good* by humans. This explicitly targets the extraction of high-quality summaries as measured by humans and limits undesired gaming of the target evaluation metric:

$$\theta \approx h \approx H \tag{4.1}$$

The learning setup is the same when an automatic metric is used instead of h, but h is presumed to be a much better indicator of the average human scoring function H. Indeed, current evaluation metrics are only weak approximations of humans. They can only display mild correlations with humans in carefully controlled settings.

Unfortunately, available human judgment datasets contain only a few data points. The learned θ might be ill-behaved, driving the optimizer to regions of the feature space unseen during training where θ wrongly assumes high scores. We examine these challenges and potential solutions in section 4.3.

4.1.2 Axis 2: Learning Constraints on θ

Once a supervision signal from axis 1 is chosen and a corresponding dataset of scored summaries is available, a scoring function θ can be learned.

Formally, let $\mathcal{D}$ be the dataset of topics $\{\mathcal{T}_1, \dots, \mathcal{T}_n\}$. Each topic $\mathcal{T}_i$ has: $i)$ sources D_i and $ii)$ a set of scored summaries $S_i = \{s_{i,1}, \dots, s_{i,m}\}$. Each summary $s_{i,j}$ has a target score $\theta^*(s_{i,j})$. For example, if θ^* is ROUGE-N, the score of a summary $s_{i,j}$ is its n-gram overlap with reference summaries. However, θ^* could be any evaluation metric or score manually given by humans (h).

We aim to learn a function θ approximating θ^* on dataset $\mathcal{D}$ without accessing the reference summaries or any other evaluation resources. Thus, θ could, for example, minimize the following loss:

$$\mathcal{L}(\theta) = \sum_{D_i \in \mathcal{D}} \sum_{s_{i,j} \in S_i} \|\theta(D_i, s_{i,j}) - \theta^*(s_{i,j}, R)\|^2 \tag{4.2}$$

Here, the θ should minimize the squared distance from θ^* over the available training data. While we stick with this loss for our experiments, several other loss functions would be possible. For instance, a max-margin loss (Sipos et al., 2012) or ranking loss could be employed.

Building from this general learning goal, we explore how the linear constraint can be imposed on θ in a general way and see how this simplifies when this constraint is removed. While we did not implement it, the same analysis could be done for the submodular constraint by using general results from Tschiatschek et al. (2018).

Learning with the linearity constraint:
Now, we present a simple but general way to learn a summary scoring function with linearity constraints. This allows comparing θ trained with the linearity constraint (optimized with ILP) and θ trained without constraint (optimized with a GPO).

By linearity, we mean that θ should be linearly factorizable with respect to sub-elements like sentences. In the simplest case, θ takes the following form:

$$\theta(S) = \sum_{e \in S} f_\omega(e) \tag{4.3}$$

The function f_ω of parameter ω scores each element e of the summary S. The overall score of S is the sum of the scores of its elements. The latter are not learned independently, instead, f_ω is learned such that θ as a whole matches the target summary score.

This view is simplistic and it is possible to maintain linearity while accounting, to some extent, for element interactions:

$$\theta(S) = \sum_{e \in S} f_\omega(e) + \sum_{i>j} g_\gamma(e_i, e_j) \tag{4.4}$$

Here, two functions are jointly learned: f_ω (of parameter ω) is a function scoring individual elements, and g_γ (of parameter γ) is a function scoring pairs of elements. This learning scenario jointly learns the *sentence relevance* f_θ and *redundancy* g_γ that make θ match the target θ^*. This scenario is rather intuitive and neatly fits within the general description proposed by McDonald (2007).

In such a case, before running an ILP solver on θ, one needs to precompute the score for each element (in a 1-dimensional array), but also the score for each pair of elements (in a 2-dimensional matrix).

In fact, one can extend this idea to the interactions between n elements, but this would require to store an n-dimensional tensor for the ILP extraction. In practice, we don't see any benefits after $n = 2$ but the runtime of the ILP explodes with n.

For learning, each element is represented by a feature set ϕ and each pair of elements by $\phi^{(2)}$. Thus, the feature set for a summary S is given by:

$$\Phi(S) = \{\Phi(s)_{s \in S}\} = \{\bigcup_{e \in S} \phi(e) \cup \bigcup_{i>j} \phi^{(2)}(e_i, e_j)\} \tag{4.5}$$

The number of features is variable and depends on the number of sentences in S.

In order to deal with a variable sized input, one could use recurrent neural networks (Hochreiter and Schmidhuber, 1997), but at the cost of losing linearity. Instead, we employ linear models for both f_θ and g_γ:

$$\theta(S) = \sum_{e \in S} \omega \cdot \phi(e) - \sum_{i>j} \gamma \cdot \phi^{(2)}(e_i, e_j) \tag{4.6}$$

By linearity, we end up with the following formulation:

$$\theta(S) = \omega \cdot \sum_{e \in s} \phi(s) - \gamma \cdot \sum_{i>j} \phi^{(2)}(e_i, e_j) \tag{4.7}$$

The resulting feature set at the summary-level is the sum of the features of its elements, together with the sum of the features of pairs of elements:

$$\Phi_+(S) = \{\phi_+(S) \cup \phi_+^{(2)}(S)\} \tag{4.8}$$

$$\text{where } \phi_+(S) = \sum_{e \in s} \phi(s) \tag{4.9}$$

$$\text{and } \phi_+^{(2)}(S) = \sum_{i>j} \phi(e_i, e_j) \tag{4.10}$$

Suppose ϕ is composed of k features and $\phi^{(2)}$ of p features. Then $\phi_+(S)$ is a vector of dimension k, and similarly $\phi_+^{(2)}(S)$ is of dimension p. Finally, Φ_+ has a fixed size of $k+p$.

The function θ as defined in equation (4.6) remains linear with respect to elements and pairs of elements. The parameters ω and γ can be estimated using the loss described by equation (4.2) with standard linear regression algorithm (Pedregosa et al., 2011).

Once θ is learned, we can extract the best scoring summary of a given topic via ILP. Let x be a binary vector indicating whether the element i is in the summary or not. Similarly, let α be a binary matrix indicating whether both elements i and j are in the summary. With L denoting the length constraint, the ILP solving "argmax θ" is given by:

$$\underset{\mathbf{x}}{\operatorname{argmax}} \sum_{e_i \in S} x_i \cdot \theta \cdot \phi(e_i) - \sum_{i \geq j} \alpha_{i,j} \cdot \gamma \cdot \phi^{(2)}(e_i, e_j) \tag{4.11}$$

$$\textbf{such that, } \sum_{i=1}^{m} x_i \cdot len(e_i) \leq K \tag{4.12}$$

$$\forall (i,j), \alpha_{i,j} - x_i \leq 0 \tag{4.13}$$

$$\forall (i,j), \alpha_{i,j} - x_j \leq 0 \tag{4.14}$$

$$\forall (i,j), x_i + x_j - \alpha_{i,j} \leq 1 \tag{4.15}$$

This derivation is a general formulation of existing ideas in optimization-based summarization. McDonald (2007) already introduced the idea of optimizing a linear objective function as the difference of *relevance* and *redundancy*.

Additionally, the idea of providing feedback at the summary level and propagating it to the smaller units originally comes from the structured output learning paradigm. Structured output learning was already used in summarization (Li et al., 2009; Sipos et al., 2012).

Thus, the framework we just presented is following the tradition of summarization systems learning scores for sub-summary units.

Removing the constraints:

In contrast to the constrained case, learning without constraints is conceptually much simpler.

It is sufficient to specify an arbitrary feature set $\Phi(S)$, which may or not include the features from the constrained case. Then, any regression algorithm can be directly applied to this feature set using the loss from equation (4.2).

In general, the resulting function does not exhibit convenient mathematical properties such as linearity or submodularity. This is not problematic since GPO can be used for the extraction. In comparison, the expressive power of an unconstrained θ is greater because the hypothesis space is larger and – more importantly – non-linear but powerful features become available.

4.1.3 Axis 0: Features

The feature choice is usually a crucial step in any ML setup. Here, in order to compare the other aspects of the θ learning framework, we selected a small but *standard* feature set inspired by existing summarization systems.

When learning with the linearity constraint, only sentence-level features are affordable. To preserve linearity, sentence-level features can offer a score at the summary-level only by (weighted) summation. Alternatively, one can define features for pairs of sentences (as described in the previous section).

However, when the constraint is removed, one has also access to features computable only at the summary-level.

TF·IDF (linear):
In section 3.1, we introduced θ_{TF*IDF}: each term in the document receives a score based on its frequency in the source (TF) and its inverse document frequency (IDF) in a background corpus.[1] This is a linear feature as the score of the summary is the sum of the sentence scores.

Document-frequency of n-grams (linear):
As described in section 3.1, θ_{ICSI} produces a score for a summary based on the coverage of frequent bigrams. It is linearly factorizable and can, therefore, be used at the sentence-level. This is known to be a strong signal correlating with importance as ICSI is capable of extracting high-quality summaries (Gillick and Favre, 2009).

Edmundson scores (linear):
In section 3.1, we also discussed the Edmundson approach (Edmundson, 1969), which consists of 4 different sentence scoring methods.

The **Cue-phrase** method counts the number of bonus and stigma words in the sentences. The **Key** method computes the frequency of words (which are not stopwords) in the sentences. The **Title** method measures the overlap between the sentence and the title. Finally, the **Location** method puts more weights on the first and last sentences as they are expected to be more relevant. It also linear because the score of the whole summary is the sum of the sentence scores.

Sentence centrality (linear):
The LexRank method (Erkan and Radev, 2004) also produces a summary scoring function $\theta_{LexRank}$ described in section 3.1. It computes sentence centrality based on a similarity graph where sentences are nodes and edges are drawn between two sentences if their TF·IDF similarity is above a given threshold. The PageRank algorithm is run on the resulting graphs which returns a score for each sentence. The $\theta_{LexRank}$ score of a summary is the sum of the scores of individual sentences.

Number of sentences (linear):
We also use the number of sentences in the summary as a feature because when a summary contains a lot of sentences, they tend to be very short and irrelevant.

[1] The IDF values are computed from the Wikipedia dump of 2017 using scikit-learn: `http://scikit-learn.org/stable/`)

Pairwise redundancy (linear with respect to pairs of sentences):
For each pair of sentences (s_a, s_b), we define the intersection $s_a \cap s_b$ as the set of n-grams appearing in both sentences. The redundancy between s_a and s_b is then the size of their intersection: $|s_a \cap s_b|$. The overall pairwise redundancy of a summary is the sum of the redundancy of all sentence pairs. This feature is used to model redundancy in the constrained case. We used $n = 1$ and $n = 2$. This corresponds to $\phi_+^{(2)}(S)$ defined in the previous section.

JS divergence with source document (non-linear):
Similar to θ_{KL} also discussed in section 3.1, we also employ θ_{JS} as feature. This is an example of a feature that cannot arise from linear (or submodular) combination of sentence scores. This feature can only be used in the unconstrained scenario but it is a strong indicator of importance, as indicated by the results of section 3.1. We chose θ_{JS} instead of θ_{KL} because of its better correlation with humans.

Intra-summary diversity (non-linear):
Let S be a summary composed of several terms w. Each term appears with a frequency $\mathbb{P}_S(w)$ in the summary. A summary with a lot of diversity and therefore low redundancy would exhibit a high entropy in the distribution of its terms. Thus, we define the diversity of S by:

$$\theta_{Div.}(S) = H(S) = \sum_{w \in S} \mathbb{P}_S(w) \cdot \log \frac{1}{\mathbb{P}_S(w)} \tag{4.16}$$

This is a measure of redundancy (as the opposite of diversity) and is also not computable from a linear or submodular combination of sentence scores. It is more meaningful than the *pairwise redundancy* feature defined for the constrained case because it can directly account for the whole summary. The pairwise redundancy feature is bound to overcount terms appearing in the intersection of three sentences.

Remark:
Even though many linear and non-linear features can be investigated, there are some dependencies between the choice of features (axis 0) and the constraints (axis 2). Indeed, non-linear features are incompatible with a linear constraint imposed on θ. Thus, we have two distinct feature sets for the two different kinds of constraints:

- **Linear constraint**: $\Phi_{lin} = \{$TF·IDF, n-gram frequency, Edmundson scores, sentence centrality, number of sentences, pairwise redundancy$\}$.
- **Unconstrained**: $\Phi = \Phi_{lin} \cup \{$ JS, diversity$\}$.

4.2 An Important Example: Exploiting ROUGE properties

One special case in the matrix presented above is learning with ROUGE as supervision in the constrained case. This scenario is the one followed by the majority of previous works (Yao et al., 2017).

Typically, there exist two ways of learning from ROUGE: *(i)* training a model that assigns ROUGE scores to individual textual units (e.g., bigrams or sentences) (Li et al., 2013) or *(ii)* performing structured output learning with supervision at the summary-level propagated to the textual units (Nishikawa et al., 2014; Takamura and Okumura, 2010). In both cases, linearity or submodularity constraints are usually imposed (Sipos et al., 2012).

While we advocate for unconstrained summary scoring functions in general, in the special case where ROUGE is the supervision, we demonstrate that one can exploit the simple mathematical structure of ROUGE to uncover a surprisingly effective linearly factorizable approximation (noted $\tilde{\theta}_R$).

Assuming that ROUGE scores of individual sentences have already been estimated, we derive an approximation of the ROUGE score of a summary from the scores of its sentences. This results in a $\tilde{\theta}_R$ approximating ROUGE, for which an ILP can be solved to extract the optimal summary.

Most importantly, the resulting framework reduces the summarization task (as evaluated by ROUGE) to the problem of scoring individual sentences with their ROUGE scores. Indeed, the overall task is converted into two sequential tasks: *(i)* scoring single sentences, and *(ii)* selecting a set of sentences by solving an optimization problem where the ROUGE score of the summary is maximized. The proposed approximation of ROUGE almost exactly solves *(ii)* when optimized with the proper ILP. Hence, solving the whole problem of summarization (as evaluated by ROUGE) is reduced to solving *(i)*.

In section 4.4, we compare this approximation (noted $\tilde{\theta}_R$) to the general θ learning with linearity constraint presented in the previous section. As hypothesized, we observe strong performances of $\tilde{\theta}_R$ when ROUGE is the evaluation metric. Thus, this derivation is a principled way of approximating ROUGE while preserving the linearity constraint. Unfortunately, such analysis relies on the simplicity of ROUGE and cannot be replicated to more interesting metrics like JS-Eval-N or PEAK. For the two latter ones, we have to use the general framework outlined in section 4.1.

4.2.1 Useful Mathematical Properties of ROUGE

Let $S = \{s_i | i \leq m\}$ be a set of m sentences which constitute a system summary. We use $\theta_{R-N}(S)$ or simply $\theta_R(S)$ to denote the ROUGE-N score of S. Let S^* denote the reference summary and R_N the number of n-gram tokens in S^*. R_N is a function of the summary length in words, in particular, R_1 is the target size of the summary in words. Finally, let $F_S(g)$ denote the number of times the n-gram type g occurs in S. For a single reference summary, ROUGE-N is computed as follows:

$$\theta_R(S) = \frac{1}{R_N} \sum_{g \in S^*} \min(F_S(g), F_{S^*}(g)) \tag{4.17}$$

For compactness, we use the following notation for any set of sentences X:

$$C_{X,S^*}(g) = \min(F_X(g), F_{S^*}(g)) \tag{4.18}$$

$C_{X,S^*}(g)$ can be understood as the contribution of the n-gram g.

ROUGE-N for a pair of sentences:
Using this notation, the ROUGE-N score of a set of two sentences a and b can be written as:

$$\theta_R(a \cup b) = \frac{1}{R_N} \sum_{g \in S^*} C_{a \cup b, S^*}(g) \tag{4.19}$$

We observe that $\theta_R(a \cup b)$ can be expressed as a function of the individual scores $\theta_R(a)$ and $\theta_R(b)$:

$$\theta_R(a \cup b) = \theta_R(a) + \theta_R(b) - \epsilon(a \cap b) \tag{4.20}$$

where $\epsilon(a \cap b)$ is an error correction term that discards overcounted n-grams from the sum of $\theta_R(a)$ and $\theta_R(b)$:

$$\epsilon(a \cap b) = \frac{1}{R_N} \sum_{g \in S^*} \max(C_{a,S^*}(g) + C_{b,S^*}(g) - F_{S^*}(g), 0) \tag{4.21}$$

A derivation of this error correction is correct is given in Appendix C.2.

General formulation of ROUGE-N:
We can extend the previous formulation of θ_R to sets of arbitrary cardinality using recursion. If $\theta_R(S)$ is given for a set of sentences S, and a is a sentence then:

$$\theta_R(S \cup a) = \theta_R(S) + \theta_R(a) - \epsilon(S \cap a) \tag{4.22}$$

We prove in Appendix C.2 that this formula is the ROUGE-N score of $S \cup a$.

Another way to obtain θ_R for an arbitrary set S is to adapt the principle of inclusion-exclusion:

$$\theta_R(S) = \sum_{i=1}^{m} \theta_R(s_i) + \sum_{k=2}^{m} (-1)^{k+1} (\sum_{1 \leq i_1 \leq \cdots \leq i_k \leq m} \epsilon^{(k)}(s_{i_1} \cap \cdots \cap s_{i_k})) \tag{4.23}$$

This formula can be understood as adding up scores of individual sentences, but n-grams appearing in the intersection of two sentences are overcounted. $\epsilon^{(2)}$ is used to account for these n-grams. But now, n-grams in the intersection of three sentences are undercounted and $\epsilon^{(3)}$ is used to correct this. Each $\epsilon^{(k)}$ contributes to improving the accuracy by refining the errors made by $\epsilon^{(k-1)}$ for the n-grams appearing in the intersection of k sentences. When $k = |S|$, $\theta_R(S)$ is exactly the ROUGE-N of S. A rigorous proof and details about $\epsilon^{(k)}$ are provided in Appendix C.3.

Approximation of ROUGE-N for a pair of sentences:
To find a valid approximation of θ_R as defined in (4.23), we first consider the $\theta_R(a \cup b)$ from equation (4.19) and then extend it to the general case.

When maximizing θ_R, scores for sentences are assumed to be given (e.g., estimated by an ML component). We still need to estimate $\epsilon(a \cap b)$, which means, according to (4.21), to estimate:

$$\sum_{g \in S^*} \max(C_{a,S^*}(g) + C_{b,S^*}(g) - F_{S^*}(g), 0) \tag{4.24}$$

At inference time, neither S^* (the reference summary) nor F_{S^*} (number of occurrences of n-grams in the reference summary) are known.

At this point, we can observe that, similar as for sentence scoring, ϵ can be estimated via a supervised ML component. In fact, equation (4.20) has the same form as equation (4.4) with one function scoring sentences et one fonction scoring pairs of sentences. Both can be learned with supervised learning as described earlier.

Thus, this would correspond exactly to the *learning with linearity constraint* scenario presented in the previous section. There, both the scores for individual sentences and the ϵ are learned empirically from data using ML.

However, we found in our experiments that a simple heuristic yields a decent approximation of ϵ. The heuristic uses the frequency $freq(g)$ of an n-gram g observed in the source documents:

$$\sum_{g \in S^*} \max(C_{a,S^*}(g) + C_{b,S^*}(g) - F_{S^*}(g), 0) \approx \sum_{g \in a \cap b} \mathbb{1}[freq(g) \geq \alpha] \tag{4.25}$$

The threshold α tells us which n-grams are likely to appear in the reference summary, and it is determined by grid-search on the training set. This is penalizing n-grams which appear twice in the candidate summary but are likely to occur in the reference summary. In practice, we used $\alpha = 0.3$. However, we experimented with various values of the hyper-parameter α and found that its value has no significant impact as long as it is fairly small (< 0.5). Higher values will ignore too many redundant n-grams and the summary will have a high redundancy.

R_N is known since it is simply the number of n-gram tokens in the summaries. We end up with the following approximation for the pairwise case:

$$\tilde{\theta}_R(a \cup b) = \theta_R(a) + \theta_R(b) - \tilde{\epsilon}(a \cup b), \text{ where} \tag{4.26}$$

$$\tilde{\epsilon}(a \cup b) = \frac{1}{R_N} \sum_{g \in a \cap b} \mathbb{1}[freq(g) \geq \alpha] \tag{4.27}$$

General approximation of ROUGE-N:

Now, we can approximate $\theta_R(S)$ for the general case defined by equation (4.23). We recall that $\theta_R(S)$ contains the sum of $\theta_R(s_i)$, the pairwise error terms $\epsilon^{(2)}(s_i \cap s_j)$, the error terms of three sentences $\epsilon^{(3)}$ and so on.

We can restrict ourselves to the individual sentences and the pairwise error corrections. Indeed, the intersection between more than two sentences is often empty, and accounting for it does not improve the accuracy significantly, but greatly increases the computational cost.

A formulation of ϵ in the case of two sentences has already been defined in (4.27). Thus, we have an approximation of the ROUGE-N function for any set of sentences that can be computed at inference time:

$$\tilde{\theta}_R(S) = \sum_{i=1}^{n} \theta_R(s_i) - \sum_{s_i, s_j \in S, s_i \neq s_j} \tilde{\epsilon}(s_i \cap s_j) \tag{4.28}$$

This is again the same form as equation (4.4). However, we have presented an approximation of the second term. Thus, only the sentence scores remain to be learned.

We empirically checked the validity of this formula. For this, we sampled 1000 sets of sentences from source documents of DUC-2003 (sets of 2 to 5 sentences) and compared their $\tilde{\theta}_R$ score to the real ROUGE-N. When the sentence scores are given, we observe a high Pearson's r correlation of 0.97, which validates $\tilde{\theta}_R$.

4.2.2 Optimizing the Approximation

$\tilde{\theta}_R$ from equation (4.28) defines a set function that scores a set of sentences. Now, the remaining task of summarization is to select the set S^* with maximal $\tilde{\theta}_R(S^*)$ under a length constraint (the component O).

Submodularity:
It has been shown that ROUGE-N is submodular (Lin and Bilmes, 2011) and one can verify that $\tilde{\theta}_R$ is submodular as well (the proof is given in Appendix C.4).

Therefore, we can apply the Greedy-M maximization algorithm to find a good set of sentences. This has the advantage of being straightforward and fast. However, it does not necessarily find the optimal solution.

Integer Linear Programming:
A common way to solve a discrete optimization problem is to formulate it as an ILP. It maximizes (or minimizes) a linear objective function with some linear constraints where the variables are integers.

We observe that it is possible to formulate the maximization of $\tilde{\theta}_R(S)$ as an ILP. Let x be the binary vector whose i-th entry indicates whether sentence i is in the summary or not, $\tilde{\theta}_R(s_i)$ the scores of sentences, and L the length constraint. We pre-compute the symmetric matrix $\tilde{P}$ where $\tilde{P}_{i,j} = \tilde{\epsilon}(s_i \cap s_j)$ and solve the following ILP:

$$\underset{\mathbf{x}}{\text{argmax}} \sum_{i=1}^{n} x_i * \tilde{\theta}_R(s_i) - d\frac{1}{R}\sum_{i \geq j} \alpha_{i,j} * \tilde{P}i,j \tag{4.29}$$

$$\textbf{such that,} \sum_{i=1}^{n} x_i * len(s_i) \leq L \tag{4.30}$$

$$\forall(i,j), \alpha_{i,j} - x_i \leq 0 \tag{4.31}$$

$$\forall(i,j), \alpha_{i,j} - x_j \leq 0 \tag{4.32}$$

$$\forall(i,j), x_i + x_j - \alpha_{i,j} \leq 1 \tag{4.33}$$

d is a damping factor that allows accounting for approximation errors. When $d = 0$, the problem becomes the maximization of "summary worthiness" under a length constraint, with "summary worthiness" being defined by $\sum \theta_R(s_i)$.

In practice, we used a value $d = 0.9$ because we observed that the learner tends to slightly overestimate the ROUGE-N scores of sentences. The mathematical deriva-

	DUC-02		DUC-03	
	R-1	R-2	R-1	R-2
($\tilde{\theta}_R$, Greedy)	.597	.414	.391	.148
($\tilde{\theta}_R$, Greedy-M)	.630	.484	.424	.160
($\tilde{\theta}_R$, ILP)	.644	.495	.447	.178
Upper-Bound	.648	.497	.452	.181

Table 4.2: Upper-bound of the proposed framework compared to the true extractive upper-bound.

tion implies $d = 1$, however, we can easily adjust for shifts in average scores of sentences from the estimation step by adjusting d.

Another option would be to post-process the scores after the estimation step to fix the average and let $d = 1$ in the optimization step. Indeed, if d moves away from 1, we move away from the mathematical framework of ROUGE-N maximization.

If $d \neq 0$, it seems intuitive to interpret the second term as minimizing the summary redundancy, which is in accordance with previous works (McDonald, 2007).

However, here, this term has a precise interpretation: it maximizes ROUGE-N scores up to the second order of precision, and the ROUGE-N formula itself already induces a notion of "summary worthiness" and redundancy, which we can empirically infer from data via supervised ML for sentence scoring and a simple heuristic for sentence intersections.

4.2.3 Approximation of ROUGE: a Problem Reduction

Now, we show that our proposed approximation is valid and actually reduces the problem of summarization (evaluated by ROUGE-N) to the problem of learning sentence scores. In section 4.4, we perform a thorough comparison of the summaries extracted by this method against baselines and other proposed approaches.

For the current experiments, we use the multi-document summarization datasets released as part of the DUC editions of 2002 and 2003 (DUC-02 and DUC-03).

Problem reduction:

Given that sentences receive scores close to their individual ROUGE-N, we presented a function $\tilde{\theta}_R$ that approximates the ROUGE-N of sets of these sentences. We also proposed an optimization to find the best scoring set under a length constraint.

To validate our framework empirically, we measured the quality of summaries extracted when $\tilde{\theta}_R$ is optimized with the real ROUGE-N scores of the individual sentences, calculated based on the reference summaries.

Then, $\tilde{\theta}_R$ can be optimized either by the Greedy algorithm ($\tilde{\theta}_R$, Greedy), by Greedy-M ($\tilde{\theta}_R$, Greedy-M), the greedy algorithm for submodular function maximization, or via the ILP proposed above ($\tilde{\theta}_R$, ILP). For reference, we report the real upper-bound for extractive summarization which is determined by solving a maximum coverage problem for n-grams from the reference summary, as it was done by Takamura and Okumura (2010).

Table 4.2 shows the results. We observe that ($\tilde{\theta}_R$, ILP) produces scores really close to the upper-bound. Thus, the problem of extractive summarization is reduced to the task of sentence scoring, because perfect scores for sentences induced near perfect extracted summaries when $\tilde{\theta}_R$ is optimized. Greedy-M seems less promising than the ILP because it greedily maximizes a function which the ILP can exactly maximize. However, Greedy-M offers a nice trade-off between performance and computation cost. The Greedy-M optimization is noticeably faster than the ILP (on average 3 times faster).

4.3 Using Human Judgments

Another especially important special case in the matrix of possible θ concerns the use of human judgments as supervision. In fact, this constitutes the ideal setup if the goal is to actually mimic humans.

However, this scenario poses several issues. First, collecting such data is difficult and expensive. As a result, the existing datasets contain few data points. Second, these datasets are limited in scope as they cover mostly average summaries. Indeed, human judgment datasets were created as a by-product of the manual evaluations performed during the shared-tasks of DUC and TAC. The scored summaries are the summaries extracted by the participating systems which are mostly average (compared to nowadays standards) and may exhibit some bias in their selection procedures.

In this section, we actually observe the impact of these problems with a small θ learning example and propose a simple regularization strategy to still leverage the existing human judgments.

Furthermore, we draw a simple connection between the task of learning the evaluation metric and learning the objective function θ. Indeed, the idea of learning from human judgments is a natural extension to the initial proposal of evaluating summary scoring functions by measuring their correlation with human judgments.

We already discussed in section 3.1 that this evaluation applies to both evaluation metrics and summarizers' internal objective functions. We briefly expand on this idea because it presents the nice conceptual advantage of putting the two main challenges of summarization (i.e., the evaluation and crafting of systems) into the same learning setup.

4.3.1 Learning the Metric – Learning the System

In section 3.1, we argued that evaluation metrics and summarizers' internal objective functions are both summary scoring functions which are expected to correlate highly with humans (i.e., the scores produced by the function should rank summaries in a similar way as humans do).

