universidad de león

UNIVERSITY OF LEÓN

DEPARTMENT OF ELECTRICAL, SYSTEMS AND AUTOMATIC ENGINEERING

DEEP LEARNING METHODS FOR EXTRACTIVE TEXT SUMMARIZATION

A dissertation supervised by

DR. EDUARDO FIDALGO FERNÁNDEZ,

AND PROF. DR. ENRIQUE ALEGRE GUTIÉRREZ,

and submitted by

AKANKSHA JOSHI

in fulfillment of the requirements for the Degree of

PHILOSOPHIÆDOCTOR (PH.D.)

León, December 2021

universidad de león

UNIVERSIDAD DE LEÓN

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y DE SISTEMAS Y AUTOMÁTICA

MÉTODOS DE APRENDIZAJE PROFUNDO PARA RESUMEN EXTRACTIVO DE TEXTO

Tesis doctoral dirigida por

EL DR. EDUARDO FIDALGO FERNÁNDEZ

Y EL PROF. DR. ENRIQUE ALEGRE GUTIÉRREZ,

y desarrollada por

AKANKSHA JOSHI

a fin de optar al grado de

DOCTOR POR LA UNIVERSIDAD DE LEÓN DEL PROGRAMA DE INGENIERÍA DE PRODUCCIÓN Y COMPUTACIÓN

León, diciembre de 2021

Abstract

This thesis presents new algorithms, methods, and datasets to solve extractive text summarization of single documents using deep learning methods and fusion-based approaches.

Our first contribution is SummCoder, an unsupervised method for extractive text summarization, unaffected by the non-availability of large labeled datasets required for supervised learning of extractive text summarization. Our proposal generates a summary according to three metrics for sentence selection: content relevance, novelty, and position relevance. The relevance of the sentence content is measured using a deep auto-encoder network. The novelty metric is derived by exploiting the similarity among sentences represented as embeddings in a distributed semantic space. And, the sentence position relevance is a hand-designed feature, which assigns more weight to the first few sentences through a dynamic weight calculation function regulated by the document length. Furthermore, we developed a sentence ranking and a selection technique for generating a document summary by ranking the sentences according to the final score obtained by fusing the three sentences selection metrics. We also introduce a new summarization benchmark, Tor Illegal Documents Summarization (TIDSumm) dataset, mainly to assist Law Enforcement Agencies (LEAs). It contains two sets of ground truth summaries, manually created, for 100 web documents extracted from onion websites in Tor (The Onion Router) network. The evaluation of SummCoder framework on DUC 2002, CNN/DailyMail, Blog Summarization and TIDSumm dataset exhibits a remarkable improvement in ROUGE scores on all of these datasets, compared to other state-of-the-art systems.

To keep enhancing the accuracy on the task of text summarization, we propose DeepSumm, a summarization framework that utilizes the topic information in documents along with sequence to sequence networks. The topic vectors capture long-range semantic information in the document that is not otherwise encapsulated using other document representations. In DeepSumm, we utilize the latent information in the document estimated via topic vectors and sequence networks to improve the quality and accuracy of the summarized text. Each sentence is encoded through two different recurrent neural networks based on probabilistic topic distributions and word embeddings. Then, a sequence to sequence network is applied to each sentence encoding. The outputs of the encoder and

the decoder in the sequence to sequence networks are combined after weighting using an attention mechanism and converted into a score through a multi-layer perceptron network. The sentence scores based on topic, sentence embeddings, position and novelty of each sentence are fused to generate a rank for each sentence indicating their importance. We empirically demonstrated that DeepSumm captures the global and local semantic information of the document, outperforming existing state-of-the-art approaches for extractive text summarization in DUC 2002 and CNN/DailyMail datasets.

Our final contribution aims to increase the accuracy of the text summarization task without any supervision. We designed RankSum, a fusion-based approach that looks at multidimensional features of sentences in the document to achieve this. The proposed methodology utilizes the heterogeneous features of sentences such as topic information, semantic content, important keywords and positional information in sentences to rank them according to their significance. We use probabilistic topic models to determine topic rank, whereas semantic information is captured using sentence embeddings. To derive rankings using sentence embeddings, we utilize Siamese networks to produce abstractive sentence representation and then we formulate a novel strategy to arrange them in their order of importance. A graph-based strategy is applied to find the significant keywords and related sentence rankings in the document. We also formulate a sentence novelty measure based on bigrams, trigrams and sentence embeddings to eliminate the redundant sentences from the summary. We compute the rank of all the sentences in the document using each of these features. The ranks of all the sentences are finally fused to get the final score for each sentence in the document. Experimental results on CNN/DailyMail and DUC 2002 dataset show that our approach is one of the best approaches compared to existing state-of-the-art summarization methods.

Resumen

Esta tesis presenta nuevos algoritmos, métodos y conjuntos de datos para realizar resúmenes de texto extractivos en documentos individuales utilizando métodos de aprendizaje profundo y enfoques basados en la fusión de puntuaciones.

Nuestra primera contribución es SummCoder, un método no supervisado que, por ese motivo, no se ve afectado por la carencia de grandes conjuntos de datos etiquetados, para el entrenamiento de modelos de resúmenes de texto extractivos. SummCoder genera un resumen de texto utilizando tres métricas de selección de oraciones: relevancia del contenido, novedad y relevancia de la posición. La relevancia del contenido de una frase se mide utilizando una red profunda de codificación automática. La métrica de novedad se calcula midiendo la similitud entre oraciones, previamente codificadas como incrustaciones en un espacio semántico distribuido. Por último, la métrica de relevancia de la posición de una frase se basa en una función diseñada que asigna más peso a las primeras oraciones a través de una función de cálculo de peso dinámico regulada por la longitud del documento. Se propone generar el resumen de texto final fusionando las tres métricas anteriores y ordenando dichas frases dentro del resumen final en base a la puntuación obtenida. Además, presentamos TIDSumm, un conjunto de datos que contiene resúmenes extractivos de 100 dominios recuperados de la red Tor (del inglés, The Onion Router). El objetivo de este dataset es comprobar la efectividad de los métodos de resumen de texto extractivos para dar un posible soporte a Fuerzas y Cuerpos de Seguridad del Estado.

Para mejorar aún más la precisión de los resúmenes de texto extractivos, proponemos DeepSumm, un método para generar resúmenes que utiliza la información de los tópicos de los documentos junto con redes profundas de secuencia a secuencia. Los vectores de los tópicos pueden capturar información semántica en el documento. Cada oración se codifica a través de dos redes neuronales recurrentes diferentes basadas en distribuciones de tópicos probabilísticos e incrustaciones de palabras, y luego aplicar una red de secuencia-a-secuencia a la codificación de cada oración. Las salidas de dicha red se combinan tras ser ponderadas utilizando un mecanismo de atención, convirtiéndose en una puntuación a través de una red neuronal de perceptrones de múltiples capas. Las puntuaciones de las oraciones basadas en el tema, la inserción de palabras, la posición y la novedad de cada oración finalmente se fusionan para generar una puntuación para cada

oración que indica su importancia dentro del resumen final. Los resultados de la experimentación demostraron que DeepSumm captura tanto la información semántica global como local del documento, y obtiene mejores resultados que los métodos del estado del arte a la hora de obtener resúmenes de texto extractivos en los conjuntos de datos DUC 2002 y CNN / DailyMail.

Finalmente, hemos abordado nuevamente la generación de resúmenes de texto extractivos sin necesidad de un proceso supervisado. En este caso, hemos propuesto RankSum, un enfoque basado en la fusión de características multidimensionales de las oraciones en el documento, como son la información del tópico, el contenido semántico, las palabras clave significativas y la posición de las oraciones, para clasificarlas según su significado. Para determinar la clasificación de los tópicos, utilizamos modelos probabilísticos, mientras que la información semántica se captura utilizando frases incrustadas. Para clasificar utilizando incrustaciones de oraciones, utilizamos redes siamesas que permiten producir una representación de oraciones abstractas y luego formulamos una nueva estrategia para ordenarlas en base a su importancia. Se aplica una estrategia basada en grafos para encontrar las palabras clave significativas y las clasificaciones de oraciones relacionadas en el documento. También formulamos una medida de novedad de oraciones basada en bigramas, trigramas e incrustaciones de oraciones para eliminar las oraciones redundantes del resumen. Calculamos el rango de todas las oraciones en el documento usando cada una de estas características. Los rangos de todas las oraciones finalmente se fusionan para obtener la puntuación final de cada oración en el documento. Los resultados experimentales muestran que nuestro enfoque obtiene resultados comparables con otros métodos existentes del estado del arte.

Contents

List of Figures

List of Tables

Índice general

Acknowledgements

I would like to express my sincere gratitude to my guide, Dr. Enrique Alegre and supervisor, Dr. Eduardo Fidalgo, for their guidance, encouragement, and expertise towards completing a PhD thesis. I do not doubt that it would be impossible to complete this thesis without their tremendous support, their patience, motivation, extensive knowledge and invaluable guidance throughout this work.

I would like to thank my colleagues in Group for Vision and Intelligent Systems (GVIS), Laura, Victor, Wesam, Surajit, Deisy, Rubel, Pablo, Javi, Fran, Manuel, and Alex for all their help and encouragement.

Also, I would like to acknowledge the University of León, to provide a nice research environment for my research studying, and INCIBE, which made our research outcome as practical tools helping the community.

I would like to express my deepest gratitude to my esteemed organization, the Center for Development of Advanced Computing (C-DAC) that permitted me to pursue my research goal.

Nobody has been more important to me in pursuing this research than the members of my family. I would like to thank my parents and brother, whose love and guidance are with me in whatever I pursue. They are the ultimate role models.

Finally and most importantly, I wish to thank my loving and supportive husband, Abhishek, and my wonderful daughter, Riyanshi, who provided me with unending inspiration.

Akanksha Joshi
León
5th November 2021

Chapter 1

Introduction

1.1. Motivation

With the advent of the Internet and the huge amount of textual data available, an immediate challenge is to develop new tools to represent the content in a concise form called summary. Automatic text summarization is an important branch of natural language processing that aims to represent long text documents in a compressed way, so that the information can be quickly understood and readable for end users. We can broadly classify text summarization techniques into two categories; extractive and abstractive text summarization Gambhir and Gupta (2017). Extractive text summarization concatenates the most relevant sentences from the document to produce a summary. It usually comprises three major steps: intermediate representation of input text, sentence scoring, and sentence selection. Conversely, abstractive text summarization generates the summary by paraphrasing the main contents of the document using natural language generation techniques.

Summarization can also be categorized as single-document and multi-document depending on the number of input documents given (Zajic et al., 2008; Fattah and Ren, 2009). Similarly, the summarization can be either generic or query-based (Gong and Liu, 2001; Dunlavy et al., 2007; Wan, 2007; Ouyang et al., 2011). The generic summarization provides an overall idea of the document content whereas the query-focused presents the relevant content of the document according to the user given queries. In this paper, our focus is on the task of generic extractive text summarization on single documents. Various traditional methods for extractive text summarization proposed in the literature are based mainly on human-engineered features, as the combination of statistical and linguistic features such as term frequency (Luhn, 1958; Nenkova and Vanderwende, 2005), sentence length and position (Erkan and Radev, 2004) or cue and stigma words (Edmundson, 1969). In these methods, a score is assigned to each sentence based on its features. To select sentences to generate a summary, various techniques have been proposed, including greedy approaches (Carbonell and Goldstein, 1998), graph-based approaches (Liu et al., 2021; Erkan and Radev, 2004; Mihalcea and Tarau, 2004; Wan, 2010; Parveen et al., 2015), and optimization-based approaches (McDonald, 2007).

In more recent works, deep learning-based methods have attained impressive accuracies in many Natural Language Processing (NLP) tasks, such as question-answering

(Bordes et al., 2014), natural language understanding (Collobert et al., 2011), sentiment analysis (dos Santos and Gatti, 2014), text classification Zhang et al. (2015) and language translation (Jean et al., 2015). Following the success of deep learning in various NLP applications, many ongoing research works are also aiming to improve the text summarization task by exploiting the capabilities of deep neural networks (Rush et al., 2015; Nallapati et al., 2017b,a, 2016; Bhargava and Sharma, 2020). Deep neural networks represent the data with multiple levels of abstraction after processing over several non-linear computation-intensive layers. To learn a good and semantically meaningful representation of input data, deep learning networks must be supplied with a huge amount of training data. Most deep learning-based approaches, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNN), require labelled data to train the deep network architectures with millions of learnable parameters.

The Onion Router (Tor)[1] is one of the most popular Darknet networks, and its websites are usually known as Hidden Services (HS). These HS can be accessed through Tor Browser software[2] or a proxy, such as Tor2Web[3]. In the last few years, projects like *TOR2WEB* were developed to allow the Surface Web users access directly to Tor content through a standard browser, instead of starting an instance of a dedicated one like the Tor browser. The concept of Darknet has existed since the establishment of the World Wide Web (WWW). However, its popularity raised in October 2013 when the FBI arrested Dread Pirate Roberts, the owner of *Silk Road* black market. The FBI estimated the sales on Silk Road to be 1,2 Billion dollars by July 2013. The trading network covered among 150,000 anonymous customers and approximately 4,000 vendors (Rudesill et al., 2015) who use Cryptocurrency for their financial transactions (Nakamoto et al., 2008; Ron and Shamir, 2014) .

The privacy and anonymity offered by Tor result in an adequate shelter for journalists who are seeking freedom of speech in their dictatorship countries. But, unfortunately, traders illegally abuse the network to promote their business, creating marketplaces and forums for unlawful activities. These activities include, but are not limited to, illegal drugs, trading weapons, and child sexual abuse (Ling et al., 2015; Gangwar et al., 2017; Norbutas, 2018; Al-Nabki et al., 2019). The Tor metrics website[4] reported that the number of unique addresses has increased from 30K to almost 160K between April 2015 and June 2021.

Law Enforcement Agencies (LEAs) monitor Tor Darknet, trying to identify, manage, and address these threats. INCIBE, the Spanish National Cybersecurity Institute[5] works with Spanish LEAs, providing them automated tools and services to fight against cybercrime. These tools allow LEAs to automatically monitor Tor HS, saving them from manually explore thousands of domains daily (Saikia et al., 2017a; FIDALGO et al., 2017; Biswas

[1] www.torproject.org
[2] https://www.torproject.org/projects/torbrowser.html.en
[3] https://tor2web.org/
[4] https://metrics.torproject.org/
[5] In Spanish, it stands for the Instituto Nacional de Ciberseguridad de España

et al., 2020, 2017; Al-Nabki et al., 2019, 2020b,c; Saikia et al., 2017b). Also, to deal with the fast proliferation of Tor hidden services, analyzing their content to capture meaningful information that will assist in fighting against cybercrimes.

We collaborate with INCIBE, developing solutions based on Artificial Intelligence for these tools and services oriented to Cybersecurity and Cybercrime. In this work, we have created a solutions for several INCIBE projects that allow them to summarize the content of Tor network documents. We designed three different approaches SummCoder, DeepSum and RankSum utilizing deep learning methods and fusion schemes to enhance the accuracy of extractive text summarization.

1. *SummCoder*: An Unsupervised Framework for Extractive Text Summarization Based on Deep Auto-encoders.

2. *DeepSumm*: Exploiting Topic Models and Sequence to Sequence Networks for Extractive Text Summarization.

3. *RankSum*: An Unsupervised Extractive Text Summarization based on Rank Fusion.

In the following, we present the motivation for each of the proposed summarization methods.

1.1.1. SummCoder

At present, one of the biggest challenges in applying supervised deep learning approaches for extractive text summarization is the unavailability of large-scale extractive summaries created manually that are needed as ground truth for training networks. SummCoder addresses this shortcoming by exploiting techniques that do not require labelled data for training. SummCoder is an unsupervised approach based on auto-encoders and sentence embeddings that generates summaries using three different scores. First, we computed the sentence saliency score via auto-encoders and compressed document representation. Then, to eliminate the redundancy in the produced summary, we formulate a novelty score, together with a position score that takes into account the positional information of each sentence in the document. Finally, the saliency, novelty and position scores of sentences are combined through a weighted fusion to generated the final score for ranking sentences and summary generation.

Deep auto-encoders and variational auto-encoders have been previously applied for query-based single-document text summarization (Yousefi-Azar and Hamey, 2017). Yousefi-Azar and Hamey (2017) applied auto-encoders for query-based single-document text summarization (2017), and Alami et al. (2018) used variational auto-encoders for Arabic text summarization. However, these approaches are different from our approach because Yousefi-Azar and Hamey (2017) and Alami et al. (2018) trained the auto-encoders by

representing the text document using term frequency-inverse document frequency (TF-IDF) vectors, which completely ignore the word order. Li et al. (2017a) proposed variational auto-encoder for latent semantic modelling of sentences to estimate their salience for multi-document summarization. In contrast, we trained the auto-encoder using sentence embeddings, which considers the order of words and thus, the meaning of the sentence is better preserved. Besides that, we represent the documents in a high-level concept space through an auto-encoder, which efficiently captures the semantics of the document and explores the inter-relationship among the sentences.

1.1.2. DeepSumm

Several authors proposed deep learning approaches using sequence networks for extractive (Ren et al., 2017; Nallapati et al., 2017a; Liu, 2019; Mutlu et al., 2020) and abstractive (Nallapati et al., 2016; Li et al., 2017a) text summarization. Despite gaining so much popularity in text summarization, neural network methods have some limitations. The main problem with recent state-of-the-art RNN-based summarization methods (Zhou et al., 2018; Zhang et al., 2018; Nallapati et al., 2017a) is that they fail to capture the latent topic information in the document that carries the significant content to summarise the text (Dieng et al., 2016). Thus, the summary lies in an embedding space that hardly contains any topic information from the document. Apart from this, the variants of Recurrent Neural Networks (RNN) such as Gated Recurrent Unit (GRU) (Chung et al., 2014) and Long Short Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) have very limited capability to retain the long-range semantics of the document (Khandelwal et al., 2018).

Thus, to overcome the shortcomings of recent literature, we introduce *DeepSumm*, a novel summarization method that uses the global semantic information jointly with both the local syntactic and semantic information in a document to produce extractive text summaries. LSTM networks are capable of extracting the local semantic and syntactic information and handling long-range dependencies to some extent. However, enriching LSTM networks with topic information enables to capture the global meaning embedded in the document, which is helpful for generating summaries. Our proposed method obtains a summary after selecting sentences ranked using the fusion of four scores: Sentence Topic Score (STS), Sentence Content Score(SCS), Sentence Novelty Score (SNS) and Sentence Position Score(SPS).

1.1.3. RankSum

Although recently proposed deep learning approaches (Ren et al., 2017; Nallapati et al., 2017a; Cheng and Lapata, 2016; Zhou et al., 2020; Joshi et al., 2019; Liu, 2019; Bahdanau et al., 2015) have promised to be a good solution for summarization, they require a lot of training data to produce good extractive summaries. This is one of the main draw-

backs of supervised learning approaches. We propose RankSum, an unsupervised approach based on the fusion of sentence ranks calculated via different mechanisms and properties of sentences such as topic, semantics, keywords and positional information.

The topic content (Blei, 2012) in a document can capture the global saliency of the document and has been implemented for understanding long-range dependencies in documents (Mikolov and Zweig, 2012a). The sentence embeddings preserve the semantic meaning of the sentences (Kiros et al., 2015). We use Siamese networks (Bromley et al., 1993) with triplet loss to derive sentence embeddings for our task. These embeddings efficiently represent the semantics of sentences (Reimers and Gurevych, 2019) and thus, can be efficiently utilized for summarization tasks .

Several approaches have been applied in literature (Jindal and Kaur, 2020; Litvak and Last, 2008; Matsuo and Ishizuka, 2003) to derive keywords in the text for summarization purposes. It is based on the assumption that significant sentences contain the significant keywords of the document. The other attributes that we employed in our approach are the relative positioning sentence in the document. To identify redundancy in the summary text, we employ sentence embeddings, bigrams and trigrams. Through experiments, We showed that each sentence feature is significant for the generation of good summaries. However, different features complement each other and can produce a more meaningful representation of a document.

1.2. Objectives

This dissertation aims to develop new methods and solutions to boost the quality and accuracy of the summarization of single text documents using extractive approaches. To this end, we propose three methods based on deep learning techniques, topic vectors and fusion schemes. Our approaches have been designed to be used in real tools and services for INCIBE.

Based on the previous general goal, we defined the following particular objectives:

1. To propose a method to summarize single-documents using unsupervised techniques.
2. To exploit topic vectors along with sequence networks to improve the quality of summarization.
3. To develop a unified summarization framework that summarizes documents based on various multi-dimensional features extracted from sentences, such as topic information, semantic content, keywords and sentence position in the document.

1.3. Main Contributions

The main contributions of this dissertation may be summarized as follows:

1. We introduce SummCoder, an unsupervised approach for extractive summarization of single documents. Since SummCoder is unsupervised, we do not need training data and it can be applied to different domains, apart from news and web documents.

2. We propose three metrics to measure the sentence quality: sentence content relevance, sentence novelty and sentence position. The computation of the content relevance is based on an auto-encoder network trained by us. The sentence novelty is determined using sentence embeddings generated by using skip-thoughts model (Kiros et al., 2015), fine-tuned on the data from publicly available summarization datasets such as CNN/DailyMail. The sentence position parameter is a handcrafted feature, which dynamically assigns weight to each sentence considering the number of total sentences in the document.

3. We create a sentence ranking and selection strategy fusing the three previous quality metrics. The output summary is produced by selecting the top-ranked sentences constrained by the pre-defined length of the summary.

4. We introduce the dataset Tor Illegal documents Summarization (TIDSumm) to evaluate the performance of SummCoder under a use case from a real tool from INCIBE that monitors Tor darknet (The Onion Router) (Wood, 2010). The dataset consists of 100 Hidden Services and two sets of human-generated ground truth summaries for these Tor domains.

5. We develop Deep Summarization (DeepSumm), a novel method that generates extractive summaries through the weighted fusion of four scores –Sentence Content Score, Sentence Topic Score, Sentence Novelty Score and Sentence Position Score–. We derive STS and SCS using Seq2Seq attention networks. In contrast, SNS is computed employing the word vector representations, and SPS reflects the relative positions of sentences in the documents.

6. We introduce Sentence Topic Embeddings and Sentence Content Embeddings to capture the long-range semantic dependencies and structural content information in the document. Our approach models sentences as functions of word embeddings and topic distributions, producing sentence saliency scores for SCS and STS, respectively. To derive Sentence Topic and Sentence Content Embeddings, LSTM networks and Seq2Seq architectures with decoder attention are applied to generate the STS and SCS scores. Thus, we can compute the saliency of sentences using their local and global semantic structures to retain the pertinent content in the document.

7. We presented a new Sentence Novelty Score (SNS) to eliminate the redundant information and introduce diversity in the output summary. SNS uses sentence rep-

resentations derived from word and topic distribution vectors to compute a novelty score for each sentence in the document.

8. We propose RankSum, a unified framework for extractive text summarization based on four multi-dimensional features extracted from sentences: topic information, semantic content, keywords and sentence position. The approach ranks the sentences of documents based on each of these features. Then, it performs a weighted rank-level fusion to generate the final summary.

9. We generate a topic rank for each sentence based on probabilistic topic models in a novel manner. The topic score of each sentence is computed by estimating the distance of topic representation of each sentence from the topic centroid of the document. Thus, the significant sentences in the document are those that fall close to the topic centroid of the document.

10. We design a novel method of ranking sentences based on semantic sentence embeddings that can efficiently capture the meaning of each sentence in the document. We recursively determine document embedding by removing each sentence from the document and calculate the difference each time with the document embedding computed using all the sentences of the document.

11. We formulate a novelty parameter based on bigrams, trigrams and sentence embeddings to eliminate the redundant sentences from the summary.

12. We use the keyword information in each sentence to produce its final rank. We make use of a graph-based approach to identify the keywords in the document. The sentences with a higher number of keywords are deemed more important than other sentences of the document.

1.4. Chapter Structure

This section describes the structure of this doctoral thesis. This first introductory chapter focused on motivating the work presented in this dissertation, its main objectives and original contributions. The remaining chapters of this thesis are organized as follows.

- Chapter 2 reviews the literature for text summarization methods and algorithms for the three objectives as proposed in Section 1.2. First, the chapter reviews supervised and unsupervised approaches for extractive text summarization and the deep learning methods developed for text summarization. Then, it references the techniques that have been utilized in the literature for various sentence/word representation techniques proposed. This chapter also references the methods that utilize the topic vectors for achieving the summarization object. It further explores

the other statistical and linguistic features exploited for text summarization such as position, term frequency, sentence centrality, keywords, relative length of sentences and numerical data in sentences. Finally, it reviews the approaches that exploit fusion schemes for summarizing text documents.

- Chapter 3 addresses the problem of the unavailability of labeled data for extractive text summarization of single documents and presents SummCoder framework. SummCoder uses skip-thoughts sentence embeddings and auto-encoders to determine the salience score of each sentence in the document. The salience score is then fused with sentence position and novelty score through weighted fusion to obtain final sentence rankings to produce a summary. The chapter also introduces the TIDSumm dataset, created to evaluate SummCoder under a use case for a tool designed by INCIBE to assist Law Enforcement Agencies.

- Chapter 4 proposes DeepSumm framework that makes use of topic vectors and word vectors along with sequence networks for text summarization. The chapter also presents a new novelty score based on topic and content embeddings to eliminate redundant sentences from the summary. It also discusses the experiments that illustrate that topic information can identify significant sentences for extractive summarization, and it provides complementary information in addition to word embeddings that boost the summarization accuracy.

- Chapter 5 introduces a unified approach for summarization that utilizes multi-dimensional features such as topic, keywords, semantic content and positional information to rank sentences for summarization. A novel method of identifying redundant sentences based on sentence embeddings, bigrams and trigrams is also presented. The chapter also discusses the rank fusion methodology to fuse the ranks obtained via different features to produce the final summarization rank.

- Chapter 6 contains a summary of the conducted work with the conclusions of this thesis and gives an outlook of possible future research lines.

- Chapter 7 presents a summary of the doctoral dissertation in Spanish to fulfill the regulations about the Ph.D. studies at the University of León.

1.5. Publications and Research Results

This section presents the research results obtained during the completion of this doctoral thesis.

1.5.1. Publications related to this manuscript

- Joshi, A., Fidalgo, E., Alegre, E., "Extractive Text Summarization in Dark Web: A Preliminary Study", *In proceedings of Applications of Intelligent Systems* , 2018, Gran Canaria, Spain.

- Joshi, A., Fidalgo, E., Alegre, E., "Deep Learning based Text Summarization: Approaches, Databases and Evaluation Measures", *In proceedings of Applications of Intelligent Systems* , 2018, Gran Canaria, Spain.

- Joshi, A., Fidalgo, E., Alegre, E., Fernandez-Robles, L., 2019. Summcoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications*, 129, 200 – 215.

- Joshi, A., Fidalgo, E., Alegre, E., Fernandez-Robles, L.,2020. DeepSumm: Exploiting Topic Models and Sequence to Sequence Networks for Extractive Text Summarization, *submitted to Expert Systems with Applications* [under review].

- Joshi, A., Fidalgo, E., Alegre, E., Fernandez-Robles, L., 2021, RankSum-An Unsupervised Extractive Text Summarization based on Rank Fusion, *submitted to Expert Systems with Applications* [under review].

1.5.2. Research projects

- "Acuerdo de Colaboración para la puesta en marcha de un equipo de investigación aplicada en visión artificial y reconocimiento de patrones". Addendum 22 to the Framework Agreement between INCIBE (Spanish National Cybersecurity Institute) and the University of Leon.

- "Acuerdo de Colaboración para la continuidad de los trabajos de un equipo de investigación aplicada en visión artificial y aprendizaje automático". Addendum 01 to the Framework Agreement between INCIBE (Spanish National Cybersecurity Institute) and the University of Leon.

1.5.3. Attended conferences

- Applications of Intelligent Systems, 2018, (8-12) January, Gran Canaria, Spain.

1.5.4. Patents and Intellectual Property

Patents

- Akanksha Joshi; Eduardo Fidalgo Fernández; Enrique Alegre Gutiérrez; Laura Fernández Robles, Procedimiento y Sistema de Generación de Resúmenes de Texto

Extractivos Utilizando Aprendizaje Profundo No Supervisado y Autocodificadores. Patent # ES2716634, Grant date: November 26, 2020[6]

Intellectual Property Registers

- Akanksha Joshi; Eduardo Fidalgo Fernández; Enrique Alegre Gutiérrez; Laura Fernández Robles, Aplicación para resumir el contenido textual de páginas web de la red Tor. Application: LE-137-2018 , grant date: December 11, 2018.
- Akanksha Joshi; Eduardo Fidalgo Fernández; Enrique Alegre Gutiérrez; Laura Fernández Robles, Aplicación para resumir video a partir de los fotogramas en los que aparecen personas, Application: LE-139-2018, grant date: December 11, 2018.
- Akanksha Joshi; Eduardo Fidalgo Fernández; Enrique Alegre Gutiérrez; Laura Fernández Robles, Aplicación para resumir el contenido textual de páginas web de la red Tor utilizando autoencoders, Application: LE-140-2018, grant date: December 11, 2018.
- Laura Fernández Robles; Enrique Alegre Gutiérrez; Abhishek Gangwar; Akanksha Joshi; Víctor González Castro; Rocío Alaíz Rodríguez; Eduardo Fidalgo Fernández, Aplicación para la monitorización automática del desgaste de plaquitas basado en descriptores de textura, Application: LE-30-2019, grant date: May 13, 2019.

[6] Spanish Patent and Trademark Office, published in Spanish

Chapter 2

State of the art

In this chapter, we will review the last published approaches and solutions that have been applied for extractive text summarization methods.

2.1. Machine Learning-based approaches

A vast corpus of extractive summarisation-related articles is available in the literature involving supervised, semi-supervised and unsupervised machine learning methods. The supervised learning-based approaches (Haghighi and Vanderwende, 2009; Fattah and Ren, 2009; Li et al., 2009; Ouyang et al., 2011; Cheng and Lapata, 2016; Nallapati et al., 2017a; Mandal et al., 2021) such as support vector machines (SVM), Naive-Bayes classifier (Fattah, 2014), mathematical regression decision trees, neural networks (Fattah and Ren, 2009) require gold summaries at the training phase. Mandal et al. (2021) demonstrated the performance of summarization using Support Vector Machine, K-Nearest Neighbor and Decision Trees. In contrast, unsupervised systems (Dunlavy et al., 2007; Wang et al., 2008; Fattah and Ren, 2009; Parveen et al., 2015; Fang et al., 2017; Padmakumar and He, 2021) mostly rank sentences using some heuristics, and they do not require manually created gold summaries for training (Yousefi-Azar and Hamey, 2017). Due to this, unsupervised methods can be easily adapted to new domains without much alteration. The earlier methods of unsupervised text summarization mainly rely on hand-crafted features, such as term frequency (Luhn, 1958), (Nenkova and Vanderwende, 2005), sentence position and length (Erkan and Radev, 2004) or cue and stigma words (Edmundson, 1969). The selected features in these approaches are used to obtain a ranking score for each sentence.

2.2. Deep Learning Methods

Methods based on deep learning for text summarization recently gained momentum by achieving state-of-the-art accuracies. For extractive summarization, most of the state-of-the-art works rely on Gated Recurrent Units (GRU) or Long Short Term Memory (LSTM) networks. Regarding GRU, the following recent works are of interest. Nallapati et al. (2017a) proposed SummaRuNNER, a GRU-RNN based sequence model that can be trained extractive and abstractively to generate summaries. Their approach used the absolute and relative position of the sentence and sentences from previously selected sum-

maries to remove redundancy. The work of Nallapati et al. (2017b) involved two architectures – classifier and selector – consisting of GRU-RNNS for extractive text summarization, which obtained state-of-the-art performance on CNN/DailyMail and DUC 2002 datasets. Zhou et al. (2018) presented NeuSum, an end-to-end hierarchical sentence and document encoder architecture that utilize GRU-RNN to score and select sentences jointly. Their RNN-based sentence extractor considers previously selected sentences while estimating sentence salience to eliminate redundancy in the output summary. Finally, Shi et al. (2019) introduced a novel extractive summarization framework, DeepChannel, which consists of a deep RNN-GRU for salience estimation and a salience-guided greedy sentence extraction strategy.

LSTM-based proposals include works like Cheng and Lapata (2016), which employed an encoder-decoder approach to extract the salient sentences and words for extractive summarization. Their encoder consists of a CNN, whereas their decoder architecture uses LSTM to classify sentences as summary and non-summary. (Jadhav and Rajan, 2018) designed SWAP-NET to model the interactions of salient sentences and keywords in documents to produce extractive summaries. Their approach also uses a bidirectional LSTM architecture with an encoder-decoder to model the interactions between salient sentences and keywords in a document. (Narayan et al., 2018b) conceptualized extractive summarization as a sentence ranking task using an LSTM-based document encoder and sentence extractor.

Narayan et al. (2017) designed a hierarchical LSTM document encoder and an extractor with attention over side information. The side information that authors considered significant in an article is its title and image captions, along with the main body of the document. They proposed a novel training algorithm which globally optimizes the ROUGE evaluation metric through a reinforcement learning objective. (Zhang et al., 2018) built a latent extractive model based on an LSTM network, which instead of maximizing the likelihood of gold standard labels, directly maximizes the likelihood of human summaries given selected sentences. (Tarnpradab et al., 2017) utilized hierarchical attention networks based on LSTM for extractive summarization of forum threads. (Liu, 2019) fine-tuned BERT, a pre-trained encoder-decoder based transformer architecture to boost the accuracies for extractive summarization. Vhatkar et al. (2020) utilized triple scoring and LSTM based sequence networks to generate summary documents.

A wide variety of other approaches, including reinforcement learning, have been applied for extractive text summarization. Feng et al. (2018) presented an attentive encoder RNN based summarization (AES) which consists of an attention-based document encoder and an attention-based sentence extractor. Wu and Hu (2018) focused on introducing coherence into the neural extractive model via reinforcement learning. Their neural coherence model comprises a word-level CNN encoder and a bidirectional GRU sentence level encoder, allowing to capture the cross-sentence local entity transitions. Yao et al. (2018) applied deep reinforcement learning for extractive text summarization in which they used Deep Q-Network (DQN). Their sentence encoder consisted of CNN

and document encoder modeled using bidirectional GRU. It can model the salience and redundancy of sentences in the Q-value approximation and learn a policy that maximizes the ROUGE score concerning gold summaries. Li and Yu (2021) presented summarization method that exploits BERT and dynamic memory networks to generate extractive summaries. Some unsupervised approaches have also been introduced by Li et al. (2017b,a); Joshi et al. (2019) for extractive summarization.

2.3. Topic-based approaches

Though RNN-based models can remember a long context, such large-scale networks are unable to capture the global information present in long documents (Pascanu et al., 2013; Sutskever et al., 2013) which can otherwise be encapsulated through topic models. Topic models have been used earlier for improving the sequence networks (Lau et al., 2017; Wang et al., 2018; Dieng et al., 2016; Mikolov and Zweig, 2012a; Le and Mikolov, 2014) but these models fail to capture the structural content of the text. Ghosh et al. (2016) presented contextual LSTM models, which incorporate the topic information of the text in LSTM networks. Ji et al. (2016) explored multi-level recurrent architectures by efficiently leveraging document context information in language models. Tang et al. (2019) tried to combine a sequence modeling component with a topic modeling component to contain the semantics and sequential structure of the texts for text generation (Tang et al., 2019).

Gialitsis et al. (2019) examined the effects of probabilistic topic model-based word representations for extractive text summarization based on supervised algorithms such as Naive Bayes, Quadratic Discriminant Analysis, or Gradient Boosting Classifiers, among others. They demonstrated that topic modeling outperforms TF-IDF for sentence classification for extractive summarization tasks. Gao et al. (2012) applied LDA to identify semantic topics in the document and then construct a bipartite graph to represent the document and further find sentence salience scores of sentences and topics simultaneously generate a summary. Hennig (2009) developed a query-focused multi-document summarization method based on latent semantic analysis to represent sentences and queries as probability distributions over latent topics. Ailem et al. (2019) explored the use of latent topic information to reveal more global content, which can be used to bias the decoder network to generate words for abstractive summarization. Nagwani (2015) designed a technique using semantic similarity-based topic modeling and topic models to summarize documents over the MapReduce framework.

Topic modeling has been used in multi-document summarization in one of the works authored by Wu et al. (2017). Narayan et al. (2018b) proposed topic-conditioned convolution Seq2Seq networks for extreme summarization – one-line summaries – of news articles. Authors experimentally demonstrated that convolution layers capture long-range dependencies in documents better than RNNs, which is useful for performing document-level abstraction and inference. Issam et al. (2021) produced extractive summaries using

latent topics generated via topic modeling tecniques.

Mehta et al. (2018) proposed LSTM-based sequence encoders that jointly use topic models to learn attention weights across sentence words to produce abstracts of scientific articles.

2.4. Fusion-based Methods

Few summarization approaches have explored the fusion of different summarization features or algorithms. Dutta et al. (2018) presented an ensemble algorithm that combined the outputs of multiple summarization algorithms to produce final summaries. They developed an unsupervised graph-based strategy and a supervised method Learning-to-Rank to fuse the output of multiple algorithms. Summcoder Joshi et al. (2019) is based on the weighted fusion of three sentence features: relevance, novelty and position.

You et al. (2020) presented topic information fusion and semantic relevance for text summarization based on Fine-tuning BERT(TIF-SR). Firstly, authors extracted topic keywords and fused them with source documents. Then they made the summary closer to the source document by computing the semantic similarity between the generated summary and the source document. This approach requires labeled data for summarization.

Wong et al. (2008) designed a learning-based approach using various sentence features such as surface, content, relevance and event. Authors combined all the features using semi-supervised learning to minimize the dependency on the labeled datasets for summarization. However, their approach needs labeled data and, thus, is dependent on the domain for which it is trained.

Mao et al. (2019) developed three methods to fuse and score sentences by combining sentence relations with statistical features of sentences using supervised and unsupervised learning. This method only explored statistical features and sentence relations with each other and would have missed other features required to build an optimal summarization system.

Our RankSum approach is quite distinct from all of the above approaches in the manner that we proposed the fusion of all the sentence features based on topic, sentence embeddings, keywords and position. Finally, we fused all of these features at rank level which is not proposed by any of the above algorithms. Moreover, our summarization method is completely unsupervised which is different from other fusion strategies that used supervised and unsupervised learning and require labeled training data

2.5. Other approaches

Other methods have been proposed in the literature following the extractive approach: statistical-based , concept-based , optimization-based (Sanchez-Gomez et al.,

2020), topic-based, graph-based (Davoodijam et al., 2021), sentence centrality-based (Erkan and Radev, 2004), semantic-based , deep learning-based (Cheng and Lapata, 2016).

Statistical-based methods (Gupta and Lehal, 2010) select important sentences and words for summary depending on their position and most frequent terms or keywords in the sentence. They require less memory and processor capacity, but missing important sentences and summaries might carry more redundancy by including similar sentences in the generated summary.

Concept-based summarization (Moratanch and Chitrakala, 2017) includes retrieving the concepts from an external knowledge source, building a graph model to find relations between concept and sentences and then applying a ranking algorithm to score sentences. It depends on external knowledge sources for the extraction of concepts, which affects the quality of the summary.

Sentence Centrality-based methods (Erkan and Radev, 2004) extract the most important and central sentence in a cluster using the centrality of words which is estimated using the centroid of a document cluster. In this technique, it is required to specify a prior number of clusters, and the generated summary may carry redundancy. Graph-based methods (Mihalcea and Tarau, 2004) build a graph of the document to identify the relationship among sentences and then using a ranking algorithm to determine summary sentences. The main drawback of graph-based approaches is that they fail to identify semantically related sentences, and it does not take into account the importance of words in the sentences (Fang et al., 2017).

Semantic-based methods (Mohamed and Oussalah, 2019) identify key sentences through methods that explore text semantics such as Latent Semantic Analysis (LSA), Semantic Role Labeling (SRL) and Explicit Semantic Analysis (ESA). They are language independent and generates semantically related sentences. However, they are time costly and the summary quality depends on on the semantic representation of input text.

Optimization-based methods(Sanchez-Gomez et al., 2020) utilize an optimization algorithm such as sub-modular programming, Multi-Object Artificial Bee Colony Algorithm to generate a summary of length L. The computational time and cost are high for optimization-based approaches.

Among other statistical features (Ko and Seo, 2008; Fattah and Ren, 2009), such as positive and negative keywords, sentence centrality, the resemblance of sentences to the title, relative length of sentences or presence of numerical data in sentences, sentence position is deemed as one of the essential features that contribute towards the accuracy of text summarization systems (Yeh et al., 2005). According to Gupta et al. (2011) and Abuobieda et al. (2012), the first sentence of a paragraph is important and a strong candidate for summary generation. In our text summarization approaches, we also take the position of the sentence into account so that the sentences, which occur in the first few lines, get higher weights than those that appear posterior in the document. Some other features used for text summarization are TF*IDF (Term Frequency-Inverse Document frequency), inform-

ation gain, mutual information and residual inverse document frequency. These features assign some weights to words and sentences, and accordingly, the sentences are scored and ranked for a summary generation.

Another critical aspect of text summarization is to minimize redundancy in the output summary. Various approaches have been introduced in the literature, which handles redundant content while summary generation Carbonell and Goldstein (1998); Huang et al. (2010); Lloret and Palomar (2013). In our proposed summarization frameworks, we calculate the novelty score of each sentence based on the similarity between sentence embeddings to handle the redundancy while creating document summaries.

Chapter 3

SummCoder: An Unsupervised Framework for Extractive Text Summarization Based on Deep Auto-encoders

This chapter focuses in the creation of summaries for documents when there is not labeled data available. As a solution to this problem, we introduce a novel summarization framework, SummCoder, an unsupervised method for extractive text summarization based on auto-encoders.

Due to copyright issues, we have deleted this chapter from thesis. Here are the details of the published paper:

Akanksha Joshi, E. Fidalgo, E. Alegre, Laura Fernández-Robles, SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders, Expert Systems with Applications, Volume 129, 2019, Pages 200-215,https://doi.org/10.1016/j.eswa.2019.03.045.

Chapter 4

DeepSumm: Exploiting Topic Models and Sequence to Sequence Networks for Extractive Text Summarization

In this chapter, we will discuss about our deep learning and topic-models based approach to boost the accuracy of text summarization.

4.1. Overview

Recently, approaches based on neural networks have gained momentum because of their high performance in many NLP tasks such as text classification (Al-Nabki et al., 2020a), machine translation (Jean et al., 2015), text generation or question answering (Bordes et al., 2014). Several authors proposed deep learning approaches using sequence networks for extractive (Ren et al., 2017; Nallapati et al., 2017a; Liu, 2019; Mutlu et al., 2020; Vhatkar et al., 2020) and abstractive (Nallapati et al., 2016; Li et al., 2017a) text summarization.

Despite gaining so much popularity in text summarization, methods based on neural networks have some limitations. These methods do not capture the latent topic information in documents (Dieng et al., 2016), and thus, the summary lies in an embedding space that hardly contains any topic information from the document. Apart from this, the variants of Recurrent Neural Networks (RNN) such as Gated Recurrent Unit (GRU) (Chung et al., 2014) and Long Short Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) have limited capability to retain the long-range semantics of the document (Khandelwal et al., 2018). In the approach, we complement neural networks with additional topic information from the document to take advantage of the latent content of documents, which otherwise is hardly captured using RNNs.