This calls for the supervised learning of summary scoring functions from human judgments. With h the implicit function explaining the human judgments, θ_{eval} an evaluation metric and θ_{sys} a scoring function for a summarizer:

$$\theta_{eval} \approx h \tag{4.34}$$

$$\theta_{sys} \approx h \tag{4.35}$$

While this idea is particularly intuitive for evaluation metrics (Conroy and Dang, 2008; Rankel et al., 2012), it now becomes obvious for a summarizers' scoring function as well. In fact, before, the evaluation metric was used as signal: $\theta_{sys} \approx \theta_{eval} \approx h$.

Interestingly, the two main challenges of summarization, the evaluation and the crafting of summarizers, are unified and framed within the same setup. The only difference concerns the features available. An evaluation metric θ_{eval} can access more information than a system's scoring function θ_{sys}.

In particular, a θ_{eval} can leverage reference summaries or previously manually constructed Pyramid sets. Let $\Phi_{sys}(S)$ be the features used by a summarization system and $\Phi_{eval}(S)$ the features used by an evaluation metric, then we have:

$$\forall S,\ \theta_{eval}(\Phi_{eval}(S)) \approx h(S) \tag{4.36}$$

$$\forall S,\ \theta_{sys}(\Phi_{sys}(S)) \approx h(S) \tag{4.37}$$

Where $\Phi_{sys}(S) \subset \Phi_{eval}(S)$, but $\Phi_{eval}(S) \not\subset \Phi_{sys}(S)$. Generally, since the learned evaluation metric has access to more information, the evaluation metric is expected to better approximate human judgments.

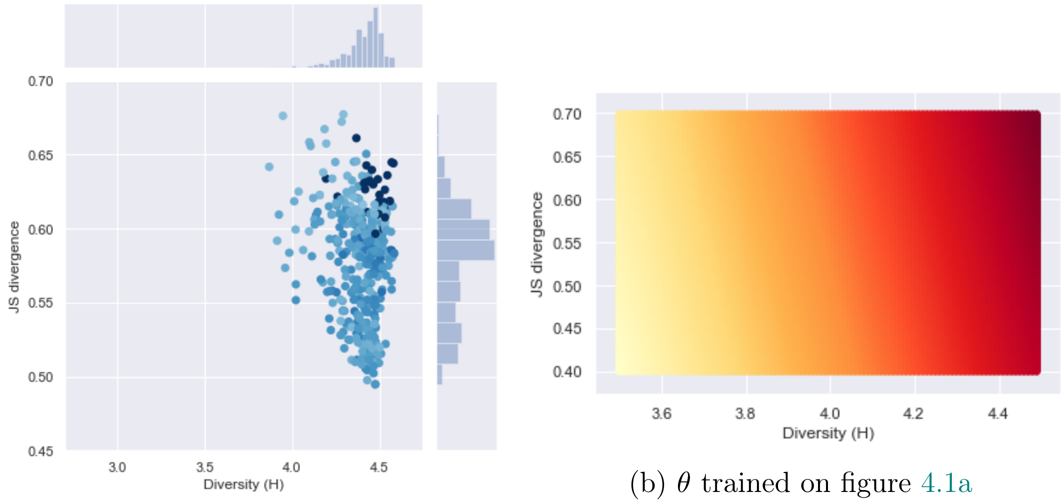
In our experiments, Φ_{eval} contains all the features between the summary and the inputs described in section 4.1 with the addition of several existing evaluation metrics described in section 2.2: ROUGE-1 (unigram overlap with reference summaries), ROUGE-2 (bigram overlap), ROUGE-L (longest sequence in common with reference summaries), ROUGE-WE (ROUGE-1 with soft matching based on word embeddings (Mikolov et al., 2013b)), JS-Eval-1 and JS-Eval-2 (JS divergence between n-gram distribution of the candidate and reference summaries):

Features for learning the evaluation metric: $\Phi_{eval} = \Phi \cup \{$ JS-1, JS-2, R-1, R-2, RL, R-WE$\}$

4.3.2 The Need for Regularization

We propose to learn the summary scoring function θ from a pool of manually annotated system summaries to ensure the extraction of summaries considered *good* by humans. With the same setup, we can train an evaluation metric that explicitly maximizes its correlation with human judgments.

Let h be the observed human judgments, which can be manual Pyramid (Nenkova et al., 2007) or overall Responsiveness (on a 0 to 5 LIKERT scale). We learn a



(a) Distribution of summaries from 5 topics from TAC-2008

(b) θ trained on figure 4.1a

Figure 4.1: Toy dataset of human judgments with diversity on the x-axis and JS divergence on the y-axis. Figure 4.1a represents the position of the annotated summaries available in the human judgments of TAC-2008, and figure 4.1b displays θ trained on summaries of TAC-2008 annotated with Pyramid.

function θ_ω with parameters ω approximating h based on a predefined feature set Φ ($\Phi(S) \in \mathbb{R}^d$ is the feature representation of a summary S). The parameters ω can be estimated by minimizing the loss defined by equation (4.2) with available human judgment datasets.

We notice that a summarizer's scoring function trained in this fashion tends to be ill-behaved under optimization. We first observe this on a toy dataset and then discuss a possible solution based on automatically generated noisy data. This approach is then tested in comparison to others in section 4.4.

Difficulties with available human judgments:
We first extracted a toy dataset from TAC-2008. We took the human annotated summaries from 5 randomly selected topics and represented them with two of the features described in section 4.1: JS divergence and Intra-summary diversity.

We kept only two features for this toy dataset in order to visualize the results in $2D$. Figure 4.1a shows the distribution of summaries from these 5 topics in the selected features space. As hypothesized, summaries available in the human judgments datasets only cover a small area of the feature space, leaving large spots without supervision.

In figure 4.1b, we illustrate the heatmap of θ trained with this toy dataset using a standard Support Vector Regression (SVR) [2] using Pyramid annotations as target scores. The red parts indicate areas of the feature space where θ assigns high scores. Therefore, the red areas are the areas of the feature space that an optimizer will explore and select summaries from.

[2] from scikit-learn: `http://scikit-learn.org/stable/index.html`

Already, we observe why learning only from human judgments poses problems. On this dataset, the learner has found the top right corner to be the best. Thus, an optimization procedure will wrongly extract summaries with high diversity and high JS divergence. While extracting summaries with high diversity makes sense, summaries with high JS divergences are expected to be bad because they will not resemble the sources. We discuss a potential solution in the next paragraphs.

Automatic data generation:
About 50 manually annotated summaries per topic are available in TAC-2008 and TAC-2009 shared tasks. They cover a small region of the feature space and the learned θ might be ill-behaved (high θ scores for bad summaries) pushing the optimizer to explore regions of the feature space unseen during training where θ wrongly assumes high scores.

To prevent this scenario, we rely on a large amount of noisy but automatic training data which provides a signal on a larger span of the feature space. Intuitively, it can be viewed as a kind of regularization.

We generate summaries distributed across the feature space. For each feature x, we sample a set of k summaries covering the range of possible values of x.

For sampling, we use the genetic algorithm introduced previously in section 3.2. The resulting population ranges from low to high values for the specific features. This process is repeated for each feature for both maximization and minimization as described by algorithm 5 and results in a dataset covering a large span of the feature space.

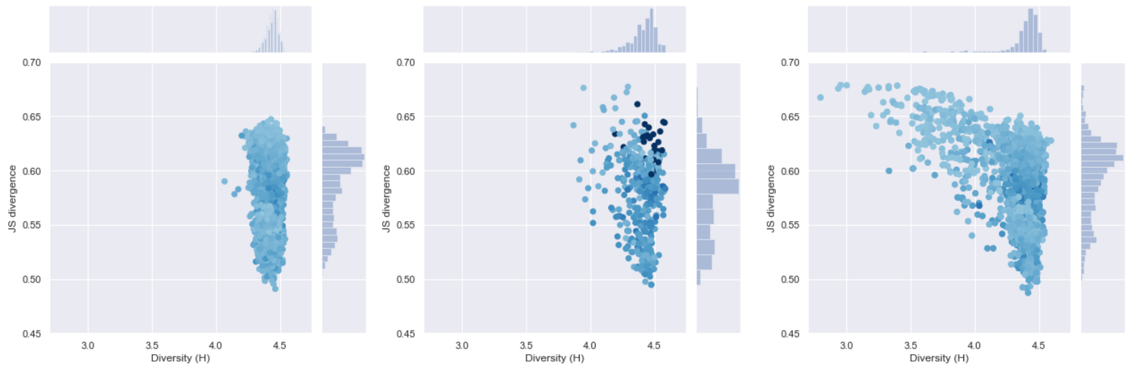
Each summary is then scored with ROUGE-N, a noisy surrogate for human judgments which can, nevertheless, discard poor regions of the feature space and regularize the behavior of the learned θ in previously unseen regions.

As an example, we project the generated summaries to the space of the 2 chosen features (JS divergence and Intra-summary diversity) for the 5 randomly selected topics of TAC-2008. Figure 4.2 represents the scatter-plot of summaries generated in parallel to summaries already available as part of the human evaluation in TAC-2008. For comparison, we also report the results of randomly sampling summaries.

Like the summaries already scored by humans, randomly selected summaries remain confined in a small region but seem uniformly sampled across this area. This reveals that some areas are more likely to be sampled than others. In contrast, the summaries generated with algorithm 5 cover a much wider spread of the feature space.

Even for data automatically generated with algorithm 5, there is still a remaining correlation between the features. Indeed, summaries with high JS divergence (intuitively *bad*) tend to also have low diversity (high redundancy).

The bottom left corner of the feature space (low JS and high-redundancy) is hard to sample because probably few summaries exist in this area. Such a summary should have a similar distribution of words as the sources but also be redundant, which is only possible if the sources are themselves highly redundant.



(a) Distribution of randomly sampled summaries

(b) Distribution of summaries from TAC-2008

(c) Distribution of our generated summaries

Figure 4.2: Comparison of dataset coverage of the feature space with diversity on the x-axis and JS divergence on the y-axis. Figure 4.2a depicts summaries randomly sampled and their position in the feature space, figure 4.2b represents the position of summaries available in the human judgments of TAC-2008 (same as figure 4.1a put here for comparison) and figure 4.2c the summaries generated according to algorithm 5.

Algorithm 5: Generate a Dataset of Diverse Summaries

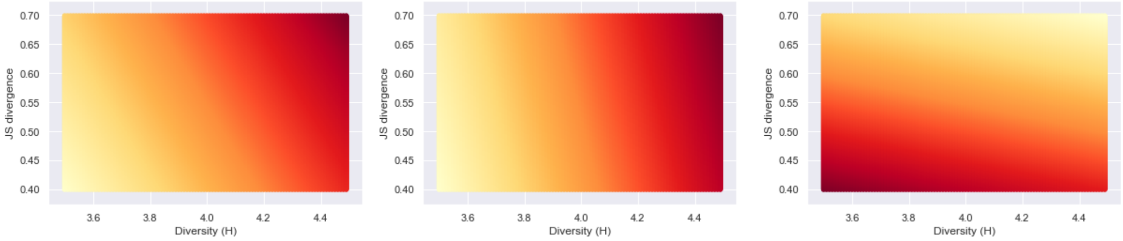
```
Input  : $D = \{s_1, \dots, s_n\}$: document as a set of sentences
         $L$: length constraint
         $k$: number of summaries to generate
         $F = \{f_1, \dots, f_e\}$: features considered
Output: $C = [S_1, \dots, S_k]$: a set of summaries
Function GenerateData($D, L, k, F$):
    $C := []$
    for $f \in F$ do
        $S := SampleSummaries(D, L, k, f)$
        $S := RemoveDuplicate(S)$
    end
```

Benefit of learning from several sources:
In figure 4.3, we illustrate the heatmap of θ trained with the respective datasets using a standard Support Vector Regression (SVR).[3]

As seen in the previous paragraph, training only on human judgments poses problems. In fact, the learner trained on automatically generated summaries has discovered that low JS divergence summaries should be extracted but failed to identify that high diversity is desired. It would itself be ill-behaved under optimization.

Thus, it seems natural to combine both learners. The simplest way is linear

[3] from scikit-learn: `http://scikit-learn.org/stable/index.html`



(a) θ trained on figure 4.2a (b) θ_h trained on figure 4.2b (c) θ_{R2} trained on figure 4.2c

Figure 4.3: Heatmaps associated with θ trained on various datasets. The more red, the higher score is given by θ to this area of the feature space. Figure 4.3a represents θ trained on the randomly generated summaries scored with ROUGE-2, figure 4.3b displays θ trained on summaries of TAC-2008 annotated with Pyramid and figure 4.3c depicts θ trained with summaries generated according to algorithm 5 scored with ROUGE-2.

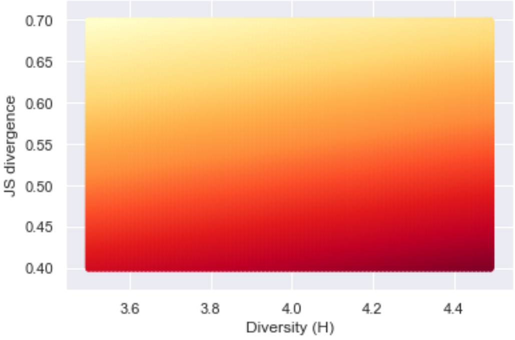


Figure 4.4: Heatmap of the combined summary scoring function: $\theta = \theta_h + \alpha \cdot \theta_{R2}$.

combination:

$$\theta(S) = \theta_h(S) + \alpha \cdot \theta_{R2}(S) \tag{4.38}$$

Where θ_h is the function trained only on summaries available in TAC-2008 with human annotations, θ_{R2} is the function trained on generated summaries scored with ROUGE-2 and α is the factor adjusting the strength of θ_{R2}. Therefore θ_{R2} acts as a regularizer for θ_h.

While ROUGE is just a poor proxy for human judgments, it provides a rough signal for unseen areas of the feature space. In general, we expect θ_{R2} to discard obviously bad regions of the feature space for which no supervision was available in the human judgment dataset.

We report the heatmap associated with the combined θ for $\alpha = 1$ in figure 4.4. Here, the resulting θ correctly identifies the bottom right corner as the area from which high-quality summaries should be extracted.

There exist many different techniques to combine different models together, such as multi-task learning, model averaging and various ensemble techniques. However, in our experiments, a simple linear combination gave good results. In this section, we presented an example on a restricted dataset and feature set. In section 4.4, we provide detailed and rigorous experiments comparing this approach against others.

4.4 Comparing Scoring Functions and Induced Summarizers

Encoding all the relevant aspects of content selection into one scoring function is a challenging problem. Instead of manually crafting this function, we investigated the automatic discovery of such functions from available data. This learning setup involves several design choices organized along 3 main axes of variation described in section 4.1.

In particular, we proposed general ways to learn both linear and unconstrained functions. We presented refinements for some important special cases: linear constraint with ROUGE as supervision in section 4.2 and involving human judgments in the supervision in section 4.3.

In this section, we compare the resulting summary scoring functions with the evaluation setup introduced in section 3.1. The summary scoring functions are evaluated by estimating their correlation with available human judgments. This concerns both learned evaluation metric and learned summarizers' internal scoring functions. The results confirm the superiority of the unconstrained case.

Finally, the learned summary scoring functions are combined with an appropriate optimization algorithm (i.e., an ILP for linear functions and a GPO otherwise). The extracted summaries are evaluated in a standard evaluation with both automatic metrics and manual evaluation.

These experiments also confirm the strength of the unconstrained case and answer **RQ2**. They also illustrate the problems discussed in section 4.3 when human judgments are used as supervision. The simple regularization strategy we proposed is able to mitigate these problems. Humans prefer the summaries extracted by the system which has been trained with human judgments.

We emphasize that the purpose of this chapter is not to claim state-of-the-art performance on summarization benchmarks but rather test our hypotheses using simple features set and basic Machine Learning algorithms.

4.4.1 Comparison of the Summary Scoring Functions

In order to compare the various summary scoring functions, we used two multi-document summarization datasets from the Text Analysis Conference (TAC) shared tasks: TAC-2008 and TAC-2009.[4]

Additionally, we used the recently created German dataset DBS-corpus (Benikova et al., 2016). It contains 10 topics consisting of 4 to 14 documents each. The reference summaries have variable sizes and are about 500 words long. For each topic,

[4] `http://tac.nist.gov/2009/Summarization/`, `http://tac.nist.gov/2008/Summarization/`

5 summaries were evaluated by trained human annotators but only for content selection with Pyramid. We experimented with this dataset because it contains heterogeneous sources (different text types) in German about the educational domain. This contrasts with the English homogeneous news documents from TAC-2008 and TAC-2009.

We trained summary scoring functions for each target data: ROUGE-2 (R-2), automatic Pyramid (PEAK), JS-Eval-2 (JS-2) and manually created Pyramid annotations (human judgments).[5]

When automatic metrics provide the supervision, we generated a dataset of 100 summaries per topic scored with the metric using the strategy presented in section 4.3 by algorithm 5.

For each supervision signal, two different θ's are trained: both with and without the linearity constraint.

For human annotations as supervision, we also trained an evaluation metric which is an unconstrained θ whose features access the reference summaries. The naming convention we used to distinguish these variations is available in table 4.3.

Combined function:
In order to regularize the summary scoring function trained with human judgments, we followed the procedure discussed in the previous section 4.3.

Thus, we trained 3 different scoring functions: θ_{pyr} with manual Pyramid annotations, θ_{resp} with Responsiveness annotations and θ_{R2} with our automatically generated data. We trained these models separately because the different annotations do not lie on the same scale.[6]

The final scoring function is a linear combination:

$$\theta_C(S) = \alpha_1 \cdot \theta_{pyr}(S) + \alpha_2 \cdot \theta_{resp}(S) + \alpha_3 \cdot \theta_{R2}(S) \tag{4.39}$$

This results in a combined summary scoring function. In the linearly constrained case, θ_{pyr}^{lin}, θ_{resp}^{lin} and θ_{R2}^{lin} are trained with the linear constraint and θ_C^{lin} is linear. To get the unconstrained θ_C, we used the unconstrained θ_{pyr}, θ_{resp} and θ_{R2}. We didn't automatically tune the different values of α_i but observed that $[1, 0.5, 0.5]$ works well in practice.

Training details:
The summary scoring functions are all trained with a linear model. For the unconstrained case, this still results in a non-linear function because it contains non-linear features. We used the same learner in order to compare the benefits of adding non-linear summary-level features. Indeed, the non-linear θ's differ only by the presence of non-linear features. Each θ is trained and evaluated in a leave-one-out cross-validation scenario.

[5] Due to the high runtime of PEAK, we perform its analysis only for one dataset: TAC-2009

[6] In DBS, no Responsiveness annotations are available. Therefore we used only 2 scoring functions: θ_{pyr} and θ_{R2}

Additionally, we evaluate $\tilde{\theta}_{R2}$ which exploits the mathematical structure of ROUGE-N as presented in section 4.2. This setup requires learning ROUGE scores for individual sentences. Thus, we train a linear model to approximate ROUGE-2 scores of sentences from the following feature set: bigram frequencies (Gillick and Favre, 2009), each of the 4 features from Edmundson (1969) and the LexRank scores of sentences (Erkan and Radev, 2004). These are all the linear features that can be defined at the sentence level (note: pairwise redundancy is a linear feature which is not defined for individual sentences).

	ROUGE-2	JS-Eval-2	PEAK	Pyr	Combination
w. linear constraint	θ^{lin}_{R2}	θ^{lin}_{JS2}	θ^{lin}_{Peak}	θ^{lin}_{Pyr}	θ^{lin}_{C}
w.o. linear constraint	θ_{R2}	θ_{JS2}	θ_{Peak}	θ_{Pyr}	θ_{C}
evaluation	-	-	-	θ^{Eval}_{Pyr}	-

Table 4.3: Notations used for various trained summary scoring functions. The functions from the first lign rely on the linear features described in section 4.1 : Φ_{lin}. On the second lign, the features used are both the linear and non linear ones: Φ. On the last line, reference summaries are available and the features used are all the previous ones and existing evaluation metrics: Φ_{eval}

Correction with human judgments:

To measure the performance of each summary scoring function, we replicate the evaluation detailed in section 3.1. Table 4.4 reports the correlation between each θ and manual Pyramid annotations for the three datasets (TAC-2008, TAC-2009 and DBS). It contains all the functions described in table 4.3. Additionally, for comparison, we reproduce the baseline from section 3.1. Also, the bottom part of the table compare θ^{Eval}_{Pyr} to existing evaluation metrics.

Performances vary across datasets as it was already observed in section 3.1. In particular, we tend to observe higher correlations in the DBS dataset. This can be explained by the smaller sample of annotated summaries (5 per topic) which are easier to distinguish than the ≈ 50 summaries per topic in TAC datasets.

DBS also contains longer summaries (500 words compared to 100 words for TAC), this explains the better performances of JS measures. Indeed, word frequency distributions are more representative for longer texts.

Generally, the learned summary scoring functions better approximate the human judgments than the baselines. It is particularly surprising to observe that learning with ROUGE-2 or JS-Eval-2 as supervision sometimes yields better approximations of Pyramid annotations than directly training with the annotations themselves.

We hypothesize that automatic metrics are more consistent than human annotations which makes the learning simpler (especially for a linear model). In contrast, Pyramid annotations may contain more inconsistencies which may confuse the learner. This motivates the use of more sophisticated techniques to deal with noise in future works (Simpson and Gurevych, 2018).

	TAC-2008			TAC-2009			DBS		
θ	r	τ	nDCG	r	τ	nDCG	r	τ	nDCG
θ_{ICSI}	.093	.169	.785	.088	.186	.772	.090	.058	.899
$\theta_{Edm.}$	.190	.155	.792	.385	.276	.804	.322	.301	.798
$\theta_{LexRank}$	.259	.152	**.826**	.390	.250	.816	.416	.343	.771
θ_{PI}	.131	.117	.779	.199	.203	.764	.170	.157	.827
θ_{JS}	.280	.230	.790	.262	.220	.760	.291	.144	.913
θ_{Div}	.288	.172	.814	.290	.155	.800	.132	.140	.853
$\tilde{\theta}_{R2}$	.362	.229	.809	.418	.278	**.830**	.553	.444	.937
θ^{lin}_{R2}	.280	.218	.801	.415	.276	.824	.325	.264	.916
θ^{lin}_{JS2}	.263	.208	.800	.403	.273	.823	.576	.426	.923
θ^{lin}_{Peak}	–	–	–	.414	.279	.826	–	–	–
θ^{lin}_{Pyr}	.280	.241	.795	.336	.258	.784	.600	.447	.924
θ^{lin}_{C}	.285	.240	.797	.370	.267	.795	.569	.424	.927
θ_{R2}	**.376**	**.281**	.823	**.428**	**.291**	**.830**	.683	.574	.957
θ_{JS2}	.320	.258	.801	.335	.246	.796	**.850**	**.755**	**.995**
θ_{Peak}	–	–	–	.426	.288	.829	–	–	–
θ_{Pyr}	.350	.267	.810	.376	.265	.793	.839	.708	.981
θ_{C}	.356	.270	.813	.400	.276	.806	.822	.679	.980
ROUGE-1	.748	.488	.961	.806	.547	.965	.702	.632	.984
ROUGE-2	.718	.490	.960	.803	.550	.963	.823	.784	.998
JS-Eval-1	.751	.495	.960	.820	.572	.965	**.971**	.754	.998
JS-Eval-2	.714	.474	.953	.775	.543	.952	.968	.766	.999
θ^{Eval}_{Pyr}	**.755**	**.501**	**.965**	**.843**	**.588**	**.976**	.908	**.820**	**.999**

Table 4.4: Correlation of θ functions with human judgments across various systems on TAC-2008 and TAC-2009.

Interestingly, learning with PEAK seems useful. This was expected as PEAK was specially designed to approximate human Pyramid annotations. This result further motivates the use of more semantically aware evaluation metrics for both the evaluation and training of summarization systems.

Additionally, $\tilde{\theta}_{R2}$ derived for approximating ROUGE-2 also yields relatively high correlation with Pyramid scores. In fact, it correlates better with humans than θ^{lin}_{R2} based on the general learning setup with linear constraint. The development made in section 4.2 is justified because a better correlation with humans is reached without losing linearity.

Most importantly, there is a large improvement when the constraint on θ is removed. This begins to answer **RQ2** by demonstrating that freeing θ from previously unjustified constraints makes it more capable of approximating human judgments.

Finally, we observe that the regularization strategy described by section 4.3 does not improve the correlation with human judgments. Indeed, the regularization term is useful during the optimization and not aimed at improving the correlation of the

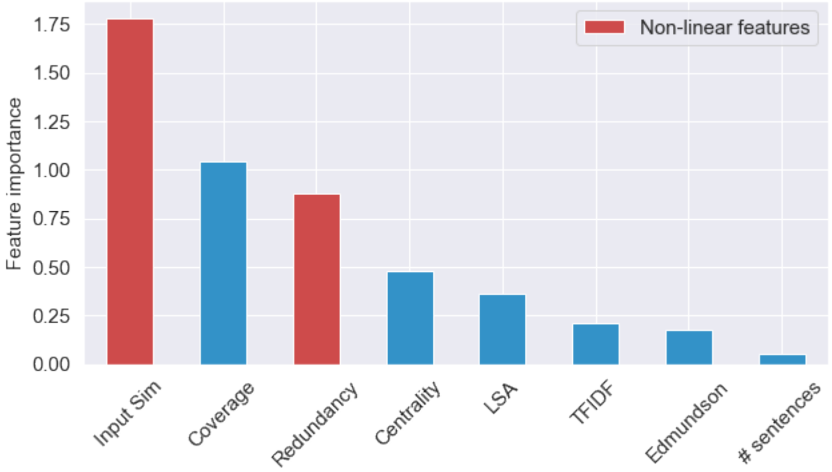


Figure 4.5: Relative importance of features in the unconstrained scenario trained with $JS-2$ as supervision.

summary scoring function with observed data. In the next section, we report the performances of extracted summaries and observe the benefits of this regularization.

The importance of non-linear features:
Since we used a linear regression for training, we can estimate the importance of a feature by the amplitude of its associated weight. For all unconstrained cases (θ_{R2}, θ_{JS2}, θ_{Peak}, θ_{Pyr}), the two non-linear features,JS divergence and Intra-summary diversity, were in the top 3 best features. This confirms the advantage of using a summary-level scoring function.

Figure 4.5 represents the relative importance of for an unconstrained learning scenario: θ_{JS2}. We observe the importance of the non-linear features as they are both in the top 3 features. Additionally, the bigram coverage is the strongest linear feature.

Analysis of the trained evaluation metric:
In this paragraph, we focus on the bottom part of table 4.4 which compares evaluation metrics. Unsurprisingly, summary scoring functions trained with features using only the source and the summary remain much worse than evaluation metrics (which use the references). θ^{Eval}_{Pyr} also gives improvements over other existing evaluation metrics.

As discussed above and demonstrated in the next section, blindly matching Pyramid annotations may result in ill-behaved summary scoring functions. Thus, we may question whether the learned θ^{Eval}_{Pyr} is actually a strong evaluation metric.

For summarizers' summary scoring functions, we can check whether they are well-behaved or not by optimizing them and evaluating the extracted summaries with automatic metrics and manual evaluation (as done in the next section).

Similarly, for an evaluation metric, we can extract summaries ranging from random to upper-bound and manually score them to check the correlation with humans on the whole scoring spectrum, i.e., what is considered upper-bound (resp. random)

	r	τ	nDCG
JS-Eval-1	.695	.620	.921
θ_{Pyr}^{Eval}	**.732**	**.643**	**.936**

Table 4.5: Correlation of automatic metrics with humans across the whole scoring spectrum of θ_{Pyr}^{Eval}.

by the metric is also considered as excellent (resp. bad) by humans.

To perform such a study, we collected summaries ranging from random to upper-bound using the data generation procedure described in section 4.3 with θ_{Pyr}^{Eval} as the fitness function for 15 topics of TAC-2009.

To select the summaries, for each topic we ranked them according to their θ_{Pyr}^{Eval} scores and, out of a population of 100, we picked 10 evenly spaced summaries (the first, the tenth and so on). Then, we asked two humans to score them following the guidelines used during DUC and TAC for assessing content selection.[7]

We observed an inter-annotator agreement of 0.74 Cohen's κ. The results of the evaluation are displayed in table 4.5 where θ_{Pyr}^{Eval} is compared to the best baseline metric: JS-Eval-1. The results indicate that the metric is reliable even outside of its training domain. It also outperforms JS-1 in this experiment.

Therefore, we release a user-friendly tool with the trained metric (called **S3**: Supervised Summary Scorer) for the community.[8]

We hypothesize that θ_{Pyr}^{Eval} (evaluation metric) is well-behaved even though θ_{Pyr} (summarizer's scoring function) is not because its feature space is more stable. Indeed, θ_{Pyr}^{Eval} is a combination of previously existing evaluation metrics which are themselves relatively well-behaved.