The main problem with recent state-of-the-art RNN-based summarization methods (Zhou et al., 2018; Zhang et al., 2018; Nallapati et al., 2017a; Mutlu et al., 2020) is that they fail to capture the latent topic information in the document that carries the significant content to summarize text. In our approach, we aim to solve this problem by incorporating topic information in sequence networks to capture the long range semantics in the document.

Another problem to overcome is that there are no works that eliminate redundant

information in the summaries using topic distribution models per words, apart from word embeddings as representations of sentences. Our approach uses sentence embeddings derived using topic and word vectors to discard redundant information and introduce diversity in the generated summary.

Topic modeling (Mikolov and Zweig, 2012a) has been applied to capture the long-range dependencies in documents via latent topics. An increase in accuracy was reported when deep learning networks were supported with topic information (Dieng et al., 2016).

Probabilistic topic models (Blei, 2012) preserve the global semantic information in a document via latent topics that can efficiently capture the global semantic information in documents. By providing the topic information directly to RNN, the global information in the document can be preserved, avoiding the long-standing vanishing gradient problem of neural networks to remember long-term information (Pascanu et al., 2013).

To this end, to combine the merits of both approaches (Zhou et al., 2018; Mikolov and Zweig, 2012b) and increase the accuracy, we introduce *DeepSumm*, a novel summarization method which uses the global semantic information jointly with both the local syntactic and semantic information in a document to produce summaries. LSTM networks are capable of extracting the local semantic and syntactic information as well as handling long-range dependencies to some extent. However, enriching LSTM networks with topic information enables to capture the global meaning embedded in the document, which is quite useful for generating summaries. Our proposed method obtains a summary after selecting sentences ranked using the fusion of four scores: Sentence Topic Score (STS), Sentence Content Score(SCS), Sentence Novelty Score (SNS) and Sentence Position Score(SPS). Our main contributions in this chapter are:

- Deep Summarization (DeepSumm), a novel method for extractive text summarization which generates summaries through the weighted fusion of four scores: SCS, STS, SNS and SPS.

- Sentence Topic and Sentence Content Embeddings, to capture the long-range semantic dependencies and structural content information in the document. Our approach models sentences as functions of word embeddings as well as of topic distributions, and produces sentence saliency scores for both of them, SCS and STS, respectively. To derive sentence topic and sentence content embeddings, LSTM networks and Seq2Seq architectures with decoder attention are applied to generate the STS and SCS scores. Thus, we are able to calculate the saliency of sentences by using both their local and global semantic structures to retain the pertinent content in the document.

- A new Sentence Novelty Score (SNS) is presented to eliminate the redundant information and to introduce diversity in a summary. Our SNS makes use of the sentence representations derived using word and topic distribution vectors to compute a novelty score for each sentence in the document.

Now, we will discuss how our approach is different from the msummairzation methods proposed in literature earlier. To the best of our knowledge, none of the approaches (Cheng and Lapata, 2016; Jadhav and Rajan, 2018; Zhang et al., 2018; Tarnpradab et al., 2017; Liu, 2019; Nallapati et al., 2017a,b; Li et al., 2017b,a; Joshi et al., 2019) based on deep neural network for text summarization made use of topic information to encode the documents and sentences in them. The approaches indicated above utilize word embeddings to represent the documents. However, they miss the global information content, which can be captured using topic vectors.

In our encoder-decoder framework, we fuse the information obtained from both word embeddings and topic vectors. Our encoder-decoder framework is also different from previous works. First, we encode our sentences using LSTM networks and produce sentence content and sentence topic embeddings. Then, the sequence to sequence LSTM network is applied to generate a score for sentences. We use scores from the LSTM network rather than a decision and finally fused it with other sentence scores to classify sentences as summary/non-summary. Moreover, to the best of our knowledge, there are no works that eliminate redundant information in the summaries using topic distribution models per word, apart from word embeddings as representations of sentences. In our work, we utilized both sentence content and sentence topic embeddings in Sentence Novelty Parameter to produce non-redundant and diverse summaries.

Narayan et al. (2018b) proposed topic-conditioned convolution Seq2Seq networks for extreme summarization – one-line summaries – of news articles. They experimentally demonstrated that convolution layers capture long-range dependencies in documents better than RNNs, which is useful to perform document level abstraction and inference. Though they utilized topic information in their framework, their approach is different from ours because they used topic information with CNN networks to generate one line abstractive summaries of documents. However, our framework is focused on extractive summarization using sequence networks rather than convolution networks.

Mehta et al. (2018) proposed LSTM based sequence encoders that jointly use topic models to learn attention weights across sentence words to produce abstracts of scientific articles. Their approach is quite different from ours as they used modified LDA to generate document context/embeddings. In contrast, we use both LDA and sequence networks to generate sentence and document vectors based on topic and word information. This gives an edge over the document context encapsulated using the LDA model only.

4.2. DeepSumm

4.2.1. Problem formulation

We formulate extractive text summarization as a combination of sentence scoring and a selection problem. Each sentence in the document is ranked based on its relevance according to the assigned score. Then, a given number of the top-ranked sentences are se-

lected to form the summary. Given a document D made up by a sequence of N sentences (S_1, S_2, ..., S_N) and sequence of M words as (w_1, w_2, ..., w_M), the summary is generated by a subset of N that contains the most relevant sentences in the document. The relevance of the sentences is determined based on their structural content, topic information, relative position and novelty of that sentence in the document.

For our framework, self-attention sequence networks (Vaswani et al., 2017) are employed to encode the information in the document. They are appropriate for identifying local structural context because of their sequential nature. The overall pipeline of the proposed DeepSumm framework is illustrated in Figure 4.1. The parameters used in the DeepSumm framework are given in Table 4.1. In the subsequent sections, we describe the key components of the proposed deep learning framework for generating an extractive single-document summary.

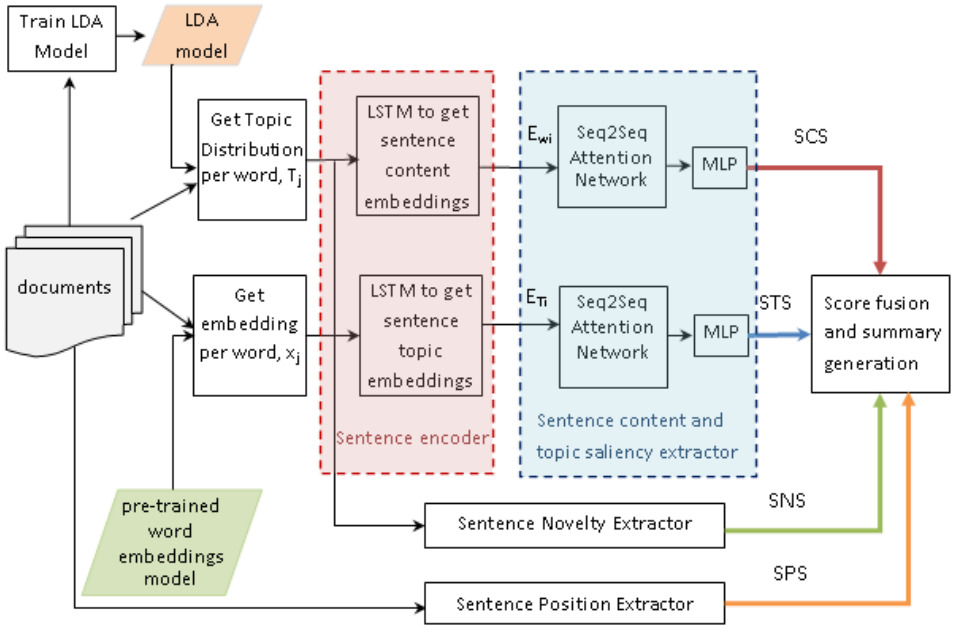


Figure 4.1: Schema of the DeepSumm architecture

Probabilistic Topic distribution per word

Probabilistic topic models were proposed by Blei (2012) to capture the global semantic structure of the documents. The main objective of topic modeling methods is to model documents as collections of multiple latent topics. Each topic can be seen as a distribu-

Table 4.1: Framework Parameters

S.no	Acronym	Parameters
1	T_j	topic vector of j^{th} word
2	x_j	word embedding of $j^{th} word$
3	E_{wi}	Sentence Content Embeddings of i^{th} sentence
4	E_{Ti}	Sentence Topic Embeddings of i^{th} sentence
5	SCS	Sentence Content Score
6	SPS	Sentence Position Score
7	SNS	Sentence Novelty Score

tion of semantically coherent terms and each document exhibits these topics with different probabilities or proportions. One of the probabilistic topic models is Latent Dirichlet Allocation (LDA) (Blei et al., 2003), whose main goal is to find the K latent topics $T = \{T_1, T_2, ..., T_k\}$ in a collection of documents where each topic is a collection of words that tend to co-occur together.

LDA is better than other topic models: LSA (Landauer et al., 1998) and pLSA (Hofmann, 1999) as LDA generalizes well for new documents and has less risk of over-fitting. We use LDA to generate topic vectors T_D for each document present in the distribution, and topic vectors for each word, as t_{w_j} for the j^{th} word, w_j, in a document. We consider topic vectors for each word T_j as point-wise addition of the word topic vector t_{w_j} generated by LDA, plus the document topic vector t_D, as: $T_j = t_{w_j} + t_D$.

Word embeddings

The word embeddings for each word are computed in the document to capture the structural content information. The pre-trained word vectors (Pennington et al., 2014) are used to represent each word as x_j in d-dimensional embedding space, $R^{M \times d}$.

Sentence encoder

We encode topic vectors per word, T_j and word vectors, x_j computed using pre-trained embeddings, as sentence vectors by means of two bidirectional LSTMs. On the one hand, a bidirectional LSTM takes the topic vectors of each word, T_j, in a sentence as input to extract the sentence embedding, termed as E_{Ti}, which relates to the topic information of i^{th} sentence. The forward LSTM reads the sentence S_i from T_{i1} to T_{im} and the backward LSTM from T_{im} to T_{i1}. E_{Ti} is produced by concatenating the final hidden output states, $\overrightarrow{h_{Tt}}$ and $\overleftarrow{h_{Tt}}$ of the forward and backward LSTMs as stated in Equations 4.1, 4.2 and 4.3.

$$\overrightarrow{h_{Tt}} = \text{LSTM}(T_t, \overrightarrow{h_{T(t-1)}}), \tag{4.1}$$

$$\overleftarrow{h_{Tt}} = \text{LSTM}(T_t, \overleftarrow{h_{T(t+1)}}), \tag{4.2}$$

$$E_{Ti} = [\overrightarrow{h_{Tt}}, \overleftarrow{h_{Tt}}], \tag{4.3}$$

On the other hand but similarly, word embeddings, x_{im}, of a sentence i, are inputted to another bidirectional LSTM to extract the sentence embedding, E_{wi}. The forward LSTM for producing E_{wi} reads the sentence S_i from x_{i1} to x_{im} and the backward LSTM from x_{im} to x_{i1}. Equations 4.4, 4.5 and 4.6 indicate the calculation of sentence embeddings E_{wi}.

$$\overrightarrow{h_{xt}} = \text{LSTM}(x_t, \overrightarrow{h_{x(t-1)}})), \tag{4.4}$$

$$\overleftarrow{h_{xt}} = \text{LSTM}(x_t, \overleftarrow{h_{x(t+1)}})), \tag{4.5}$$

$$E_{wi} = [\overrightarrow{h_{xt}}, \overleftarrow{h_{xt}}], \tag{4.6}$$

Sentence content and topic saliency extractor

As in the previous Section 4.2.4, two similar pipelines are designed for computing sentence salience and scores from sentence vectors; one for sentence topic embeddings, E_{Ti}, and the other one for sentence embeddings, E_{wi} based on word vectors. The Sequence to Sequence (seq2seq) attention networks are employed to obtain the sentence saliency based on topic and word vectors. The proposed Seq2Seq architecture consists of an LSTM encoder that reads the sentences one by one and an LSTM decoder that tries to generate the target sequence through an attention mechanism (Bahdanau et al., 2015). The objective of the encoder is to derive a document representation based on the sentences and words present in it.

In the following, we formulate the pipeline that inputs sentence embeddings, E_{wi} obtained using word vectors in the previous section. The encoder consists of a bidirectional LSTM that takes sentence embeddings E_{wi} as input to generate an encoded document representation as described in Equations 4.7 and 4.8.

$$\overrightarrow{h_{E_w i}} = \text{LSTM}_{\text{enc}}(E_{wi}, \overrightarrow{h_{E_w(i-1)}}), \tag{4.7}$$

$$\overleftarrow{h_{E_w i}} = \text{LSTM}_{\text{enc}}(E_{wi}, \overleftarrow{h_{E_w(i+1)}}), \tag{4.8}$$

The decoder is also composed by a bidirectional LSTM that takes the sentence embeddings and attention weighted encoder outputs into consideration to produce decoder hidden states, $\overrightarrow{h_{D_w i}}$ and $\overleftarrow{h_{D_w i}}$ as given in Equation 4.9 and 4.10.

$$\overrightarrow{h_{D_w i}} = \text{LSTM}_{\text{dec}}(E_{wi}, \overrightarrow{h_{D_w(i-1)}}), \tag{4.9}$$

$$\overleftarrow{h_{D_w i}} = \text{LSTM}_{\text{dec}}(E_{wi}, \overleftarrow{h_{D_w(i+1)}}), \tag{4.10}$$

The encoder and decoder outputs e_i,d_i of our pipeline consist of the following hidden states as given in Equation 4.11 and 4.12, respectively.

$$e_i = (\overrightarrow{h_{E_w i}}, \overleftarrow{h_{E_w i}}), \tag{4.11}$$

$$d_i = (\overrightarrow{h_{D_w i}}, \overleftarrow{h_{D_w i}}), \tag{4.12}$$

Then, an attention mechanism is applied to find the global sentence saliency for each sentence using the following Equations 4.13 and 4.14.

$$\alpha_{ij} = \frac{\exp(d_i, e_j)}{\sum_{j=1}^{N} \exp(d_i \cdot e_j)}, \tag{4.13}$$

$$\overline{e_i} = \sum_{j=1}^{N} \alpha_{ij} \cdot e_j, \tag{4.14}$$

Where α_{ij} is a scalar value indicating the importance of i^{th} sentence and $\overline{e_i}$ is the weighted sum of sentence vectors.

The decoder and attention weighted encoder outputs are finally fed into a MLP network to generate scores for each sentence in the document as in Equations 4.15 and 4.16.

$$a_i = ReLU(U \cdot [\overline{e_j}; d_i] + u), \tag{4.15}$$

$$P(y_i = 1 | E_{wi}) = \sigma(V \cdot a_i + v), \tag{4.16}$$

where, U, V are the learned weights of the encoder and decoder, respectively, and u, v are the bias parameters of the encoder and decoder, respectively.

Thus, Sentence Content Score (SCS) is computed using Equation 4.17– by inputting the sentence embeddings E_{wi} to the pipeline described in this section. The Sentence Topic Score (STS) is obtained as given in –Equation 4.18– by inputting the topic distribution sentence encodings E_{Ti} to another seq2seq network designed for encoding sentence topic vectors. STS can capture the global semantics in the document whereas, using SCS, we are able to apprehend the local syntactic information in the document.

$$SCS_i = P(y_i = 1 | E_{wi}), \tag{4.17}$$

$$STS_i = P(y_i = 1 | E_{Ti}), \tag{4.18}$$

Sentence novelty extractor

We propose a new Sentence Novelty Score (SNS) that progressively scans the document one sentence at a time and assigns a score to each sentence depending on the novelty of the sentence for all the previous ones. The novelty of each sentence is calculated based on the sentence embeddings E_{wi} and the topic distribution sentence encodings E_{Ti} as given in Equation 4.19.

$$SNS_i = \begin{cases} 1 & \text{if } i = 1 \\ \frac{1}{i-1} \sum_{j=1}^{i-1} \frac{\left(1-(\text{Sim}(E_{wi},E_{wj})+\text{Sim}(E_{Ti},E_{Tj}))\right)}{2} & \text{otherwise} \end{cases}, \tag{4.19}$$

where, $\text{Sim}(x, y)$ is the cosine similarity between vectors x and y. SNS_1 is set to 1 considering the first sentence of the article as the most significant and novel to be included in the summary. To obtain the sentence novelty, both sentence content and topic representations as generated in section 4.2.4 are used. Through sentence content embeddings, we can find the sentences which are semantically similar to each other and thus can eliminate redundancy in summary. Enriching the novelty calculation with sentence topic embeddings, those sentences, in summary, can be discarded that discuss similar topics and sometimes may not be captured with sentence content embeddings only. Experimentally on a small test dataset, it was found that the average of both scores works better in computing novelty scores for each sentence. The SNS is low for redundant sentences in the document. However, it is robust enough to introduce diversity in the output summary by producing a high score for sentences not covered in the previous text of the document.

Sentence position extractor

In news documents, the sentences which occur earlier in the document are deemed more significant in comparison to other sentences in the document (Luhn, 1958; Edmundson, 1969). Therefore, our Sentence Position Score (SPS) assigns to each sentence a relative score based on its relative position on the document and computed as given in Equation 4.20.

$$SPS_i = \frac{N - P_i}{N}, \tag{4.20}$$

where, P_i is the absolute position of sentence i in the document. The SPS will assign higher scores to the sentences which are in the beginning of the document compared to those which occur in later part of the document.

Scores fusion and summary generation

We finally fused SCS, STS, SNS and SPS to obtain a final sentence score (FSS) for a sentence i, as given in Equation 4.21.

$$FSS_i = \alpha \cdot SCS_i + \beta \cdot STS_i + \gamma \cdot SNS_i + \delta \cdot SPS_i, \tag{4.21}$$

In FSS_i, the sentence with highest score is considered as the most significant to be included in the summary. The values of α, β, γ, and δ are determined empirically. Finally, the sentences of a document are arranged in descending order with respect to their FSS and the top k sentences or words of the list are picked to form the extractive text summary of the document.

4.3. Experimental analysis and results

4.3.1. Datasets

For the supervised - training - of our method, we need a large annotated dataset for text summarization. CNN/DailyMail (Hermann et al., 2015) is the biggest dataset that contains news articles and is frequently used in question-answering research. CNN and DailyMail comprise 197,000 and 90,000 stories, respectively. As extractive summaries of news documents are not available, we utilize the highlights, which are actually abstractive summaries, given along with the news articles to produce their extractive summaries.

Those sentences are greedily added from the document to the gold summary that maximizes the ROUGE-1 and ROUGE-2 scores when matching them with the highlights. A similar approach was followed by (Cheng and Lapata, 2016) to obtain summaries from CNN/DailyMail datasets to train their extractive summarization methods. The standard train, test and validation split for the dataset as given in Table 4.2 are used for evaluation.

To validate our approach on a different dataset, we also use the standard summarization benchmark dataset DUC2002. DUC 2002 [1] was created by the National Institute of Standards and Technology (NIST) for Document Understanding Conferences (DUC) to evaluate and analyze the advances in the field of text summarization. The DUC 2002 dataset consists of 567 news articles from 59 news categories. There are two or more human summaries given for each of the news articles.

Table 4.2: Databases information. # stands for 'number of'.

Dataset	Type	Usage	# Documents	# Categories
CNN/DailyMail	News	Training	287,227	-
		Validation	13,368	-
		Testing	11,490	-
DUC 2002	News	Testing	567	59

[1] https://duc.nist.gov/data.html

4.3.2. Experimental set up

We first split the document into sentences and tokenized them into words. Then, the words were represented using 100-dimensional GloVe embeddings (Pennington et al., 2014). The length of topic vectors for each word and document extracted using LDA is 432. For LDA, different dimensions were tried on the CNN/DailyMail validation set. The best accuracy is reported at 432 dimensions.

The size of the hidden layer of LSTM was set to 256 and of MLP to 128. We used a 0.0001 learning rate and employed gradient clipping of ± 0.5. The learning rate was initially set to 0.01 and reduced by a factor of 10 first after 50^{th} iteration and then after 75^{th} iteration. The batch size was kept at 64, and we trained our network using stochastic gradient descent, and Adam optimization algorithm Kingma and Ba (2014). The dropout probability of 0.5 has been applied in the encoder and 0.25 in the decoder.

Our network was trained for a maximum of 100 epochs, and the best model was selected based on the validation accuracy metric. We set values of α, β, γ and δ in Equation 4.21 to $0.45, 0.45, 0.05$ and 0.05. The values of these parameters were determined empirically on a set of 5000 news documents randomly selected from the CNN/DailyMail validation data. The experiments have been carried out in a machine with two Tesla K40M GPUs with $12GB$ memory, an Intel Xeon processor with 3.00 GHz frequency and 64 GB RAM.

4.3.3. Evaluation

The ROUGE-1, ROUGE-2 and ROUGE-L metrics (Lin, 2004) have been used to evaluate and compare our approach with other state-of-the-art methods. ROUGE metrics are computed by matching unigrams, bigrams and the longest common subsequences between the system and gold summaries. For the DUC 2002 dataset, we kept the summary length to 100 words. For the CNN/DailyMail dataset, the full-length ROUGE metric is used to compare our results with other approaches.

We selected the following state-of-the-art methods to carry out a comparative analysis of the results achieved with our method. For both datasets, we considered the following methods are considered:

NN-SE (Cheng and Lapata, 2016) consists of a hierarchical document encoder and attention-based content extractor that jointly scores and select sentences to generate extractive summaries.

SummaRuNNer (Nallapati et al., 2017a) is a simple RNN-based sequence classifier for extractive summarization. It employed a novel training mechanism to train the network using abstractive summaries.

HSSAS (Al-Sabahi et al., 2018) is a general neural network-based approach that extracts sentences from a document by treating a summarization problem as a classification task. Their network follows a hierarchical structure to reflect the hierarchical nature of documents and used two levels of self-attention mechanism to attend to more import-

ant content for summarization.

LEAD baseline selects the first three leading sentences from the document to produce a summary for comparison.