4.4.2 Comparison of the Summarization Systems

A summary scoring function alone does not make a summarizer, it also requires an optimization technique to actually select one summary for each topic. Thus, we performed an evaluation of the summaries extracted by the genetic optimizer for unconstrained θ's and by an ILP for the linear θ's.

To evaluate summaries, we report the ROUGE variant identified by Owczarzak et al. (2012) as strongly correlating with human evaluation methods: ROUGE-2 (R2). We also report JS-2, the Jensen-Shannon divergence between bigrams in the reference summaries and the candidate system summary (Lin et al., 2006). The last metric is S3, the combination of several existing metrics trained explicitly to maximize its correlation with human judgments (it is θ_{Pyr}^{Eval} from the previous section).

[7] We used LIKERT scale instead of Pyramid because they required less training for the annotators.

[8] `https://github.com/UKPLab/emnlp-ws-2017-s3` as part of the publication Peyrard et al. (2017)

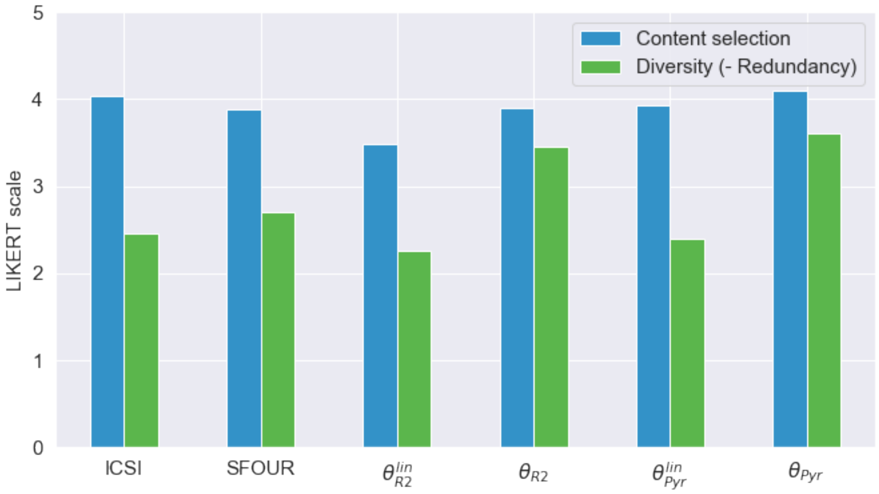


Figure 4.6: Manual evaluation of the main summarization systems for both content selection and diversity (minus redundancy) on a 5-point LIKERT scale.

Additionally, we set up a manual evaluation for the two English datasets. Two annotators were given the summaries of every system for 10 randomly selected topics of both TAC-2008 and TAC-2009. They annotated the content selection and redundancy level of summaries on a 5-points LIKERT scale. The inter-annotator agreement was 0.68 Cohen's kappa for content selection and 0.71 Cohen's kappa for redundancy.

Furthermore, for comparison, we include the scores from the baselines we previously considered: LexRank, ICSI, (KL, Greedy) and (JS, Greedy). Also, we report the scores of SFOUR (Sipos et al., 2012), a structured prediction approach that trains an end-to-end system with a large-margin method to optimize a convex relaxation of ROUGE. SFOUR optimizes a submodular function and acts as a supervised baseline that is not using linear functions. We use the publicly available implementation.[9]

Results:
The results of both manual and automatic evaluations of content selection are reported in table 4.6. The results of the manual evaluation for both content selection and redundancy are depicted in figure 4.6.

While θ's trained on human judgments have high correlations with humans, they behave badly under optimization. This effect is much less visible for θ's trained on ROUGE and JS-Eval because they have been trained on a dataset especially sampled to cover the whole feature space.

However, the regularization strategy introduced in section 4.3 mitigates these issues. Indeed, the combined θ_C performs better than each individual scoring function.

We also note that (θ_{R2}, Gen) performs on par with the other supervised baseline SFOUR but both are outperformed by exploiting human judgments. (θ_C, Gen) is

[9] http://www.cs.cornell.edu/~rs/sfour/

consistently better than baselines across datasets and metrics. In particular, humans tend to prefer the summaries extracted by (θ_C, Gen).

Manual inspection of summaries and figure 4.6 reveal that (θ_C, Gen) has lower redundancy than previous baselines thanks to summary-level features. In fact, figure 4.6 shows that the unconstrained θ, which uses non-linear features, are less redundant than constrained ones.

This was already hinted by the importance of the non-linear diversity feature in figure 4.5. As an example, we provide two generated summaries, one extracted from θ_C and another extracted from the best linear method: θ_C^{lin} (topic: D0831). We can see a sentence almost fully repeated in θ_C^{lin}.

> **Extracted from θ_C**
> *The FARC with about 17,000 fighters and a smaller leftist guerrilla group, the National Liberation Army (ELN) with some 6,000 members, have been locked in a 40-year civil war against the Colombian government. The FARC, the largest guerrilla force in Colombia, is accused of committing selective murders and slaughters, as well as terrorist acts in different parts of the South American country. President Alvaro Uribe said late Thursday he was ready to negotiate a prisoner swap with the country's largest rebel group, the Revolutionary Armed Forces of Colombia (FARC). The government turned down the rebel demand.*

> **Extracted from θ_C^{lin}**
> *The FARC kidnaps hundreds of people a year for ransom, and also holds dozens of so-called "exchangeables," including Colombian politicians, police officers, soldiers and the three U.S. contractors. President Alvaro Uribe said late Thursday he was ready to negotiate a prisoner swap with the country's largest rebel group, the Revolutionary Armed Forces of Colombia (FARC). The government turned down the rebel demand. The FARC kidnaps hundreds of people a year for ransom, and also holds dozens of so-called "exchangeables,"*

Furthermore, our ROUGE approximation $\tilde{\theta}_{R2}$ derived in section 4.2 is also a strong summarizer, better than its counter-part θ_{R2}^{lin} following the general *learning with linearity constraint* setup described in section 4.1. This further confirms the intuitions developed in section 4.2.

Finally, we can conclude that an unconstrained θ optimized with GPO is better than a constrained one optimized exactly with ILP. However, the gap between the two diminishes greatly after optimization. This resolves **RQ2**, as the unconstrained summary scoring function optimized with GPO is capable of producing high-quality summaries.

	TAC-2008				TAC-2009				DBS		
	R-2↑	JS-2↓	S3↑	H↑	R-2↑	JS-2↓	S3↑	H↑	R-2↑	JS-2↓	S3↑
LexRank	.078	.635	.336	3.74	.090	.625	.360	3.75	.105	.594	.354
(KL, Greedy)	.068	.644	.294	3.42	.061	.648	.288	3.21	.078	.620	.293
(JS, Gen)	.098	.618	.376	3.99	.101	.618	.370	3.89	.112	**.584**	.362
SFOUR	.101	.623	.372	3.88	.101	.622	.367	3.85	.114	.591	.357
ICSI	.101	.620	.377	4.03	.103	.619	.369	3.91	.115	.586	.361
($\tilde{\theta}_{R2}$, ILP)	.099	.622	.369	3.75	.100	.621	.368	3.63	.110	.592	.359
(θ^{lin}_{R2}, ILP)	.091	.629	.365	3.49	.097	.622	.366	3.43	.108	.595	.355
(θ^{lin}_{JS2}, ILP)	.090	.625	.363	3.48	.095	.624	.361	3.32	.102	.598	.351
(θ^{lin}_{Peak}, ILP)	–	–	–	–	.087	.619	.359	3.50	–	–	–
(θ^{lin}_{pyr}, ILP)	.084	.630	.348	3.44	.081	.636	.331	3.34	.075	.623	.305
(θ^{lin}_{C}, ILP)	.101	.622	.372	3.92	.102	.620	.361	3.85	.110	.592	.354
(θ_{R2}, Gen)	.100	.620	.375	3.89	**.104**	.618	.373	3.82	.116	.585	.363
(θ_{JS2},Gen)	.092	.621	.369	3.79	.098	.620	.368	3.52	.109	.593	.360
(θ_{Peak},Gen)	–	–	–	–	.092	.618	.367	3.88	–	–	–
(θ_{pyr}, Gen)	.096	.623	.369	3.65	.085	.631	.339	3.77	.078	.615	.312
(θ_{C}, Gen)	**.105**	**.615**	**.382**	**4.09**	**.104**	**.617**	**.376**	**4.03**	**.117**	**.584**	**.367**

Table 4.6: Comparison of systems across 3 datasets evaluated with ROUGE-2 recall, JS divergence on bigrams, S3 and Human annotations.

Chapter Summary

- When learning the summary scoring function from data, several design choices can be made along 3 dimensions: **supervision signal** (ROUGE, PEAK, human judgments, etc.), **learning constraints** (e.g., linearity) and **feature space**.

- With a fixed feature space, we compared the remaining dimension based on their ability to correlate with human judgments. The result confirms the **superiority of the unconstrained case**.

- When learning from the small and biased human judgments, θ is **not well-behaved** under optimization.

- This can be addressed by training a complementary scoring function on automatically generated data covering the whole feature space (with noisy signals like ROUGE scores).

- An evaluation metric is a summary scoring function and can also be trained to maximize its correlation with human judgments. This results in a new evaluation metric: **S3**.

- When ROUGE is used as supervision, an **almost perfect linear approximation** can be derived (provided sentence scores are available). This reduces the task of summarization (as evaluated by ROUGE) to the task of learning sentence scores.

- However, given recent advances in the automatic evaluation, we believe that empirical research in summarization should progressively move away from ROUGE towards more meaningful metrics for both evaluating and training systems like PEAK (Yang et al., 2016) or PyrEval (Gao et al., 2018).

Chapter 5

Theoretical Approach

In the previous chapters, we introduced the (θ, O) decomposition. The inherent question of summarization is that of finding strong summary scoring function θ, i.e., identifying an appropriate input representation $\mathbf{I}$ together with a simplification strategy $\mathbf{T}$ guided by a notion of *Importance*. In the previous chapter, we followed the path of empirically discovering summary scoring functions from observed data using tools from statistical analysis. Now, we explore a theoretical path to define a notion of *Importance* from an abstract framework rooted in information theory.

In general, automatic text summarization research has heavily focused on empirical developments, crafting summarization systems to perform well on standard datasets leaving the formal definition of *Importance* latent (Das and Martins, 2010; Nenkova and McKeown, 2012). This view entails collecting datasets, defining evaluation metrics and iteratively selecting the best-performing systems either via supervised learning or via repeated comparison of unsupervised systems (Yao et al., 2017). Our contributions of chapter 4 also follow this paradigm.

Such solely empirical studies may lack guidance as they are often not motivated by more general theoretical frameworks. While empirical approaches have facilitated the development of practical solutions, they mostly identify signals correlating with the vague human intuition of *Importance*. For instance, even nowadays, structural features like centrality and repetitions are still among the most used proxies for *Importance* (Yao et al., 2017). However, such features may just correlate with *Importance* in standard datasets. Unsurprisingly, simple adversarial attacks reveal their weaknesses (Zopf et al., 2016a).

We postulate that establishing formal theories of *Importance* will advance our understanding of the task and further improve summarization systems. One can draw inspiration from physics, arguably one of the most successful scientific developments, which fosters both empirical and theoretical works with strong interactions between the two. Empirical studies test hypotheses designed to falsify working theories, while theories are refined to account for new empirical results (Kuhn, 1970). In summarization, the lack of efforts to produce abstract theoretical frameworks might impede the progress.

A theory provides a frame of reference for interpreting observations, defining new concepts, generalizing knowledge and understanding complex logical relationships between variables. It forms an interrelated, coherent set of ideas and models which is refined upon new empirical observations (Kuhn, 1970). Hence, it is, by design, more internally consistent than common sense and intuition.

In symbiosis with empirical works, theories are particularly useful because they provide a common language to ground research. They describe how different approaches relate to each other, pinpoint dark zones and promising areas. Theoretically motivated experiments are always beneficial; even if the outcome of an experiment is unexpected, it is an opportunity to revise and improve the theory in a fundamental way (Kuhn, 1970).

In this chapter, we propose a possible definition of *Importance* within an abstract theoretical framework. This requires the notion of *information*, which has received a lot of attention since the work of Shannon (1948) in the context of communication theory. The subsequent theory produced powerful tools applied successfully in various domains like physics (Jaynes, 1957), economics (Maasoumi, 1993), evolutionary biology (Adami, 2012), or even the study of consciousness (Tononi et al., 2016). Information theory provides the means to rigorously discuss the abstract concept of information, which seems particularly well suited as an entry point for a theory of summarization.

However, information theory concentrates on uncertainty (entropy) about which message was chosen from a set of possible messages, ignoring the semantics of messages (Shannon, 1948). Yet, summarization is a lossy semantic compression depending on background knowledge.

In order to apply information theory to summarization, we assume the existence of a semantic representation of texts over a set of semantic units. This assumption is motivated by previous works on semantic information theory (Carnap and Bar-Hillel, 1953; Zhong, 2017). When applied to semantic symbols, the tools of information theory indirectly operate at the semantic level.

Within this framework, we define several concepts intuitively connected to summarization: *Redundancy*, *Relevance* and *Informativeness*. From these intuitive definitions, we can formulate properties required from a useful notion of *Importance*. In this view, *Importance* is not an intrinsic property of a semantic unit, it depends on which other units are present within some contextual boundaries: *Redundancy* in the context of the summary only, *Relevance* in the context of the source document(s) and *Informativeness* in the context of background knowledge and preconceptions of the user. *Importance* encompasses these three levels. Finally, whenever one compresses with loss of information one must make choices about what to discard. *Importance* is the measure that guides these choices. This chapter answers **RQ5**.

5.1 Semantic Units: Terminology and Assumptions

In the previous chapter, we proposed techniques to infer the summary scoring function from observations via machine learning techniques. In contrast, now, we aim to derive a theoretical framework governing the information selection step. However, any meaningful notion of *Importance* has to account for meaning and operate with semantic symbols. Thus, we have to make a choice of the input representation but this choice must be as general and as simple as possible in order to make the

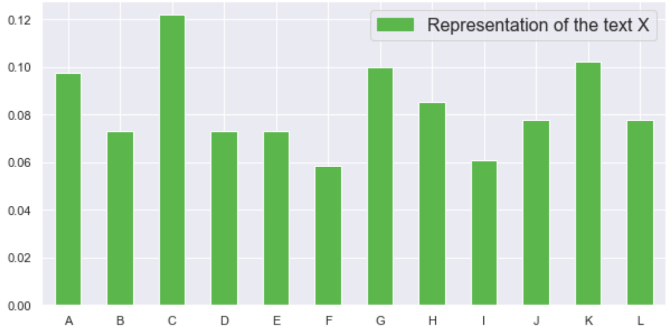


Figure 5.1: Possible representation of a text X over a set of semantic units: $\{A, B, \dots L\}$.

framework broadly applicable. Indeed, to be useful, it should encompass most of the practical textual representations.

With these requirements in mind, we present *semantic units*, which are general enough to account for most of the practical approaches to semantics and are well motivated by existing theoretical frameworks.

5.1.1 Introduction to Semantic Units

We call *semantic unit* an atomic piece of information which is independent of every other semantic unit. *Atomic* and *Independent* mean that knowing or observing one semantic unit w_i gives no information about the existence or the content of a different unit w_j (Zhong, 2017). Formally, this states that semantic units form a set Ω. Indeed, the elements of a set do not share any dependencies other than belonging to the same set.

A text X is considered as a semantic source emitting semantic units as envisioned by Weaver (1953) and recently discussed by Bao et al. (2011). Hence, we assume that X can be represented by a probability distribution $\mathbb{P}_X$ over the semantic units Ω. This is the input representation defining the step **I** of summarization, it is illustrated by figure 5.1.

Possible interpretations of the representation over semantic units: One can interpret $\mathbb{P}_X$ as the frequency distribution of semantic units in the text. Alternatively, $\mathbb{P}_X(\omega_i)$ can be seen as the (normalized) likelihood that a text X entails an atomic information ω_i (Carnap and Bar-Hillel, 1953). Another interpretation is to view $\mathbb{P}_X(\omega_i)$ as the normalized contribution (utility) of ω_i to the overall meaning of X (Zhong, 2017) For practical considerations, these interpretations are equivalent. Nevertheless, we discuss the motivations for semantic units in greater detail in the next section.

5.1.2 Motivation for Semantic Units

In this section, we examine the initial assumption that texts can be represented by distributions over semantic units. We observe that the existence of semantic units is well-motivated by prior work on semantic information theory and fits within several computational approaches to semantics.

Soon after information theory was introduced, Weaver (1953) mentioned that it only tackles what he called level A: the problem of accurately transmitting symbols of communication (the technical problem). He then discusses two other levels: (B) "How precisely do the transmitted symbols convey the desired meaning?" (the semantic problem) and (C) "How effectively does the received meaning affect conduct in the desired way?" (the effectiveness or pragmatic problem). Tackling problems (B) and/or (C) formally is the focus of *Semantic Information Theory*.

Theoretical motivations for semantic units:
Carnap and Bar-Hillel (1953) delivered one of the first and most prominent attempts at semantic information theory: in a simple formal language, the semantic information of a proposition p is determined by the number of propositions implied by p. Thus, semantic information has an underlying representation on a discrete and finite set of elements (akin to semantic units). Dunn (1976) discusses the atomicity of the state description of propositions described by Carnap and Bar-Hillel (1953) (i.e., the atomicity of semantic units). Recently, Bao et al. (2011) extended this work and argued that the size of Ω should be finite even for a language with infinite syntactic variability.

While previous works aimed at measuring the amount of semantic information, Zhong (2017) examined what is the essence of semantic information. He proposes a notion that encompasses syntax, semantics, and pragmatics. In his view, a piece of semantic information X is represented by an N-dimensional vector v, where an element v_i is a number between 0 and 1 representing the likelihood that X implies v_i. This directly supports the idea of a probability distribution over semantic units.

From philosophy, the *Theory of Strongly Semantic Information* produced by Floridi (2009) also implies the existence of semantic units (called information units in his work). Based on this work, Tsvetkov (2014) argued that the original theory of Shannon can operate at the semantic level by relying on semantic units.

In fact, by viewing summarization as semantic data compression, our framework proposes an operational approach to semantic information theory. A rigorous treatment of semantic compression based on semantic units and its connection to existing semantic and pragmatic information theories is out of the scope of this work. However, it is a promising direction for future work.

Semantic units within approaches to semantics:
Previous semantic information theories already justify the existence of semantic units in formal semantics, which treat natural languages as formal languages (Montague, 1970). In general, lexical semantics (Cruse, 1986) also postulates the existence of elementary constituents called minimal semantic constituents. For instance, with frame semantics (Fillmore, 1976), frames can directly act as semantic units.

Recently, distributional semantics approaches have received a lot of attention (Turian et al., 2010; Mikolov et al., 2013b). They are based on the distributional hypothesis (Harris, 1954) and the assumption that meaning can be encoded in a vector space (Turney and Pantel, 2010; Erk, 2010). These approaches search latent and independent components that correlate with the behavior of words (Gábor et al., 2017; Mikolov et al., 2013a).

While different approaches to semantics postulate different basic units and different properties for them, they have in common that *meaning arises from a set of independent and discrete units*. Thus, the semantic units assumption is general and has minimal commitment to the actual nature of semantics. This makes the framework compatible with most existing semantic representation approaches. Each approach specifies these units and can be plugged in the framework, e.g., frame semantics would define units as frames, topic models (Allahyari et al., 2017) would define units as topics and distributional representations would define units as dimensions of a vector space.

Other approximations to semantic units:
Characters, character n-grams, morphemes, words, n-grams, phrases, and sentences do not actually qualify as semantic units. Even though previous works which relied on information-theoretic motivations (Lin et al., 2006; Haghighi and Vanderwende, 2009; Louis and Nenkova, 2013) used some of them as support for probability distributions, they are neither atomic nor independent. It is mainly because they are surface forms whereas semantic units are abstract and operate at the semantic level.

However, they might serve as convenient approximations. Then, interesting research questions arise like *Which granularity offers a good approximation of semantic units?*, *Can we automatically learn good approximations?* In summarization, n-grams are known to be useful, but other granularities have rarely been considered together with information-theoretic tools.

5.2 Summarization Quantities

After introducing the notion of probability distributions over semantic units as a general representation of meaning, we focus on developing the framework useful for defining the notion of *Importance*.

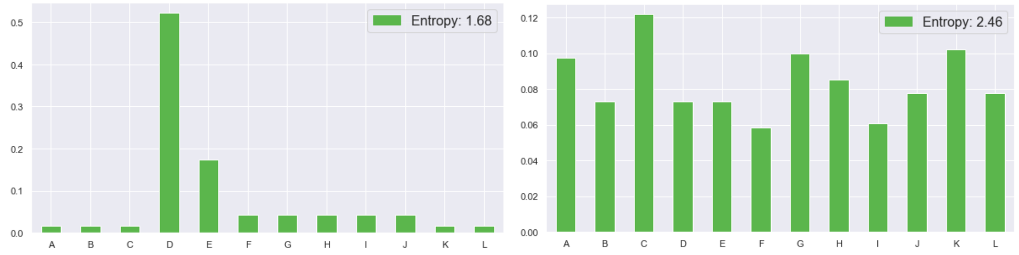
In the following section, we represent the source texts D and the candidate summary S by their respective distribution $\mathbb{P}_D$ and $\mathbb{P}_S$. Also, we sometimes note X instead of $\mathbb{P}_X$ when it is not ambiguous.

Then, we propose intuitive definitions for *Redundancy*, *Relevance* and *Informativeness*, by relying on the well-established field of information theory to provide sound theoretical motivations. Interestingly, by applying the same tools, we can also come up with a notion of *Potential Information* which connects background knowledge with the content of the sources.

5.2.1 Redundancy

Intuitively, a summary should contain a lot of information. In information-theoretic terms, the *amount of information* is measured by Shannon's entropy. For a summary S represented by $\mathbb{P}_S$, the entropy is given by:

$$H(S) = -\sum_{\omega_i} \mathbb{P}_S(\omega_i) \cdot \log(\mathbb{P}_S(\omega_i)) \tag{5.1}$$



(a) Summary represented by a low-entropy distribution

(b) Summary represented by a high-entropy distribution

Figure 5.2: Comparison between an example of a redundant summary (figure 5.2a) and an example of a non-redundant one (figure 5.2b).

$H(S)$ is maximized for a uniform probability distribution when every semantic unit is present only once in S: $\forall(i,j), \mathbb{P}_S(\omega_i) = \mathbb{P}_S(\omega_j)$. Therefore, we define *Redundancy*, our first quantity relevant to summarization, via entropy:

$$Red(S) = H_{max} - H(S) \tag{5.2}$$

Since $H_{max} = \log|\Omega|$ is a constant indepedent of S, we can simply write: $Red(S) = -H(S)$. A high-entropy distribution and a low-entropy one are compared in figure 5.2.

Intuitively, a summary S maximizes the information content if it displays many semantic units but once. Indeed, S should not be redundant but also contain as many semantic units as possible. The two following summaries: $S_1 = (a,b)$ and $S_2 = (a,b,c)$ are both non-redundant but S_2 is intuitively better because it contains more information. This is captured by entropy because $H(S_1) = \log(2) \leq \log(3) = H(S_2)$.

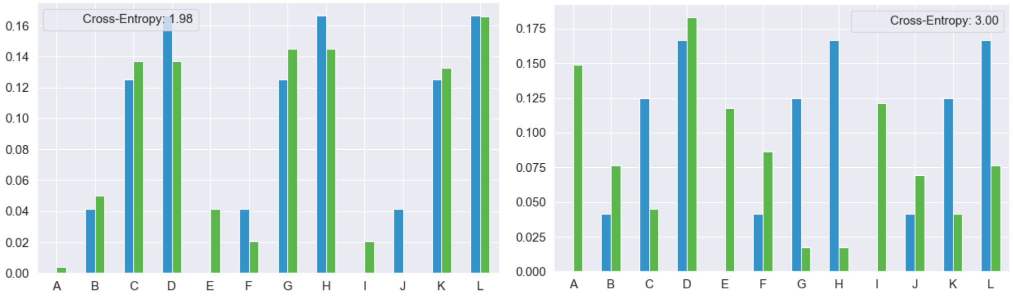
Redundancy in Previous Works:
By definition, entropy encompasses the notion of maximum coverage. Low redundancy via maximum coverage is the main idea behind the use of submodularity (Lin and Bilmes, 2011). Submodular functions are generalizations of coverage functions which can be optimized greedily with guarantees that the result would not be far from optimal (Krause and Golovin, 2014). Thus, they have been used extensively in summarization (Sipos et al., 2012; Yogatama et al., 2015). Otherwise, low redundancy is usually enforced during the extraction/generation procedures like MMR (Carbonell and Goldstein, 1998).

5.2.2 Relevance

Intuitively, observing a summary should reduce our uncertainty about the original text. A summary approximates the original sources and this approximation should incur a minimum loss of information. We call this property *Relevance*.

Within information theory, estimating *Relevance* boils down to comparing the distributions $\mathbb{P}_S$ and $\mathbb{P}_D$, which is done via the cross-entropy $Rel(S,D) = -CE(S,D)$:

$$Rel(S,D) = \sum_{\omega_i} \mathbb{P}_S(\omega_i) \cdot \log(\mathbb{P}_D(\omega_i)) \tag{5.3}$$



(a) Relevant summary (in green)

(b) Non-relevant summary (in green)

Figure 5.3: Compared to the input in blue, figure 5.3a presents an example of a relevant summary (with low cross-entropy); figure 5.3b depicts an example of non-relevant summary (with high cross-entropy).

The cross-entropy is interpreted as the average surprise of observing S while expecting T. A summary with a low expected surprise produces low uncertainty about what were the original sources. This is achieved by exhibiting a distribution of semantic units similar to the distribution of semantic units of the source documents: $\mathbb{P}_S \approx \mathbb{P}_D$. This is illustrated by figure 5.3.

Furthermore, we observe the following connection with *Redundancy*:

$$KL(S||D) = CE(S, D) - H(S) \tag{5.4}$$

$$-KL(S||D) = \mathit{Rel}(S, D) - \mathit{Red}(S) \tag{5.5}$$

Where the KL divergence is interpreted as the information loss incurred by using D as an approximation of S (i.e., the uncertainty about D arising from observing S instead of D). A summarizer that minimizes the KL divergence minimizes *Redundancy* while maximizing *Relevance*.

In fact, this is an instance of the Kullback Minimum Description Principle (MDI) (Kullback and Leibler, 1951), a generalization of the maximum entropy principle (Jaynes, 1957) with non-uniform prior: the summary minimizing the KL divergence is the least biased (i.e, least redundant or with highest entropy) summary matching D. In other words, this summary fits D while inducing a minimum amount of new information (any new information is necessarily biased since it does not arise from observations). The MDI principle and KL divergence unify *Redundancy* with *Relevance*.

In summarization, McDonald (2007) already observed creating high-quality summaries corresponds to maximizing some notion of relevance while minimizing redundancy. As it can be seen in section 2.3, many works followed this approach. Here, we propose a formal definition of these quantities which were extensively used intuitively in summarization.

Relevance in Previous Works:
Relevance is the most heavily studied aspect of summarization. In fact, by design, most unsupervised systems model *Relevance*. Usually, they used the idea of *topical frequency* where the most frequent topics from the sources must be extracted. Then,

different notions of *topics* and counting heuristics have been proposed. We briefly discuss these developments here.

Luhn (1958) introduced the simple but influential idea that sentences containing the most important words are most likely to embody the original document. Later, Nenkova et al. (2006) showed experimentally that humans tend to use words appearing frequently in the sources to produce their summaries. Then, Vanderwende et al. (2007) developed the system *SumBasic*, which scores each sentence by the average probability of its words.

The same ideas can be generalized to n-grams. A prominent example is the ICSI system (Gillick and Favre, 2009) which extracts frequent bigrams.

Different but similar words may refer to the same topic and should not be counted separately. This observation gave rise to a set of important techniques based on topic models (Allahyari et al., 2017). These approaches cover sentence clustering (McKeown et al., 1999; Radev et al., 2000; Zhang et al., 2015), lexical chains (Barzilay and Elhadad, 1999), Latent Semantic Analysis (Deerwester et al., 1990) or Latent Dirichlet Allocation (Blei et al., 2003) adapted to summarization (Hachey et al., 2006; Daumé III and Marcu, 2006; Wang et al., 2009; Davis et al., 2012).

Graph-based methods form another particularly powerful class of techniques to estimate the frequency of topics, e.g., via the notion of centrality (Mani and Bloedorn, 1997; Mihalcea and Tarau, 2004; Erkan and Radev, 2004).

Therefore, in existing approaches, the topics (i.e., atomic units) were words, n-grams, sentences or combinations of these. The general idea of preferring *frequent topics* based on various counting heuristics is formalized by cross-entropy. Indeed, requiring the summary to minimize the cross-entropy with the source documents implies that frequent topics in the sources should be extracted first.

An interesting line of work is based on the assumption that the best sentences are the ones that permit the best reconstruction of the input documents (He et al., 2012). It was refined by a stream of works using distributional similarities (Li et al., 2015; Liu et al., 2015b; Ma et al., 2016). There, the atomic units are the dimensions of the vector spaces. This information bottleneck idea is also neatly captured by the notion of cross-entropy which is a measure of information loss.

5.2.3 Informativeness

Relevance still ignores other potential sources of information such as previous knowledge or preconceptions about the task. We need to further extend the contextual boundary. Intuitively, a summary is informative if it induces, for a user, a great change in her knowledge about the world. Therefore, we introduce K as the background knowledge (or preconceptions about the task). K is also represented by a probability distribution $\mathbb{P}_K$ over semantic units Ω.

Formally, the amount of *new* information contained in a summary S is given by the cross-entropy $Inf(S, K) = CE(S, K)$:

$$Inf(S,K) = -\sum_{\omega_i} \mathbb{P}_S(\omega_i) \cdot \log(\mathbb{P}_K(\omega_i)) \tag{5.6}$$

For *Relevance*, the cross-entropy between S and D should be low. However, for *Informativeness*, the cross-entropy between S and K should be high because we measure the amount of new information induced by the summary in our knowledge.

Informativeness is also connected to entropy via KL divergence:

$$KL(S||K) = CE(S, K) - H(S) \tag{5.7}$$

$$KL(S||K) = Inf(S, K) - Red(S) \tag{5.8}$$

KL maximization unifies *Redundancy* and *Informativeness.*

Remark:
The background knowledge can be modeled by assigning a high probability to known semantic units. These probabilities correspond to the strengths of ω_i in the user's memory. A simple model could be the uniform distribution over known information where $\mathbb{P}_K(\omega_i)$ is $\frac{1}{n}$ if the user knows ω_i, and 0 otherwise.

However, K can control many variants of summarization tasks:

A personalized K_p models the preferences of a user by setting low probabilities to the semantic units of interest.

Similarly, a query Q can be encoded by setting low probability to semantic units related to Q.

Finally, there is a natural formulation of update summarization. Let U and D be two sets of documents. Update summarization consists in summarizing D given that the user has already seen U. This is modeled by setting $K = U$, considering U as previous knowledge.

Informativeness in Previous Works:
The modelling of *Informativeness* has received less attention by the summarization community. The problem of identifying stopwords originally faced by Luhn (1958) could be addressed by developments in the field of information retrieval using background corpora like TF·IDF (Sparck Jones, 1972). Based on the same intuition, Dunning (1993) outlined an alternative way of identifying highly descriptive words: the *log-likelihood ratio* test. Words identified with such techniques are known to be useful in news summarization (Harabagiu and Lacatusu, 2005).

Furthermore, Conroy et al. (2006) proposed to model background knowledge by a large random set of news articles. In update summarization, Delort and Alfonseca (2012) used Bayesian topic models to ensure the extraction of informative summaries. Louis (2014) investigated background knowledge for update summarization with Bayesian surprise. This is comparable to the combination of *Informativeness* and *Redundancy* in our framework when semantic units are n-grams. Thus, previous approaches to *Informativeness* generally craft an alternative background distribution to model the *a-priori* importance of units. Then, units from the document rare in the background are preferred, which is captured by maximizing the cross-entropy between the summary and K.

5.2.4 Potential Information

Relevance relates S and D, *Informativeness* relates S and K, but what connects D and K? Intuitively, only when K and D are different, we can extract a lot of new information from D.

With the same argument we laid out for *Informativeness*, we can define the amount of potential information as the average surprise of observing D while already knowing K. Again, this is given by the cross-entropy $PI_K(D) = CE(D, K)$:

$$PI(D, K) = -\sum_{\omega_i} \mathbb{P}_D(\omega_i) \cdot \log(\mathbb{P}_K(\omega_i)) \tag{5.9}$$

Previously, we stated that a summary should aim, using only information from D, to offer the maximum amount of new information with respect to K. $PI(D, K)$ can be understood as *Potential Information*, the maximum amount of new information that a summary can extract from D while knowing K. It can be viewed as the maximum *Informativeness* available in D.

5.3 Formal Definition of Importance

Summarization is a lossy semantic compression and whenever one compresses with loss of information one must make choices about what to discard. Informally, *Importance* is the measure that guides these choices. However, *Importance* is hard to define because of its inherent vagueness and subjectivity.

Instead, we establish simple properties required from a meaningful measure of *Importance* and search for quantities satisfying these specifications. To this end, we introduce the importance-encoding distribution unifying *Relevance* and *Informativeness*. Then, the Kullback MDI principle is employed to naturally incorporate *Redundancy* into a final summary scoring function.

5.3.1 Importance

Since *Importance* is a measure that guides which choices to make when discarding semantic units, we must devise a way to encode the relative importance of semantic units. Here, this means finding a probability distribution unifying D and K by encoding expectations about which semantic units should appear in a summary.

Informativeness requires a biased summary (w.r.t. K) and *Relevance* requires an unbiased summary (w.r.t. D). Thus, a summary should, by using only information available in D, produce what brings the most new information to a user with knowledge K. This could formalize a common intuition in summarization that units frequent in the source(s) but rare in the background are important.

Formally, let $d_i = \mathbb{P}_D(\omega_i)$, the probability of the unit ω_i in the source D. Similarly, we note $k_i = \mathbb{P}_K(\omega_i)$. We seek a function $f(d_i, k_i)$. From the previous insights, we formulate simple requirements that the function f should satisfy:

- Informativeness: $\forall i \neq j$, if $d_i = d_j$ and $k_i > k_j$ then $f(d_i, k_i) < f(d_j, k_j)$
- Relevance: $\forall i \neq j$, if $d_i > d_j$ and $k_i = k_j$ then $f(d_i, k_i) > f(d_j, k_j)$
- Additivity: $I(f(d_i, k_i)) \equiv \alpha I(d_i) + \beta I(k_i)$ (I is the information measure from Shannon's theory (Shannon, 1948))
- Normalization: $\sum_i f(d_i, k_i) = 1$

The first requirement states that, for two semantic units equally represented in the source, we prefer the more informative one. The second requirement is an analogous statement for *Relevance*. The third requirement is a consistency constraint to preserve additivity of the information measures, as initially proposed by Shannon (1948). The fourth requirement ensures that f is a valid distribution.

Theorem 2. *The functions satisfying the previous requirements are of the form:*

$$\mathbb{P}_{\frac{D}{K}}(\omega_i) = \frac{1}{C} \cdot \frac{d_i^\alpha}{k_i^\beta} \tag{5.10}$$

$$C = \sum_i \frac{d_i^\alpha}{k_i^\beta} \text{ , } \alpha, \beta \in \mathbb{R}^+ \tag{5.11}$$

C is the normalizing constant. The parameters α and β represent the strength given to *Relevance* and *Informativeness* respectively (this is made clearer by equation (5.15)). The proof is provided in Appendix C.5.

Summary scoring function:
By construction, a candidate summary should approximate $\mathbb{P}_{\frac{D}{K}}$, which encodes relative expectations about which semantic units should appear in a summary. Furthermore, the summary should be non-redundant (i.e., high entropy). These two requirements are unified by the Kullback MDI principle: The least biased summary S^* that best approximates the distribution $\mathbb{P}_{\frac{D}{K}}$ is the solution of:

$$S^* = \operatorname*{argmax}_S \theta_I = \operatorname*{argmax}_S -KL(S || \mathbb{P}_{\frac{D}{K}}) \tag{5.12}$$

Thus, we note θ_I as the quantity that measures the quality of a summary:

$$\theta_I(S, D, K) = -KL(\mathbb{P}_S, || \mathbb{P}_{\frac{D}{K}}) \tag{5.13}$$

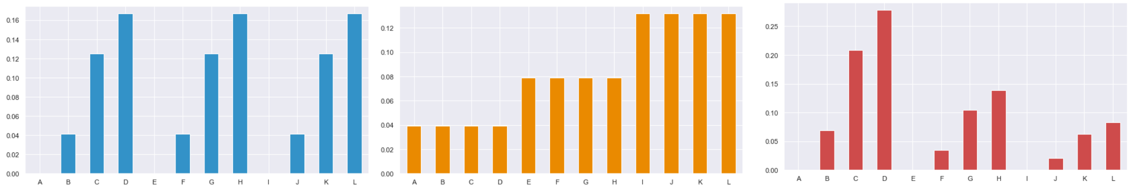
Remark: We note that a summary maximizing the *Relevance* also follows the Kullback MDI and is the least biased summary fiting the expectations encoded by D. Similary, a summary maximizing the *Informativeness* is an instance of the Kullback MDI where the expectations are given by $\frac{1}{K}$. Trivially, a summary minimizing its redundancy (maximizing its entropy) is an instance of the maximum entropy principle.

Interpretation of $\mathbb{P}_{\frac{D}{K}}$:
$\mathbb{P}_{\frac{D}{K}}$ can be viewed as an *importance-encoding distribution* because it encodes the relative importance of semantic units and gives an overall target for the summary.

For example, if a semantic unit ω_i is prominent in D ($\mathbb{P}_D(\omega_i)$ is high) and not known in K ($\mathbb{P}_D(\omega_i)$ is low), then $\mathbb{P}_{\frac{D}{K}}(\omega_i)$ is very high, which means very desired in the summary. Indeed, choosing this unit will fill the gap in the knowledge K while matching the sources.

Figure 5.4 illustrates how this distribution behaves with respect to D and K (with $\alpha = \beta = 1$).

Summarizability:
The target distribution $\mathbb{P}_{\frac{D}{K}}$ may exhibit different properties. For example, it might



(a) ditribution $\mathbb{P}_D$ (b) distribution $\mathbb{P}_K$ (c) distribution $\mathbb{P}_{\frac{D}{K}}$

Figure 5.4: Figure 5.4a represents an example distribution of sources, figure 5.4b an example distribution of background knowledge and figure 5.4c is the resulting target distribution that summaries should approximate.

be clear which semantic units should be extracted (i.e., a spiky probability distribution) or it might be unclear (i.e., many units have more or less the same importance score). This can be quantified by the entropy of the importance-encoding distribution:

$$H_{\frac{D}{K}} = H(\mathbb{P}_{\frac{D}{K}}) \tag{5.14}$$

Intuitively, this quantifies the number of possibly good summaries. If $H_{\frac{D}{K}}$ is low then $\mathbb{P}_{\frac{D}{S}}$ is spiky and there is little uncertainty about which semantic units to extract (few possible *good* summaries). Conversely, if the entropy is high, many equivalently *good* summaries are possible.

Interpretation of θ_I:
Maximizing θ_I not only encourages the selection of high scoring semantic units, it also indicates which choices and trade-offs are more beneficial. θ_I covers the overall selection of several semantic units together, it is not restricted to independently choosing individual ones. To better understand θ_I, we remark that it can be expressed in terms of the previously defined quantities:

$$\begin{align} \theta_I(S, D, K) &= H(S) - \alpha CE(S, D) + \beta CE(S, K) + \log C \tag{5.15} \\ \theta_I(S, D, K) &\equiv -Red(S) + \alpha Rel(S, D) + \beta Inf(S, K) \tag{5.16} \end{align}$$

Equality holds up to a constant term $\log C$ independent from S. From now on, we omit the constant term $\log C$ as it does not depend on S. Thus, maximizing θ_I is equivalent to maximizing *Relevance* and *Informativeness* while minimizing *Redundancy*.

Finally, we can say that $H(S)$, $CE(S, D)$ and $CE(S, K)$ are the three independent components of *Importance*.

It is worth noting that each previously defined quantity: *Red*, *Rel* and Inf are measured in bits (using base 2 for the logarithm). Shannon initially axiomatized that information quantities should be additive (Shannon, 1948) and therefore θ_I arising as the sum of other information quantities is unsurprising. Moreover, we ensured additivity with the third requirement of $\mathbb{P}_{\frac{D}{K}}$.

5.3.2 Examples

To further illustrate the workings of the formula, we provide examples of experiments done with a simplistic choice for semantic units: words. These show that even with simplistic assumptions, θ_I and $\mathbb{P}_{\frac{D}{K}}$ are meaningful and interpretable quantities which correlate well with human judgments.

Setup and assumptions:
We experiment with TAC-2008 and TAC-2009 for two different summarization tasks: generic and update multi-document summarization.

To keep the experiments simple and focused on illustrating the formulas, we make several simplistic assumptions. First, we choose words as semantic units and therefore texts are represented as frequency distributions over words. While it is limiting, this remains a simple approximation letting us observe the quantities in action.

K, α and β are the parameters of the theory and their choice is subject to investigation. Here, we made simple choices: for update summarization, K is the frequency distribution over words in the background documents (A). For generic summarization, K is the uniform probability distribution over all words from the source documents. Furthermore, we use $\alpha = \beta = 1$.

Correlation with humans:
First, we measure how well the different quantities correlate with human judgments. We compute the score of each system summary according to each quantity defined in the previous section: *Red*, *Rel*, Inf, $\theta_I(S, D, K)$. We then compute the correlations between these scores and the manual Pyramid scores, as described in section 3.1.

We measure the correlation with kendall's τ, a rank correlation metric which compares the orders induced by both scored lists. We report results for both generic and update summarization over the two datasets TAC-2008 and TAC-2009 in table 5.1. Thus, in the generic case, the results are comparable with table 3.1 and we include the baselines already considered there. Furthermore, we report two baselines from Louis (2014) to model the informativeness: KL_{back} which measures the divergence between the distribution of the summary and the background knowledge K. JS_{back} does the same with JS divergence instead of KL.

In general, the modelling of *Relevance* (based only on the sources) correlate more with human judgments than other quantities. This justifies why most summarization approaches focused on this aspect. In general, metrics accounting for background knowledge work better in the update scenario. It is not suprising as the background knowledge K is more meaningful in this case (using the previous document set).

In general, we observe that JS divergence gives slightly better results than KL. Even though KL is more theoretically appealing, JS is smoother and usually works better in practice when distributions have different supports (Louis and Nenkova, 2013).

Finally, θ_I significantly[1] outperforms all baselines in both the generic and update case. *Red*, *Rel* and *Inf* are not particularly strong on their own, but combined

[1] at 0.01 with significance testing done with t-test to compare two means

together they yield a strong summary scoring function θ_I. Indeed, each quantity models only one aspect of content selection, they are the three independent components of *Importance.*

We need to be careful when interpreting these results because we made several strong assumptions: by choosing n-grams as semantic units and by choosing K rather arbitrarily. Nevertheless, these are promising results. Should we craft better text representations and come-up with more suitable K, we would expect even higher correlations. We already observe better correlation for θ_I in the update summarization scenario, which comes from a more natural choice of K. In the generic case, the uninformative uniform distribution is a weaker approximation of background knowledge.

	Generic	Update
ICSI	.178	.139
Edm.	.215	.205
LexRank	.201	.164
TFIDF	.227	.182
KL	.204	.176
JS	.225	.189
KL_{back}	.110	.167
JS_{back}	.066	.187
Red	.098	.096
Rel	.212	.192
Inf	.091	.086
θ_I	**.294**	**.211**

Table 5.1: Correlation of various information-theoretic quantities with human judgments measured by Kendall's τ on generic and update summarization.

Comparison with reference summaries:
Intuitively, the distribution $\mathbb{P}_{\frac{D}{K}}$ should be similar to the probability distribution $\mathbb{P}_R$ of the human-written reference summaries.

To verify this, we scored the system summaries and the reference summaries with θ_I and checked whether there is a significant difference between the two lists.[2] We found that θ_I scores reference summaries significantly higher than system summaries. The $p-$value, for the generic case, is 9.2e−6 and 1.1e−3 for the update case. Both are much smaller than the 1e−2 significance level. Therefore, θ_I is capable of distinguishing systems summaries from human written ones.

Example on a topic:
As an example, for one selected topic of TAC-2008 update track, we computed the $\mathbb{P}_{\frac{D}{K}}$ and compare it to the distribution of the 4 reference summaries.

[2] with standard t-test for comparing two related means.

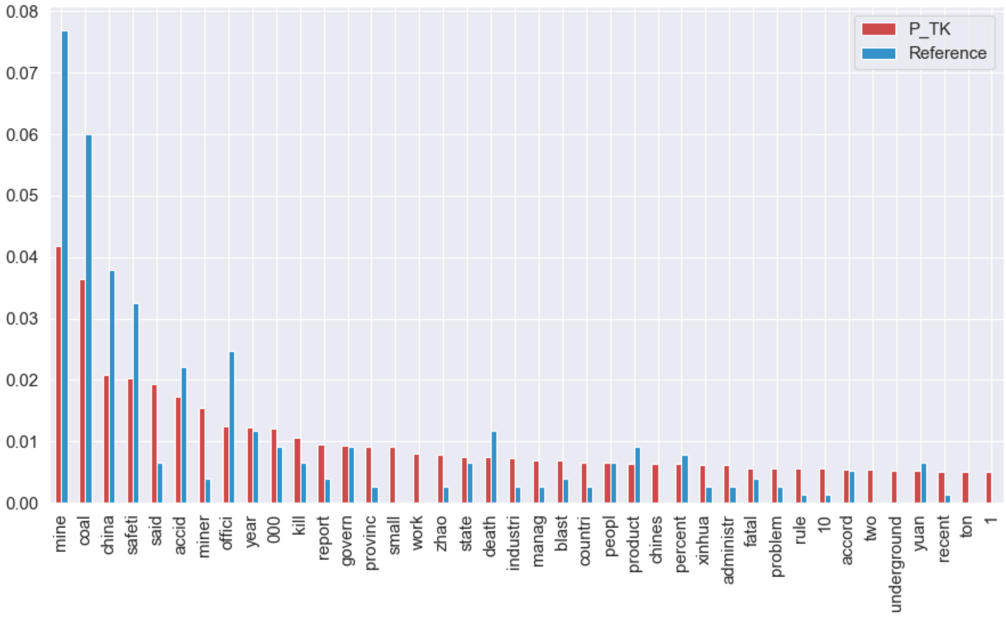


Figure 5.5: Example of $\mathbb{P}_{\frac{D}{K}}$ in comparison to the word distribution of reference summaries for one topic of TAC-2008 (D0803).

We report the two distributions together in figure 5.5. For visibility, only the top 50 words according to $\mathbb{P}_{\frac{D}{K}}$ are considered. However, we observe a good match between the distribution of the reference summaries and the *ideal* distribution as defined by $\mathbb{P}_{\frac{D}{K}}$.

Furthermore, the most desired words according to $\mathbb{P}_{\frac{D}{K}}$ make sense. This can be seen by looking at one of the human-written reference summary of this topic:

> **Reference summary for topic D0803**
> *China sacrificed coal mine safety in its massive demand for energy. Gas explosions, flooding, fires, and cave-ins cause most accidents. The mining industry is riddled with corruption from mining officials to owners. Officials are often illegally invested in mines and ignore safety procedures for production. South Africa recently provided China with information on mining safety and technology during a conference. China is beginning enforcement of safety regulations. Over 12,000 mines have been ordered to suspend operations and 4,000 others ordered closed. This year 4,228 miners were killed in 2,337 coal mine accidents. China's mines are the most dangerous worldwide.*

Chapter Summary

- The notion of **semantic units** is introduced as a general way of encoding meaning. It is supported by previous theoretical works and encompasses most of practical approaches to semantics.

- Based on semantic units and information theoretic tools, several notions are formally defined: **Redundancy**, **Relevance** and **Informativeness**.

- **Importance** can be interpreted as the quantity unifying these concepts. **Importance** is not an intrinsic property of a semantic unit, it depends on which other units are present within some contextual boundaries: **Redundancy** in the context of the summary only, **Relevance** in the context of the source documents and **Informativeness** in the context of background knowledge.

- Whenever one compresses with loss of information, one must make choices about what to discard. **Importance** is the measure that guides these choices.

- The notion of **Importance** induces a summary scoring function which, under simplifying assumptions, is shown to correlate with human judgments and is capable of discriminating human summaries from system summaries.

Chapter 6

Limitations of Human Judgment Datasets

In the previous chapters, we emphasized the role played by humans in defining what are *good* and *bad* summaries. Here, we discuss some limitations of existing human judgment datasets. More specifically, the experiments of this chapter motivate the collection of manual annotations for high-scoring summaries to further advance summarization.

Human judgments play an important role in evaluation. Indeed, evaluation metrics are compared based on their ability to correlate with humans (Lin and Hovy, 2003). Then, the selected metrics heavily influence summarization research by guiding progress (Lloret et al., 2018). These metrics also provide supervision for training summarization systems (see section 2.2). Human judgments can even play a more direct role in the training of systems. For example, in section 4.3, we demonstrated how a supervised system can be trained directly with human scores.

Despite their central role, few human judgment datasets have been created. The existing ones were collected as by-products of the manual evaluations performed during the shared-tasks of DUC and TAC. As an example of limitation of these datasets, section 4.3 already observed the lack of diversity in the annotated summaries.

We first make another observation: the annotated summaries are mostly average compared to nowadays standards. Indeed, the best systems submitted at the time of these shared-tasks have typically been used as baselines for subsequent works. This is illustrated by figure 6.1, which compares the score distribution of summaries in the human judgment datasets with the score distribution of modern summarization systems.[1] The distribution of scores on which evaluation metrics are tested (blue zone) differs from the one in which they now operate (red zone). Thus, there is no guarantee that evaluation metrics behave according to human judgments in the high-scoring range. Yet, whether the supervision comes from human judgments directly or from evaluation metrics, summarization systems aim to generate high-scoring summaries. As observed previously by Radev et al. (2003), the high-scoring regime is of great importance. This is the particular range we study here.

In this chapter, we demonstrate that current evaluation metrics disagree with each other in the high-scoring range. Even though they correlate well in the average

[1] scores for modern systems are obtained from the various systems presented in this thesis and other systems mentioned by Hong et al. (2014).

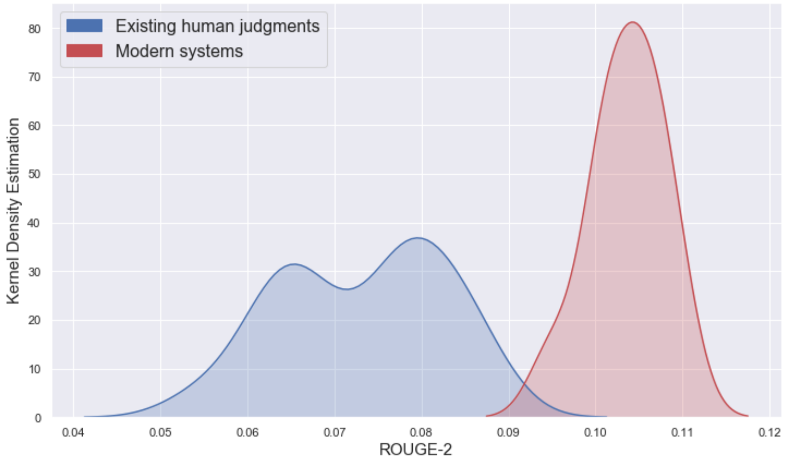


Figure 6.1: The blue distribution represents the score distribution of summaries available in the human judgment datasets of TAC-2008 and TAC-2009. The red distribution is the score distribution of summaries generated by mordern systems.

range, they present low and even negative correlations for high-scoring summaries. This is highly problematic because current metrics cannot be distinguished based solely on analysis on available human judgments. Indeed, in practice, they behave similarly on these datasets. Nevertheless, they will promote vastly different summaries and systems.

It is a common good practice to report several metrics together. In the average scoring range, where metrics generally agree, this creates a robust measure of progress. The specificities of each metric are averaged-out putting the focus on the general trend. However, if the metrics do not correlate, they do not share a common trend. In fact, we show that, in the high-scoring range, it becomes almost impossible to find summaries which provide improvements according to all metrics. Indeed, with disagreeing metrics, it becomes difficult to find summaries for which metrics agree.

Since metrics disagree strongly in the high-scoring regime, at least some of them are deviating largely from humans. Should we collect more human judgments, we could identify the best ones and develop new and better evaluation methodologies. Thus, our work motivates the collection of human judgments for high-scoring summaries.

Furthermore, the analysis we provide here could be duplicated for related fields like *Machine Translation* or *Natural Language Generation*.

6.1 Data Collection

In this work, we study the following metrics:

- **ROUGE-2** (**R-2**): measures the bigram overlap between the candidate summary and the pool of reference summaries (Lin, 2004b).

- **ROUGE-L** (**R-L**): a variant of ROUGE which measures the size of the longest common sub-sequence between candidate and reference summaries.
- **ROUGE-WE** (**R-WE**): instead of hard lexical matching of bigrams, **R-WE** uses soft matching based on the cosine similarity of word embeddings (Ng and Abrecht, 2015).
- **JS divergence** (**JS-2**): uses Jensen-Shannon divergence between bigram distributions of references and candidate summaries (Lin et al., 2006).
- **S3**: the metric we introduced and evaluated in chapter 4. It is trained explicitly to maximize its correlation with manual Pyramid annotations.

We chose these metrics because they correlate well with available human judgments. We didn't include PEAK (Yang et al., 2016) because it remains slow to compute and difficult to use which decreases its chances of being used as a standard metric. However, for future analysis, it might be interesting to consider the more recent version of automatic Pyramid: PyrEval (Gao et al., 2018).

Once an evaluation metric becomes standard, it is optimized, either directly by supervised methods or indirectly via repeated comparisons of unsupervised systems. To mimic this procedure, we optimized each metric using the genetic algorithm for summarization described in section 3.2. The metric m is used as the fitness function. The resulting population is a set of summaries ranging from random to upper-bound according to m. For both TAC-2008 and TAC-2009, we used a population of 400 summaries per topic (per metric). The final dataset contains $160,523$ summaries for an average of $1,763$ summaries per topic (less than $5 * 400$ due to removed duplicates).

The summary generation procedure is described by algorithm 6. The function $Score(S, M)$ takes a list S of summaries and a list M of evaluation metrics and outputs a list where each summary has been scored by each evaluation metric in M. The $SampleSummaries$ function is the genetic algorithm.

Algorithm 6: Generate a Dataset of Scored Summaries

```
Input : $D = \{s_1, \dots, s_n\}$: document as a set of sentences
        $L$: length constraint
        $k$: number of summaries to generate
        $M = \{m_1, \dots, m_e\}$: evaluation metrics considered
Output: $C = [S_1, \dots, S_k]$: a set of scored summaries
Function $GenerateData(D, L, k, M)$:
    $C := []$
    for $m \in M$ do
        $S := SampleSummaries(D, L, k, m)$
        $S := RemoveDuplicate(S)$
        $C \leftarrow Score(S, M)$
    end
```

6.2 Correlation Analysis

The dataset described above does not include gold scores given by humans, it cannot be used to decide which candidate metric is better. However, we can compare the behavior of candidate evaluation metrics, especially in the scoring ranges of interest. We first observe that evaluation metrics correlate with each other when the whole scoring range is considered. Then, we observe low and even negative correlations for high-scoring summaries.

This is related to the *Simpson paradox*, where different conclusions are drawn depending on which slice of the population is considered (Wagner, 1982). In fact, it is simple to distinguish obviously bad from obviously good summaries, which results in superficially high correlations when the whole scoring range is considered.

6.2.1 When Metrics Correlate with Each Others

We first compute the pairwise correlation between metrics using the existing human judgments (TAC-2008 and TAC-2009). Figure 6.2 is the scatter matrix plot describing the correlations between pairs of candidate metrics. The number and the cell background color indicate the Kendall's τ between the two metrics. This measures the proportion of pairs of summaries ranked in the same order by both metrics. Diagonal cells represent the score distribution of summaries for the given metric.

The correlations between pairs of metrics are high: metrics behave similarly in the average-scoring range and cannot be easily distinguished based only on analysis of standard human judgment datasets.

We then replicated the same pairwise correlation analysis, this time using the large dataset generated according to algorithm 6. The scatter matrix plot in figure 6.3 depicts the relationships between each pair of metrics using all summaries in this generated dataset.