Specifically for CNN/DailyMail dataset, the results are also reported on following methods:

REFRESH (Narayan et al., 2018a), that globally optimize the ROUGE evaluation metric rather than cross-entropy objective and produces extractive summaries using learning to rank sentences via a reinforcement learning algorithm.

Bi-AES (Feng et al., 2018) is an attentive bi-directional encoder based extractive summarization technique to generate summaries. Bi-AES can generate a rich document representation by considering both the global information of a document and the relationships of sentences in the document

RNES (Wu and Hu, 2018) is a neural coherence model to capture the cross-sentence semantic and syntactic coherence patterns. The model obviates the need for feature engineering and can be trained end-to-end using unlabeled data. Furthermore, the RNES model learns to optimize coherence and the informative importance of the summary simultaneously using reinforcement learning.

NeuSum (Zhou et al., 2020) is a neural network framework for extractive summarization that jointly scores and select the summary sentences. It uses a document encoder to encode each sentence of the document and then an RNN-based sentence extractor to score sentences with their representations while remembering the partial output summary.

BERTSum (Liu, 2019) is an extractive summarization technique that fine-tuned BERT architecture for extractive summarization. Authors tried several summarization layers over BERT architecture and found BERTSum with inter-sentence transformer layers achieve the best performance.

PACSUM(BERT) (Zheng and Lapata, 2019) is an unsupervised summarization algorithm that employed BERT to capture sentence similarity and built graphs with directed edges, arguing the contribution of any two nodes to their respective centrality is influenced by their relative position in the document.

JECS (Xu and Durrett, 2019) consists of a sentence extraction model joined with a compression classifier that decides whether or not to delete syntax-derived compression for each sentence.

For DUC 2002 dataset, we took into consideration:

Integer linear programming (ILP) is a phrase-based summarization system proposed by Woodsend and Lapata (2012) that attempts to cover multiple aspects of summarization such as content selection, surface realization, paraphrasing, and stylist conventions. These features are learned separately using specific "expert" predictors but are optimized jointly using the ILP model to generate summaries.

Egraph (Parveen and Strube, 2015) is an entity graph-based method for extractive single-document summarization that considers importance, non-redundancy and local

coherence simultaneously. A bipartite graph represents the input documents, and sentences are ranked based on importance by applying a graph-based ranking algorithm.

Tgraph (Parveen et al., 2015) is another unsupervised entity graph-based system, wherein the nodes are represented using topics rather than entities. The graph is weighted and dense as compared to the Egraph method (Parveen and Strube, 2015).

URANK (Wan, 2010) is a unified rank methodology that simultaneously performs single and multi-document summarization. The mutual influences between the two tasks are incorporated into a graph model, and the ranking scores of a sentence for the two tasks can be obtained in a unified ranking process.

CoRank (Fang et al., 2017) is an unsupervised summary extraction method that combines word-sentence relationship into the graph-based ranking model, such that the mutual influence can convey the intrinsic status of words and sentences accurately.

SummCoder (Joshi et al., 2019) is an auto-encoder based unsupervised extractive summarization method. The authors did a weighted fusion of sentence scores based on its saliency derived using auto-encoders, sentence position and novelty parameters to get the final scores for ranking sentences for generating extractive summaries.

4.3.4. Results

As shown in Table 4.3, DeepSumm achieves outstanding accuracy for the task of single-document extractive summarization on the DUC 2002 dataset. The ROUGE-1, ROUGE-2 and ROUGE-L scores of 53.2, 28.7 and 49.2 yielded by DeepSumm outperformed all the considered state-of-the-art approaches. Furthermore, none of the state-of-the-art RNN-based summarization approaches such as NN-SE, SummaRuNNer, SummCoder, HSSAS utilizes latent topic information in the document, making our proposed method superior to them.The latent topic information in the document along with sequence is able to capture the local as well as global features to identify significant content in the document. This supports the efficacy of our proposed framework that utilizes both topic distribution vectors and language models to derive extractive summaries of the document.

Table 4.3: Comparative analysis of DeepSumm with state-of-the-art algorithms on DUC 2002

Method	ROUGE-1	ROUGE-2	ROUGE-L
LEAD	43.6	21.0	40.2
ILP	45.4	21.3	42.8
NN-SE	47.4	23.0	-
SummaRuNNer	47.4	24.0	14.7
Egraph+coh	47.9	23.8	-
Tgraph+coh	48.1	24.3	
URANK	48.5	21.5	-
SummCoder	51.7	27.5	44.6
HSSAS	52.1	24.5	48.8
CoRank	52.6	25.8	-
DeepSumm	**53.2**	**28.7**	**49.2**

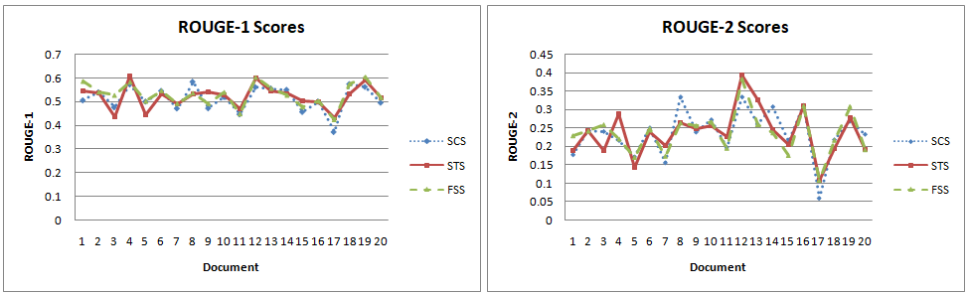
A comparative analysis of the performance of different sentence scores, SCS, STS and FSS, from 20 randomly selected documents on DUC 2002 dataset is illustrated in Figure 4.2. ROUGE-1, ROUGE-2 and ROUGE-L metrics were computed when raking and extracting the sentences of the documents to generate the extractive summaries using SCS and STS, besides the default score FSS. It can be seen that STS, which is computed using topic distribution sentence encodings, achieves as good ROUGE scores on the documents as SCS, which is based on word embeddings. Even in some documents, STS yielded higher ROUGE scores than SCS. It depicts that probabilistic topic distribution encodings can extract the latent topic information of the document, which is complementary to the information captured using word embeddings. The global semantic information encapsulated using topic distribution encodings is relevant for generating good summaries and contributing to better summarization systems. The final sentence score –FSS– generated using the fusion of SCS, STS, SNS and SPS scores can accomplish a good overall accuracy on all the documents. Table 4.4 presents an example of the summary generated by our proposed method for a DUC 2002 document.

We analyzed few summaries from the dataset for which our algorithm scored less. One of the reasons, they scored low is that, they are getting good score from either one from topic or word embeddings and simultaneously scoring low from the other features which makes the overall rouge score fall. The other reason that we found that some sentences are too long to be included in summary as a whole but their fragment. As our algorithm works on sentence extraction rather phrase extraction that makes the summaries scored low compared to human summaries, where only fragments of sentences are picked from the document to create summary. An example of such a summary has been shown in Table 4.5

Table 4.4: Gold summary and DeepSumm generated summary for a document from DUC 2002 dataset

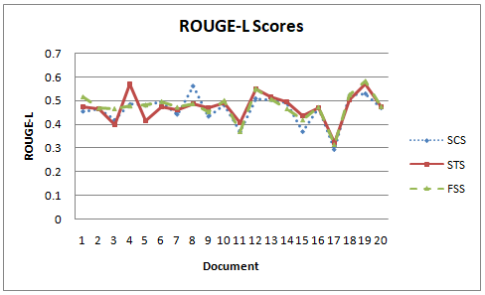
Gold Summary
President Bush named career diplomat Deane Hinton as ambassador to Panama as a recess appointment since Congress is not in session. Hinton, currently ambassador to Costa Rica, replaces Arthur Davis who had been recalled in protest of what the administration\considered the stealing of the Panamanian elections by General Manuel Noriega. Davis was later returned to Panama after US forces invaded Panama and Guillermo Endara was installed as president. Hinton has also been ambassador to El Salvador and Pakistan. Senate Majority Leader George Mitchellcalled Hinton highly qualified because of his "wide-ranging experience and expertise in Central America".
DeepSumm Summary
President Bush has named career diplomat Deane Hinton as ambassador to Panama, the White House announced Tuesday. Hinton, currently ambassador to Costa Rica, replaces Ambassador Arthur H. Davis, who was recalled by Bush in protest of what the administration considered the stealing of the Panamanian elections last May by Gen. Manuel Antonio Noriega. Bush sent Davis back to Panama City after the Dec. 20 invasion of Panama by U.S. forces and installation of Guillermo Endara as president. Independent observers mostly concluded Endara had won the elections by a hefty margin.

On CNN/DailyMail dataset, our method obtained the highest ROUGE-1 score and ROUGE-2 and ROUGE-L scores comparable to the best extractive summarization ap-



(a) ROUGE-1 graph

(b) ROUGE-2 graph



(c) ROUGE-L graph

Figure 4.2: Illustration of ROUGE-1, ROUGE-2 and ROUGE-L metrics considering SCS, STS and FSS scores for the ranking of sentences on 20 randomly selected documents of DUC 2002

Table 4.5: An example of Gold summary and DeepSumm generated summary for a document from DUC 2002 dataset where DeepSumm achieved low ROUGE scores

Gold Summary
On Friday a crane lifted Checkpoint Charlie, the Allied border crossing on the west side of the Berlin Wall, and placed it on a flatbed truck. The guardhouse was taken away in an elaborate ceremony attended by six foreign ministers, a brass band and hundreds of reporters. The public was not invited to the ceremony but many Berliners viewed it from nearby, reliving memories of attempted escapes through the checkpoint that went up with the Berlin Wall in 1961. Now with the wall being dismantled daily in anticipation of German unification, U.S. officials decided to remove Checkpoint Charlie with a grand flourish.
DeepSumm Summary
Maik Polster was a stern-faced member of the East German secret police. Patrick Gainey took pictures for the U.S. Army. Andreas Bratke was an East German who wanted to be a West German. Illa Wobig saw eight people just like him die on her street. What they all had in common was a little white shack on the west side of the Berlin Wall, a guardhouse with a funny name that stood as a chilling symbol of the tensions that divided a street, a city, a nation, the world.

proaches of the literature, as it can be seen in Table 4.6.

Our algorithm ranked at first place for ROUGE-1 score of 43.3 and at second place for ROUGE-2 and ROUGE-L scores of 19.0 and 38.9. Our algorithm surpassed NN-SE, SummaRuNNer, Bi-AES with a very high margin of 4, 3.3, 3.4 for ROUGE-1, ROUGE-2 and ROUGE-L scores. Other state-of-the-art methods such as REFERESH and RNES that

used reinforcement learning also lag in performance in comparison to our summarization proposal. We also got better ROUGE scores than those of NeuSum and HSSAS, which are based on sequence networks. Though we fall behind BERTSum for ROUGE-2 and ROUGE-L score of 20.24 and 39.6, we attained a better ROUGE-1 score with lesser resources in terms of memory and computation as BERTSum architecture is more complex and uses more number of layers comparatively. The notable increase in accuracy compared to most recent approaches proved that our method is quite robust in grasping the pertinent content in the document. The proposed DeepSumm method can condense the salient information from the document, which is otherwise not captured alone using language models. Thus, it boosts the overall accuracy of extractive summarization.

Table 4.6: Comparative analysis of DeepSumm with state-of-the-art algorithms on CNN/DailyMail

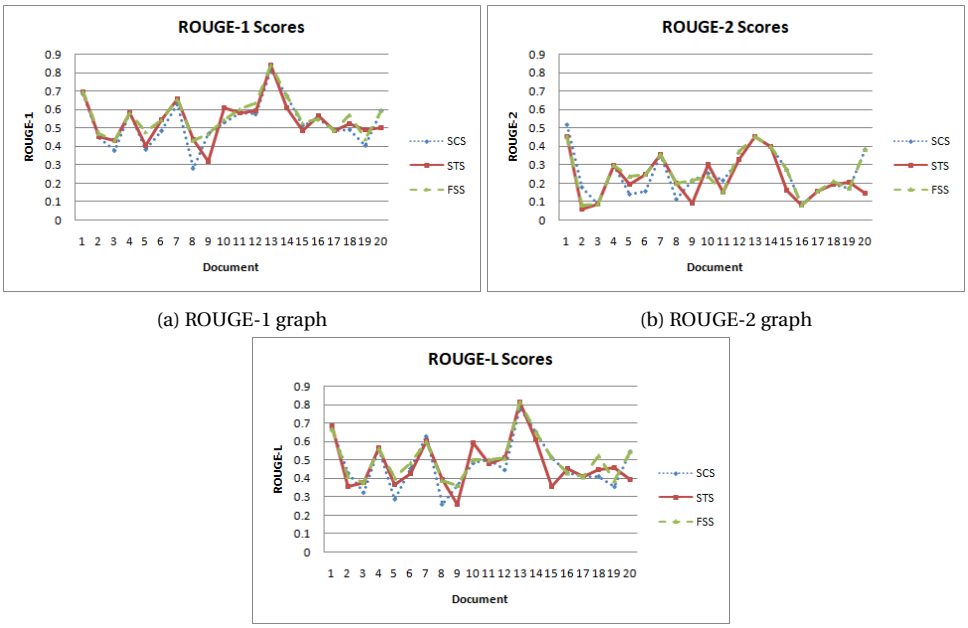
Methods	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
NN-SE	35.5	14.7	32.2
Bi-AES	38.8	12.6	33.85
LEAD	39.2	15.7	35.5
SummaRuNNer	39.6	16.2	35.3
REFRESH	40.0	18.2	36.6
PACSUM(BERT)	40.7	17.8	36.9
RNES w/o coherence	41.2	18.8	37.7
NeuSum	41.5	19.0	37.9
JECS	41.7	18.5	37.9
HSSAS	42.3	17.8	37.6
BertSum	43.2	**20.2**	**39.6**
DeepSum	**43.3**	19.0	38.9

We also illustrated in Figure 4.3 a comparative analysis of the performance obtained with sentence scores SCS, STS and FSS for the ranking and extraction summary sentences of documents on the CNN/DailyMail dataset. Similarly to DUC 2002 dataset, it can be seen that STS yielded as good ROUGE scores on the documents as SCS did. Therefore, topic distribution sentence encodings are quite relevant for finding the pertinent content in the document to obtain semantically coherent and meaningful summaries. A summary of a CNN/DailyMail document produced by the DeepSumm method is shown in Table 4.7.

4.4. Conclusions

In this Chapter of the thesis dissertation, we have presented DeepSumm, a novel method for extractive summarization which produces compact single-document summaries. DeepSumm captures structural and semantic features of the document by utilizing a combination of topic and language vector encodings.

First, we encoded the document sentences using word embeddings and word probabilistic topic distributions, creating their corresponding sentence representations. The inclusion of probabilistic topic distributions in our method considers the latent semantic structure of the document, which is otherwise not captured in the word embedding space.



(a) ROUGE-1 graph

(b) ROUGE-2 graph

(c) ROUGE-L graph

Figure 4.3: Illustration of ROUGE-1, ROUGE-2 and ROUGE-L metrics considering SCS, STS and FSS scores for the ranking of sentences on 20 randomly selected documents of CNN/DailyMail

Table 4.7: Gold summary and DeepSumm generated summary for a document from CNN/DailyMail dataset

Gold Summary
And this week its lyrics , hand-written in 1971 by a young folk singer called Don McLean , were sold at auction in New York for more than $ 1 million . Don McLean (pictured) is responsible American Pie , the lyrics of which have been puzzled over for decades . Argued over by generations of geeky fans , deciphered and re-deciphered by code-breaking rock nerds and considered to be poetic reflections on mid-20_{th} century U.S. social history by even groovier academics , it 's called American Pie . For more than 40 years , its lyrics have been an enigma wrapped in an eight-and-a-half minute long rock 'n' roll puzzle.
DeepSumm Summary
Don Mclean pictured is responsible American pie the lyrics of which have been puzzled over for decades. Argued over by generations of geeky fans deciphered and re-deciphered by code breaking rock nerds and considered to be poetic reflections on century us social history by even groovier academics its called American pie and this week its lyrics handwritten in by a young folk singer called don mclean were sold at auction in new York for more than million. Its also a paean to education mclean loves words he says almost as much as life that may be a slight overstatement but it shows of course like all poets mclean didn't give us a key to the riddle of what his song was about when he released his multi million selling single that would have spoiled it.

Next, seq2Seq attention networks were applied over the sentence embeddings and encodings to extract salient sentences based on their content and topic scores. We also introduced a new novelty computation measure, SNS, to generate a non-redundant and

diversified summary of the document. The position of the sentence in the document was also taken into consideration using the Sentence Position Score. Finally, a weighted fusion of the Sentence Content, Topic, Novelty and Position scores was used to determine the salient sentences in the document.

The experimental results demonstrated how DeepSumm outperformed all the state-of-the-art baselines evaluated on DUC 2002 dataset and achieved a competitive performance on CNN/DailyMail dataset. It has also been illustrated that high-level document features extracted using probabilistic topic distribution models are relevant to generating informative summaries. In the future, we will explore methods to derive other abstract features that contribute to summarization. We will also use probabilistic topic distributions for abstractive text summarization.

Chapter 5

RankSum: An Unsupervised Extractive Text Summarization Method based on Rank Fusion

This chapter focuses on RankSum, an unsupervised approach for extractive text summarization. The proposed framework (Figure 5.1) obtains ranks of sentences from different methods and finally fuse them to generate the summary.

5.1. Overview

Several methods have been proposed in the literature following extractive approaches for text summarization: statistical-based (Gupta and Lehal, 2010; Gambhir and Gupta, 2017), concept-based (Moratanch and Chitrakala, 2017), optimization-based (Sanchez-Gomez et al., 2020), topic-based (Nagwani, 2015; Issam et al., 2021), graph-based (Mihalcea and Tarau, 2004; Liu et al., 2021), sentence centrality-based (Erkan and Radev, 2004), semantic-based (Mohamed and Oussalah, 2019) and deep learning-based (Cheng and Lapata, 2016; Mutlu et al., 2020) among others. Each method has its advantages and limitations and different methods produce different summaries from the same original text.

The topic content (Blei, 2012) captures the global saliency of a document. It has been implemented for understanding long-range dependencies inside documents (Mikolov and Zweig, 2012a). The sentence embeddings preserve the semantic meaning of the sentences. We use siamese networks (Bromley et al., 1993) with triplet loss to derive sentence embeddings for our task. These embeddings efficiently represent the semantics of sentences and can be efficiently utilized for summarization tasks.

Several approaches have been applied in literature (Jindal and Kaur, 2020; Litvak and Last, 2008; Matsuo and Ishizuka, 2003) to derive keywords in the text for summarization purposes. It is based on the assumption that significant sentences contain the significant keywords of the document. The other attribute we employed in our approach is the relative positioning of a sentence in the document. Additionally, to identify redundancy in the summary text, we use sentence embedding, bigrams and trigrams.

We propose combining different techniques to benefit from their specific advantages and reduce their limitations, which should enable the generation of better summaries. Several systems that combine approaches, techniques or features have been pro-

posed for text summarization, such as (Mao et al., 2019; Alami et al., 2019; Moratanch and Chitrakala, 2017; Rahman et al., 2019; Mohd et al., 2020), showing promising results. However, these approaches failed to consider all the features together and none of the methods explored the fusion of topic information in the document. Moreover, few approaches exploited techniques that require supervision which is a limitation when sufficient labeled data is not available for training.

Our aim is to develop a unified framework for extractive text summarization that is fully unsupervised and merges the ranks obtained by different techniques. It relies on methods based on topic, keywords, semantics and positional information. Unlike other fusion strategies, the merging of features occurs at the rank level. It is easy to fuse at a rank level instead of score level due to incompatibility and normalization issues present at score level. Through experiments, we showed that each sentence feature is significant for generating good summaries. However, different features complement each other and can produce a more meaningful representation of the document.

Our main contributions are summarized as follows:

1. We propose RankSum, a unified framework for extractive text summarization that summarizes documents based on multi-dimensional sentences features: topic information, semantic content, keywords and sentence position. RankSum ranks the sentences of documents based on each of these features and then performs a weighted rank level fusion to generate a final summary.

2. We generate a novel topic rank for each sentence based on probabilistic topic models. The topic score of each sentence is computed by estimating the distance of topic representation of each sentence from the topic centroid of the document. The significant sentences in the document fall close to the topic centroid of the document.

3. We introduce a new method for ranking sentences based on semantic sentence embeddings that can efficiently capture the meaning of each sentence in the document. We recursively determine document embedding by removing each sentence from the document and calculate the difference each time with the document embedding computed using all the sentences of the document.

4. We also formulated a novelty parameter based on bigrams, trigrams and sentence embeddings to eliminate the redundant sentences from the summary.

   Finally, we evaluated our summarization method on two publicly available summarization datasets: DUC 2002 and CNN/DailyMail. Empirically, we demonstrated that our unsupervised summarization approach is quite robust compared to other state-of-the-art proposals, including the supervised methods, on both datasets.

5.2. RankSum

5.2.1. Problem formulation

Let each document D consist of N sentences (S_1, S_2, ..., S_N) and M words as (w_1, w_2, ..., w_M). The goal of a summarization framework is to extract a ranked set of the top-L most significant sentences from the document to represent it in a compressed form, i.e. a summary of L sentences. Our proposal, RankSum, uses four multi-dimensional sentence features (topics, keywords, semantics and position) to rank sentences in a document. First, we generate a rank for each sentence in the document according to each feature, gathering information from different aspects of the sentences. Then, we compute a weighted rank fusion derived from the four generated ranks. We hypothesize that each sentence feature contributes to generate a good summary. For this reason, we assume that different features complement each other and can produce a more meaningful representation of a document.

The overall pipeline of the proposed RankSum framework is given in Figure 5.1. It comprises four rank extractors- Topic, Semantic, Keyword and Position Rank Extractor. In sections below, we discuss each of the rank extractors used to rank sentences and the fusion methodology applied, in more detail.

5.2.2. Topic Rank Extractor

In this section, we present a new method to rank sentences based on their topic vectors. We assume that topic vectors provide a latent representation of document which is quite significant while summarising documents. The topic information has been used in query-oriented (Hennig, 2009), abstractive (Ailem et al., 2019) and multi-document summarization (Nagwani, 2015) to boost the summarization accuracy and to complement their method with additional information. Topic information can preserve the global meaning of a document, which is helpful in summarization to understand the long-range semantic information in the text.

We employ Latent Dirichlet Allocation (LDA)(Blei et al., 2003), to find the topics in the document. LDA models text documents as a mixture of topics. Each topic is a collection of words that tend to co-occur together. Each topic is represented as a probability distribution of key terms in the text document. Each document is modelled as a probability distribution of topics. Thus, LDA computes the topic-term distribution and documents topic distribution from an extensive collection of documents using Dirichlet priors for distributions over a fixed number of topics.

By applying LDA, firstly we calculate the topic vectors T_D for each document in the corpus and topic vectors T_w for each word in the document. Then, we obtain the sentence topic vector, $T_S i$ by averaging the topic vectors of each word present in the sentence. To rank the sentences in the document, we compute the euclidean distance, $ED_i(T_{S_i}, T_D)$ between the topic vector of each sentence, T_{S_i} and the topic vector of the document T_D

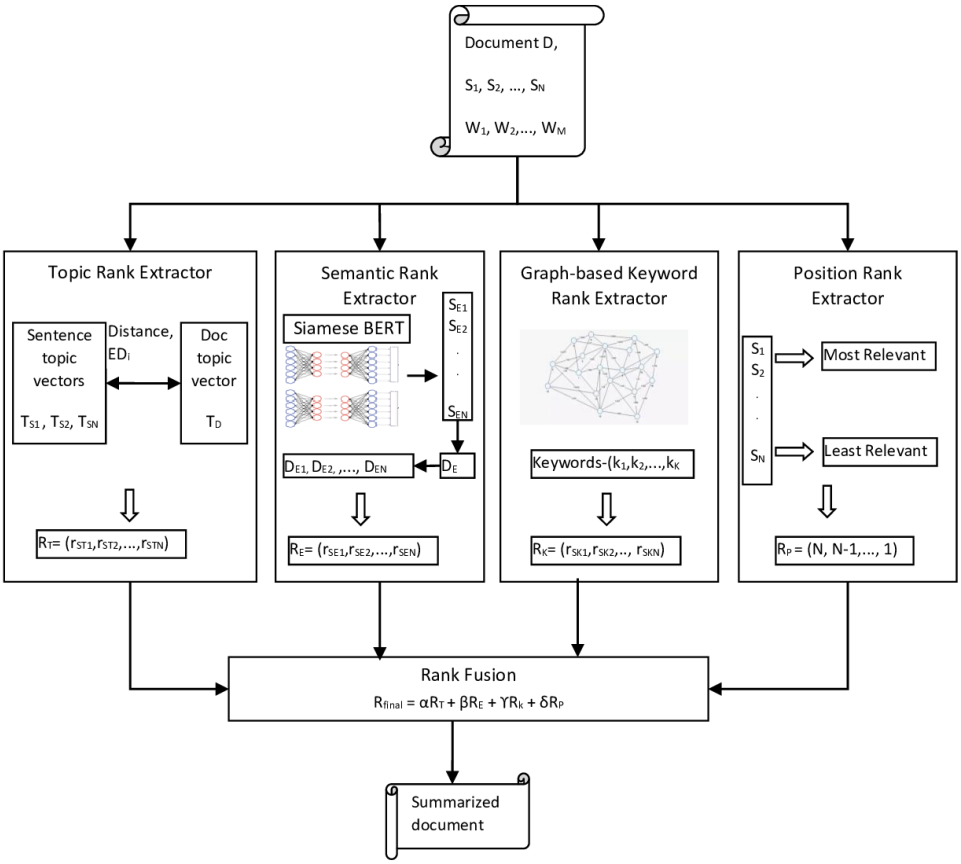


Figure 5.1: Overview of the RankSum architecture. The framework consists of four different rank extractiors- Topic, Semantic, keyword and Position Rank Extractor

as given in Equation 5.1

$$ED_i(T_{S_i}, T_D) = \sqrt{\sum_{q=1}^{Q} (T_{S_{iq}} - T_{D_q})^2}, \tag{5.1}$$

where Q is the length topic vector for sentence, i and document, D. The sentences which are more important will fall close to the document topic vector and ranked accordingly. We represent the rank topic vector generated for a document as

$$\mathbf{R_T} = (r_{ST_1}, r_{ST_2}, \ldots, r_{ST_N}), \tag{5.2}$$

where r_{ST_i} is the topic rank associated with each sentence i, $1 \leq r_{ST_i} \leq N$ and N is the

total number of sentences. Note that 1 is considered the lowest rank.

5.2.3. Embedding-based Semantic Rank Extractor

In order to identify significant sentences based on their semantics, we exploited sentence embeddings. We use SBERT (Reimers and Gurevych, 2019) to obtain the sentence embeddings for each sentence in the document. SBERT is a BERT based architecture that utilises Siamese and triplet networks to derive semantically meaningful embeddings. SBERT has shown to outperform other state-of-the-art embeddings (Devlin et al., 2019; Conneau et al., 2017; Liu et al., 2019) on seven Semantic Textual Similarity (STS) tasks. SBERT is also computationally efficient compared to other sentence embeddings.

We develop a novel algorithm to find the ranking of sentences based on their respective embeddings. Let S_{E_i} represents the embeddings obtained using SBERT architecture for each sentence S_i in the document. We calculate the document embedding D_E by averaging the embeddings of all the sentences in the document. To identify the saliency of each sentence in the document, we remove that sentence from the document. We again obtain a new document embedding D_{E_i}. Next, to measure the saliency of the sentence S_i in the document, we calculate the Euclidean distance, d_E between document vectors, D_E and D_{E_i}. The notable sentence will generate a high value of d_{E_i} as compared to the sentences which do not express the meaning of the document. Thus, we produce the rank vector $\mathbf{R_E}$ for all sentences of the document based on their d_{E_i} scores.

5.2.4. Keyword Rank Extractor

Keywords capture the structural content information in a document. The sentences containing keywords carry significant information compared to other sentences. To compute the set of keywords $K = (k_1, k_2, ...k_K)$ in a document, we firstly remove the stop words and apply lemmatization. Then, we follow a graph-based strategy (Brin and Page, 1998) to identify the key terms in a document.

Then, we generate the rank $\mathbf{R_K}$ regarding the keywords within each sentence in the document. We choose to give a higher rank to the sentences that consist of more keywords. If some sentences contain the same number of keywords, we ranked them according to their positions. We assume that important sentences contain more keywords than sentences with less number of important words.

5.2.5. Position Rank Extractor

The relative position of a sentence in a document indicates the importance of sentence for summary generation (Luhn, 1958; Edmundson, 1969). Therefore, we use it as one relevant attribute for ranking sentences in the document. The sentences that appear at the beginning of a document are more relevant compared to the sentences which appear later in the document (Gupta et al., 2011). We generate the position rank vector $\mathbf{R_P}$

by assigning a rank to sentences depending on their position. The sentence in the first position is given the highest rank, and the last sentence is given the lowest rank as given in Equation 5.3.

$$\mathbf{R_p} = N, N-1, .., 1, \tag{5.3}$$

where, N is the total number of sentences in the document

5.2.6. Sentence Novelty Extractor

To eliminate redundant sentences while producing extractive summaries of the document, we propose a new sentence novelty extractor that makes use of sentence representations S_{E_i} as obtained in Section 5.2.2 as well as bigrams and trigrams present in the sentences. By finding the number of bigrams and trigrams, we can predict which two sentences are similar to each other. However it ignores the semantics of the sentences. To overcome this issue, we complement our novelty extractor with sentence embeddings that are quite good at finding out semantically similar sentences. Embeddings generated using the SBERT network (Reimers and Gurevych, 2019) are quite robust and do well while predicting similar sentences during summary generation. We estimate the sentence novelty, S_{Nov_i} as given in Equation 5.4

$$S_{Nov_i} = \begin{cases} 1, & Sim(S_{E_i}, S_{E_j}) < t1 \quad or \quad Count_{bigrams,trigrams}(i,j)) < t2, \\ & 1 \le j \le V, \qquad i \ne j \\ 0, & otherwise, \end{cases} \tag{5.4}$$

where, 1 indicates the sentence is novel and 0 tells that the sentence is redundant. V is the number of sentences that have been already added to the summary, $t1$ and $t2$ are the thresholds set experimentally to identify similar sentences, $Count_{bigrams,trigrams}(i,j)$ is the number of bigrams and trigrams that match between sentence $S_i and S_j$ and $Sim(S_i, S_j)$ is the cosine distance between sentence S_i and S_j given by

$$Sim(S_i, S_j) = \frac{\overrightarrow{S_i}, \overrightarrow{S_j}}{\| \overrightarrow{S_i} \| \| \overrightarrow{S_j} \|}, \tag{5.5}$$

5.2.7. Rank fusion and summary generation

We finally combine the information provided by the aforementioned modules at the rank level. We fuse all the rank vectors, $\mathbf{R_T}$, $\mathbf{R_K}$, $\mathbf{R_E}$, $\mathbf{R_P}$ and generate a final rank for each sentence in the document. The rankings are merged following

$$\mathbf{Rank_{final}} = \alpha \cdot \mathbf{R_T} + \beta \cdot \mathbf{R_K} + \gamma \cdot \mathbf{R_E} + \delta \cdot \mathbf{R_P}, \tag{5.6}$$

where the values of α, β, γ, and δ are determined empirically. The final rank $\mathbf{Rank_{final}}$ determines the order in which the sentences will be added to the summary. Following an

iterative process, a new sentence is added to the summary if it is distinct from the already added summary sentences based on the novelty extractor, S_{Nov_i} given by Equation (5.4) and defined in Section 5.2.5.

5.3. Experimental results and analysis

5.3.1. Datasets

We used CNN/DailyMail datasets (Hermann et al., 2015) for training the Siamese network for sentence embeddings. This dataset comprises 197,000 and 90,000 news articles from CNN and DailyMail, used frequently in question-answering tasks. We divided CNN/DailyMail into training, validation and test as indicated in Table 5.1. We used the CNN/DailyMail test set and DUC 2002[1] dataset for evaluating our proposed approach and other algorithms for extractive text summarization. DUC 2002 is a standard summarization benchmark consisting of 567 news articles from 59 categories with at least two gold summaries for each article.

Table 5.1: Datasets used for training and evaluation of RankSum Framework

Dataset	Type	Usage	# Documents	# Categories
CNN/DailyMail	News	Training	287,227	-
		Validation	13,368	-
		Testing	11,490	-
DUC 2002	News	Testing	567	59

5.3.2. Experimental set up

Firstly, we split the document into sentences and tokenize it into words. We removed the stop words and applied lemmatization for keyword rank generation. We kept the length of the topic vector to 512 dimensions. The values of α, β, γ and δ are set to 0.3, 0.35, 0.34 and .01 in Equation 5.6, which are determined empirically using CNN/DailyMail Validation set.

5.3.3. Evaluation

To evaluate RankSum, we used ROUGE-1, ROUGE-2 and ROUGE-L metrics (Lin, 2004). ROUGE-1 and ROUGE-2 measure the unigram and bigram matches between the candidate and gold summary, whereas ROUGE-L gives the longest common subsequence matches between the candidate and gold summary.

[1] https://duc.nist.gov/data.html

We compared RankSum with other widely known state-of-the-art approaches for automatic summarization. We used a summary length of 100 words for DUC 2002 and a full-length ROUGE metric for CNN/DailyMail dataset. We picked the results for comparison directly from the papers. Not all the approaches reported accuracies on both datasets. Therefore, we illustrate the results accordingly on CNN/DailyMail and DUC 2002 datasets.

The following techniques are used for comparison with both the datasets.

LEAD selects the first three leading sentences from the document to generate a summary.

NN-SE (Cheng and Lapata, 2016) is a method to jointly score and select sentences using hierarchical encoder-decoder network.

SummaRuNNEr (Nallapati et al., 2017a) is an extractive summarization method based on Recurrent Neural Networks.

HSSAS is a hierarchically structured encoder-decoder network for self attention proposed by Al-Sabahi et al. (2018).

Additionally, we reported the accuracy on the CNN/DailyMail dataset for the following methods:

Bi-AES (Feng et al., 2018) used bi-directional encoder with attention to find extractive summaries of the documents.

REFERESH (Narayan et al., 2018a) is another reinforcement learning based approach that globally optimizes ROUGE evaluation metric.

PACSUM(BERT) (Zheng and Lapata, 2019) is an unsupervised summarization algorithm that employed BERT to capture sentence similarity. It builds graphs with directed edges, arguing that the contribution of any two nodes to their respective centrality is influenced by their relative position in the document.

RNES, developed by Wu and Hu (2018), is a reinforced learning-based extractive summarization for producing coherent summaries .

JECS (Xu and Durrett, 2019) consists of a sentence extraction model joined with a compression classifier that decides whether or not to delete syntax-derived compression for each sentence.

NeuSum presented by Zhou et al. (2020) is a neural network framework to score and select sentences for summary generation jointly.

HIBERTM (Zhang et al., 2019b) is Hierarchical Bidirectional Encoder Representations from Transformers for document encoding and a method to pre-train using unlabeled data for text summarization.

BERTSum proposed by Liu (2019) fine-tuned BERT architecture for extracting meaningful summaries from the document.

BERTSUM+Classifier (Liu, 2019) is a simple classifier developed for extractive summarization based on BERT architecture with inter-sentence transformer layers.

BART (Lewis et al., 2020) is a denoising auto-encoder built using sequence to sequence models that can can be applied to wide variety of tasks including summarization.

PEGASUSLARGE (Zhang et al., 2019a) is a large transformer-based encoder-decoder model pre-trained on massive text corpora where meaningful sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary.

MATCHSUM (Zhong et al., 2020) is summary-level framework that conceptualized extractive summarization as a semantic text matching problem.

For the DUC 2002 dataset we report the performance of the following approaches:

Integer Linear Programming (ILP) proposed by Woodsend and Lapata (2012) an ILP formulation to efficiently search through Quasi-synchronous grammar rules to provide a globally optimal solution for coherent and grammatical summaries.

Tgraph (Parveen et al., 2015) is a graph based approach that uses topical information to compress the document with relevant information.

URANK is a unified ranking methodology presented by Woodsend and Lapata (2012) to simultaneously summarize both single and multiple documents.

SummCoder, an unsupervised auto-encoder based approach to find extractive summaries of the documents(Joshi et al., 2019).

CoRank is proposed by Fang et al. (2017) that explores the word-sentence relationship for unsupervised summary extraction.

5.3.4. Results

Table 5.2 shows the results of our proposed RankSum framework and other state-of-the-art on the DUC 2002 dataset using ROUGE metrics. RankSum achieves ROUGE-1, ROUGE-2 and ROUGE-L scores of 53.2, 27.9 and 49.3, respectively, outperforming all the recently methods analyzed for extractive text summarization dataset.

Table 5.2 shows the results of our proposed RankSum framework and other state-of-the-art on the DUC 2002 dataset using ROUGE metrics. Our proposal achieves ROUGE-1, ROUGE-2 and ROUGE-L scores of 53.2, 27.9 and 49.3, respectively, outperforming all the

recently methods analyzed for the extractive text summarization dataset. We exceeded the highly accurate summarization system, HSSAS and Co-Rank, with a very high margin of 0.6, 0.8 and 0.5 for ROUGE-1, ROUGE-2 ROUGE-L scores. Our unsupervised approach does not require any labeled data to train the system, which is the requirement of most recently proposed summarization methods such as SummaRuNNer, HSSAS and NN-SE. The proposed RankSum summarization method covers every critical aspect of summarization. Rather than focusing on just one feature, we make use of different features of sentences. The topic vectors provide the global content of the document, whereas the keywords can capture the local structural information. The semantics of the sentences are well captured using sentence embeddings. Therefore, the RankSum summarization strategy can supersede the results of other state-of-the-art methods, even supervised ones. The other methods do not consider the topic content in the document. Since we are using the improved SBERT embedding to represent the sentences, we can get better accuracies in comparison to other techniques.

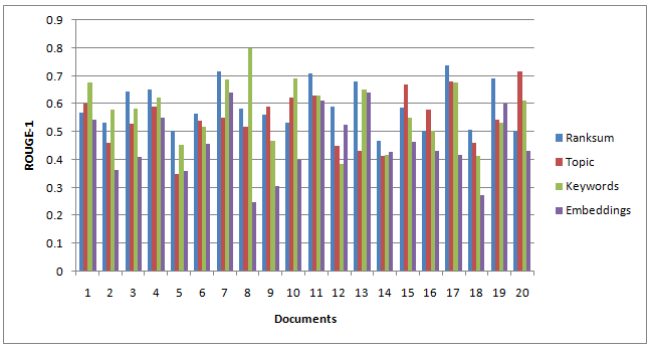
Table 5.2: Comparative analysis of RankSum with state-of-the-art algorithms on the DUC 2002 dataset

Method	ROUGE-1	ROUGE-2	ROUGE-L
LEAD	43.6	21.0	40.2
ILP	45.4	21.3	42.8
NN-SE	47.4	23.0	-
SummaRuNNer	47.4	24.0	14.7
Egraph+coh	47.9	23.8	-
Tgraph+coh	48.1	24.3	-
URANK	48.5	21.5	-
SummCoder	51.7	27.5	44.6
HSSAS	52.1	24.5	48.8
CoRank	52.6	25.8	-
RankSum	**53.2**	**27.9**	**49.3**

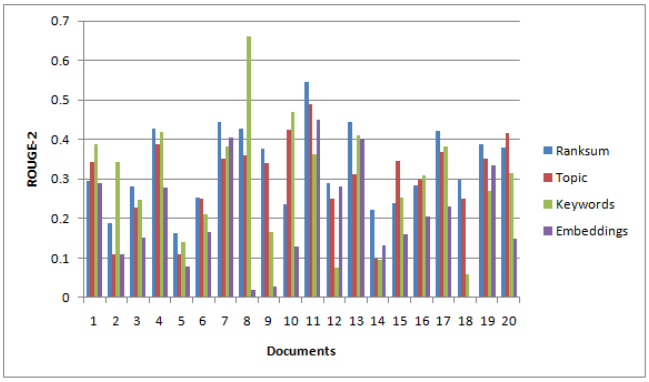
Figure 5.2 illustrates the ROUGE scores generated for 20 randomly selected documents from DUC 2002 dataset. The rouge scores are computed using the rank produced using the topic, keywords, embeddings and RankSum algorithm. The graph depicts that the fusion of ranks of several features such as topic, keywords, embeddings and position generate better ROUGE scores than any of the features individually. This shows that our RankSum algorithm is quite robust to capture the multiple aspects of words/sentences in the document, boosting the overall summarization accuracy of documents.

It can be observed from the graph that individual features cannot attain good ROUGE scores constantly and thus cannot produce a good compression of the query document. However, the fusion of various sentence ranking methods can yield better accuracy in most documents and thus retain the relevant information in a document generating better and more valuable summaries. We also present the summarization of a randomly selected DUC 2002 document using the RankSum algorithm in Table 5.3. As can be seen, that many of the phrases/sentences that appear in the Gold summary are similar to those

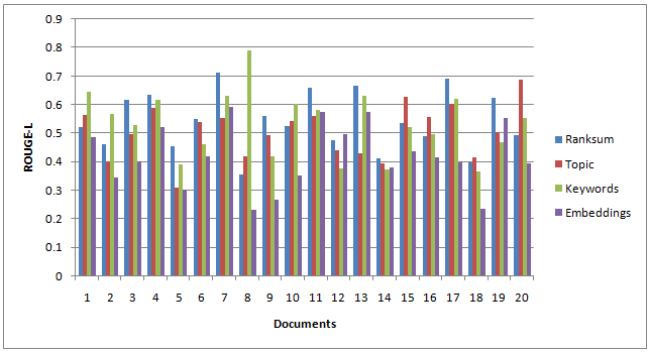
of the ones generated using the RankSum framework.



(a) ROUGE-1 graph



(b) ROUGE-2 graph



(c) ROUGE-L graph

Figure 5.2: Illustration of ROUGE-1, ROUGE-2 and ROUGE-L metrics considering ranking methodology based on topics, keywords, embeddings and RankSum on 20 randomly selected documents of DUC 2002

Regarding the CNN/DailyMail dataset, we observed an improvement in accuracy with

Table 5.3: Gold summary and RankSum generated summary for a document from DUC 2002 dataset

Gold Summary
An overloaded ferry taking 183 Qiongzhong County students and teachers on a field trip to visit a hydroelectric power station capsized Wednesday, killing 55. The passengers and four crew members exceeded the ferry's capacity. The ship sank before it had sailed 200 yards, off Hainan Island in southern China. On September 22, another overloaded ferry sank in the Guangxi Zhuang Autonomous Region bordering Vietnam, killing 61. After 133 died July 21st, when a ferry capsized on the Min River in SW Sichuan Province, and 71 died July 25th, when a passenger boat sank on the Yangtze River, the Ministry of Communications began investigating river vessel safety.
RankSum Summary
On September 22, another overloaded ferry sank in the Guangxi Zhuang Autonomous Region bordering Vietnam, leaving 61 people dead and one missing. The accident occurred off Hainan island, the report said. It did not say how many people the boat was designed to hold. The move followed the July 21 capsizing of a ferry on the Min River in Southwestern Sichuan province in which 133 people died, and the July 25 sinking of a passenger boat on the Yangtze River in which 71 people drowned. The Ministry of Communications, which is responsible for inland water navigation, announced in August it had begun an investigation into the safety of China's river vessels.