Unsurprisingly, we again observe high positive correlations. As in TAC annotations (figure 6.2), the correlations are about .7 Kendall's τ. **JS-2** and **R-2** have the strongest correlation, while **R-L** seems less correlated with the others. It is worth remembering that **JS-2** and **R-2** both operate on bigrams which also explain their stronger connection.

The diagonal cells display the score distribution of summaries for each metric. In our generated dataset, each evaluation metric was sampled evenly across its scoring range. However, when summaries sampled for every metric are put together, **R-2** and **JS-2** exhibit distributions skewed toward lower scores. It suggests that average summaries for other metrics tend to have below average **R-2** and **JS-2** scores.

6.2.2 When Metrics Do Not Correlate

Poor summaries are easily dismissed by metrics and summarization systems alike. Hence, the main focus of summarization progress is towards increasingly high-scoring summaries (Radev et al., 2003). We focus our next experiments on this relevant scoring range.

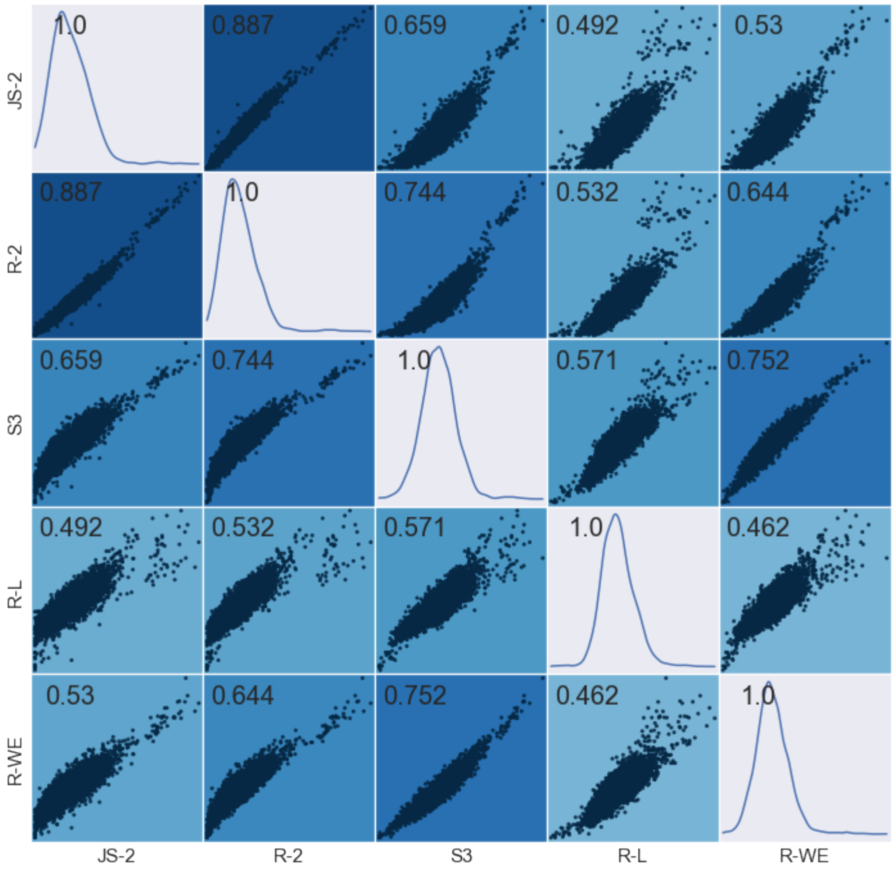


Figure 6.2: Scatter matrix plot over all summaries available as part of the human evaluation in TAC-2008 and TAC-2009.

To do so, we selected the top summaries from our generated dataset following the procedure described in algorithm 7. In this algorithm, the function $Score(\mathcal{T}, m)$ returns a list of all the summaries in the topic $\mathcal{T}$ scored by the metric m. The baseline B is an existing algorithm used as a threshold: for each metric, we keep every summary scoring higher than B. The final set of top-scoring summaries is the union of the top-scoring summaries of each metric.

For the thresholding, we chose LexRank (Erkan and Radev, 2004), because it is a heavily used baseline. Therefore, most current and future summarization systems should perform better and should be covered by the selected scoring range. Besides, LexRank is strong enough to discard a large number of average scoring summaries. After the selection, we ended up with an average of 102 summaries kept per topic.

We performed the same correlation analysis as the ones presented before in figure 6.2 and figure 6.3, but on this restricted dataset of high-scoring summaries. Figure 6.4 is the scatter matrix plot depicting the relationship between pairs of metrics. Again, the cell background color and the number indicate the pairwise correlation (Kendall's τ) averaged over all topics.

Surprisingly, we observe low and even several negative correlations.[2] Even, **R-2**

[2] We see the same behavior for each dataset taken independently

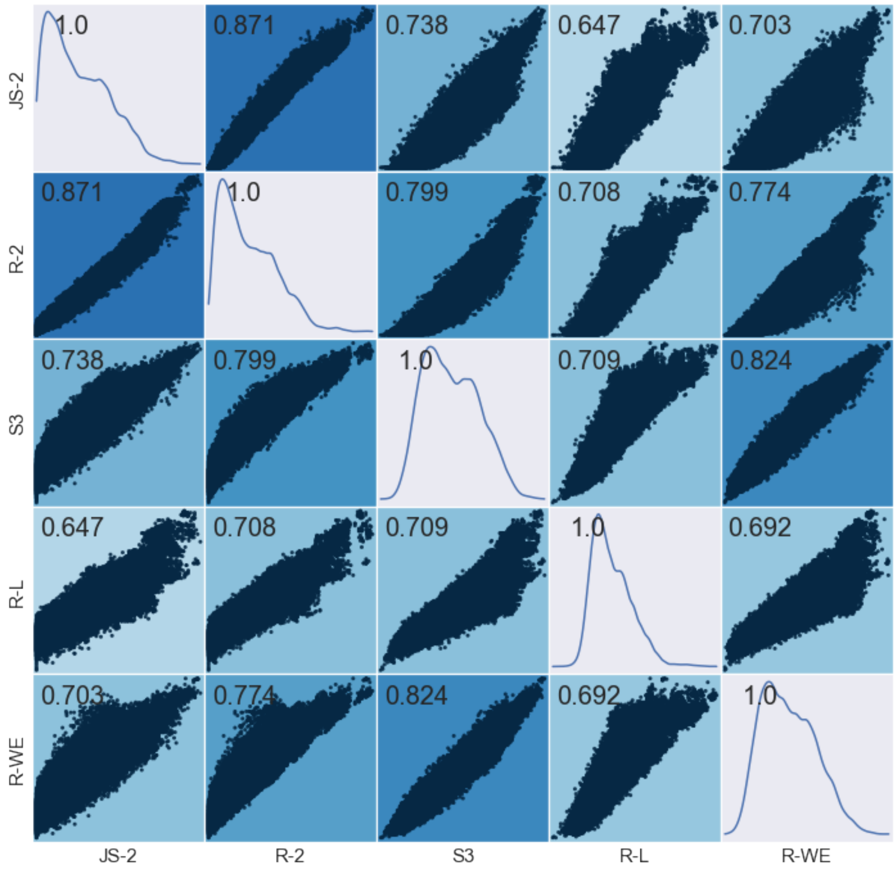


Figure 6.3: Scatter matrix plot over all summaries available in the generated dataset.

and **JS-2** which had the strongest connection in figure 6.2 only retain little correlation ($< 0.3\ \tau$). For most pairs, the correlations are close to what would be expected from random behavior. Additionally, **R-L** seems to even have negative correlation with other metrics. It indicates that there is no global agreement on what constitutes improvements when the summaries are already better than the baseline. This greatly undermines the possibility of constructing a more robust evaluation methodology simply by reporting several metrics. This problem is discussed in the next paragraphs.

On the diagonal, as a result of the selection process, we obtain more spread score distributions. Note that low-scoring summaries are possible even when selecting only the top-scoring summaries of each topic. Indeed, for some difficult topic, even the best summaries have low scores compared to other summaries drawn from easier topics. Also, top-scoring summaries according to one metric may be low-scoring according to another. Figure 6.4 indicates that this actually happens.

Disagreement increases with higher-scoring summaries:
In this experiment, we measured the percentage of disagreement between two metrics as a function of the average score of summaries. In figure 6.5, the y coordinate represents the percentage of disagreement for pairs of summaries which have an average score higher or equal to the x coordinate. We observe that the disagreement is increasing for higher scoring summary pairs. (Note the x axis has been normalized

```
Algorithm 7: Select Top-Scoring Summaries
Input  : $D = \{\mathcal{T}_1, \dots, \mathcal{T}_n\}$: dataset as a list of topics (each topic contains a
         list of summaries)
         $B$: baseline algorithm used to decide the high-scoring summaries
         $M = \{m_1, \dots, m_e\}$: evaluation metrics considered
Output: $D^{(top)}$: dataset which contain only top-scoring summaries
Function SelectTopSummaries($D, B, M$):
    $D^{(top)} := []$
    for $\mathcal{T} \in D$ do
        $T^{(top)} := []$
        for $m \in M$ do
            $S := []$
            for $s \in Score(\mathcal{T}, m)$ do
                if $m(s) > m(B(\mathcal{T}.source))$ then
                    $S \leftarrow s$
                end
            end
            $T^{(top)} := T^{(top)} \cup S$
        end
        $D^{(top)} \leftarrow T^{(top)}$
    end
```

because different metrics have different scales). This confirms the hypothesis that metrics disagree more for increasingly high-scoring summaries.

The problem with reporting several disagreeing metrics:
It is common to report the results of several evaluation metrics. However, when the metrics do not correlate, it becomes difficult to identify consistent improvements. Indeed, when metrics do not share some common patterns, improvements according to one metric is almost never an improvement for another metric. We illustrate this problem with the following experiment:

We first fix a summary s. We then consider the set of summaries from the same topic which are better than s according to at least one metric. We note N the size of this set. Then, we count the number of summaries which are better than s according to all metrics. We note F this number. Finally, $\frac{F}{N}$ is the proportion of summaries which are better than s for all metrics. This measures the difficulty of finding consistent improvements across metrics. In other words, we ask: among the summaries which are better than s for one metric, how many are better for all metrics?

The process is done for $5,000$ randomly sampled summary s in the sources for which $N > 20$. Indeed, if N is too small, $\frac{F}{N}$ is not a representative proportion. The figure 6.6, represents the results of this experiments, where the x-axis is the score of the summary s (average of all the metrics after they have been normalized between 0 and 1) and the y-axis is the corresponding proportion $\frac{F}{N}$.

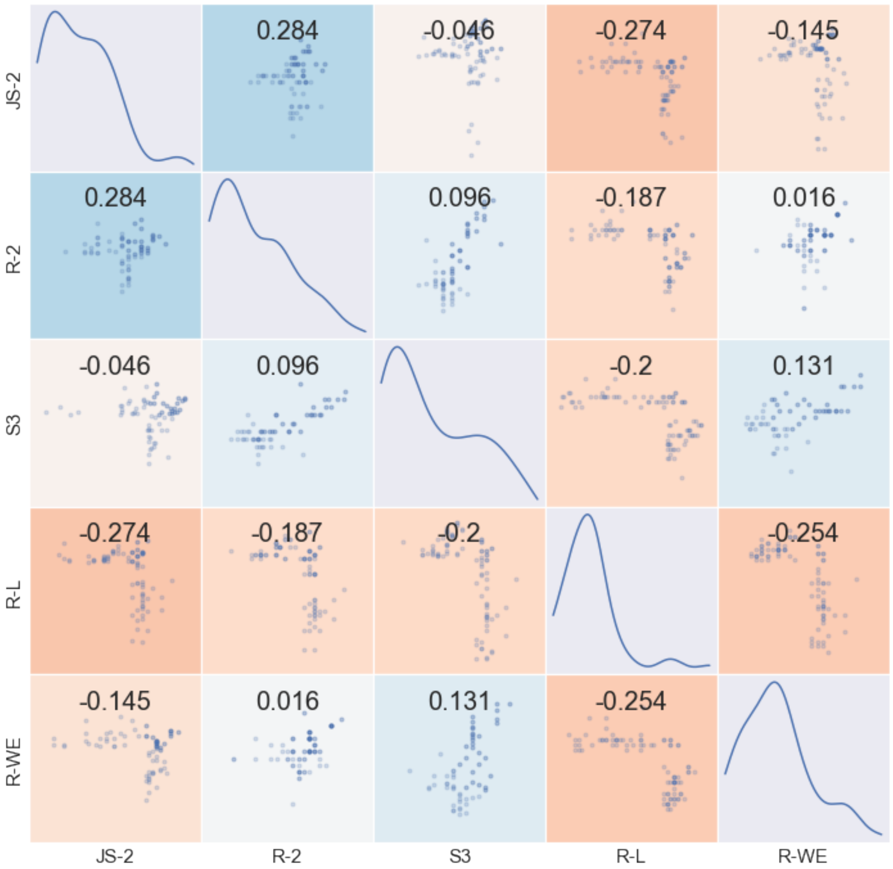


Figure 6.4: Scatter matrix plot on high-scoring summaries selected with the procedure described by algorithm 7.

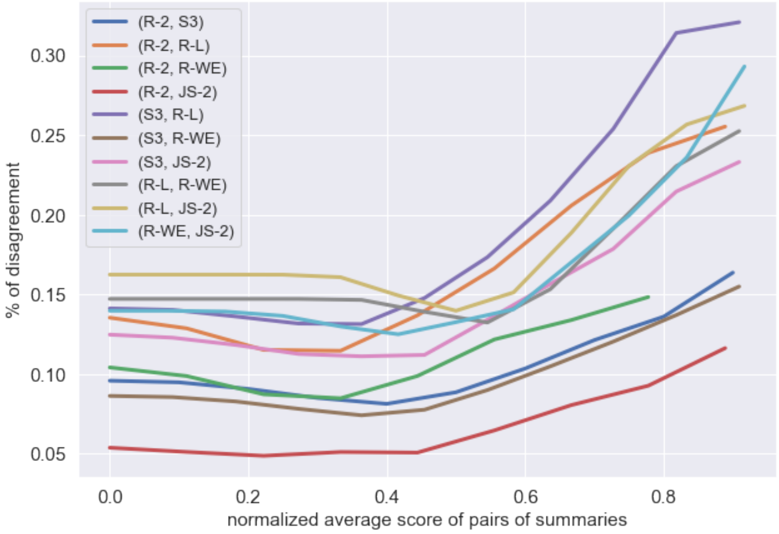


Figure 6.5: Percentage of disagreement between metrics for increasing scores of summary pairs.

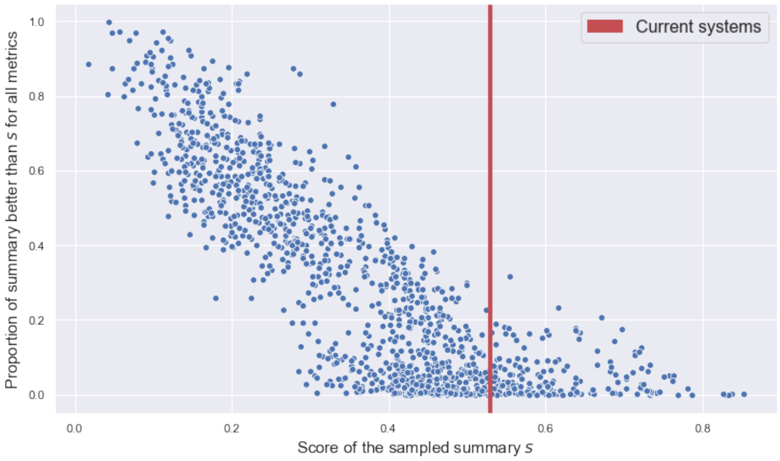


Figure 6.6: On the x axis: the score of the sampled summary s is computed by averaging the score assigned to s by every metric. For meaningful averaging, all metrics have been previously normalized such that 0 is the score of the worst summary and 1 the score of the best summary. We also report the average performance of current systems. On the y axis: $\frac{F}{N}$ associated to the sampled summary s.

We observe a quick decrease in the ratio $\frac{F}{N}$. The proportion of consistent improvements (agreed by all metrics) is quickly decreasing with the average score of summaries. When the baseline scores go up, the disagreement between metrics is strong enough that we cannot identify summaries which are considered better than the baseline for each metric. This is problematic because we don't know which metrics is the best. We also observe that this behavior starts to be problematic considering the current systems average performances.

Discussion:
Intuitively, a smaller population can also lead to lower correlations. However, in the high-scoring range, there are 102 summaries per topic and we observe very low correlations whereas in the average-scoring range (human judgments) there are around 50 summaries per topic but we observe strong correlations.

Furthermore, the plot of figure 6.5 is normalized according to population size and the disagreement still increases with average scores. This rules out the possibility of explaining the low correlations simply by population size.

Also, the high-scoring range covers 38% of the full scoring range (from LexRank to upper-bound), while human judgments covers 35% of the full scoring range. This rules out the possibility of explaining the low correlations by the width of the scoring range.

Chapter Summary

- Existing human judgment datasets are not suitable to properly compare evaluation metrics because they do not cover the high-scoring range in which summarization systems and metrics operate.
- Existing evaluation metrics do not correlate in the high-scoring range. We cannot measure improvements reliably because metrics disagree and we don't know which one to trust.
- This motivates efforts to collect human judgments for high-scoring summaries as this would be necessary to settle the debate over which metric to use. This would also be greatly beneficial for improving summarization systems and metrics alike.

Chapter 7

Conclusion

Automatic text summarization is a complex NLP task that requires natural language understanding, content selection and natural language generation capabilities. In this thesis, we concentrated on content selection, the inherent challenge of summarization which is controlled by the notion of *Importance* and encoded in summary scoring functions. We introduced several interconnected frameworks to model the summarization task and guide the search for summary scoring functions. Within these frameworks, we investigated both empirical and theoretical techniques to discover some strong scoring functions. We now summarize the main contributions and findings.

We introduced the (θ, O) framework which views, without loss of generality, the summarization task as two components: a summary scoring function θ and an optimization technique O. Every summarizer implicitly or explicitly implements a summary scoring function θ which is subject to an evaluation of its own. By analogy with the evaluation of evaluation metrics, different summary scoring functions can be compared based on their ability to correlate with human judgments.

We conducted this evaluation on existing summarizers and discovered surprisingly low correlations. This suggests that current summarization systems do not use the same strategy as humans during the summarization process. This θ evaluation is also shown to be complementary to the conventional evaluation of extracted summaries with automatic metrics as they provide different rankings of systems.

In fact, the evaluation of summary scoring functions can be used as a way to guide us when crafting summary scoring functions. This can be useful to pinpoint potential areas of improvements. These contributions particularly address **Research Question 3**.

Once the (θ, O) perspective is adopted, one might be motivated to develop both components independently. Hence, from the optimization perspective, we were interested in the most general scenario where θ does not exhibit exploitable mathematical properties. In such case, in order to extract a summary out of it, θ must be optimized approximately via heuristic search algorithms (or GPOs). In this work, we adapted and implemented several techniques which can optimize arbitrary objective functions. We demonstrated that they are both efficient and effective enough for the summarization use-case. This answers **Research Question 1** and allows us to search arbitrarily complex summary scoring functions.

Furthermore, existing summarizers which greedily optimized their internal summary scoring function can be significantly improved simply by switching to a more powerful optimization techique like the genetic algorithm. Also, for some evaluation metrics like JS-Eval and PEAK, it is impossible to find the exact upper-bound efficiently. By leveraging the complementarity of several optimization techniques, we computed better upper-bound estimates for both JS-Eval and PEAK.

An analysis of previous works revealed that they have heavily constrained the scoring function θ in order to solve convenient optimization problems. However, we showed that a θ relieved from constraints is better able to match human judgments. To do so, we trained various summary scoring functions with and without linearity constraints and observed a large gap in favor of the unconstrained functions. When such summary scoring functions are optimized by an appropriate optimization technique (e.g., genetic algorithm), the unconstrained functions are still capable of extracting high-scoring summaries. This answers **Research Question 2**.

When the summary scoring function is learned from data, we investigated another dimension of variation: the supervision signal. Traditionally, ROUGE provides this supervision. In such case, based on the mathematical structure of ROUGE, we could derive an almost perfect linear approximation (provided sentence scores are available). However, recent improvements in evaluation metrics should encourage us to progressively move away from ROUGE towards more semantically motivated metrics. In particular, we trained summary scoring functions using alternative metrics like JS-Eval and PEAK and observed good performances when PEAK was used.

Ideally, summary scoring functions should mimic humans and, thus, human scores would be the best possible signal. Unfortunately, human annotations are rare and expensive to obtain. We demonstrated a simple strategy to leverage the small human judgment datasets for training. This contributes to **Research Question 4**.

When human judgments are used for supervision, we saw that training the evaluation metric and training the summarizer's scoring function is the same learning problem, with the difference that the evaluation metric can leverage the reference summaries. Hence, we could train a new evaluation metric **S3** and released it for the community.

Apart from empirically learning the summary scoring function from statistical analysis, we also investigated a theoretical formulation of the notion of *Importance*. In a framework rooted in information theory, we formalized several summary-related quantities like: *Redundancy*, *Relevance* and *Informativeness. Importance* arises as the notion unifying these concepts. More generally, *Importance* is the measure that guides which choices to make when information must be discarded. This contribution answers **Research Question 5** and provides promising directions for future work.

Finally, evaluation remains an open-problem with a massive impact on summarization progress. We conducted experiments on available human judgment datasets commonly used to compare evaluation metrics and discovered that they do not cover the high-scoring range in which summarization systems and evaluation metrics operate. A series of experiments motivates efforts to collect human judgments for high-scoring summaries as this would be necessary to settle the debate over which metric to use. This would also be greatly beneficial for improving summarization systems and evaluation metrics alike.

Future Work

To conclude this thesis, we provide interesting directions for future works which can built upon the work established in this thesis.

First, the (θ, O) framework from chapter 3 provides a general and universal interpretation of the summarization task. This creates two distinct branches of potential improvements: (i) investigating new optimization techniques O particularly tailored to the summarization task, and (ii) exploring better learning scenario for the summary scoring function θ. Furthermore, the theoretical framework introduced in chapter 5 presents a high-level view of summarization. It can be extended to a broader set of problems but can also inform the development of new summarization systems.

Learning and Optimizing Summary Scoring Functions

With the aim of training better summarization systems, one could consider the contributions made by chapter 4. In this chapter, we only scratched the surface of what is possible within the (θ, O) framework. Several subsequent works may investigate more sophisticated techniques to learn θ and also craft summarization-specific optimization procedures.

Exploring learning possibilities for θ:
In chapter 4, we introduced the matrix of possible learning scenarios for θ and proposed an initial study of the two main axes of variations: the learning constraints and the supervision signal. The feature set is also an important design choice, but we left it fixed in our experiments. In fact, this matrix hints at many possible future works along each axis:

- **Axis 0: Features** While we restricted ourselves to a simple feature set, future works can clearly benefit from a larger set of more sophisticated features including semantically motivated quantities. For instance, one could include the signal from distributional representations of meaning like word vectors (Mikolov et al., 2013b) or sentence vectors (Conneau et al., 2017). In particular, He et al. (2012) introduced the idea that a summary should be made of the sentences which allow the best reconstruction of the input documents. Several works (Li et al., 2015; Liu et al., 2015b; Ma et al., 2016) refined this idea with distributional similarities. We believe that such scoring functions can provide useful signal in a learning scenario.

  In general, any existing summary scoring function can become a feature in the θ learning setup. Indeed, with the constraint on θ removed, more meaningful and complex summary scoring functions can be employed. This opens the possibility to include any existing previous works in the feature set.

Finally, provided enough training data is available, one could learn the scoring function from deep learning architectures by representing texts as sequences of word vectors without specifying any other features. This would generalize the approach proposed by Nallapati et al. (2017) by providing supervision at the summary-level.

- **Axis 1: Supervision signal** Evaluation becomes a more pressing topic in the summarization community. New automatic evaluation metrics are regularly introduced to address the shortcomings of existing ones. When new promising metrics arise, they become great candidates to be used as a supervision signal for training summary scoring functions.

  Alternatively, since human judgments would be the ideal signal, one could allocate resources to collect larger sets of manual annotations. We made a case for this in chapter 6.

  Finally, one could use partial human feedback and include them in the learning loop with (inter-)active learning scenarios (P.V.S. and Meyer, 2017).

- **Axis 2: Learning constraints and learning algorithms** We studied the two main kinds of constraints: linear and *no constraint*. In summarization, submodularity has also been extensively used for learning ROUGE (Lin and Bilmes, 2011; Sipos et al., 2012). It would be interesting to extend the comparison of section 4.4 to also consider the submodularity constraint. Recently, Tschiatschek et al. (2018) proposed a methodology to learn submodular functions which is akin to the derivation we made for linear functions in section 4.1.

  More generally, one can investigate learning algorithms because different learning algorithms come with different assumptions. For example, it might be particularly interesting to learn θ with a ranking loss instead of a regression loss. Indeed, the scores available for summaries may suffer from inconsistency when going from one topic to another. In contrast, the relative ordering of summaries may be a more robust signal (Fürnkranz and Hüllermeier, 2003).

  Bayesian approaches capable of dealing with noisy annotations (Chu and Ghahramani, 2005) could be employed to better learn human judgments or to combine noisy automatic scores with human ones (Simpson and Gurevych, 2018).

From GPO to summarization specific global optimization:
Because of the *no free lunch theorem* (Wolpert and Macready, 1997), we do not have guarantees about the performances of General Purpose Optimization techniques. Thus, optimization strategies informed with knowledge about language and the structure of summarization could be investigated.

Even though we viewed the extraction part of summarization as a generic optimization task in this thesis, it actually happens on a specific kind of data – texts – which might present exploitable regularities. Such regularities can be identified and leveraged to design more efficient optimizers. They can be discovered by statistical analysis as part of the *learning to optimize* paradigm (Li and Malik, 2016). Optimizers can be either partially or completely learned from the kind of data on which they are expected to perform optimization.

In fact, there are simple improvements that can be easily incorporated into optimizers. For example, by knowing that redundant summaries correlate negatively with quality, the neighbor search can be incentivized to generate non-redundant summaries. Another example is that frequency is known to have a positive correlation with *Importance* on standard datasets. The stochastic search could be biased to sample summaries with frequent *terms* more often. Such modifications are expected to improve the convergence speed as they avoid evaluating summaries unlikely to provide improvements.

In general, learning representations useful for the optimization heuristic might be beneficial (Li and Malik, 2016). For example, one could change the neighbor search from a simplex on sentences (every summary is a binary vector of the sentences it contains) to a semantic space. In this case, neighbors are semantically similar summaries instead of summaries which differ by one sentence.

Finally, while we argued for the study of θ and O independently, remerging and jointly learning them in end-to-end scenario might be investigated under the constraint of preserving an interpretable notion of θ. Indeed, this is the central concept of summarization and it should be evaluated on its own. Approaches like Inverse Reinforcement Learning (Ng and Russell, 2000) might then be of interest. Such approaches try to infer the reward function (here θ) from observed outcomes (here reference summaries).

Extensions and Applications of the Theoretical Framework

Now, we discuss interesting research directions stemming from the framework outlined in chapter 5.

Text representations:
The notion of semantic units introduced in section 5.1 is a very general representation of meaning supported by previous works. Enforcing text representations (e.g., distributional representations) to be probability distributions over independent units may result in better semantic textual representations. Furthermore, this would allow the use of information-theoretic tools at the semantic level for a wide range of NLP problems.

Semantic information theory:
In section 5.3, we mentioned that the *Importance* framework provides an entry point to develop an operational semantic information theory and semantic data compression theory.

Indeed, summarization is a lossy semantic compression which draws an analogy with *Rate-Distortion Theory* initiated by Shannon (1948) to address lossy coding compression schemes. Prominent examples of applications are *JPEG* for lossy compression of images or *MP3* for lossy compression of audio signals (Ortega and Ramchandran, 1998). *Rate-Distortion Theory* builds upon the notion of distortion functions which measure the human-perceived discrepancies between the lossy compressed and original uncompressed data.

While these distortion functions operate at the syntactic level (Shannon, 1948), the notion of *Importance* aims to operate at the semantic level. This poses the bases of lossy semantic compression schemes which could be further studied and made into a wider theory of semantic compression.

Practical applications in summarization:
Conceptually, it is straightforward to build a system out of θ_I (the summary scoring function induced by *Importance*) once a semantic units representation and a K have been chosen. A summarizer intends to extract or generate a summary maximizing θ_I. Indeed, θ_I is a summary scoring function which fits within the (θ, O) framework.