ROUGE-1, ROUGE-2 and ROUGE-L scores of 44.5, 24.0 and 41.0 in comparison to other summarization methods. As can be seen in Table 5.4, we obtained the best or comparable ROUGE scores with other state-of-the-art methods on CNN/DailyMail dataset. Our method ranked first for ROUGE-1 and ROUGE-2 scores whereas lags behind PEGASUSLARGE for ROUGE-L with a minimal margin of 0.1.

RankSum is quite robust as it outperforms all the state-of-the-art supervised methods, including PACSUM, HIBERTM, HSSAS, BertSum, PEGASUSLARGES, JECS and BART. This gives our framework an edge over existing summarization techniques. It does not require any ground truth data for producing extractive summaries. In contrast, all the other recently proposed methods require supervision with labeled data. Our summarization method is quite comprehensive as it explores the several significant aspects of sentences in a document required for summarization, such as topic, keywords, semantics and position.

A graphical comparison of ROUGE scores obtained through the ranking based on individual parameters and their fusion using the RankSum methodology is depicted in Figure 5.3. The fusion of all the parameters using the RankSum framework can increase the accuracy on the randomly selected 20 documents and the whole testing CNN/DailyMail dataset. This shows that the weighted rank fusion of different aspects of documents can provide us with a broader and abstractive view of the document to produce better summaries than the individual features. This is because the pertinent content encapsulated using one feature in a document is missed and can be captured via other parameters.

Thus, the fusion of several document features can better understand the syntax and semantics of the document to generate optimal extractive summaries. The RankSum summary of a randomly selected CNN/DailyMail document is shown in Table 5.5.

Table 5.4: Comparative analysis of RankSum with state-of-the-art algorithms on CNN/DailyMail

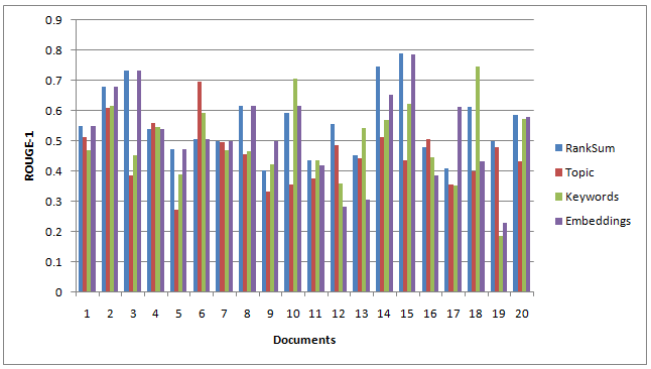
Methods	ROUGE-1	ROUGE-2	ROUGE-L
NN-SE	35.5	14.7	32.2
Bi-AES	38.8	12.6	33.85
LEAD	39.2	15.7	35.5
SummaRuNNer	39.6	16.2	35.3
REFRESH	40.0	18.2	36.6
PACSUM(BERT)	40.7	17.8	36.9
RNES w/o coherence	41.2	18.8	37.7
JECS	41.7	18.5	37.9
NeuSum	41.5	19.0	37.9
HSSAS	42.3	17.8	37.6
HIBERTM	42.3	19.9	38.8
BertSum	43.2	20.2	39.6
BERTSUM+Classifier	43.2	20.2	39.6
BART	44.1	21.2	40.9
PEGASUSLARGE	44.1	21.4	**41.1**
MATCHSUM	44.4	20.8	40.5
RankSum	**44.5**	**24.0**	41.0

Table 5.5: Gold summary and RankSum generated summary for a document from CNN/DailyMail dataset

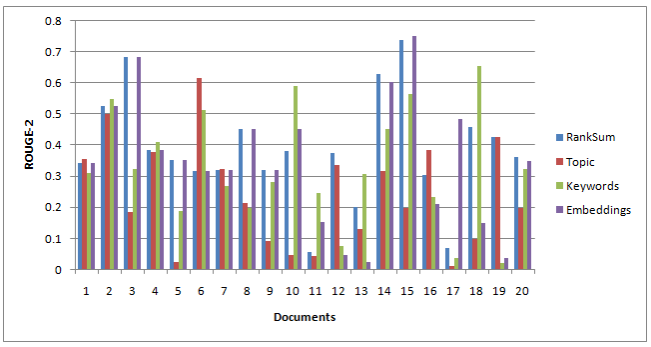
Gold Summary
Astonishing images have emerged of hollywood legend mickey rooney having a gash to his head stitched amid allegations he suffered elder abuse before his death. the actor who died earlier this year at the age of is shown in one picture having a large wound on his head treated by medics. In a second image taken in january mickey rooney is pictured with a missing tooth and other facial injuries. The shocking photos were revealed for the first time amid claims the star may have suffered abuse in the years before he died in april this year.
RankSum Summary
In the enquirers article rooneys eighth wife jan chamberlin vehemently denies any suggestion that she may have abused the star. In a second image he is pictured with a missing tooth and other facial injuries. According to the national enquirer some members of the stars family are preparing to hand a file over to law enforcement chiefs which they believe may explain his death. Astonishing images have emerged of hollywood legend mickey rooney having a gash to his head stitched amid allegations he suffered elder abuse before his death. In a second image taken in january mickey rooney is pictured with a missing tooth and other facial injuries.

5.4. Conclusions

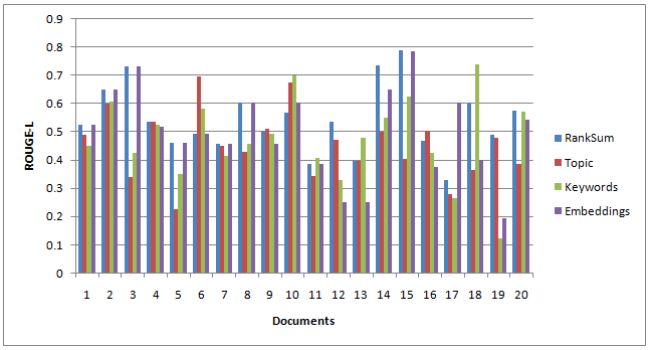
In this Chapter, we presented RankSum, a unified, integrated and unsupervised framework for extractive text summarization. Bearing in mind that humans combine different characteristics for text summarization tasks, RankSum is based on combining several structural and semantic features of a document. Our proposal captures multi-dimensional information from the document using keywords, signature topics, sentence embeddings, and a sentence's position in the document. All of the features individually are capable of extracting important content from the document. However, when combined through a rank fusion scheme, they can cover different aspects of a document to



(a) ROUGE-1 graph



(b) ROUGE-2 graph



(c) ROUGE-L graph

Figure 5.3: Illustration of ROUGE-1, ROUGE-2 and ROUGE-L metrics considering RankSum, topic, semantics and keyword approach for the ranking sentences on 20 randomly selected documents of CNN/DailyMail

summarize it adequately.

We designed a ranking method of sentences for summarization based on topic vectors

estimated using probabilistic topic vectors. A novel method for ranking sentences based on sentence embeddings computed through Siamese networks has also been introduced. Based on document graph representation, the keyword information derived has also been exploited to rank the summary generation sentences. We also developed a novel scheme to eliminate redundant sentences by using bigrams, trigrams and sentence embeddings.

Experimentally, it has been shown that the RankSum method yields a more robust description and outperforms most of the existing state-of-the-art approaches. It was also illustrated that different ranking strategies formulated via topic, keywords and embeddings may not capture the important content in the document, individually, in most of the cases. However, a combination and fusion of all the presented ranking strategies can deliver better ROUGE scores comparatively and, thus, can be a beneficial addition for summarization. Another benefit of our approach is that it does not require labelled data for training.

In the future, we will explore the applicability of the RankSum framework for abstractive summarization.

Chapter 6

Conclusions and Outlook

6.1. Work Summary

In the information age, we experience "information overload", and text summarization produces a concise summary that helps quickly get the critical information, saving the reader a lot of time and effort. On the other hand, manual text summarization consumes a lot of time, effort, cost and becomes unfeasible for many tasks. Therefore, there is a need to research, contribute and obtain improvements in the field of automatic text summarization.

In this thesis, we were focused mainly on extractive text summarization of single documents. Most state-of-the-art deep learning methods rely on the labeled data for summarization to achieve good quality extractive summaries. However, there is no such big summarization dataset available in the text summarization community. Therefore, we proposed SummCoder, an unsupervised deep auto-encoders based framework for text summarization. Secondly, we proposed DeepSumm, a novel approach based on sequence to sequence attention networks for extractive summarization that utilizes topic information from the document to boost the summarization accuracy. Thirdly, we presented RankSum, an unsupervised approach that combined multi-dimensional sentence features such as topic, semantics, keywords and position through rank fusion to achieve optimal text summaries.

Thanks to our collaboration with the Spanish National Cybersecurity Institute, we have the chance to collaborate in the enhancement of products and services to support the work of Law and Enforcement Agencies (LEA). In their effort to tracking unlawful or criminal activities in the darknet, LEA daily work require for tools and services that can help them to quickly identify these activities. Automatic text summarization would be helpful in these tools to present a summary of the Tor domain content that is being analyzed. For that purpose, we also introduced a novel summarization dataset, TIDSumm, created from domains in the Tor network, and we evaluate one of our contributions, SummCoder, on it.

We evaluated our summarization frameworks on publicly available CNN/DailyMail, DUC 2002, Blog Summarization and TIDSumm dataset and observed to achieve state-of-the-art accuracies at the moment they were obtained.

The remainder of this chapter presents a summary of the contributions and possible

areas that may expand to our research.

6.2. Summary of Contributions

In this dissertation, we presented three different frameworks (Summcoder, DeepSumm and RankSum) for extractive text summarization of single documents. The summarization methods exploit deep learning methods such as auto-encoders, sequence networks to boost the summarization accuracies. Fusion based strategies have also been applied to further assist in generating extractive summaries. We also explored unsupervised methods to overcome the limitation of availability of labeled data for extractive summarization. We present our three lines of research as given below.

1. SummCoder, Deep-autoencoders based Unsupervised Approach for Extractive Text Summarization:

   - *We introduced SummCoder, a novel and unsupervised approach for single-document extractive text summarization.* In Chapter 3, we propose three sentence quality metrics: sentence content relevance, sentence novelty and sentence position in the document. The computation of the content relevance parameter is based on an auto-encoder network trained by us and the sentence novelty is determined using sentence embeddings generated by using skip-thoughts model (Kiros et al., 2015), fine-tuned on our data. The sentence position parameter is a hand-crafted feature, which dynamically assigns weight to each sentence taking into consideration the number of total sentences in the document.
   - *A sentence ranking and selection strategy that is derived based on the fusion of scores obtained from the three proposed sentence features.* The output summary is produced by selecting the top-ranked sentences constrained by the pre-defined length of the summary.
     - **Experimental Demonstration.** *To evaluate our approach more comprehensively and to verify the summarization results in a new domain, we introduced a new text summarization benchmark dataset, named Tor Illegal documents Summarization (TIDSumm).* The dataset consists of web documents related to illegal hidden services from the Tor (The Onion Router) Dark Net . The dataset contains documents along with two sets of human-generated ground truth summaries for 100 onion web documents.

2. DeepSumm, Deep Summarization based on Topic Models and Sequence to Sequence Networks:

- *We developed a Deep Summarization (DeepSumm), a novel method for extractive text summarization which generates summaries through the weighted fusion of four scores –Sentence Content Score(SCS), Sentence Topic Score (STS), Sentence Novelty Score (SNS) and Sentence Position Score (SPS)–.* We derive STS and SCS using Seq2Seq attention networks, whereas SNS is computed by means of the word vector representations and SPS reflects the relative positions of sentences in the documents.
- *We introduced Sentence Topic Embeddings and Sentence Content Embeddings to capture the long-range semantic dependencies and structural content information in the document.* Our approach models sentences as functions of word embeddings as well as of topic distributions, and produces sentence saliency scores for both of them, SCS and STS, respectively. To derive sentence topic and sentence content embeddings, LSTM networks and Seq2Seq architectures with decoder attention are applied to generate the STS and SCS scores. Thus, we are able to calculate the saliency of sentences by using both their local and global semantic structures to retain the pertinent content in the document..
- *A new Sentence Novelty Score (SNS) is presented to eliminate the redundant information and to introduce diversity in a summary.* Our SNS makes use of the sentence representations derived using word and topic distribution vectors to compute a novelty score for each sentence in the document.
    - **Experimental Demonstration.** *The DeepSumm summarization framework has been evaluated on standard DUC 2002 summarization benchmark and CNN/DailyMail corpus.* The DeepSumm framework achieves a very good accuracy in single-document extractive text summarization task surpassing several state-of-the-art neural network-based summarizers.

3. RankSum, A Rank Fusion based Framework for Extractive Text Summarization:

- *We proposed a unified framework, RankSum, for extractive text summarization that summarizes documents based on various multi-dimensional sentential features – topic information, semantic content, keywords and sentence position – in the document.* The approach primarily ranks the sentences of documents based on each of these features and then finally performs a weighted rank level fusion to generate final summary.
- *We generate a topic rank for each sentence based on probabilistic topic models in a novel manner.* The topic score of each sentence is computed by estimating the distance of topic representation of each sentence from the topic vector of

the document. The significant sentences in the document falls close the topic vectors of the document.

- *We designed a novel method of ranking sentences based on sentence semantic embeddings that can efficiently capture the meaning of each sentence in the document.* We recursively determine document embedding by removing each sentence from the document and calculate the difference each time with the document embedding computed using all the sentences of the document. We also formulated a novelty parameter based on bigrams, trigrams and sentence embeddings to eliminate the redundant sentences from the summary.

  - **Experimental Demonstration.** *We evaluated our summarization methods on publicly available summarization datasets: DUC 2002 and CNN/DailyMail.* Empirically, we demonstrated that our unsupervised summarization approach is quite robust as compared to other state-of-the-art accuracies including the supervised methods on both datasets.

6.3. Open Problems and Future Work

To end this chapter, we include some research lines arisen during this work, which could be interesting to be addressed in the future.

- Explore additional variations of auto-encoder networks for summarization task.
- Extend our summarization systems for query-based text summarization and multi-document text summarization.
- Incorporate and explore other methods to derive other abstractive features that contribute towards summarization.
- Use topic information derived from probabilistic topic distributions for abstractive text summarization to generate appropriate abstracts of the documents.
- Utilize transfer learning so that our training based summarization system, DeepSumm, can be extended to documents from other domains.

Capítulo 7

Conclusiones y perspectiva

7.1. Resumen del trabajo

En la era de la información experimentamos una "sobrecarga de información", y a través de los resúmenes de texto automáticos podríamos obtener información más concisa que nos permitiría obtener rápidamente la información crítica, ahorrándole al lector mucho tiempo y esfuerzo. Realizar un resumen de texto manual es una tarea muy costosa, en términos de tiempo y esfuerzo, y es inviable para muchas tareas. Por tanto, surge la necesidad de investigar, contribuir y obtener mejoras en el campo del resumen automático de textos.

En esta tesis, nos centramos principalmente en el resumen de texto extractivo de documentos individuales. La mayoría de los métodos recientes de aprendizaje profundo se basan en los datos etiquetados para el resumen a fin de lograr resúmenes extractivos de buena calidad. Sin embargo, no existe un conjunto de datos de resumen tan grande disponible en la comunidad de resumen de texto. Por lo tanto, propusimos un marco de trabajo basado en codificadores automáticos profundos no supervisados para el resumen de texto. En segundo lugar, propusimos un enfoque novedoso basado en redes de atención de secuencia a secuencia para el resumen extractivo que utiliza la información del tema del documento para aumentar la precisión del resumen. En tercer lugar, presentamos un enfoque no supervisado que combinó características de oraciones multidimensionales como tema, semántica, palabras clave y posición a través de la fusión de rangos para lograr resúmenes de texto óptimos.

Gracias a nuestra colaboración con el Instituto Nacional de Ciberseguridad de España, tenemos la oportunidad de colaborar en la mejora de productos y servicios para apoyar la labor de las Agencias de Seguridad y Ejecución (LEA). En su esfuerzo por rastrear actividades ilegales o delictivas en la red oscura, el trabajo diario de LEA requiere herramientas y servicios que puedan ayudarlos a identificar rápidamente estas actividades. El resumen de texto automático sería útil en estas herramientas para presentar un resumen del contenido del dominio Tor que se está analizando. Para ese propósito, también introdujimos un nuevo conjunto de datos de resumen, TIDSumm, creado a partir de dominios en la red Tor, y evaluamos una de nuestras contribuciones, SummCoder, en él.

Todas las contribuciones se han evaluado en los conjuntos de datos CNN / DailyMail, DUC 2002, Blog Summarization y TIDSumm, que están disponibles públicamente y sobre

los que dichos métodos han obtenido resultados del estado del arte en el momento de la realización de los experimentos.

El resto de este capítulo presenta un resumen de las contribuciones y posibles áreas que pueden expandirse a nuestra investigación.

7.2. Resumen de contribuciones

En esta tesis se han presentado tres métodos para el resumen de texto extractivo de documentos individuales: Summcoder, DeepSumm y RankSum. En dichos métodos hemos utilizado técnicas del aprendizaje aprendizaje profundo, como los codificadores automáticos y redes de secuencia a secuencia, para aumentar la precisión del resumen. También se han aplicado estrategias basadas en la fusión de puntuaciones para tratar de obtener resúmenes extractivos de mayor calidad, así como métodos no supervisados para superar la limitación de la disponibilidad de datos etiquetados para el resumen extractivo.

A continuación, se presenta un resumen de las contribuciones de esta tesis:

- *Presentamos SummCoder, un método no supervisado para la realización de resúmenes de texto extractivos de un solo documento.* Junto con SummCoder, en el Capítulo 3 proponemos tres métricas que utilizaremos para la selección de oraciones dentro de un documento: relevancia del contenido de la oración, novedad de la oración y posición de la oración en el documento. El cálculo del parámetro de relevancia del contenido se basa en una red de codificación automática entrenada por nosotros y la métrica de la novedad de la oración se determina utilizando incrustaciones de oraciones generadas mediante el modelo de skip-thoughts (Kiros et al., 2015), ajustado en nuestros conjuntos de datos. La métrica de posición de la oración es una característica diseñada a mano, que asigna dinámicamente peso a cada oración tomando en consideración el número total de oraciones en el documento.

- *Proponemos una estrategia de clasificación y selección de oraciones en base a la fusión de las tres métricas anteriores.* El resumen del texto se obtiene seleccionando aquellas oraciones que hayan obtenido una mejor puntuación tras la fusión de dichas métricas.

- *Presentamos un nuevo conjunto de datos para evaluar métodos de resúmenes de texto extractivos de un solo documento, llamado Resumen de Documentos Ilegales de Tor (TIDSumm).* El conjunto de datos contiene dominios relacionados con servicios ocultos de la red oscura Tor (The Onion Router), junto con dos conjuntos de 100 resúmenes extractivos cada uno, generados manualmente por dos etiquetadores humanos.

- *Introducimos "Resumen Profundo" (DeepSumm), un método novedoso que realiza resúmenes de texto extractivos mediante la fusión ponderada de cuatro métricas: puntuación de contenido de frases (SCS), puntuación de tópicos de frases (STS), puntación de la novedad de la frase (SNS) y puntuación de la posición de la frase (SPS).* STS y SCS se obtienen a través de redes de secuencia a secuencia, mientras que SNS se calcula mediante las representaciones de vectores de palabras y SPS refleja las posiciones relativas de las oraciones en los documentos.

- *Proponemos dos nuevas codificaciones de frases: incrustaciones de tópicos y de contenidos, para capturar las dependencias semánticas de largo alcance y la información de contenido estructural en el documento.*

- *Nuestro enfoque modela las oraciones como funciones de incrustaciones de palabras, así como de distribuciones de temas, y produce puntuaciones de prominencia de las frases para SCS y STS, respectivamente.* Para obtener ambos tipos de incrustación, se utilizan redes LSTM y arquitecturas Seq2Seq con atención del decodificador para generar las puntuaciones de STS y SCS.

- *Se presenta un nuevo Puntaje de Novedad de Oraciones (SNS) para eliminar la información redundante e introducir diversidad en el resumen generado.* Este puntuaje, SNS, utiliza representaciones de oraciones derivadas usando vectores de distribución de palabras y temas para calcular un puntaje de novedad para cada frase del documento.

- *Finalmente, presentamos RankSum, una aproximación para el resumen extractivo de textos usando cuatro características extraídas de las oraciones del documento: información del tema, contenido semántico, palabras clave y posición de la oración.* RankSum clasifica las oraciones de los documentos en función de cada una de estas características y, finalmente, realiza una fusión a nivel de ranking ponderado para generar un resumen final.

- *Generamos un ranking de tópicos para cada frase basado en modelos de tópicos probabilísticos.* La puntuación del tópico de cada oración se calcula estimando la distancia de la representación del tópico de cada oración con respecto al tópico del documento.

- *Diseñamos un método novedoso para clasificar oraciones basado en incrustaciones semánticas de oraciones que pueden capturar de manera eficiente el significado de cada oración en el documento.* Determinamos la representaciones incrustadas de las frases del documento del siguiente modo. recursivamente, eliminamos una a una todas las frases del documento y calculamos la distancia entre el documento incrustado resultante tras haber eliminado una frase y el documento incrustado completo.

- *Formulamos un parámetro de novedad basado en bigramas, trigramas e incrustaciones de oraciones para eliminar las oraciones redundantes del resumen.*

7.3. Problemas abiertos y trabajo futuro

A continuación, presentamos algunas líneas futuras de investigación surgidas durante este trabajo.

- Explorar variaciones de redes de codificación automáticas para la tarea de resumen automático de textos.
- Ampliar los métodos propuestos para la tarea de resúmenes de texto basados en consultas y respuestas aplicado a múltiples documentos.
- Incorporar y explorar otros métodos para obtener características abstractivas que pudieran contribuir al proceso de resumen.
- Usar la información del tópico derivada de las distribuciones probabilísticas del tópico para realizar resúmenes de texto abstractivos.
- Utilizar el aprendizaje por transferencia para que nuestro método de Resumen Profundo (DeepSum), pueda extenderse a documentos de otros dominios.

Bibliography

Abuobieda, A., Salim, N., Albaham, A. T., Osman, A., and Kumar, Y. J. (2012). Text summarization features selection method using pseudo genetic-based model. *International Conference on Information Retrieval & Knowledge Management*, pages 193–197.

Ailem, M., Zhang, B., and Sha, F. (2019). Topic augmented generator for abstractive summarization. *arXiv*.

Al-Nabki, M. W., Fidalgo, E., Alegre, E., and Aláiz-Rodrıguez, R. (2020a). File name classification approach to identify child sexual abuse. In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods*, volume 1, pages 228–234.

Al-Nabki, M. W., Fidalgo, E., Alegre, E., and Fernández-Robles, L. (2019). Torank: Identifying the most influential suspicious domains in the tor network. *Expert Systems with Applications*, 123:212–226.

Al-Nabki, M. W., Fidalgo, E., Alegre, E., and Fernández-Robles, L. (2020b). Improving named entity recognition in noisy user-generated text with local distance neighbor feature. *Neurocomputing*, 382:1 – 11.

Al-Nabki, M. W., Fidalgo, E., Alegre, E., and Fernández-Robles, L. (2020c). Improving named entity recognition in noisy user-generated text with local distance neighbor feature. *Neurocomputing*, 382:1–11.

Al-Sabahi, K., Zu-ping, Z., and Nadher, M. (2018). A hierarchical structured self-attentive model for extractive document summarization (HSSAS). *IEEE Access*, 6:24205–24212.

Alami, N., Ennahnahi, N., Ouatik, S. A., and Meknassi, M. (2018). Using unsupervised deep learning for automatic summarization of arabic documents. *Arabian Journal for Science and Engineering*, 43:7803–7815.

Alami, N., Meknassi, M., and En-nahnahi, N. (2019). Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. *Expert Systems with Applications*, 123:195–211.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, pages 1–15.

Bhargava, R. and Sharma, Y. (2020). Deep extractive text summarization. *Procedia Computer Science*, 167:138–146. International Conference on Computational Intelligence and Data Science.

Biswas, R., Fidalgo, E., and Alegre, E. (2017). Recognition of service domains on tor dark net using perceptual hashing and image classification techniques. In *8th International Conference on Imaging for Crime Detection and Prevention (ICDP 2017)*, pages 7–12.

Biswas, R., González-Castro, V., Fidalgo, E., and Alegre, E. (2020). Perceptual image hashing based on frequency dominant neighborhood structure applied to tor domains recognition. *Neurocomputing*, 383:24–38.

Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, 55(4):77–84.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Bordes, A., Chopra, S., and Weston, J. (2014). Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620, Doha, Qatar. Association for Computational Linguistics.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107 – 117. Proceedings of the Seventh International World Wide Web Conference.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Carbonell, J. and Goldstein, J. (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*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Cheng, J. and Lapata, M. (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, Berlin, Germany. Association for Computational Linguistics.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (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.

Davoodijam, E., Ghadiri, N., Lotfi Shahreza, M., and Rinaldi, F. (2021). Multigbs: A multi-layer graph approach to biomedical summarization. *Journal of Biomedical Informatics*, 116:103706.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dieng, A. B., Wang, C., Gao, J., and Paisley, J. (2016). TopicRNN: A Recurrent Neural Network with Long-Range Semantic Dependency. *arXiv e-prints*, page arXiv:1611.01702.

dos Santos, C. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Dunlavy, D. M., O'Leary, D. P., Conroy, J. M., and Schlesinger, J. D. (2007). Qcs: A system for querying, clustering and summarizing documents. *Inf. Process. Manage.*, 43(6):1588–1605.

Dutta, S., Chandra, V., Mehra, K., Das, A. K., Chakraborty, T., and Ghosh, S. (2018). Ensemble algorithms for microblog summarization. *IEEE Intelligent Systems*, 33(3):4–14.

Edmundson, H. P. (1969). New methods in automatic extracting. *J. ACM*, 16(2):264–285.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

Fang, C., Mu, D., Deng, Z., and Wu, Z. (2017). Word-sentence co-ranking for automatic extractive text summarization. *Expert Systems with Applications*, 72:189 – 195.

Fattah, M. (2014). A hybrid machine learning model for multi-document summarization. *Applied Intelligence*, 40:592–600.

Fattah, M. A. and Ren, F. (2009). Ga, mr, ffnn, pnn and gmm based models for automatic text summarization. *Computer Speech & Language*, 23(1):126 – 144.

Feng, C., Cai, F., Chen, H., and de Rijke, M. (2018). Attentive encoder-based extractive text summarization. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1499–1502, New York, NY, USA. Association for Computing Machinery.

FIDALGO, E., Alegre, E., González-Castro, V., and Fernández-Robles, L. (2017). Illegal activity categorisation in darknet based on image classification using creic method. In *SOCO-CISIS-ICEUTE*.

Gambhir, M. and Gupta, V. (2017). Recent automatic text summarization techniques: A survey. *Artif. Intell. Rev.*, 47(1):1–66.

Gangwar, A. K., FIDALGO, E., Alegre, E., and González-Castro, V. (2017). Pornography and child sexual abuse detection in image and video: A comparative evaluation. In *Proceedings of 8th International Conference on Imaging for Crime Detection and Prevention*.

Gao, D., Li, W., You, O., and Zhang, R. (2012). Lda-based topic formation and topic-sentence reinforcement for graph-based multi-document summarization. In Hou, Y., Nie, J., Sun, L., Wang, B., and Zhang, P., editors, *Information Retrieval Technology, 8th Asia Information Retrieval Societies Conference, AIRS 2012, Tianjin, China, December 17-19, 2012. Proceedings*, volume 7675 of *Lecture Notes in Computer Science*, pages 376–385. Springer.

Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., and Heck, L. (2016). Contextual lstm (clstm) models for large scale nlp tasks. *ArXiv*, abs/1602.06291.

Gialitsis, N., Pittaras, N., and Stamatopoulos, P. (2019). A topic-based sentence representation for extractive text summarization. In *Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources*, pages 26–34, Varna, Bulgaria. INCOMA Ltd.

Gong, Y. and Liu, X. (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*, SIGIR '01, page 19–25, New York, NY, USA. Association for Computing Machinery.

Gupta, P., Pendluri, V. S., and Vats, I. (2011). Summarizing text by ranking text units according to shallow linguistic features. *13th International Conference on Advanced Communication Technology (ICACT2011)*, pages 1620–1625.

Gupta, V. and Lehal, G. S. (2010). A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3):258–268.

Haghighi, A. and Vanderwende, L. (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, Boulder, Colorado. Association for Computational Linguistics.

Hennig, L. (2009). Topic-based multi-document summarization with probabilistic latent semantic analysis. In *Proceedings of the International Conference RANLP-2009*, pages 144–149, Borovets, Bulgaria. Association for Computational Linguistics.

Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 50–57, New York, NY, USA. Association for Computing Machinery.

Huang, L., He, Y., Wei, F., and Li, W. (2010). Modeling document summarization as multi-objective optimization. In *Proceedings of the 2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, IITSI '10, page 382–386, USA. IEEE Computer Society.

Issam, K. A. R., Patel, S., and N, S. C. (2021). Topic modeling based extractive text summarization. *CoRR*, abs/2106.15313.

Jadhav, A. and Rajan, V. (2018). Extractive summarization with SWAP-NET: Sentences and words from alternating pointer networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 142–151, Melbourne, Australia. Association for Computational Linguistics.

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2015). On using very large target vocabulary for neural machine translation. 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: Long Papers)*, pages 1–10. Association for Computational Linguistics.

Ji, Y., Cohn, T., Kong, L., Dyer, C., and Eisenstein, J. (2016). Document context language models. In Bengio, Y. and LeCun, Y., editors, *4rd International Conference on Learning Representations, ICLR 2016, Puerto Rico, May 2-4, 2016, Workshop Track*, pages 1–10.

Jindal, S. G. and Kaur, A. (2020). Automatic keyword and sentence-based text summarization for software bug reports. *IEEE Access*, 8:65352–65370.

Joshi, A., Fidalgo, E., Alegre, E., and Fernández-Robles, L. (2019). Summcoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications*, 129:200 – 215.

Khandelwal, U., He, H., Qi, P., and Jurafsky, D. (2018). Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294, Melbourne, Australia. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28, pages 3294–3302. Curran Associates, Inc.

Ko, Y. and Seo, J. (2008). An effective sentence-extraction technique using contextual information and statistical approaches for text summarization. *Pattern Recognit. Lett.*, 29:1366–1371.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

Lau, J. H., Baldwin, T., and Cohn, T. (2017). Topically driven neural language model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 355–365, Vancouver, Canada. Association for Computational Linguistics.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Li, L., Zhou, K., Xue, G.-R., Zha, H., and Yu, Y. (2009). Enhancing diversity, coverage and balance for summarization through structure learning. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, page 71–80, New York, NY, USA. Association for Computing Machinery.

Li, P., Lam, W., Bing, L., Guo, W., and Li, H. (2017a). Cascaded attention based unsupervised information distillation for compressive summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2081–2090, Copenhagen, Denmark. Association for Computational Linguistics.

Li, P., Wang, Z., Ren, Z., Bing, L., and Lam, W. (2017b). Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 345–354, New York, NY, USA. Association for Computing Machinery.

Li, P. and Yu, J. (2021). Extractive summarization based on dynamic memory network. *Symmetry*, 13(4).

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ling, Z., Luo, J., Wu, K., Yu, W., and Fu, X. (2015). Torward: Discovery, blocking, and traceback of malicious traffic over tor. *IEEE Transactions on Information Forensics and Security*, 10(12):2515–2530.

Litvak, M. and Last, M. (2008). Graph-based keyword extraction for single-document summarization. In *COLING 2008*.

Liu, J., Hughes, D. J. D., and Yang, Y. (2021). Unsupervised extractive text summarization with distance-augmented sentence graphs. In *SIGIR '21*, page 2313–2317, New York, NY, USA. Association for Computing Machinery.

Liu, Y. (2019). Fine-tune BERT for Extractive Summarization. *arXiv e-prints*, page arXiv:1903.10318.

Liu, Y. (2019). Fine-tune BERT for extractive summarization. *CoRR*, abs/1903.10318.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *cite arxiv:1907.11692*.

Lloret, E. and Palomar, M. (2013). Tackling redundancy in text summarization through different levels of language analysis. *Computer Standards and Interfaces*, 35(5):507 – 518.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Mandal, S., Achary, P., Phalke, S., Poorvaja, K., and Kulkarni, M. (2021). Extractive text summarization using supervised learning and natural language processing. In *2021 International Conference on Intelligent Technologies (CONIT)*, pages 1–7.

Mao, X., Yang, H., Huang, S., Liu, Y., and Li, R. (2019). Extractive summarization using supervised and unsupervised learning. *Expert Systems with Applications*, 133:173 – 181.

Matsuo, Y. and Ishizuka, M. (2003). Keyword extraction from a single document using word co-occurrence statistical information. *Int. J. Artif. Intell. Tools*, 13:157–169.

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, page 557–564, Berlin, Heidelberg. Springer-Verlag.

Mehta, P., Arora, G., and Majumder, P. (2018). Attention based Sentence Extraction from Scientific Articles using Pseudo-Labeled data. *arXiv e-prints*, page arXiv:1802.04675.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Mikolov, T. and Zweig, G. (2012a). Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 234–239.

Mikolov, T. and Zweig, G. (2012b). Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 234–239.

Mohamed, M. A. and Oussalah, M. (2019). Srl-esa-textsum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Inf. Process. Manag.*, 56:1356–1372.

Mohd, M., Jan, R., and Shah, M. (2020). Text document summarization using word embedding. *Expert Systems with Applications*, 143:112958.

Moratanch, N. and Chitrakala, S. (2017). A survey on extractive text summarization. In *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pages 1–6.

Mutlu, B., Sezer, E. A., and Akcayol, M. A. (2020). Candidate sentence selection for extractive text summarization. *Information Processing & Management*, 57(6):102359.

Nagwani, N. K. (2015). Summarizing large text collection using topic modeling and clustering based on mapreduce framework. *J. Big Data*, 2:6.

Nakamoto, S. et al. (2008). Bitcoin: A peer-to-peer electronic cash system. *NA*.

Nallapati, R., Zhai, F., and Zhou, B. (2017a). SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3075–3081. AAAI Press.

Nallapati, R., Zhou, B., dos Santos, C., Gúlçehre, Ç., and Xiang, B. (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, Berlin, Germany. Association for Computational Linguistics.

Nallapati, R., Zhou, B., and Ma, M. (2017b). Classify or select: Neural architectures for extractive document summarization. *ArXiv*, abs/1611.04244.

Narayan, S., Cohen, S. B., and Lapata, M. (2018a). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Narayan, S., Cohen, S. B., and Lapata, M. (2018b). Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Narayan, S., Papasarantopoulos, N., Lapata, M., and Cohen, S. B. (2017). Neural extractive summarization with side information. *ArXiv*, abs/1704.04530.

Nenkova, A. and Vanderwende, L. (2005). The impact of frequency on summarization. Technical report, Microsoft Research.

Norbutas, L. (2018). Offline constraints in online drug marketplaces: An exploratory analysis of a cryptomarket trade network. *The International journal on drug policy*, 56:92–100.

Ouyang, Y., Li, W., Li, S., and Lu, Q. (2011). Applying regression models to query-focused multi-document summarization. *Information Processing and Management*, 47(2):227–237.

Padmakumar, V. and He, H. (2021). Unsupervised extractive summarization using pointwise mutual information. *CoRR*, abs/2102.06272.

Parveen, D., Ramsl, H.-M., and Strube, M. (2015). Topical coherence for graph-based extractive summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1949–1954, Lisbon, Portugal. Association for Computational Linguistics.

Parveen, D. and Strube, M. (2015). Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 1298–1304. AAAI Press.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1310–III–1318. JMLR.org.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Rahman, A., Rafiq, F. M., Saha, R., Rafian, R., and Arif, H. (2019). Bengali text summarization using textrank, fuzzy c-means and aggregate scoring methods. In *2019 IEEE Region 10 Symposium (TENSYMP)*, pages 331–336.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ren, P., Chen, Z., Ren, Z., Wei, F., Ma, J., and de Rijke, M. (2017). Leveraging contextual sentence relations for extractive summarization using a neural attention model. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 95–104, New York, NY, USA. Association for Computing Machinery.

Ron, D. and Shamir, A. (2014). How did dread pirate roberts acquire and protect his bitcoin wealth? In *International Conference on Financial Cryptography and Data Security*, pages 3–15. Springer.

Rudesill, D. S., Caverlee, J., and Sui, D. (2015). The deep web and the darknet: A look inside the internet's massive black box. *Woodrow Wilson International Center for Scholars, STIP*, 3.

Rush, A. M., Chopra, S., and Weston, J. (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, Lisbon, Portugal. Association for Computational Linguistics.

Saikia, S., FIDALGO, E., Alegre, E., and Fernández-Robles, L. (2017a). Object detection for crime scene evidence analysis using deep learning. In *ICIAP*.

Saikia, S., Fidalgo, E., Alegre, E., and Fernández-Robles, L. (2017b). Query based object retrieval using neural codes. In García, H. P., Alfonso-Cendón, J., Sánchez-González, L., Quintián, H., and Corchado, E., editors, *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17, León, Spain, September 6-8, 2017, Proceedings*, volume 649 of *Advances in Intelligent Systems and Computing*, pages 513–523. Springer.

Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., and Pérez, C. J. (2020). Experimental analysis of multiple criteria for extractive multi-document text summarization. *Expert Systems with Applications*, 140:112904.

Shi, J., Liang, C., Hou, L., Li, J., Liu, Z., and Zhang, H. (2019). Deepchannel: Salience estimation by contrastive learning for extractive document summarization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6999–7006. AAAI Press.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA. PMLR.

Tang, H., Li, M., and Jin, B. (2019). A topic augmented text generation model: Joint learning of semantics and structural features. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5090–5099, Hong Kong, China. Association for Computational Linguistics.

Tarnpradab, S., Liu, F., and Hua, K. A. (2017). Toward extractive summarization of online forum discussions via hierarchical attention networks. In Rus, V. and Markov, Z., editors, *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2017, Marco Island, Florida, USA, May 22-24, 2017*, pages 288–292. AAAI Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Vhatkar, A., Bhattacharyya, P., and Arya, K. (2020). Knowledge graph and deep neural network for extractive text summarization by utilizing triples. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 130–136, Barcelona, Spain (Online). COLING.

Wan, X. (2007). Using only cross-document relationships for both generic and topic-focused multi-document summarizations. *Information Retrieval*, 11:25–49.

Wan, X. (2010). Towards a unified approach to simultaneous single-document and multi-document summarizations. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1137–1145, Beijing, China. Coling 2010 Organizing Committee.

Wang, D., Li, T., Zhu, S., and Ding, C. (2008). Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 307–314, New York, NY, USA. Association for Computing Machinery.

Wang, F., Orton, K., Wagenseller, P., and Xu, K. (2018). Towards understanding community interests with topic modeling. *IEEE Access*, 6:24660–24668.

Wong, K.-F., Wu, M., and Li, W. (2008). Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, page 985–992, USA. Association for Computational Linguistics.

Wood, J. (2010). The darknet: A digital copyright revolution. *Richmond Journal of Law and Technology*, 16:14.

Woodsend, K. and Lapata, M. (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*, EMNLP-CoNLL '12, page 233–243, USA. Association for Computational Linguistics.

Wu, Y. and Hu, B. (2018). Learning to extract coherent summary via deep reinforcement learning. In *AAAI*.

Wu, Z., Lei, L., Li, G., Huang, H., Zheng, C., Chen, E., and Xu, G. (2017). A topic modeling based approach to novel document automatic summarization. *Expert Systems with Applications*, 84:12 – 23.

Xu, J. and Durrett, G. (2019). Neural extractive text summarization with syntactic compression. *arXiv*, 1902.00863.

Yao, K., Zhang, L., Luo, T., and Wu, Y. (2018). Deep reinforcement learning for extractive document summarization. *Neurocomputing*, 284:52 – 62.

Yeh, J.-Y., Ke, H.-R., Yang, W.-P., and Meng, I.-H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing & Management*, 41(1):75 – 95. An Asian Digital Libraries Perspective.

You, F., Zhao, S., and Chen, J. (2020). A topic information fusion and semantic relevance for text summarization. *IEEE Access*, 8:178946–178953.

Yousefi-Azar, M. and Hamey, L. (2017). Text summarization using unsupervised deep learning. *Expert Systems with Applications*, 68:93 – 105.

Zajic, D. M., Dorr, B. J., and Lin, J. (2008). Single-document and multi-document summarization techniques for email threads using sentence compression. *Information Processing & Management*, 44(4):1600 – 1610.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019a). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *ArXiv*, abs/1912.08777.

Zhang, X., Lapata, M., Wei, F., and Zhou, M. (2018). Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 779–784, Brussels, Belgium. Association for Computational Linguistics.

Zhang, X., Wei, F., and Zhou, M. (2019b). HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

Zhang, X., Zhao, J. J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.

Zheng, H. and Lapata, M. (2019). Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., and Huang, X. (2020). Extractive summarization as text matching. *arXiv*, 2004.08795.

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., and Zhao, T. (2018). Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia. Association for Computational Linguistics.

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., and Zhao, T. (2020). A joint sentence scoring and selection framework for neural extractive document summarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:671–681.

Annex B

SUMMARY OF THE THESIS IN SPANISH

RESUMEN DE LA TESIS EN CASTELLANO

En cumplimiento del punto 7 de la normativa complementaria del Real Decreto 778/1998, de 30 de Abril y de las normas para la aplicación del mismo, aprobadas por acuerdo de la Junta de Gobierno de fecha 10 de mayo de 1999, se adjunta un resumen en castellano de cada uno de los capítulos de esta tesis doctoral para que pueda admitirse a trámite.

1 Introducción

1.1 Motivación

Con la llegada de Internet, se generan a diario una cantidad masiva de datos textuales. Por este motivo, surge la necesidad de generar herramientas automáticas que sean capaces de procesar y extraer información de interés de dichos datos, así como de representar dichos contenidos textuales en formas más reducidas y concisas llamadas resumen de texto.

El resumen de texto automático es una rama muy importante del Procesamiento del Lenguaje Natural. Su objetivo es la representación de documentos de texto en una forma más reducida, conteniendo la información más relevante para un usuario final. Los resúmenes de texto se clasifican en dos categorías; extractivos y abstractivos(Gambhir and Gupta, 2017) (Gambhir and Gupta, 2017). Los resúmenes extractivos están formados por la concatenación de las oraciones más relevantes del documento original. Estos resúmenes se generan habitualmente siguiendo tres fases: representación en forma de vector de características del texto de entrada, asignación de puntuaciones a cada oración del texto, y selección de dichas oraciones en base a las puntuaciones obtenidas. En cambio, los resúmenes de texto abstractivos generan resúmenes parafraseando el contenido principal del documento utilizando técnicas de generación de lenguaje natural.

El resumen de texto se puede clasificar también como resumen de un documento o de varios documentos, en base al número de documentos de entrada proporcionados (Zajic et al., 2008; Fattah and Ren, 2009). También, los resúmenes pueden ser genéricos o basado en consultas (Gong and Liu, 2001; Dunlavy et al., 2007; Wan, 2007; Ouyang et al., 2011). El resumen genérico proporciona una idea general del contenido del documento, mientras que el resumen basado en consultas presenta contenido relevante del documento en base a las consultas realizadas por un usuario. En esta tesis, nos hemos centrado en la tarea de resúmenes extractivos de texto de un único documento.