Therefore, in extractive summarization, this can naturally be cast as a discrete optimization problem where the text source is considered as a set of sentences and the summary is created by selecting an optimal subset of the sentences under a length constraint (McDonald, 2007). In abstractive summarization, a language-aware decoder needs to be employed to also guarantee linguistic qualities.

In fact, the background knowledge and semantic units are free parameters of the theory. They are design choices which can be explored empirically. Then, interesting research questions arise like *Which granularity offers a good approximation of semantic units?*, *Can we automatically learn good approximations?* In summarization, n-grams are known to be useful, but other granularities have rarely been considered together with information theoretic tools.

Discovering K:
The background knowledge K was introduced, but its practical implementation remains a design choice. A promising direction would be to use the framework to actually learn K from data. In particular, one can apply supervised techniques to automatically search for K, α and β: finding the values of these parameters such that θ_I has the best correlation with human judgments. By aggregating over many users and many topics one can find a generic K: what, on average, people consider as known when summarizing a document. By aggregating over different persons but in one domain, one can uncover a domain-specific K. Similarly, by aggregating over many topics for one person, one would find a personalized K.

Appendix

A Corpora and Implementation Details

We reproduce in table A.1 here the part of table 2.1 from section 2.1 that contains the description of the datasets used throughout the thesis. We also describe further details concerning these datasets.

A.1 DUC Datasets

In experiments from chapter 4, we used two datasets from the Document Understanding Conference (DUC) shared task: DUC-2002 [1] and DUC-2003 (Over, 2003). DUC-2002 and DUC-2003 contain about 60 and 30 topics, respectively. Each topic consists of 10 news articles to be summarized in a maximum of 100 words for DUC-2003 and 200 words for DUC-2002. In the official DUC-03 competitions, summaries of length 665 bytes were expected. Systems could produce different numbers of words. The variation in length has a noticeable impact on ROUGE recall scores.

A.2 TAC Datasets

We experiment with standard datasets for two different summarization tasks: generic and update multi-document summarization from the Text Analysis Conference (TAC) shared task: TAC-2008 and TAC-2009.[2] . The generic part is used for the experiments of chapter 3, chapter 4, chapter 5 and chapter 6. The update part is used for the experiments of chapter 5.

TAC-2008 and TAC-2009 contain 48 and 44 topics, respectively. Each topic consists of 10 news articles to be summarized in a maximum of 100 words. In the update part, 10 new documents (B documents) are to be summarized assuming that the first 10 documents (A documents) have already been seen.

For each topic, there are 4 human reference summaries along with a manually created Pyramid set (Nenkova et al., 2007). In both editions, all system summaries and the 4 reference summaries were manually evaluated by NIST assessors for readability, content selection (with Pyramid) and overall Responsiveness. At the time of the shared tasks, 57 systems were submitted to TAC-2008 and 55 to TAC-2009. The Responsiveness annotations follow a 5 point LIKERT scale in TAC-2008 but a 10 point LIKERT scale in TAC-2009.

[1] https://www-nlpir.nist.gov/projects/duc/pubs/2002slides/overview.02.pdf

[2] http://tac.nist.gov/2009/Summarization/, http://tac.nist.gov/2008/

Dataset	Creation	Input			Purpose	Output		Size	
	Man./Auto.	Type	Genre	Lang		Type	Length	Topics	Doc/Topic
DUC-2002	M	SDS, MDS	News	en	Gen.	Abs.	10-400	60	≈10
DUC-2003	M	SDS, MDS	News	en	Gen.	Abs./Ext.	10, 100	30	≈10
TAC-2008	M	MDS	News	en	Gen. Upd. Opi.	Abs.	100	48	10
TAC-2009	M	MDS	News	en	Gen. Upd.	Abs.	100	44	10
DBS	M	MDS	Heter.	de	Gen.	Ext.	≈500	30	4-14

Table A.1: Description of the datasets used during the thesis

A.3 DBS

We also use the recently created German dataset DBS-corpus (Benikova et al., 2016). It contains 10 topics consisting of 4 to 14 documents each. The summaries have variable sizes and are about 500 words long. For each topic, 5 summaries were evaluated by trained human annotators but only for content selection with Pyramid.

We experiment with this dataset because it contains heterogeneous sources (different text types) in German about the educational domain. This contrasts with the English homogeneous news documents from DUC and TAC.

A.4 Other Details

In this section, we describe some useful implementation details concerning the computation of evaluation metrics and features.

ROUGE:
ROUGE-N (Lin, 2004b) is an evaluation metric used throughout the thesis (chapter 3 and chapter 4).

For ROUGE-N, we used the variants of ROUGE identified by Owczarzak et al. (2012) as strongly correlating with human evaluation methods: with stemming and stopwords not removed (giving the best agreement with human evaluation). The truncation of system summaries is done automatically by ROUGE using the official Perl script.[3]

JS-Eval:
JS-Eval (Lin et al., 2006) is also an evaluation metric that was used throughout the thesis (chapter 3 and chapter 4) Like ROUGE-N, JS-Eval operates on n-grams. Thus, we used the same specifications: stemming and stopwords not removed.

Details about features and baselines:
In order to get the scores of baselines (LexRank, TF·IDF and Edmundson) in chapter 3 and chapter 4, we used the freely available *sumy* package.[4]. Similarly, in order to extract the features based on these systems – as described in section 4.1 – we also adapted the *sumy* implementation.

For the baseline systems *ICSI* (Gillick and Favre, 2009) and *SFOUR* (Sipos et al., 2012), we used the freely available implementations.[5]

[3] ROUGE-1.5.5 with the parameters: -n 2 -m -a -l 100 -x -c 95 -r 1000 -f A -p 0.5 -t 0

[4] `https://github.com/miso-belica/sumy`

[5] For ICSI, we used the later implementation from Boudin et al. (2015): `https://github.com/`

The remaining baselines and features (KL, JS, n-gram coverage and diversity) rely on n-grams. Contrary to ROUGE and JS-Eval, we removed the stopwords as we observed better performances without them. In particular, whenever $n > 1$, the n-grams composed of only stopwords are removed.

KL divergence and cross-entropy (used in chapter 5) implicitly assume that the two input distributions P and Q have the same support. However, in summarization, it often happens that n-grams appear in one but not the other. One solution is to add a smoothing factor to prevent divisions by 0. Alternatively, one can compute KL and cross-entropy only the shared support. We found the latter to give better performances; it also does not require to choose a smoothing factor.

`boudinfl/sume`. For SFOUR: `http://www.cs.cornell.edu/~rs/sfour/`

B Standard Optimization Algorithms

Greedy:
The greedy algorithm selects the sentence with the best score at each steps. We refer to this algorithm as *Greedy* in the following sections. A convenient improvement is the greedy with marginal gains (*Greedy-M*) which selects the sentence which incurs the best increase in the overall score. The pseudo-code of Greedy-M is described in algorithm 8.

Algorithm 8: Greedy Algorithm for Extractive Summarization

```
Input  : $D = \{s_1, \dots, s_n\}$: document as a set of sentences
         $\theta$: objective function
         $L$: length constaint
Output: $S = \{s_j\}$: summary as a set of sentences
Function Greedy-M $(D, \theta, L)$:
    $S := \{\}$
    while 1 do
        $C := \{s \in D \mid s \notin S,\ len(S \cup \{s\}) \leq L\}$
        if $C = \emptyset$ then
            return $S$
        end
        $c^* := \underset{c \in C}{\operatorname{argmax}}\, \theta(S \cup c)$
        $S := S \cup \{c^*\}$
    end
```

Beam Search:
The pseudo-code for the *Beam Search* is given by the algorithm 9. Note the notation $\overset{(k)}{\operatorname{argmax}}$ which denotes the operator returning the top k elements, i.e., the k elements which have the best scores in the list. N is the set of candidate answer and n is a candidate summary.

Suppose the final summary contains m sentences, then this algorithm keeps k candidates for each of the m decision steps and therefore considers $k \cdot m$ candidates.

Random Search:
Based on the sampling function, the random search follows a simple described by the pseudo-code in algorithm 10. It has the advantage of having a fixed and predefined complexity set by B the budget of allowed candidate evaluation.

Simulated Annealing:
The pseudo-code for *Simulated Annealing* is given by the algorithm 11. The function *Random*$(0, 1)$ generates samples from the uniform distribution between 0 and 1. Simulated annealing also has a fixed complexity predefined by the number of steps allowed k_{max}.

Algorithm 9: Beam Search for Extractive Summarization

```
Input  : $D = \{s_1, \dots, s_n\}$: document as a set of sentences
         $\theta$: objective function
         $L$: length constaint
         $k$: size of the beam
Output: $S = \{s_j\}$: summary as a set of sentences
Function BeamSearch ($D, \theta, L, k$):
    $N \coloneqq \{\}$
    while 1 do
        $C \coloneqq \{\}$
        for $n \in N$ do
            $C \coloneqq C \cup \{s \in D \mid s \notin n,\ len(n \cup \{s\}) \leq L\}$
        end
        if $C = \emptyset$ then
            $S \coloneqq \operatorname{argmax}_{n \in N} \theta(n)$
            return $S$
        end
        $\mathbf{c}^* \coloneqq \overset{(k)}{\operatorname{argmax}}_{c \in C} \theta(n \cup c)$
        $N \coloneqq N \cup \mathbf{c}^*$
    end
```

Algorithm 10: Random Search for Extractive Summarization

```
Input  : $D = \{s_1, \dots, s_n\}$: document as a set of sentences
         $\theta$: objective function
         $L$: length constaint
Output: $S = \{s_j\}$: summary as a set of sentences
Function RandomSearch ($D, \theta, L, B$):
    $c^* = \{\}$
    evaluations $= 0$
    while $evaluations < B$ do
        $c = SampleCandidate(D, L)$
        if $\theta(c) > c^*_{score}$ then
            $c^* \coloneqq c$
            evaluations++
        end
    end
```

Algorithm 11: Simulated Annealing for Extractive Summarization

```
Input  : $D = \{s_1, \dots, s_n\}$: document as a set of sentences
         $\theta$: objective function
         $L$: length constaint
         $T$: temperature
         $k_{max}$: maximum number of iteration
Output: $S = \{s_j\}$: summary as a set of sentences
Function SimulatedAnnealing $(D, \theta, L, T, k_{max})$:
    $S := SampleCandidate(D, L)$
    for $k \in \{1, \dots, k_{max}\}$ do
        $N := Mutate(S, D, L)$
        if $P(N, S, T) \geq Random(0, 1)$ then
            $S := N$
        end
    end
```

Genetic Algorithm:
Algorithm 12 describes the optimization process via the genetic algorithm. The function *RandomChoose*(P, n) randomly selects n elements from the list P. We note that in nature, the fertilized egg cell undergoes a process known as embryogenesis before becoming a mature embryo. This is believed to make the genetic search more robust by reducing the probability of fatal mutation. At each step, we only consider valid summaries, which is the direct analogy to embryogenesis.

Artificial Bee Colony:
The overall execution of the ABC algorithm is presented in the algorithm 14.

Algorithm 12: Genetic Algorithm Optimization for Extractive Summarization

```
Input  : $D = \{s_1, \dots, s_n\}$: document as a set of sentences
         $\theta$: objective function
         $L$: length constaint
         $popsize$: number of candidate in the population
         $S_r$: survival rate
         $M_r$: mutation rate
         $R_r$: reproduction rate
         $e_{max}$: maximum number of epochs
Output: $S = \{s_j\}$: summary as a set of sentences
Function $GeneticAlgorithm(D, \theta, L, popsize, S_r, M_r, R_r, e_{max})$:
    $C \coloneqq []$
    for $i \in \{1, \dots, popsize\}$ do
        $C \leftarrow SampleCandidate(D, L)$
    end
    for $e \in \{1, \dots, e_{max}\}$ do
        $newpop \coloneqq []$
        $survivors \leftarrow argmax_{c \in C}^{(popsize \cdot S_r)} \theta(c)$
        $n = len(survivors)$
        $newpop \leftarrow survivors$
        $M \coloneqq RandomChoose(survivors, n \cdot M_r)$
        for $m \in M$ do
            $newpop \leftarrow Mutate(m, D, L)$
        end
        for $j \in \{1, \dots, n \cdot R_r$ do
            $P \coloneqq RandomChoose(survivors, 2)$
            $newpop \leftarrow CrossOver(P, L)$
        end
    end
```

Algorithm 13: Choose a Location for the Artificial Bee Colony Algorithm

```
Input   : $S$: vector of scores
Output: $idx$: index of the chosen location
Function ChooseLocation($D, \theta, L, n, T, mfe$):
    $Z \coloneqq sum(S)$
    $P \coloneqq []$
    for $i \in \{1, \ldots, len(S)\}$ do
        $P \leftarrow \frac{S[i]}{Z}$
    end
    while True do
        for $j \in \{1, \ldots, len(P)\}$ do
            if $P[j] \geq Random(0, 1)$ then
                return $j$
            end
        end
    end
```

Algorithm 14: Artificial Bee Colony Algorithm for Extractive Summarization

```
Input  : $D = \{s_1, \ldots, s_n\}$: document as a set of sentences
         $\theta$: objective function
         $L$: length constaint
         $n$: number of employed bees
         $tl$: number of trial before giving up a location
         $mfe$: maximum function call
Output: $S = \{s_j\}$: summary as a set of sentences
Function $ABC(D, \theta, L, n, T, mfe)$:
    $C := []$
    $S := []$
    $T := []$
    for $i \in \{1, \ldots, popsize\}$ do
        $c = SampleCandidate(D, L)$
        $C \leftarrow c$
        $S \leftarrow \theta(c)$
        $T \leftarrow 0$ // number of trials at this location
    end
    $nevals := 0$
    while $nevals \leq mfe$ do
        $C_c := [1] * len(C)$ // How many bees will work on each location
        for $i \in \{1, \ldots, n\}$ do
            $idx := ChooseLocation(S)$
            $C_c[idx] + +$
        end
        for $i \in \{1, \ldots, n\}$ do
            // Employed and Onlooker bees phase
            for $j \in C_c[i]$ do
                $M := Mutate(C[i], L)$
                if $\theta(M) \geq \theta(C[i]$ then
                    $C[i] := M$
                    $S[i] := \theta(M)$
                    $T[i] := 0$
                end
                else
                    $T[i] + +$
                end
            end
            // Scout bees phase
            if $T[i] \geq tl$ then
                $c = SampleCandidate(D, L)$ $C \leftarrow c$
                $S \leftarrow \theta(c)$
                $T \leftarrow 0$
            end
        end
    end
```

C Proofs

C.1 Proof of the Universality Theorem

As described in section 3.1.1, σ designs a summarization system, θ an objective function and O an optimization strategy. The statement of Theorem 1 is:

theorem

$$\forall \sigma, \exists (\theta, O) \text{ such that:} \tag{1}$$

$$\forall D \in \mathcal{D}, \sigma(D) = O(\theta, D) \tag{2}$$

theorem

Proof. We can construct a function θ_σ from σ which reconstructs the exact same summaries as σ when optimized by O.

For a given document collection D, suppose that $\sigma(D) = S_\sigma$. We define θ_σ to be the following function:

$$\theta_\sigma(S) = \begin{cases} 1, \text{ if } S = S_\sigma \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

It is clear that $\forall D \in \mathcal{D} : \sigma(D) = O(\theta_\sigma, D)$, because the optimal summaries according to θ_σ are precisely the summaries produced by σ. □

C.2 Recursive Expression of ROUGE-N

Let $S = \{s_i | i \leq m\}$ and $T = \{t_i | i \leq l\}$ be two sets of sentences, S^* the reference summary, and $\rho(X)$ denote the ROUGE-N score of the set of sentences X. Assuming that $\rho(S)$ and $\rho(T)$ are given, we prove the following recursive formula:

$$\rho(S \cup T) = \rho(S) + \rho(T) - \epsilon(S \cap T) \tag{4}$$

For compactness, we use the following notation as well:

$$C_{X,S^*}(g) = \min(F_X(g), F_{S^*}(g)) \tag{5}$$

Proof. We have the following definitions:

$$\rho(S) = \frac{1}{R_N} \sum_{g \in S^*} C_{S,S^*}(g) \tag{6}$$

$$\rho(T) = \frac{1}{R_N} \sum_{g \in S^*} C_{T,S^*}(g) \tag{7}$$

$$\epsilon(S \cap T) = \frac{1}{R_N} \sum_{g \in S^*} \max(C_{S,S^*}(g) + C_{T,S^*}(g) - F_{S^*}(g), 0) \tag{8}$$

And by definition of ROUGE, the formula of $S \cup T$:

$$\rho(S \cup T) = \frac{1}{R_N} \sum_{g \in S^*} \min(F_{S \cup T}(g), F_{S^*}(g)) \tag{9}$$

In order to prove equation (4), we have to show that the following equation holds:

$$\sum_{g \in S^*} C_{S,S^*}(g) + \sum_{g \in S^*} C_{T,S^*}(g) - \sum_{g \in S^*} \max(C_{S,S^*}(g) + C_{T,S^*}(g) - F_{S^*}(g), 0) = \sum_{g \in S^*} \min(F_{S \cup T}(g), F_{S^*}(g)) \quad (10)$$

It is sufficient to show:

$$\forall g \in S^*, C_{S,S^*}(g) + C_{T,S^*}(g) - \max(C_{S,S^*}(g) + C_{T,S^*}(g) - F_{S^*}(g), 0) = \min(F_{S \cup T}(g), F_{S^*}(g)) \quad (11)$$

Let $g \in S^*$ be a n-gram. There are two possibilities:

- $F_S(g) + F_T(g) \leq F_{S^*}(g)$: g appears less frequently in $S \cup T$ than in the reference summary. It implies: $\min(F_{S \cup T}(g), F_{S^*}(g)) = F_{S \cup T}(g) = F_S(g) + F_T(g)$. Moreover, all $F_X(g)$ are positive numbers by definition, and $F_S(g) \leq F_{S^*}(g)$ is equivalent to: $C_{S,S^*}(g) = \min(F_S(g), F_{S^*}(g)) = F_S(g)$. Similarly, we have: $C_{T,S^*}(g) = \min(F_T(g), F_{S^*}(g)) = F_T(g)$. Since $\max(C_{S,S^*}(g) + C_{T,S^*}(g) - F_{S^*}(g), 0) = 0$, the equation (11) holds in this case.

- $F_S(g) + F_T(g) \geq F_{S^*}(g)$: g appears more frequently in $S \cup T$ than in the reference summary. It implies: $\min(F_{S \cup T}(g), F_{S^*}(g)) = F_{S^*}(g)$. Here we have: $\max(C_{S,S^*}(g) + C_{T,S^*}(g) - F_{S^*}(g), 0) = C_{S,S^*}(g) + C_{T,S^*}(g) - F_{S^*}(g)$, and it directly follows that equation (11) holds in this case as well.

Equation (11) has been proved, which proves (4) as well. □

C.3 Expanded Expression of ROUGE-N

Let $S = \{s_i | i \leq m\}$ be a set of sentences and $\theta_R(S)$ its ROUGE-N score. We prove the following formula:

$$\theta_R(S) = \sum_{i=1}^{m} \theta_R(s_i) + \sum_{k=2}^{m} (-1)^{k+1} \Big(\sum_{1 \leq i_1 \leq \cdots \leq i_k \leq m} \epsilon^{(k)}(s_{i_1} \cap \cdots \cap s_{i_k}) \Big) \quad (12)$$

Proof. Let $g \in S^*$ be a n-gram in the reference summary, and $k \in [1, m]$ the number of sentences in which it appears. Specifically, $\exists \{s_{i_1}, \cdots, s_{i_k}\}, \forall s_{i_j} \in \{s_{i_1}, \ldots, s_{i_k}\}, g \in s_{i_j}$. In order to prove the formula (12), we have to find an expression for the $\epsilon^{(k)}$ that gives to g the correct contribution to the formula:

$$\frac{1}{R_N} \min(F_S(g), F_{S^*}(g)) \quad (13)$$

First, we observe that g does not appear in the terms that contain the intersection of more than k sentences. Specifically, $\epsilon^{(t)}$ is not affected by g if $t \geq k$. However, g is affected by all the $\epsilon^{(t)}$ for which $t \leq k$.

Given that g appears in the sentences $\{s_{i_1}, \dots, s_{i_k}\}$, we can determine the score attributed to g by the previous $\epsilon^{(t)}$ $(t \leq k)$:

$$S^{(k-1)}(g) = \sum_{s \in \{s_{i_1}, \dots, s_{i_k}\}} \theta_R(s) + \sum_{l=2}^{k} (-1)^{(l+1)} \sum_{1 \leq i_1 \leq \dots \leq i_l \leq l} \epsilon^{(l)}(s_{i_1} \cap \dots \cap s_{i_l})) \tag{14}$$

Now, g receives the correct contribution to the overall scores if $\epsilon^{(k)}$ is defined as follows:

$$\epsilon^{(k)}(s_{i_1} \cap \dots \cap s_{i_j}) = \frac{1}{R_N} \sum_{g \in s_{i_1} \cap \dots \cap s_{i_j}} C_{\{s_{i_1}, \dots, s_{i_k}\}}(g) - S^{(k-1)}(g) \tag{15}$$

Indeed, with this expression for $\epsilon^{(k)}$, the score of g is:

$$S^{(k-1)}(g) + \frac{1}{R_N} C_{\{s_{i_1}, \dots, s_{i_k}\}}(g) - S^{(k-1)}(g) \tag{16}$$

$$= \frac{1}{R_N} C_{\{s_{i_1}, \dots, s_{i_k}\}}(g) \tag{17}$$

Since g appears only in the sentences $\{s_{i_1}, \dots, s_{i_k}\}$, $F_{\{s_{i_1}, \dots, s_{i_k}\}}(g) = F_S(g)$ and it follows that:

$$\frac{1}{R_N} C_{\{s_{i_1}, \dots, s_{i_k}\}}(g) = \frac{1}{R_N} \min(F_S(g), F_{S^*}(g)) \tag{18}$$

This proves equation (12). Every $\epsilon^{(t)}$ for $t \leq k$ including g is counted by $S^{(k-1)}$, and no other terms from $\epsilon^{(k)}$ will affect g because all the other terms $\epsilon^{(k)}$ should contain at least one sentence that is not in $\{s_{i_1}, \dots, s_{i_k}\}$ and g would not belong to this intersection by definition.

Finally, it has been proved in the appendix C.2 that for $k = 2$, $\epsilon^{(2)}$ has a reduced form:

$$\epsilon^{(2)}(s_a \cap s_b) = \frac{1}{R_N} \sum_{g \in S^*} \max(C_{s_a, S^*}(g) + C_{s_b, S^*}(g) - F_{S^*}(g), 0) \tag{19}$$

In the paper, we ignore the terms for $k \geq 2$. □

C.4 Submodularity of $\tilde{\theta}_R$

Let $S = \{s_i | i \leq m\}$ be a set of sentences and $\tilde{\theta}_R(S)$ its ROUGE-N approximation:

$$\tilde{\theta}_R(S) = \sum_{i=1}^{n} \theta_R(s_i) - \sum_{s_i, s_j \in S, s_i \neq s_j} \tilde{\epsilon}(s_i \cap s_j) \tag{20}$$

To prove the submodularity of $\tilde{\theta}_R$, we prove that $\tilde{\theta}_R$ follows the diminishing returns property: $\forall S \subseteq T$ and a sentence a: $\tilde{\theta}_R(S \cup a) - \tilde{\theta}_R(S) \geq \tilde{\theta}_R(T \cup a) - \tilde{\theta}_R(T)$

Proof. Let $S = \{s_i | i \leq m\}$ and $T = \{t_i | i \leq l\}$ be two sets of sentences such that $S \subseteq T$. We study the following difference:

$$\tilde{\theta}_R(S \cup a) - \tilde{\theta}_R(S) - (\tilde{\theta}_R(T \cup a) - \tilde{\theta}_R(T)) \tag{21}$$

We recall the formula for any set X and any sentence a:

$$\tilde{\theta}_R(X \cup a) = \tilde{\theta}_R(X) + \tilde{\theta}_R(a) - \tilde{\epsilon}(X \cap a) \tag{22}$$

When applied to equation (21) we obtain:

$$\tilde{\epsilon}(T \cap a) - \tilde{\epsilon}(S \cap a) \tag{23}$$

We recall the definition of $\tilde{\epsilon}(X)$ for any set X:

$$\tilde{\epsilon}(X) = \frac{1}{R_N} \sum_{g \in X} \mathbb{1}[freq(g) \geq \alpha] \tag{24}$$

$S \subseteq T$ implies $S \cap a \subseteq T \cap a$. Therefore we can split $\tilde{\epsilon}(T \cap a)$ and the expression (23) becomes:

$$\frac{1}{R_N}\Big(\sum_{g \in S \cap a} \mathbb{1}[freq(g) \geq \alpha] + \sum_{g \in T \cap a \setminus S \cap a} \mathbb{1}[freq(g) \geq \alpha] - \sum_{g \in S \cap a} \mathbb{1}[freq(g) \geq \alpha]\Big) \tag{25}$$

$$= \frac{1}{R_N} \sum_{g \in T \cap a \setminus S \cap b} \mathbb{1}[freq(g) \geq \alpha] \tag{26}$$

This is a sum of positive terms and it is therefore positive. This proves that expression (21) is ≥ 0 which is equivalent to:

$$\tilde{\theta}_R(S \cup a) - \tilde{\theta}_R(S) \geq \tilde{\theta}_R(T \cup a) - \tilde{\theta}_R(T) \tag{27}$$

This concludes the proof of the submodularity of $\tilde{\theta}_R$. □

C.5 Proof of Importance-Encoding Theorem

Let Ω be the set of semantic units. The notation ω_i represents one unit. Let $\mathbb{P}_T$, and $\mathbb{P}_K$ be the text representations of the source documents and background knowledge as probability distributions over semantic units.

We note $t_i = \mathbb{P}_T(\omega_i)$, the probability of the unit ω_i in the source T. Similarly, we note $k_i = \mathbb{P}_K(\omega_i)$. We seek a function f unifying T and K such that: $f(\omega_i) = f(t_i, k_i)$.

We remind the simple requirements that f should satisfy:

- Informativeness: $\forall i \neq j$, if $t_i = t_j$ and $k_i > k_j$ then $f(t_i, k_i) < f(t_j, k_j)$
- Relevance: $\forall i \neq j$, if $t_i > t_j$ and $k_i = k_j$ then $f(t_i, k_i) > f(t_j, k_j)$
- Additivity: $I(f(t_i, k_i)) \equiv \alpha I(t_i) + \beta I(k_i)$ (I is the information measure from Shannon's theory (Shannon, 1948))
- Normalization: $\sum_i f(t_i, k_i) = 1$

Theorem 2 states that the functions satisfying the previous requirements are:

$$\begin{gathered}\mathbb{P}_{\frac{T}{K}}(\omega_i) = \frac{1}{C} \cdot \frac{t_i^\alpha}{k_i^\beta} \\ C = \sum_i \frac{t_i^\alpha}{k_i^\beta} \text{ , } \alpha, \beta \in \mathbb{R}^+\end{gathered} \tag{28}$$

with C the normalizing constant.

Proof. The information function defined by Shannon (1948) is the logarithm: $I = \log$. Then, the *Additivity* criterion can be written:

$$\log(f(t_i, k_i)) = \alpha \log(t_i) + \beta \log(k_i) + A$$

with A a constant independent of t_i and k_i

Since log is monotonous and increasing, the *Informativeness* and *Additivity* criteria can be combined:

$\forall i \neq j$, if $t_i = t_j$ and $k_i > k_j$ then:

$$\begin{gathered}\log f(t_i, k_i) < \log f(t_j, k_j) \\ \alpha \log(t_i) + \beta \log(k_i) + A < \alpha \log(t_j) + \beta \log(k_j) + A \\ \beta \log(k_i) < \beta \log(k_j) \\ \text{But } k_i > k_j \text{, therefore:} \\ \beta < 0\end{gathered}$$

For clarity, we can now use $-\beta$ with $\beta \in \mathbb{R}^+$.

Similarly, we can combine the *Relevance* and *Additivity* criteria: $\forall i \neq j$, if $t_i > t_j$ and $k_i = k_j$ then:

$$\begin{gathered}\log f(t_i, k_i) > \log f(t_j, k_j) \\ \alpha \log(t_i) + \beta \log(k_i) + A > \alpha \log(t_j) + \beta \log(k_j) + A \\ \alpha \log(t_i) > \alpha \log(t_j) \\ \text{But } t_i > t_j \text{, therefore:} \\ \alpha > 0\end{gathered}$$

Then, we have the following form from the *Additivity* criterion:

$$\begin{aligned}\log f(t_i, k_i) &= \alpha \log(t_i) - \beta \log(k_i) + A \\ f(t_i, k_i) &= \exp[\alpha \log(t_i) - \beta \log(k_i)] \cdot \exp(A) \\ f(t_i, k_i) &= \frac{t_i^\alpha}{k_i^\beta} \cdot \exp(A)\end{aligned}$$

Finally, the *Normalization* constraint specifies the constant $\exp(A)$:

$$\begin{aligned} C &= \frac{1}{\exp(A)} \\ \text{and } C &= \sum_i \frac{t_i^\alpha}{k_i^\beta} \\ \text{then: } A &= -\log(\sum_i \frac{t_i^\alpha}{k_i^\beta}) \end{aligned}$$

□

List of Figures

1.1 Illustration of the (θ, O) framework (second line) in comparison to the ($\mathbf{I}$, $\mathbf{T}$, $\mathbf{G}$) perspective (first line). Here the input representation of step $\mathbf{I}$ is a graph but, in practice, it could be any kind of mathematical representation. 3

1.2 Illustration of the (θ, O) framework and some of its consequences. The green summary is the summary being scored. The gold summaries are the human-written reference summaries. Summary scoring functions can be trained and evaluated with human judgments (whether they are evaluation metrics or summarizers' internal scoring functions). Optimizing a θ_{sys} results in extracted summaries. This constitutes a summarizer if θ does not use any evaluation resources. If θ uses evaluation resources, then it is an evaluation metric and optimizing it means computing its upper-bound. 5

3.1 Illustration of evaluation setup for one topic $\mathcal{T}_i$ for $n = 3$. The scores produced by humans and by the summary scoring function are compared with a correlation measure c. These correlations are averaged over all topics. 44

3.2 The figure 3.2a is the fitness vs. time comparison of the various optimization procedure when ROUGE is used as the fitness function; figure 3.2b is the same comparison when JS is used as the fitness function. 56

3.3 Performance improvements when the genetic algorithm is used instead of greedy for TAC-2008 (measured with ROUGE-2). 57

4.1 Toy dataset of human judgments with diversity on the x-axis and JS divergence on the y-axis. Figure 4.1a represents the position of the annotated summaries available in the human judgments of TAC-2008, and figure 4.1b displays θ trained on summaries of TAC-2008 annotated with Pyramid. 81

4.2 Comparison of dataset coverage of the feature space with diversity on the x-axis and JS divergence on the y-axis. Figure 4.2a depicts summaries randomly sampled and their position in the feature space, figure 4.2b represents the position of summaries available in the human judgments of TAC-2008 (same as figure 4.1a put here for comparison) and figure 4.2c the summaries generated according to algorithm 5. . . 83

4.3 Heatmaps associated with θ trained on various datasets. The more red, the higher score is given by θ to this area of the feature space. Figure 4.3a represents θ trained on the randomly generated summaries scored with ROUGE-2, figure 4.3b displays θ trained on summaries of TAC-2008 annotated with Pyramid and figure 4.3c depicts θ trained with summaries generated according to algorithm 5 scored with ROUGE-2. 84
4.4 Heatmap of the combined summary scoring function: $\theta = \theta_h + \alpha \cdot \theta_{R2}$. 84
4.5 Relative importance of features in the unconstrained scenario trained with $JS-2$ as supervision. 89
4.6 Manual evaluation of the main summarization systems for both content selection and diversity (minus redundancy) on a 5-point LIKERT scale. 91

5.1 Possible representation of a text X over a set of semantic units: $\{A, B, \dots L\}$. 97
5.2 Comparison between an example of a redundant summary (figure 5.2a) and an example of a non-redundant one (figure 5.2b). 100
5.3 Compared to the input in blue, figure 5.3a presents an example of a relevant summary (with low cross-entropy); figure 5.3b depicts an example of non-relevant summary (with high cross-entropy). 101
5.4 Figure 5.4a represents an example distribution of sources, figure 5.4b an example distribution of background knowledge and figure 5.4c is the resulting target distribution that summaries should approximate. 106
5.5 Example of $\mathbb{P}_{\frac{D}{K}}$ in comparison to the word distribution of reference summaries for one topic of TAC-2008 (D0803). 109

6.1 The blue distribution represents the score distribution of summaries available in the human judgment datasets of TAC-2008 and TAC-2009. The red distribution is the score distribution of summaries generated by mordern systems. 112
6.2 Scatter matrix plot over all summaries available as part of the human evaluation in TAC-2008 and TAC-2009. 115
6.3 Scatter matrix plot over all summaries available in the generated dataset. 116
6.4 Scatter matrix plot on high-scoring summaries selected with the procedure described by algorithm 7. 118
6.5 Percentage of disagreement between metrics for increasing scores of summary pairs. 118
6.6 On the x axis: the score of the sampled summary s is computed by averaging the score assigned to s by every metric. For meaningful averaging, all metrics have been previously normalized such that 0 is the score of the worst summary and 1 the score of the best summary. We also report the average performance of current systems. On the y axis: $\frac{F}{N}$ associated to the sampled summary s. 119

List of Tables

2.1 Description of existing datasets 17

2.2 Rough classification of existing summarization approaches. The unsupervised approaches typically rely on a notion of *topic frequency* which is to be maximized in the final summary. Statistical learning approaches which use extractive summarization usually learn scores for sub-elements like sentences and extract a set of high-scoring sentences. Recently, end-to-end abstractive summarization simply use a loss function on the final summary to maximize its similarity with the reference. 25

3.1 Correlation of θ functions with human judgments across various systems on TAC-2008 and TAC-2009. 46

3.2 Evaluation summaries extracted by $O(\sigma, \cdot)$ for several systems TAC-2008 and TAC-2009. 46

3.3 The average improvement in extracted summaries observed when switching form Greedy to another optimization algorithm (reported in pp). The average is taken over every θ's and both datasets (TAC-2008 and TAC-2009). 58

3.4 Comparison of three methods to compute the JS-Eval-1 upper-bound on TAC-2008 and TAC-2009: a) computing the ROUGE-2 upper-bound, b) JS-Eval-1 optimized with the Greedy algorithm and c) JS-Eval-1 optimized with AUB described in algorithm 4. (All approaches are scored with JS-Eval-1) 60

3.5 Comparison of three methods to compute the JS-Eval-2 upper-bound on TAC-2008 and TAC-2009: a) computing the ROUGE-2 upper-bound, b) JS-Eval-2 optimized with the Greedy algorithm and c) JS-Eval-2 optimized with AUB described in algorithm 4. (All approaches are scored with JS-Eval-2) 61

3.6 Comparison of three methods to compute the PEAK upper-bound on TAC-2009: a) computing the ROUGE-2 upper-bound, b) PEAK optimized with the Greedy algorithm and c) PEAK optimized with AUB described in algorithm 4. (All approaches are scored with PEAK using $t = .6$ (PEAK-60) and $t = .7$ (PEAK-70)) 63

4.1 Summary of the two main axes of variations: i) the supervision signal which may come from various approximation of humans up until actually using human judgment datasets as target scores. ii) the constraints imposed on θ: either linear or no constraint. The top left corner is where most previous works lie. For the second row, non-linear features become available. 67
4.2 Upper-bound of the proposed framework compared to the true extractive upper-bound. 78
4.3 Notations used for various trained summary scoring functions. The functions from the first lign rely on the linear features described in section 4.1 : Φ_{lin}. On the second lign, the features used are both the linear and non linear ones: Φ. On the last line, reference summaries are available and the features used are all the previous ones and existing evaluation metrics: Φ_{eval} 87
4.4 Correlation of θ functions with human judgments across various systems on TAC-2008 and TAC-2009. 88
4.5 Correlation of automatic metrics with humans across the whole scoring spectrum of θ_{Pyr}^{Eval} 90
4.6 Comparison of systems across 3 datasets evaluated with ROUGE-2 recall, JS divergence on bigrams, S3 and Human annotations. 93

5.1 Correlation of various information-theoretic quantities with human judgments measured by Kendall's τ on generic and update summarization. 108

A.1 Description of the datasets used during the thesis 128

Bibliography

Christoph Adami. 2012. The Use of Information Theory in Evolutionary Biology. *Annals of the New York Academy of Sciences*, 1256(1):49–65.

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. 2017. Text Summarization Techniques: A Brief Survey. *International Journal of Advanced Computer Science and Applications*, 8(10).

Einat Amitay and Cécile Paris. 2000. Automatically Summarising Web Sites: Is There a Way Around It? In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pages 173–179.

Chinatsu Aone, Mary Ellen Okurowski, James Gorlinsky, and Bjornar Larsen. 1995. A Trainable Summarizer with Knowledge Acquired from Robust NLP Techniques. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 68–73. MIT Press, Cambridge, MA, USA.

Ayana, Shiqi Shen, Zhiyuan Liu, and Maosong Sun. 2016. Neural Headline Generation with Minimum Risk Training. *CoRR*, abs/1604.01904.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document Abstractive Summarization Using ILP Based Multi-sentence Compression. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1208–1214.

Jie Bao, Prithwish Basu, Mike Dean, Craig Partridge, Ananthram Swami, Will Leland, and James A Hendler. 2011. Towards a Theory of Semantic Communication. In *Network Science Workshop (NSW), 2011 IEEE*, pages 110–117.

Regina Barzilay and Michael Elhadad. 1999. Using Lexical Chains for Text Summarization. *Advances in Automatic Text Summarization*, pages 111–121.

Regina Barzilay and Noemie Elhadad. 2002. Inferring Strategies for Sentence Ordering in Multidocument News Summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Regina Barzilay and Mirella Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics*, 34(1):1–34.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence Fusion for Multidocument News Summarization. *Computational Linguistics*, 31(3):297–328.

Gerardo Beni and Jing Wang. 1993. Swarm Intelligence in Cellular Robotic Systems. In *Robots and Biological Systems: Towards a New Bionics?*, pages 703–712, Berlin, Heidelberg. Springer.

Darina Benikova, Margot Mieskes, Christian M. Meyer, and Iryna Gurevych. 2016. Bridging the Gap Between Extractive and Abstractive Summaries: Creation and Evaluation of Coherent Extracts from Heterogeneous Sources. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pages 1039–1050.

Leonora Bianchi, Marco Dorigo, Luca Maria Gambardella, and Walter J. Gutjahr. 2009. A Survey on Metaheuristics for Stochastic Combinatorial Optimization. *Natural Computing*, 8(2):239–287.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J. Passonneau. 2015. Abstractive Multi-Document Summarization via Phrase Selection and Merging. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 1587–1597, Beijing, China. Association for Computational Linguistics.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

David M Blei. 2012. Probabilistic Topic Models. *Communications of the ACM*, 55(4):77–84.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Christian Blum and Andrea Roli. 2003. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 35(3):268–308.

Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. 1999. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Inc., New York, NY, USA.

Florian Boudin, Hugo Mougard, and Benoît Favre. 2015. Concept-based Summarization using Integer Linear Programming: From Concept Pruning to Multiple Optimal Solutions. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1914–1918, Lisbon, Portugal.

Jill Burstein and Daniel Marcu. 2000. Toward Using Text Summarization for Essay-Based Feedback. In *Encyclopedia of Library and Information Science*, pages 245–256.

Ziqiang Cao, Chengyao Chen, Wenjie Li, Sujian Li, Furu Wei, and Ming Zhou. 2016a. TGSum: Build Tweet Guided Multi-document Summarization Dataset. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2906–2912.

Ziqiang Cao, Wenjie Li, Sujian Li, Furu Wei, and Yanran Li. 2016b. Attsum: Joint learning of focusing and summarization with neural attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 547–556, Osaka, Japan. The COLING 2016 Organizing Committee.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015a. Ranking with Recursive Neural Networks and Its Application to Multi-document Summarization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2153–2159.

Ziqiang Cao, Furu Wei, Sujian Li, Wenjie Li, Ming Zhou, and WANG Houfeng. 2015b. Learning Summary Prior Representation for Extractive Summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 2, pages 829–833.

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

Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. 2007. Summarizing Email Conversations with Clue Words. In *Proceedings of the 16th International Conference on World Wide Web*, pages 91–100.

Rudolf Carnap and Yehoshua Bar-Hillel. 1953. An Outline of a Theory of Semantic Information. *British Journal for the Philosophy of Science.*, 4.

Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. A Hybrid Hierarchical Model for Multi-Document Summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 815–824, Uppsala, Sweden. Association for Computational Linguistics.

Yllias Chali and Shafiq R. Joty. 2008. Improving the performance of the random walk model for answering complex questions. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 9–12. Association for Computational Linguistics.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367. Association for Computational Linguistics.

Jianpeng Cheng and Mirella Lapata. 2016. Neural Summarization by Extracting Sentences and Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494. Association for Computational Linguistics.

Jackie Chi Kit Cheung and Gerald Penn. 2013. Towards Robust Abstractive Multi-Document Summarization: A Caseframe Analysis of Centrality and Domain. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1233–1242, Sofia, Bulgaria. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98. Association for Computational Linguistics.

Wei Chu and Zoubin Ghahramani. 2005. Preference Learning with Gaussian Processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 137–144. ACM.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

John M. Conroy and Hoa Trang Dang. 2008. Mind the Gap: Dangers of Divorcing Evaluations of Summary Content from Linguistic Quality. In *Proceedings of the 22Nd International Conference on Computational Linguistics (COLING)*, volume 1, pages 145–152.

John M. Conroy and Dianne P. O'leary. 2001. Text Summarization via Hidden Markov Models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 406–407. ACM.

John M. Conroy, Judith D. Schlesinger, Jeff Kubina, Peter A. Rankel, and Dianne P. O'Leary. 2011. CLASSY 2011 at TAC: Guided and Multi-lingual Summaries and Evaluation Metrics. *Proceedings of the text analysing conference, (TAC 2011)*, 11:1–8.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2006. Topic-Focused Multi-Document Summarization Using an Approximate Oracle Score. In *Proceedings of the International Conference on Computational Linguistics and the annual meeting of the Association for Computational Linguistics*, pages 152–159.

John M. Conroy, Judith D. Schlesinger, Peter A. Rankel, and Dianne P. O'Leary. 2010. Guiding CLASSY Toward More Responsive Summaries. In *Proceedings of the text analysing conference, (TAC 2010)*.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms, Third Edition*, 3rd edition. The MIT Press.

D.A. Cruse. 1986. *Lexical Semantics*. Cambridge University Press, Cambridge, UK.

Hoa Trang Dang. 2005. Overview of DUC 2005. In *Proceedings of the Document Understanding Conference (DUC 2005)*, volume 2005, pages 1–12.

Hoa Trang Dang. 2006. Overview of DUC 2006. In *Proceedings of the Document Understanding Conference (DUC 2006)*, volume 2005, pages 1–12.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 Update Summarization Task. In *Proceedings of the First Text Analysis Conference (TAC 2008)*, pages 1–16.

Hoa Trang Dang and Karolina Owczarzak. 2009. Overview of the TAC 2009 Summarization Track. In *Proceedings of the First Text Analysis Conference (TAC 2009)*, pages 1–12.

Dipanjan Das and André F. T. Martins. 2010. A Survey on Automatic Text Summarization. *Literature Survey for the Language and Statistics II Course at CMU*.

Hal Daumé III and Daniel Marcu. 2006. Bayesian Query-Focused Summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312. Association for Computational Linguistics.

Sashka T. Davis, John M. Conroy, and Judith D. Schlesinger. 2012. OCCAMS–An Optimal Combinatorial Covering Algorithm for Multi-document Summarization. In *Proceeding of the 12th International Conference on Data Mining Workshops (ICDMW)*, pages 454–463. IEEE.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Luciano Del Corro and Rainer Gemulla. 2013. ClausIE: Clause-based Open Information Extraction. In *Proceedings of the 22Nd International Conference on World Wide Web*, pages 355–366, Rio de Janeiro, Brazil. ACM.

Jean-Yves Delort and Enrique Alfonseca. 2012. DualSum: A Topic-model Based Approach for Update Summarization. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 214–223.

Jean-Yves Delort, Bernadette Bouchon-Meunier, and Maria Rifqi. 2003. Enhanced Web Document Summarization Using Hyperlinks. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, pages 208–215.

Robert L. Donaway, Kevin W. Drummey, and Laura A. Mather. 2000. A Comparison of Rankings Produced by Summarization Evaluation Measures. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, pages 69–78.

Michael J. Dunn. 1976. Intuitive Semantics for First-Degree Entailments and 'Coupled Trees'. *Philosophical studies*, 29(3):149–168.

Ted Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational linguistics*, 19(1):61–74.

H. P. Edmundson. 1969. New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285.

Noemie Elhadad, Man-Yee Kan, Judith L. Klavans, and Kathleen R. McKeown. 2005. Customization in a unified framework for summarizing medical literature. *Artificial Intelligence in Medicine*, 33(2):179–198.

Katrin Erk. 2010. What is Word Meaning, Really? (and How Can Distributional Models Help Us Describe It?). In *Proceedings of the 2010 workshop on geometrical models of natural language semantics*, pages 17–26. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based Lexical Centrality As Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Tobias Falke and Iryna Gurevych. 2017. Bringing Structure into Summaries: Crowdsourcing a Benchmark Corpus of Concept Maps. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2951–2961. Association for Computational Linguistics.

Yimai Fang and Simone Teufel. 2016. Improving Argument Overlap for Proposition-Based Summarisation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 479–485. Association for Computational Linguistics.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. A Formal Model for Information Selection in Multi-sentence Text Extraction. In *Proceedings of the 20th international Conference on Computational Linguistics*, pages 397–403. Association for Computational Linguistics.

Katja Filippova. 2010a. Multi-sentence Compression: Finding Shortest Paths in Word Graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 322–330.

Katja Filippova. 2010b. Multi-Sentence Compression: Finding Shortest Paths in Word Graphs.

Katja Filippova and Michael Strube. 2008. Sentence Fusion via Dependency Graph Compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 177–185.

Katja Filippova, Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2009. Company-oriented Extractive Summarization of Financial News. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 246–254.

Charles J. Fillmore. 1976. Frame Semantics And the Nature of Language. *Annals of the New York Academy of Sciences*, 280(1):20–32.

Alessandro Fiori. 2014. *Innovative Document Summarization Techniques: Revolutionizing Knowledge Understanding*. Hershey, PA : Information Science Reference, Cambridge, MA, USA.

Luciano Floridi. 2009. Philosophical Conceptions of Information. In *Formal Theories of Information*, pages 13–53. Springer.

Maria Fuentes, Enrique Alfonseca, and Horacio Rodríguez. 2007. Support Vector Machines for Query-Focused Summarization Trained and Evaluated on Pyramid Data. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 57–60. Association for Computational Linguistics.

Pascale Fung and Grace Ngai. 2006. One Story, One Flow: Hidden Markov Story Models for Multilingual Multi-document Summarization. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(2):1–16.

Johannes Fürnkranz and Eyke Hüllermeier. 2003. Pairwise preference learning and ranking. In *Proceedings of the 14th European Conference on Machine Learning*, pages 145–156. Springer-Verlag.

Kata Gábor, Haifa Zargayouna, Isabelle Tellier, Davide Buscaldi, and Thierry Charnois. 2017. Exploring Vector Spaces for Semantic Relations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1814–1823, Copenhagen, Denmark. Association for Computational Linguistics.

Michel Galley and Kathleen R. McKeown. 2003. Improving Word Sense Disambiguation in Lexical Chaining. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1486–1488.

Michel Galley and Kathleen R. McKeown. 2007. Lexicalized Markov Grammars for Sentence Compression. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 180–187.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348.

Yanjun Gao, Andrew Warner, and Rebecca J. Passonneau. 2018. Pyreval: An automated method for summary content analysis. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*.

George Giannakopoulos. 2013. Multi-Document Multilingual Summarization and Evaluation Tracks in ACL 2013 MultiLing Workshop. pages 20–28.

George Giannakopoulos, John M. Conroy, Jeff Kubina, Peter A. Rankel, Elena Lloret, Josef Steinberger, Marina Litvak, and Benoît Favre. 2017. MultiLing 2017 Overview. In *Proceedings of the Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 1–6.

George Giannakopoulos, Mahmoud El-Haj, Benoit Favre, Marianna Litvak, Josef Steinberger, and Vasudeva Varma. 2011. TAC 2011 MultiLing pilot overview.

George Giannakopoulos and Vangelis Karkaletsis. 2011. AutoSummENG and MeMoG in evaluating guided summaries. In *Proceedings of the text analysing conference, (TAC 2011).*

George Giannakopoulos and Vangelis Karkaletsis. 2013. Together We Stand NPowered. In *Proceedings of International Conference on Computational Linguistics and Intelligent Text Processing.*

George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. 2008. Summarization System Evaluation Revisited: N-gram Graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5:5–10.

George Giannakopoulos, Jeff Kubina, John M. Conroy, Josef Steinberger, Benoît Favre, Mijail A. Kabadjov, Udo Kruschwitz, and Massimo Poesio. 2015. MultiLing 2015: Multilingual Summarization of Single and Multi-Documents, On-line Fora, and Call-center Conversations. In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 270–274.

Dan Gillick and Benoit Favre. 2009. A Scalable Global Model for Summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 10–18, Boulder, Colorado.

David E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edition. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Jamie Callan. 2000. Creating and Evaluating Multi-document Sentence Extract Summaries. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pages 165–172.

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

Yvette Graham. 2015. Re-evaluating Automatic Summarization with BLEU and 192 Shades of ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 128–137. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. NEWSROOM: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1631–1640. Association for Computational Linguistics.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the Unknown Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 140–149, Berlin, Germany. Association for Computational Linguistics.

Gregory Gutin, Anders Yeo, and Alexey Zverovich. 2002. Traveling Salesman Should Not Be Greedy: Domination Analysis of Greedy-type Heuristics for the TSP. *Discrete Applied Mathematics*, 117(1):81–86.

Ben Hachey, Gabriel Murray, and David Reitter. 2006. Dimensionality Reduction Aids Term Co-Occurrence Based Multi-Document Summarization. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 1–7. Association for Computational Linguistics.

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

Dilek Hakkani-Tur and Gokhan Tur. 2007. Statistical Sentence Extraction for Information Distillation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 1–4. IEEE.

Hans van Halteren and Simone 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, pages 57–64.

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

Aaron Harnly, Rebecca J. Passonneau, and Owen Rambow. 2005. Automation of Summary Evaluation by the Pyramid Method. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 226–232, Borovets, Bulgaria.

Zellig Harris. 1954. Distributional Structure. *Word*, 10(23):146–162.

Laura Hasler, Constantin Orasan, and Ruslan Mitkov. 2003. Building Better Corpora for Summarization. In *Proceedings of Corpus Linguistics*, pages 309–319.

Tingting He, Jinguang Chen, Liang Ma, Zhuoming Gui, Fang Li, Wei Shao, and Qian Wang. 2008. ROUGE-C: A Fully Automated Evaluation Method for Multi-document Summarization. In *IEEE International Conference on Granular Computing*, pages 269–274.

Zhanying He, Chun Chen, Jiajun Bu, Can Wang, Lijun Zhang, Deng Cai, and Xiaofei He. 2012. Document Summarization Based on Data Reconstruction. In *Proceeding of the Twenty-Sixth Conference on Artificial Intelligence*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems 28*, pages 1693–1701.

Tsutomu Hirao, Manabu Okumura, Norihito Yasuda, and Hideki Isozaki. 2007. Supervised Automatic Evaluation for Summarization with Voted Regression Model. *Information Processing and Management*, 43(6):1521–1535.

Graeme Hirst, David St-Onge, et al. 1998. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. *WordNet: An Electronic Lexical Database*, 305:305–332.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kai Hong, John M. Conroy, benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1608–1616.

Kai Hong and Ani Nenkova. 2014. Improving the Estimation of Word Importance for News Multi-Document Summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721, Gothenburg, Sweden. Association for Computational Linguistics.

Eduard Hovy and Chin-Yew Lin. 1999. Automated Text Summarization and the SUMMARIST System. *Advances in Automatic Text Summarization*, pages 82–94.

Eduard Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. 2006. Automated summarization evaluation with basic elements. In *Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC 2006)*, pages 604–611.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale chinese short text summarization dataset. *CoRR*, abs/1506.05865.

Meishan Hu, Aixin Sun, and Ee-Peng Lim. 2007. Comments-oriented Blog Summarization by Sentence Extraction. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, pages 901–904.

Edwin T. Jaynes. 1957. Information Theory and Statistical Mechanics. *Physical Review*, 106:620–630.

Paul Ji. 2006. Multi-document Summarization Based on Unsupervised Clustering. In *Proceedings of the Third Asia Conference on Information Retrieval Technology*, pages 560–566.

Hongyan Jing. 2000. Sentence Reduction for Automatic Text Summarization. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 310–315. Association for Computational Linguistics.

Hongyan Jing, Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1998. Summarization evaluation methods: Experiments and analysis. In *AAAI symposium on intelligent summarization*, pages 51–59.

Hongyan Jing and Kathleen R. McKeown. 1999. The Decomposition of Human-Written Summary Sentences. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 129–136. ACM.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and Paste Based Text Summarization. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, pages 178–185.

Karen Sparck Jones and Julia R. Galliers. 1995. *Evaluating Natural Language Processing Systems: An Analysis and Review*, volume 1083. Springer Science & Business Media.

Mikael Kågebäck, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. 2014. Extractive Summarization Using Continuous Vector Space Models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 31–39.

Dervis Karaboga and Bahriye Basturk. 2007. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *Journal of Global Optimization*, 39(3):459–471.

Dervis Karaboga, Beyza Gorkemli, Celal Ozturk, and Nurhan Karaboga. 2014. A Comprehensive Survey: Artificial Bee Colony (ABC) Algorithm and Applications. *Artificial Intelligence Review*, 42(1):21–57.

Chris Kedzie, Fernando Diaz, and Kathleen R. McKeown. 2016. Real-Time Web Scale Event Summarization Using Sequential Decision Making. pages 3754–3760.

Chris Kedzie, Kathleen McKeown, and Hal Daume III. 2018. Content Selection in Deep Learning Models of Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828. Association for Computational Linguistics.

Atif Khan, Naomie Salim, and Haleem Farman. 2016. Clustered genetic semantic graph approach for multi-document abstractive summarization. In *International Conference on Intelligent Systems Engineering (ICISE)*, pages 63–70.

Khaled Khelif, Rose Dieng, and Pascal Barbry. 2007. An Ontology-based Approach to Support Text Mining and Information Retrieval in the Biological Domain. *Journal of Universal Computer Science*, 13:1881–1907.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling Output Length in Neural Encoder-Decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

Walter Kintsch and Teun A. Van Dijk. 1978. Toward a Model of Text Comprehension and Production. *Psychological Review*, pages 363–394.

Scott Kirkpatrick, Daniel C. Gelatt, and Mario P. Vecchi. 1983. Optimization by Simulated Annealing. *science*, 220(4598):671–680.

Kevin Knight and Daniel Marcu. 2000. Statistics-based Summarization-step One: Sentence Compression. *AAAI/IAAI*, 2000:703–710.

Kevin Knight and Daniel Marcu. 2002. Summarization Beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. *Artificial Intelligence*, 139(1):91–107.

Youngjoong Ko and Jungyun Seo. 2008. An Effective Sentence-Extraction Technique using Contextual Information and Statistical Approaches for Text Summarization. *Pattern Recognition Letters*, 29(9):1366–1371.

Hayato Kobayashi, Masaki Noguchi, and Taichi Yatsuka. 2015. Summarization Based on Embedding Distributions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1984–1989, Lisbon, Portugal. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Andreas Krause and Daniel Golovin. 2014. Submodular Function Maximization. *Tractability: Practical Approaches to Hard Problems*, pages 71–104.

Thomas S. Kuhn. 1970. *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago.

Alex Kulesza, Ben Taskar, et al. 2012. Determinantal Point Processes for Machine Learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

Solomon Kullback and Richard A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A Trainable Document Summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, Seattle, Washington, USA. Association for Computing Machinery.

Mirella Lapata and Regina Barzilay. 2005. Automatic Evaluation of Text Coherence: Models and Representations. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 1085–1090.

Jure Leskovec, Natasa Milic-Frayling, and Marko Grobelnik. 2005. Impact of Linguistic Analysis on the Semantic Graph Coverage and Learning of Document Extracts. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1069–1074.

Chen Li, Xian Qian, and Yang Liu. 2013. Using Supervised Bigram-based ILP for Extractive Summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1004–1013, Sofia, Bulgaria.

Ke Li and Jitendra Malik. 2016. Learning to Optimize. *CoRR*, abs/1606.01885.

Liangda Li, Ke Zhou, Gui-Rong Xue, Hongyuan Zha, and Yong Yu. 2009. Enhancing Diversity, Coverage and Balance for Summarization Through Structure Learning. In *Proceedings of the 18th International Conference on World Wide Web*, pages 71–80.

Piji Li, Lidong Bing, and Wai Lam. 2017. Reader-Aware Multi-Document Summarization: An Enhanced Model and The First Dataset. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 91–99. Association for Computational Linguistics.

Piji Li, Lidong Bing, Wai Lam, Hang Li, and Yi Liao. 2015. Reader-Aware Multi-document Summarization via Sparse Coding. In *Proceedings of the 24th International Conference on Artificial Intelligence* , pages 1270–1276.

Chin-Yew Lin. 2004a. Looking for a Few Good Metrics: Automatic Summarization Evaluation - How Many Samples Are Enough? In *Proceedings of the NII Testbeds and Community Information access Research (NTCIR 2004)*.

Chin-Yew Lin. 2004b. ROUGE: A Package for Automatic Evaluation of summaries. In *Proceedings of ACL workshop on Text Summarization Branches Out*, pages 74–81, Barcelona, Spain.

Chin-Yew Lin, Guihong Cao, Jianfeng Gao, and Jian-Yun Nie. 2006. An Information-Theoretic Approach to Automatic Evaluation of Summaries. In *Proceedings of the Human Language Technology Conference at NAACL*, pages 463–470, New York City, USA.

Chin-Yew Lin and Eduard Hovy. 2000. The Automated Acquisition of Topic Signatures for Text Summarization. In *Proceedings of the 18th conference on Computational linguistics*, volume 1, pages 495–501.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and Automatic Evaluation of Summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4*, pages 45–51.

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

Hui Lin and Jeff A. Bilmes. 2011. A Class of Submodular Functions for Document Summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 510–520, Portland, Oregon.

Jimmy Lin, Richard McCreadie, Ellen Voorhees, and Fernando Diaz. 2016. Overview of the TREC 2016 Real-Time Summarization Track. In *Proceedings of The Twenty-Fifth Text REtrieval Conference (TREC).*

Jimmy Lin, Mohammed Salman, Royal Sequiera, Luchen Tan, Nimesh Ghelani, Mustafa Abualsaud, Richard McCreadie, Dmitrijs Milajevs, and Ellen Voorhees. 2017. Overview of the TREC 2017 Real-Time Summarization Track. In *Proceedings of The Twenty-Sixth Text REtrieval Conference (TREC).*

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015a. Toward Abstractive Summarization Using Semantic Representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

He Liu, Hongliang Yu, and Zhi-Hong Deng. 2015b. Multi-document Summarization Based on Two-level Sparse Representation Model. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 196–202.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv:1801.10198 [cs].*

Elena Lloret, Laura Plaza, and Ahmet Aker. 2013. Analyzing the Capabilities of Crowdsourcing Services for Text Summarization. *Language Resources and Evaluation*, 47(2):337–369.

Elena Lloret, Laura Plaza, and Ahmet Aker. 2018. The Challenging Task of Summary Evaluation: An Overview. *Language Resources and Evaluation*, 52(1):101–148.

Elena Lloret and Manuel Sanz. 2013. Towards Automatic Tweet Generation: A Comparative Study from the Text Summarization Perspective in the Journalism Genre. *Expert Systems with Applications: An International Journal*, pages 6624–6630.

Manuel J. Maña López, Manuel De Buenaga, and José M. Gómez-Hidalgo. 2004. Multidocument Summarization: An Added Value to Clustering in Interactive Retrieval. *ACM Transactions on Information Systems*, 22(2):215–241.

Annie Louis. 2014. A Bayesian Method to Incorporate Background Knowledge during Automatic Text Summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 333–338, Baltimore, Maryland.

Annie Louis and Ani Nenkova. 2008. Automatic Summary Evaluation without Human Models. In *Proceedings of the text analysing conference, (TAC 2008).*

Annie Louis and Ani Nenkova. 2013. Automatically Assessing Machine Summary Content Without a Gold Standard. *Computational Linguistics*, 39(2):267–300.

Claude de Loupy, Marie Guégan, Christelle Ayache, Somara Seng, and Juan-Manuel Torres Moreno. 2010. A French Human Reference Corpus for Multi-Document Summarization and Sentence Compression. In *Proceedings of the Seventh conference on International Language Resources and Evaluation.*

Hans Peter Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research Development*, 2:159–165.

Shulei Ma, Zhi-Hong Deng, and Yunlun Yang. 2016. An Unsupervised Multi-Document Summarization Framework Based on Neural Document Model. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1514–1523. The COLING 2016 Organizing Committee.

Esfandiar Maasoumi. 1993. A Compendium to Information Theory in Economics and Econometrics. *Econometric Reviews*, 12(2):137–181.

Inderjeet Mani. 1999. *Advances in Automatic Text Summarization.* MIT Press, Cambridge, MA, USA.

Inderjeet Mani and Eric Bloedorn. 1997. Multi-document Summarization by Graph Search and Matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, pages 622–628, Providence, Rhode Island. AAAI Press.

Inderjeet Mani and Eric Bloedorn. 1999. Summarizing Similarities and Differences among Related Documents. *Information Retrieval*, 1(1-2):35–67.

Inderjeet Mani, Gary Klein, David House, Lynette Hirschman, Therese Firmin, and Beth Sundheim. 2002. SUMMAC: A Text Summarization Evaluation. *Natural Language Engineering*, 8(1):43–68.

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

Daniel Marcu. 1997. From Discourse Structures to Text Summaries. *Intelligent Scalable Text Summarization.*

Daniel Marcu. 1998. To Build Text Summaries of High Quality, Nuclearity Is Not Sufficient. In *Working Notes of the AAAI-98 Spring Symposium on Intelligent Text Summarization*, pages 1–8.

Erwin Marsi and Emiel Krahmer. 2005. Explorations in Sentence Fusion. In *Proceedingsof the European Workshop on Natural Language Generation*, pages 109–117.

Iain McCowan, Jean Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, et al. 2005. The AMI meeting corpus. In *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, volume 88.

Ryan McDonald. 2007. A Study of Global Inference Algorithms in Multi-document Summarization. In *Proceedings of the 29th European Conference on IR Research*, pages 557–564, Rome, Italy. Springer-Verlag.

Kathleen R. McKeown, Judith L. Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards Multidocument Summarization by Reformulation: Progress and Prospects. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, pages 453–460.

Kathleen R. Mckeown, Rebecca J. Passonneau, David K. Elson, Ani Nenkova, and Julia Hirschberg. 2005. Do Summaries Help? A Task-Based Evaluation of Multi-Document Summarization. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval.*

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

Qiaozhu Mei and ChengXiang Zhai. 2008. Generating Impact-Based Summaries for Scientific Literature. In *Proceedings of ACL-08: HLT*, pages 816–824. Association for Computational Linguistics.

Donald Metzler and Tapas Kanungo. 2008. Machine Learned Sentence Selection Strategies for Query-Based Summarization. In *SIGIR Learning to Rank Workshop*, pages 40–47.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing.*

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, Nevada, USA.

George Aa Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–244.

Richard Montague. 1970. English as a Formal Language. In Bruno Visentini, editor, *Linguaggi nella societa e nella tecnica*, pages 188–221. Edizioni di Communita.

Gabriel Murray, Steve Renals, and Jean Carletta. 2005. Extractive Summarization of Meeting Recordings. In *Proceedings of 9th European Conference on Speech Communication and Technology*, pages 593–596.

Masahiro Nakano, Hideyuki Shibuki, Rintaro Miyazaki, Madoka Ishioroshi, Koichi Kaneko, and Tatsunori Mori. 2010. Construction of Text Summarization Corpus for the Credibility of Information on the Web. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *AAAI*, pages 3075–3081.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. Association for Computational Linguistics.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100.

George L. Nemhauser and Laurence A. Wolsey. 1978. Best Algorithms for Approximating the Maximum of a Submodular Set Function. *Mathematics of Operations Research*, 3(3):177–188.

Ani Nenkova. 2006. *Understanding the Process of Multi-document Summarization: Content Selection, Rewriting and Evaluation.* Ph.D. thesis, New York, NY, USA. AAI3203761.

Ani. Nenkova and Amit Bagga. 2003. Facilitating email thread access by extractive summary generation. In *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, pages 287–296.

Ani Nenkova and Kathleen R. McKeown. 2011. Automatic Summarization. *Foundations and Trends in Information Retrieval*, 5(2–3):103–233.

Ani Nenkova and Kathleen R. McKeown. 2012. *A Survey of Text Summarization Techniques*, pages 43–76. Springer US, Boston, MA.

Ani Nenkova and Rebecca J. Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the 2004 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 145–152. Association for Computational Linguistics.

Ani Nenkova, Rebecca J. Passonneau, and Kathleen McKeown. 2007. The Pyramid Method: Incorporating Human Content Selection Variation in Summarization Evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(2).

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors That Influence Summarization. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 573–580.

Paula S. Newman and John C. Blitzer. 2003. Summarizing Archived Discussions: A Beginning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 273–276.

Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 663–670.

Jun-Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for rouge. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, Lisbon, Portugal. Association for Computational Linguistics.

Hitoshi Nishikawa, Kazuho Arita, Katsumi Tanaka, Tsutomu Hirao, Toshiro Makino, and Yoshihiro Matsuo. 2014. Learning to Generate Coherent Summary with Discriminative Hidden Semi-Markov Model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1648–1659.

Kenji Ono, Kazuo Sumita, and Seiji Miike. 1994. Abstract Generation Based on Rhetorical Structure Extraction. In *Proceedings of the 15th Conference on Computational Linguistics*, volume 1, pages 344–348.

Antonio Ortega and Kannan Ramchandran. 1998. Rate-distortion methods for image and video compression. *IEEE signal processing magazine*, 15(6):23–50.

Miles Osborne. 2002. Using Maximum Entropy for Sentence Extraction. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, volume 4, pages 1–8. Association for Computational Linguistics.

Paul Over. 2003. An introduction to DUC 2003: Intrinsic Evaluation of Generic News Text Summarization Systems. In *Proceedings of the Document Understanding Conference (DUC 2003)*.

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An Assessment of the Accuracy of Automatic Evaluation in Summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9. Association for Computational Linguistics.

Karolina Owczarzak and Hoa Trang Dang. 2010. Overview of the TAC 2010 Summarization Track. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*.

Karolina Owczarzak and Hoa Trang Dang. 2011. Overview of the TAC 2011 summarization track: Guided task and AESOP task. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)*.

Christos H. Papadimitriou and Kenneth Steiglitz. 1982. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Konstantinos E. Parsopoulos and Michael N. Vrahatis. 2002. Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural Computing*, 1(2):235–306.

Rebecca J. Passonneau, Ani Nenkova, Kathleen McKeown, and Sergey Sigelman. 2005. Applying the pyramid method in DUC 2005. In *Proceedings of the Document Understanding Conference (DUC 05)*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A Deep Reinforced Model for Abstractive Summarization. *CoRR abs/1705.04304*.

F. Pedregosa, 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*, 12:2825–2830.

Maxime Peyrard. 2019a. A Simple Theoretical Model of Importance for Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1059–1073, Florence, Italy. Association for Computational Linguistics.

Maxime Peyrard. 2019b. Studying Summarization Evaluation Metrics in the Appropriate Scoring Range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 5093–5100, Florence, Italy. Association for Computational Linguistics.

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. 2017. Learning to Score System Summaries for Better Content Selection Evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84.

Maxime Peyrard and Judith Eckle-Kohler. 2016a. A General Optimization Framework for Multi-Document Summarization Using Genetic Algorithms and Swarm Intelligence. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pages 247 – 257.

Maxime Peyrard and Judith Eckle-Kohler. 2016b. Optimizing an Approximation of ROUGE - a Problem-Reduction Approach to Extractive Multi-Document Summarization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1836, Berlin, Germany. Association for Computational Linguistics.

Maxime Peyrard and Judith Eckle-Kohler. 2017a. A principled framework for evaluating summarizers: Comparing models of summary quality against human judgments. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, volume Volume 2: Short Papers, pages 26–31. Association for Computational Linguistics.

Maxime Peyrard and Judith Eckle-Kohler. 2017b. Supervised learning of automatic pyramid for optimization-based multi-document summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, volume Volume 1: Long Papers, pages 1084–1094. Association for Computational Linguistics.

Maxime Peyrard and Iryna Gurevych. 2018. Objective function learning to match human judgements for optimization-based summarization. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 654–660. Association for Computational Linguistics.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1341–1351, Sofia, Bulgaria. Association for Computational Linguistics.

Avinesh P.V.S. and Christian M. Meyer. 2017. Joint Optimization of User-desired Content in Multi-document Summaries by Learning from User Feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, volume Volume 1: Long Paper, pages 1353–1363. Association for Computational Linguistics.

Avinesh P.V.S., Maxime Peyrard, and Christian M. Meyer. 2018. Live Blog Corpus for Summarization. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, pages 3197–3203.

Vahed Qazvinian and Dragomir R. Radev. 2008. Scientific Paper Summarization Using Citation Summary Networks. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, pages 689–696.

Dragomir R. Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, et al. 2004. MEAD-A Platform for Multidocument Multilingual Text Summarization. In *Proceedings of the Inter- national Conference on Language Resources and Evaluation.*

Dragomir R. Radev, Eduard Hovy, and Kathleen R. McKeown. 2002. Introduction to the Special Issue on Summarization. *Computational Linguistics*, (4):399–408.

Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based Summarization of Multiple Documents: Sentence Extraction, Utility-based Evaluation, and User Studies. In *Proceedings of the NAACL-ANLP Workshop on Automatic Summarization*, volume 4, pages 21–30, Seattle, Washington.

Dragomir R Radev and Daniel Tam. 2003. Summarization Evaluation Using Relative Utility. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 508–511.

Dragomir R. Radev, Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Hong Qi, Arda Çelebi, Danyu Liu, and Elliott Drabek. 2003. Evaluation Challenges in Large-scale Document Summarization. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 375–382.

Peter A. Rankel, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2013. A Decade of Automatic Content Evaluation of News Summaries: Reassessing the State of the Art. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 131–136, Sofia, Bulgaria. Association for Computational Linguistics.

Peter A. Rankel, John M. Conroy, and Judith D. Schlesinger. 2012. Better Metrics to Automatically Predict the Quality of a Text Summary. *Algorithms*, 5(4):398–420.

Pengjie Ren, Furu Wei, Zhumin Chen, Jun Ma, and Ming Zhou. 2016. A Redundancy-Aware Sentence Regression Framework for Extractive Summarization. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 33–43.

Philip Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of international joint conferences on artificial intelligence (IJCAI)*, pages 448—-453.

Cody Rioux, Sadid A Hasan, and Yllias Chali. 2014. Fear the Reaper: A System for Automatic Multi-Document Summarization with Reinforcement Learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 681–690.

Dmitri G. Roussinov and Hsinchun Chen. 2001. Information navigation on the web by clustering and summarizing query results. *Information Processing and Management*, 37(6):789–816.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* , pages 379–389. Association for Computational Linguistics.

Horacio Saggion and Thierry Poibeau. 2013. *Automatic Text Summarization: Past, Present and Future*, pages 3–21. Springer Berlin Heidelberg, Berlin, Heidelberg.

Horacio Saggion, Simone Teufel, Dragomir R. Radev, and Wai Lam. 2002. Meta-evaluation of Summaries in a Cross-lingual Environment Using Content-based Metrics. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7.

Horacio Saggion, Juan-Manuel Torres-Moreno, Iria da Cunha, and Eric SanJuan. 2010. Multilingual Summarization Evaluation Without Human Models. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1059–1067. Association for Computational Linguistics.

Tetsuya Sakai and Karen Sparck Jones. 2001. Generic Summaries for Indexing in Information Retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 190–198.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 6(12).

Natalie Schluter. 2017. The Limits of Automatic Summarisation According to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 2, pages 41–45.

Alexander Schrijver. 1986. *Theory of Linear and Integer Programming.* John Wiley & Sons, Inc., New York, NY, USA.

Alexander Schrijver. 2003. *Combinatorial Optimization - Polyhedra and Efficiency.* Springer, New York, NY, USA.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Claude E. Shannon. 1948. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27:623–656.

Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. 2010. Summarizing Microblogs Automatically. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 685–688.

Chao Shen and Tao Li. 2011. Learning to Rank for Query-Focused Multi-Document Summarization. In *Proceedings of IEEE 11th International Conference on Data Mining (ICDM)*, pages 626–634. IEEE.

Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. 2007. Document Summarization Using Conditional Random Fields. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 2862–2867.

Advaith Siddharthan, Ani Nenkova, and Kathleen R. McKeown. 2004. Syntactic Simplification for Improving Content Selection in Multi-Document Summarization. In *Proceedings of the 20th international conference on Computational Linguistics*, page 896. Association for Computational Linguistics.

Gregory H. Silber and Kathleen F. McCoy. 2002. Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization. *Computational Linguistics*, 28(4):487–496.

Edwin Simpson and Iryna Gurevych. 2018. Finding convincing arguments using scalable bayesian preference learning. *Transactions of the Association for Computational Linguistics*, 6:357–371.

Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Large-margin Learning of Submodular Summarization Models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 224–233, Avignon, France.

Jonas Sjöbergh. 2007. Older Versions of the ROUGEeval Summarization Evaluation System Were Easier to Fool. *Information Processing & Management*, 43(6):1500–1505.

Karen Sparck Jones. 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of documentation*, 28(1):11–21.

Karen Sparck Jones. 1999. Automatic Summarising: Factors and Directions. In *Advances in Automatic Text Summarization*, pages 1–12. MIT Press.

Josef Steinberger and Karel Jezek. 2004. Using Latent Semantic Analysis in Text Summarization and Summary Evaluation. In *Proceedings of the 7th International Conference on Information Systems Implementation and Modelling*, pages 93–100.

Josef Steinberger and Karel Ježek. 2012. Evaluation Measures for Text Summarization. *Computing and Informatics*, 28(2):251–275.

Josef Steinberger, Mijail A Kabadjov, Ralf Steinberger, Bruno Pouliquen, and Massimo Poesio. 2009. Wb-jrc-ut's participation in tac 2009: Update summarization and aesop tasks. In *Proceedings of the text analysing conference, (TAC 2009).*

Josef Steinberger, Massimo Poesio, Mijail A. Kabadjov, and Karel Ježek. 2007. Two Uses of Anaphora Resolution in Summarization. *Information Processing & Management*, 43(6):1663–1680.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 3104–3112.

Hiroya Takamura and Manabu Okumura. 2010. Learning to Generate Summary as Structured Output. In *Proceedings of the 19th ACM international Conference on Information and Knowledge Management*, pages 1437–1440. Association for Computing Machinery.

Simone Teufel. 2001. Task-Based Evaluation of Summary Quality: Describing Relationships between Scientific Papers. In *Proceedings of the NAACL Workshop on Automatic Summarization*, pages 12–21.

Giulio Tononi, Melanie Boly, Marcello Massimini, and Christof Koch. 2016. Integrated Information Theory: From Consciousness to its Physical Substrate. *Nature Reviews Neuroscience*, 17(2).

Juan-Manuel Torres-Moreno. 2014a. *Automatic text summarization*. John Wiley & Sons.

Juan-Manuel Torres-Moreno. 2014b. *Single-Document Summarization*, chapter 3. Wiley-Blackwell.

Juan-Manuel Torres-Moreno, Horacio Saggion, Iria da Cunha, Eric SanJuan, and Patricia Velázquez-Morales. 2010. Summary Evaluation with and without References. *Polibits: Research Journal on Computer Science and Computer Engineering with Applications*, (42):13–20.

Stephen Tratz and Eduard H Hovy. 2008. Summarization Evaluation Using Transformed Basic Elements. In *Proceedings of the text analysing conference, (TAC 2008)*.

Sebastian Tschiatschek, Aytunc Sahin, and Andreas Krause. 2018. Differentiable submodular maximization. In *Joint Conference on Artificial Intelligence (IJCAI)*, pages 2731–2738.

Victor Yakovlevich Tsvetkov. 2014. The KE Shannon and L. Floridi's Amount of Information. *Life Science Journal*, 11(11):667–671.

Marco Turchi, Josef Steinberger, Mijail Kabadjov, and Ralf Steinberger. 2010. Using Parallel Corpora for Multilingual (Multi-Document) Summarisation Evaluation. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 52–63. Springer.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word Representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

Jenine Turner and Eugene Charniak. 2005. Supervised and Unsupervised Learning for Sentence Compression. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 290–297. Association for Computational Linguistics.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of artificial intelligence research*, 37:141–188.

Jan Ulrich, Gabriel Murray, and Giuseppe Carenini. 2008. A Publicly Available Annotated Corpus for Supervised Email Summarization. In *Enhanced Messaging: Papers from the 2008 AAAI Workshop*, pages 77–82.

Alejandro Valerio and David Leake. 2006. Jump-Starting Concept Map Construction with Knowledge Extracted From Documents. In *Proceedings of the Second International Conference on Concept Mapping*, pages 296–303.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond SumBasic: Task-focused Summarization with Sentence Simplification and Lexical Expansion. *Information Processing & Management*, 43(6):1606–1618.

Jorge Villalon and Rafael Calvo. 2010. Analysis of a Gold Standard for Concept Map Mining – How Humans Summarize Text Using Concept Maps. In *Proceedings of the 4th International Conference on Concept Mapping*, pages 14–22.

Clifford H Wagner. 1982. Simpson's Paradox in Real Life. *The American Statistician*, 36(1):46–48.

Xiaojun Wan and Jianwu Yang. 2006. Improved Affinity Graph Based Multi-Document Summarization. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 181–184. Association for Computational Linguistics.

Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. 2009. Multi-document Summarization Using Sentence-based Topic Models. In *Proceedings of the ACL-IJCNLP 2009*, pages 297–300. Association for Computational Linguistics.

Warren Weaver. 1953. Recent Contributions to the Mathematical Theory of Communication. *ETC: a review of general semantics*, pages 261–281.

Michael White, Tanya Korelsky, Claire Cardie, Vincent Ng, David Pierce, and Kiri Wagstaff. 2001. Multidocument Summarization via Information Extraction. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–7.

Florian Wolf and Edward Gibson. 2004. Paragraph-, Word-, and Coherence-based Approaches to Sentence Ranking: A Comparison of Algorithm and Human Performance. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*.

David H. Wolpert and William G. Macready. 1997. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics*, volume 1, pages 985–992. Association for Computational Linguistics.

Kristian Woodsend and Mirella Lapata. 2012. Multiple Aspect Summarization Using Integer Linear Programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 233–243. Association for Computational Linguistics.

Sewall Wright. 1932. The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution. *Proceedings of the Sixth International Congress of Genetics*, 1:356–66.

Shasha Xie and Yang Liu. 2008. Using Corpus and Knowledge-Based Similarity Measure in Maximum Marginal Relevance for Meeting Summarization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4985–4988. IEEE.

Qian Yang, Rebecca J. Passonneau, and Gerard de Melo. 2016. PEAK: Pyramid Evaluation via Automated Knowledge Extraction. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*, Phoenix, AZ, USA. AAAI Press.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2017. Recent Advances in Document Summarization. *Knowledge and Information Systems*, 53(2):297–336.

Shiren Ye, Tat-Seng Chua, Min-Yen Kan, and Long Qiu. 2007. Document Concept Lattice for Text Understanding and Summarization. *Information Processing & Management*, 43(6):1643–1662.

Wen-tau Yih, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki. 2007. Multi-Document Summarization by Maximizing Informative Content-Words. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 7, pages 1776–1782.

Wenpeng Yin and Yulong Pei. 2015. Optimizing Sentence Modeling and Selection for Document Summarization. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1383–1389.

Jaya Kumar Yogan, Ong Sing Goh, Basiron Halizah, Hea Choon Ngo, and C Puspalata. 2016. A Review on Automatic Text Summarization Approaches. *Journal of Computer Science*, 12(4):178–190.

Dani Yogatama, Fei Liu, and Noah A. Smith. 2015. Extractive Summarization by Maximizing Semantic Volume. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1961–1966. Association for Computational Linguistics.

David Zajic, Bonnie J Dorr, Jimmy Lin, and Richard Schwartz. 2007. Multi-Candidate Reduction: Sentence Compression as a Tool for Document Summarization Tasks. *Information Processing & Management*, 43(6):1549–1570.

Klaus Zechner. 2002a. Automatic Summarization of Open-Domain Multiparty Dialogues in Diverse Genres. *Computational Linguistics*, 28:447–485.

Klaus Zechner. 2002b. Summarization of Spoken Language-Challenges, Methods, and Prospects. *Speech Technology Expert eZine*, 6.

Yang Zhang, Yunqing Xia, Yi Liu, and Wenmin Wang. 2015. Clustering Sentences with Density Peaks for Multi-document Summarization. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1262–1267, Denver, Colorado. Association for Computational Linguistics.

Sheng-hua Zhong, Yan Liu, Bin Li, and Jing Long. 2015. Query-oriented Unsupervised Multi-document Summarization via Deep Learning Model. *Expert Syst. Appl.*, 42(21):8146–8155.

Yixin Zhong. 2017. A Theory of Semantic Information. 129.

Liang Zhou and Eduard Hovy. 2003. A web-trained extraction summarization system. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 205–211. Association for Computational Linguistics.

Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. 2006. Paraeval: Using Paraphrases to Evaluate Summaries Automatically. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 447–454.

Markus Zopf. 2018. Auto-hMDS: Automatic Construction of a Large Heterogeneous Multilingual Multi-Document Summarization Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018.*

Markus Zopf, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016a. Beyond Centrality and Structural Features: Learning Information Importance for Text Summarization. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016)*, pages 84–94.

Markus Zopf, Maxime Peyrard, and Judith Eckle-Kohler. 2016b. The Next Step for Multi-Document Summarization: A Heterogeneous Multi-Genre Corpus Built with a Novel Construction Approach. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1535–1545.

Anmerkungen zum Umgang mit Forschungsdaten

Gemäß der "Leitlinien zum Umgang mit Forschungsdate" der Deutschen Forschungsgemeinschaft[6] wurden alle im Zusammenhang mit dieser Dissertation entstandenen Forschungsdaten langfristig archiviert und sofern möglich öffentlich zugänglich gemacht. Folgende Forschungsdaten wurden frei verfügbar gemacht:

- Software
  - Das in Abschnitt 3.2 beschriebene `GPO` steht unter der Apache-Lizenz 2.0 unter `https://github.com/UKPLab/coling2016-genetic-swarm-MDS` zur Verfügung.
  - Das in Abschnitt 3.1 beschriebene θ `Evaluation` steht unter der Apache-Lizenz 2.0 unter `https://github.com/UKPLab/acl2017-theta_evaluation_summarization` zur Verfügung.
  - Das in Abschnitt 4.4 beschriebene `S3` steht unter der Apache-Lizenz 2.0 unter `https://github.com/UKPLab/emnlp-ws-2017-s3` zur Verfügung.
  - Das in Abschnitt 4.4 beschriebene θ_{Pyr} steht unter der Apache-Lizenz 2.0 unter `https://github.com/UKPLab/acl2017-optimize_pyramid` zur Verfügung.
  - Die für die in Abschnitt 4.2 beschriebenen Experimente notwendige Software steht unter der Apache-Lizenz 2.0 unter `https://github.com/UKPLab/acl2016-optimizing-rouge` zur Verfügung.
- Forschungsergebnisse
  - Alle im Zusammenhang mit dieser Dissertation stehenden Publikationen sind in der ACL Anthology (`https://aclanthology.coli.uni-saarland.de/`) verfügbar.
  - Alle Forschungsergebnisse sind zudem auch in dieser Dissertation selbst dokumentiert, die von der Universitäts- und Landesbibliothek Darmstadt zur Verfügung gestellt wird.

Weitere in dieser Dissertation beschriebene Embeddings können aus urheberrechtlichen Gründen nicht frei verfügbar gemacht werden. Entsprechend der DFG-Leitlinien sind diese Daten sowie damit zusammenhängende Software intern unter Nutzung der Infrastruktur der Universitäts- und Landesbibliothek Darmstadt archiviert, so dass eine Archivierung für mindestens 10 Jahre gewährleistet ist.

[6] `http://dfg.de/download/pdf/foerderung/antragstellung/forschungsdaten/richtlinien_forschungsdaten.pdf`