Varios métodos tradicionales para el resumen de textos extractivo propuestos en la literatura se basan principalmente en características diseñadas por humanos, como por ejemplo, la combinación de características estadísticas y lingüísticas como la frecuencia de términos (Luhn, 1958; Nenkova and Vanderwende, 2005), la longitud y la posición de la oración (Erkan and Radev, 2004) o palabras clave y de estigma (Edmundson, 1969). En estos métodos, se asigna una puntuación a cada oración en función de sus características. Para seleccionar oraciones para generar un resumen, se han propuesto varias técnicas,

incluyendo enfoques codiciosos (Carbonell and Goldstein, 1998), enfoques basados en grafos (Erkan and Radev, 2004; Mihalcea and Tarau, 2004; Wan, 2010; Parveen et al., 2015) y enfoques basados en optimización (McDonald, 2007), entre otros.

Desde hace unos años, los métodos basados en aprendizaje profundo han alcanzado resultados del estado de la técnica en muchas tareas de Procesamiento del Lenguaje Natural (PNL), como responder preguntas (Bordes et al., 2014), comprensión del lenguaje natural (Collobert et al., 2011), análisis de sentimientos (dos Santos and Gatti, 2014), clasificación de texto Zhang et al. (2015), traducción de idiomas (Jean et al., 2015), e incluso resúmenes de texto (Rush et al., 2015; Nallapati et al., 2017b,a, 2016). Las redes neuronales profundas representan los datos con múltiples niveles de abstracción después de procesarlos en varias capas de cálculo intensivo no lineal. Para obtener una representación buena y semánticamente significativa de los datos de entrada, las redes de aprendizaje profundo deben recibir una gran cantidad de datos de entrenamiento. La mayoría de los enfoques basados en el aprendizaje profundo, como las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN), requieren datos etiquetados para entrenar las arquitecturas de redes profundas.

La Web superficial está formada por todos aquellos dominios que están indexados por motores de búsqueda, como Google, Yahoo o Bing. En cambio, la Web Profunda (del inglés, Deep Web) está representada por todo aquel contenido no indexado por dichos motores, como puede ser el almacenado en redes privadas de empresa o nuestras cuentas de Google Drive. Dentro de la Web Profund, existen un conjunto de redes, denominadas Redes Oscuras (del inglés, DarkNets) que además de hospedar contenido no indexado requieren de métodos adicionales para acceder a las mismas. Tor (del inglés, The Onion Router [1] es una de las Redes Oscuras más populares, debido a la capa de anonimato que ofrece a sus usuarios, y sus dominios o sitios web se denominan Servicios Ocultos (del inglés, Hidden Services). Se puede acceder a estos Servicios a través del software Tor Browser (https://www.torproject.org/projects/torbrowser.html.en), o a través de un proxy, como Tor2Web (https://tor2web.org/).

El Instituto Nacional de Ciberseguridad (INCIBE) trabaja con Fuerzas y Cuerpos de Seguridad del Estado (FFCCSE), proporcionándoles herramientas y servicios que faciliten su lucha contra el cibercrimen. Gracias a nuestra colaboración con INCIBE, hemos tenido la oportunidad de integrar parte de la investigación realizada en esta tesis en dichas herramientas, permitiendo que las FFCCSE puedan automatizar tareas que habitualmente realizan durante la monitorización de la red oscura Tor, en búsqueda de actividades que podrían ser sospechosas. Más específicamente, en esta tesis se ha trabajado en resolver el caso de uso de la obtención de resúmenes automáticos del contenido textual de dominios de la red Tor.

Hemos contribuido con tres enfoques diferentes (SummCoder, DeepSumm y RankSum) utilizando métodos de aprendizaje profundo y esquemas de fusión para mejorar la

[1] www.torproject.org

precisión de los resúmenes de texto extractivos. A continuación, presentamos la motivación para cada uno de los marcos de resumen propuestos.

1.1.1 SummCoder

En la actualidad, uno de los mayores desafíos del aprendizaje profundo supervisado para la obtención de resúmenes de texto extractivos es la falta de disponibilidad de grandes cantidades de resúmenes creados manualmente para entrenar las redes. El método que proponemos, SummCoder, aborda esta deficiencia explotando técnicas que no requieren datos etiquetados para el entrenamiento a través del aprendizaje profundo no supervisado basado en codificadores automáticos e incrustaciones de oraciones. Los resúmenes se generan con la combinación de tres tipos de puntuaciones: de prominencia, posición, y novedad. La puntuación de prominencia, novedad y posición de las oraciones se combinan mediante una fusión ponderada para generar la puntuación final para clasificar las oraciones y la generación de resúmenes.

1.1.2 DeepSumm

A pesar de tener tanta popularidad, los métodos de redes neuronales tienen algunas limitaciones a la hora de realizar resúmenes de texto extractivos. Estos métodos no capturan la información latente del tópico de los documentos (Dieng et al., 2016) y, por lo tanto, el resumen se encuentra en un espacio de incrustación que apenas contiene información sobre dicho tópico. Aparte de esto, las variantes de Redes Neuronales Recurrentes (RNN), como la unidad recurrente cerrada (GRU) (Chung et al., 2014) y las redes de memoria a largo y corto plazo (LSTM) (Hochreiter and Schmidhuber, 1997) tienen una capacidad muy limitada para retener relaciones semánticas durante todo el documento (Khandelwal et al., 2018).

Para tratar de superar las carencias presentadas de la literatura, presentamos *DeepSumm*, un método para generar resúmenes de texto automáticos utilizando la información semántica global, junto con la información semántica y sintáctica local del documento . Las redes LSTM son capaces de extraer la información semántica y sintáctica local, así como manejar dependencias de largo alcance hasta cierto punto. Sin embargo, enriquecer las redes LSTM con información sobre tópicos permite capturar el significado global incrustado en el documento, lo cual es bastante útil para generar resúmenes. Nuestro método propuesto obtiene un resumen después de seleccionar oraciones del documento original, una vez han sido puntuadas usando la fusión de cuatro puntuaciones: Puntuación de tópico de oración (STS), Puntuación de contenido de oración (SCS), Puntuación de novedad de oración (SNS) y Puntuación de posición de oración (SPS).

1.1.3 RankSum

Aunque los enfoques de aprendizaje profundo propuestos recientemente (Ren et al., 2017; Nallapati et al., 2017a; Cheng and Lapata, 2016; Zhou et al., 2020; Joshi et al., 2019; Liu, 2019; Bahdanau et al., 2015) ofrecen buenas soluciones para generar resúmenes de texto abstractivo, la mayoría están basados en enfoques de aprendizaje supervisado. En el último capítulo de la tesis proponemos RankSum, un enfoque no supervisado basado en la fusión de rankings de las oraciones del documento calculados a través de cuatro características: tópico, semántica, palabras clave e información posicional de la oración.

El contenido del tópico (Blei, 2012) de un documento puede capturar la prominencia global del documento, y se ha implementado para comprender las dependencias de largo alcance en el documento (Mikolov and Zweig, 2012a). Hacemos uso de redes siameses (Bromley et al., 1993) con una función de pérdida de triplete para obtener incrustaciones de oraciones que representen eficientemente la semántica de las oraciones.

En cuanto a las palabras clave, trabajamos con la hipótesis de que las oraciones más significativas de un documento contendrían las palabras clave más significativas de dicho documento. Por ello, hemos seguido otros trabajos de la literatura (Jindal and Kaur, 2020; Litvak and Last, 2008; Matsuo and Ishizuka, 2003) para obtener palabras clave en el texto que se puedan utilizar en tareas de resúmenes de textos.

Para identificar la posible redundancia en el texto de salida resultante utilizamos la incrustación de oraciones, bigramas y trigramas. A través de experimentos, demostramos que cada característica de la oración es significativa para la generación de buenos resúmenes, sin embargo, diferentes características se complementan entre sí y pueden producir una representación más significativa del documento.

2 Revisión del Estado de la Técnica

En las últimas décadas, ha habido una enorme cantidad de trabajo en los métodos de resumen de texto. En este capítulo, revisaremos los enfoques más avanzados que se han aplicado para los métodos de resumen extractivos de texto.

2.1 Enfoques basados en aprendizaje automático

En la literatura existe una amplia investigación relacionada con la obtención de resúmenes extractivos de texto a través de métodos de aprendizaje automático supervisados, semi-supervisados y no supervisados.

En los enfoques basados en el aprendizaje supervisado (Haghighi and Vanderwende, 2009; Fattah and Ren, 2009; Li et al., 2009; Ouyang et al., 2011; Cheng and Lapata, 2016; Nallapati et al., 2017a), como las Máquinas de Vectores de Soporte (SVM), los árboles de regresión o las redes neuronales (Fattah and Ren, 2009) necesitan datos etiquetados (denominados resúmenes dorados en esta tarea) para poder entrenar modelos.

Por el contrario, los sistemas no supervisados (Dunlavy et al., 2007; Wang et al., 2008; Fattah and Ren, 2009; Parveen et al., 2015; Fang et al., 2017) clasifican o rankean las oraciones utilizando técnicas heurísticas, y no necesitan resúmenes dorados creados manualmente para realizar el entrenamiento (Yousefi-Azar and Hamey, 2017). Por este motivo, los métodos no supervisados se pueden adaptar fácilmente a nuevos dominios sin mucha alteración. Los métodos de aprendizaje no supervisado se basan principalmente en características hechas a mano, como la frecuencia de los términos como Luhn (Luhn, 1958), SumBasic (Nenkova and Vanderwende, 2005), la posición y la longitud de la oración, como LexRank (Erkan and Radev, 2004) o palabras clave y estigmatizadas. (Edmundson, 1969). Las características seleccionadas en estos enfoques se utilizan para obtener una puntuación de clasificación para cada oración.

2.2 Métodos de aprendizaje profundo

Para realizar resúmenes de texto extractivos, la mayoría de la literatura reciente utiliza Redes de Secuencia GRU o de memoria a largo-corto plazo LSTM. Nallapati et al. (2017a) propuso SummaRuNNER, un modelo de secuencia basado en GRU-RNN que puede entrenarse de forma extractiva y abstractiva para generar resúmenes. El trabajo de Nallapati et al. (2017b) involucró dos arquitecturas, clasificador y selector, basadas en GRU-RNNS para realizar resúmenes de texto extractivos. Zhou et al. (2018) presentó NeuSum, una arquitectura de codificador de documentos y frases jerárquicas de extremo a extremo que utiliza GRU-RNN para puntuar y seleccionar frases de forma conjunta. Shi et al. (2019) introdujo DeepChannel, un marco de resumen extractivo que consiste en una combinación de RNN-GRU que permite estimar la prominencia y una estrategia de extracción de oraciones codiciosas guiada por la prominencia.

Las propuestas basadas en redes LSTM incluyen trabajos como Cheng and Lapata (2016), que empleó un enfoque de codificador-decodificador para seleccionar las oraciones y palabras más relevantes para componer el resumen extractivo de salida. (Jadhav and Rajan, 2018) diseñó SWAP-NET, para modelar las interacciones de oraciones destacadas y palabras clave en documentos para producir resúmenes extractivos. (Narayan et al., 2018b) conceptualizó el resumen extractivo como una tarea de clasificación de oraciones utilizando un codificador de documentos y un extractor de oraciones basados en LSTM.

Narayan et al. (2017) diseñó un codificador de documentos LSTM jerárquico y un extractor basado en la atención con atención sobre información lateral. (Zhang et al., 2018) construyó un modelo extractivo latente basado en una red LSTM, que en lugar de maximizar la probabilidad de pertenencia al etiquetado de los resúmenes dorados, maximiza directamente la probabilidad de resúmenes humanos dadas oraciones seleccionadas. (Tarnpradab et al., 2017) utilizó redes de atención jerárquicas basadas en LSTM para el resumen extractivo de foros. (Liu, 2019) ajustó un modelo BERT, que es una arquitectura de transformadores basada en un codificador-decodificador previamente entrenado para aumentar la precisión de la tarea de resúmenes extractivos.

2.3 Enfoques basados en tópicos

Narayan et al. (2018b) propuso redes de secuencia a secuencia (Seq2Seq) para realizar resúmenes extremos (resúmenes de una línea) de artículos de noticias. Los autores demostraron experimentalmente que las capas de convolución capturan las dependencias de largo alcance en el documento mejor que las RNN, lo cual es útil para realizar abstracciones e inferencias a nivel de documento.

Mehta et al. (2018) propuso codificadores de secuencia basados en LSTM que utilizan conjuntamente modelos de tópicos para aprender los pesos de atención en las palabras de las oraciones para producir resúmenes de artículos científicos.

2.4 Métodos basados en Fusion

Existen muchos menos trabajos que exploren la fusión de diferentes características o algoritmos para realizar resúmenes de textos. Dutta et al. (2018) presentó un algoritmo que combinó las salidas de varios algoritmos de resúmenes de textos para producir resúmenes finales. Joshi et al. (2019) introdujo un método basado en la fusión ponderada de tres características de la oración: relevancia, novedad y posición. You et al. (2020) presentó la fusión de información del tópico y la relevancia semántica para el resumen de texto basado en el ajuste fino de BERT.

Wong et al. (2008) diseñó un enfoque basado en el aprendizaje utilizando varias características de la oración como la superficie, el contenido, la relevancia y el evento. Los autores combinaron todas las características mediante el aprendizaje semi-supervisado para minimizar la dependencia del conjunto de datos etiquetado para el resumen. Sin

embargo, su enfoque necesita datos etiquetados y, por lo tanto, depende del dominio para el que está capacitado. Mao et al. (2019) desarrolló tres métodos para fusionar y puntuar oraciones combinando relaciones de oraciones con características estadísticas de oraciones utilizando aprendizaje supervisado y no supervisado.

3 SummCoder: marco basado en aprendizaje no supervisado para el resumen extractivo de textos basado en codificadores automáticos profundos

Debido a problemas de derechos de autor, hemos eliminado este capítulo de la tesis. Aquí están los detalles del artículo publicado:
Akanksha Joshi, E. Fidalgo, E. Alegre, Laura Fernández-Robles, SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders, Expert Systems with Applications, Volume 129, 2019, Pages 200-215,https://doi.org/10.1016/j.eswa.2019.03.045.

4 DeepSumm: Explotación de modelos de tópicos y redes de secuencia para realizar resúmenes de texto extractivos

En este capítulo, analizaremos nuestro enfoque basado en modelos de tópicos y aprendizaje profundo para aumentar la precisión del resumen de texto.

4.1 DeepSumm

4.1.1 Formulación del problema

Formulamos el problema del resumen de textos extractivo como una selección de oraciones, en base a unas puntuaciones previas asignadas a las mismas, para formar el resumen final.

Dado un documento D compuesto por una secuencia de N oraciones (S_1, S_2, ..., S_N) y una secuencia de M palabras como (w_1, w_2, ..., w_M), el resumen se genera mediante un subconjunto de N que contiene las oraciones más relevantes del documento. La relevancia de las oraciones se determina en función de su contenido estructural, información del tópico, posición relativa y novedad de esa oración en el documento. En este trabajo, las redes de secuencia de auto-atención (Vaswani et al., 2017) se emplean para codificar la información en el documento. Dichas redes son apropiadas para identificar el contexto estructural local debido a su naturaleza secuencial. En las sub-secciones siguientes, describiremos brevemente los componentes clave de DeepSumm, cuyo resumen gráfico se presenta en la Figura 1.

4.1.2 Distribución probabilística de tópicos por palabra

Blei (2012) propuso los modelos probabilísticos de tópicos para capturar la estructura semántica global de los documentos. El objetivo principal de los métodos de modelado de tópicos es modelar documentos como colecciones de múltiples tópicos latentes. Cada tópico puede verse como una distribución de términos semánticamente coherentes y cada documento exhibe estos tópicos con diferentes probabilidades o proporciones.

Uno de los modelos de tópicos probabilísticos es la asignación de Dirichlet latente (LDA) (Blei et al., 2003), cuyo objetivo principal es encontrar los K tópicos latentes $T = \{T_1, T_2, ..., T_k\}$ en un colección de documentos donde cada tópico es una colección de palabras que tienden a coexistir juntas. LDA es mejor que otros modelos de tópicos: LSA (Landauer et al., 1998) y pLSA (Hofmann, 1999) ya que LDA se generaliza bien para documentos nuevos y tiene menos riesgo de sobreajuste. Usamos LDA para generar vectores de tópico T_D para cada documento presente en la distribución y vectores de tópico para cada palabra, como t_{w_j} para j^{th} palabra, w_j, en un documento. En este trabajo, consideramos los vectores de tópico para cada palabra T_j como una suma puntual del vector de tópico de palabra t_{w_j} generado por LDA, más el vector de tópico del documento t_D, como: $T_j = t_{w_j} + t_D$.

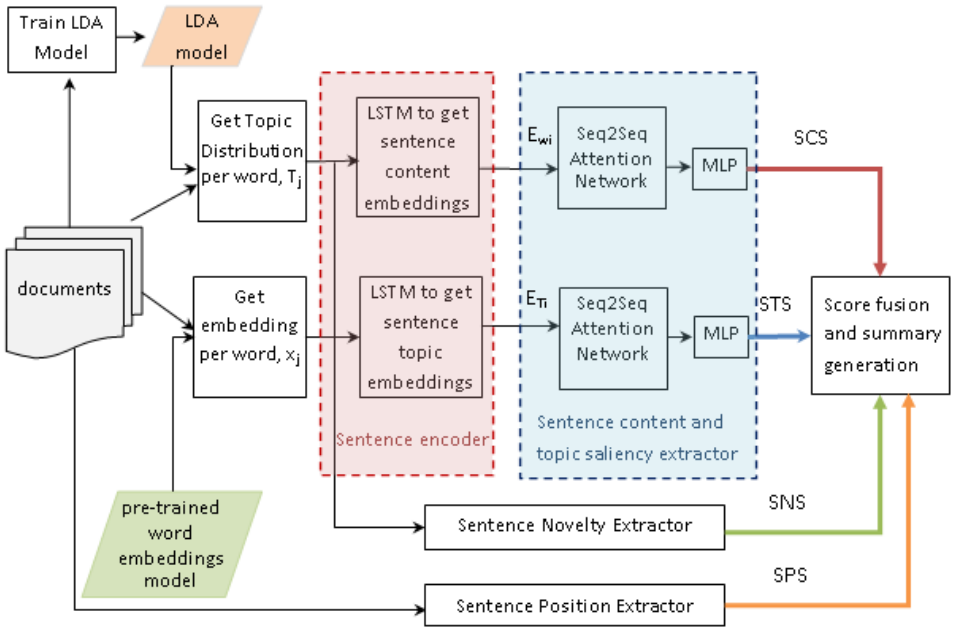


Figura 1: Esquema de la arquitectura DeepSumm

4.1.3 Incrustaciones de palabras

En este trabajo hemos calculado las incrustaciones de palabras en el documento para capturar la información del contenido estructural. Hemos utilizado los vectores de palabras pre-entrenados GloVe (Pennington et al., 2014) para representar cada palabra del documento como x_j en un espacio de incrustación d-dimensional, $R^{M\times d}$.

4.1.4 Codificador de frases

Codificamos vectores de tópico por palabra T_j y los vectores de palabra x_j como vectores de oración por medio de dos redes LSTM bidireccionales. Por un lado, una red LSTM bidireccional utiliza los vectores de tópico de cada palabra, T_j, en una oración como entrada para extraer la inserción de la oración, denominada E_{Ti}, que se relaciona con la información del tópico de la oración i^{th}. La red LSTM lee en una dirección (hacia adelante) la oración S_i desde T_{i1} a T_{im} y también hacia atrás desde T_{im} hasta T_{i1}. La codificación E_{Ti} se produce concatenando los estados de salida ocultos finales de la red, $\overrightarrow{h_{Tt}}$ y $\overleftarrow{h_{Tt}}$ en ambas direcciones (hacia adelante/atrás).

$$\overrightarrow{h_{Tt}} = \text{LSTM}(T_t, \overrightarrow{h_{T(t-1)}}) \tag{1}$$

$$\overleftarrow{h_{Tt}} = \text{LSTM}(T_t, \overleftarrow{h_{T(t+1)}})) \tag{2}$$

$$E_{Ti} = [\overrightarrow{h_{Tt}}, \overleftarrow{h_{Tt}}] \tag{3}$$

Por otro lado, pero de manera similar, las incrustaciones de palabras x_{im}, de una oración i se utilizan como entrada a otra red LSTM bidireccional para extraer la incrustación de la oración, E_{wi}. El LSTM hacia adelante (del inglés, forward LSTM) usado para producir E_{wi} lee la oración S_i desde x_{i1} a x_{im} y el LSTM hacia atrás (del inglés, backward LSTM) desde x_{im} a x_{i1}.

$$\overrightarrow{h_{xt}} = \text{LSTM}(x_t, \overrightarrow{h_{x(t-1)}})) \tag{4}$$

$$\overleftarrow{h_{xt}} = \text{LSTM}(x_t, \overleftarrow{h_{x(t+1)}})) \tag{5}$$

$$E_{wi} = [\overrightarrow{h_{xt}}, \overleftarrow{h_{xt}}] \tag{6}$$

4.1.5 Extractor de contenido de oraciones y saliencia de tópicos

En esta sección, se diseñan dos metodologías para realizar el cálculo de las incrustaciones de tópicos de oraciones, E_{Ti}, y el otro para incrustaciones de oraciones, E_{wi}, ambos basados en vectores de palabras. Se usarán redes de Secuencia a Secuencia (seq2seq), formadas por un codificador y decodificador LSTM, para obtener la prominencia de las oraciones utilizando vectores de palabras y tópicos.

El codificador consta de una red LSTM bidireccional que toma las incrustaciones de oraciones E_{wi} como entrada para generar una representación de documento codificada. El decodificador también está compuesto por un LSTM bidireccional que tiene en cuenta las incrustaciones de oraciones y las salidas del codificador ponderadas por atención para producir estados ocultos del decodificador, $\overrightarrow{h_{D_w i}}$ y $\overleftarrow{h_{D_w i}}$. Las salidas del decodificador y del codificador ponderado por atención finalmente se alimentan a una red MLP para generar puntuaciones para cada oración en el documento.

Por lo tanto, la puntuación del contenido de la oración (SCS) se calcula ingresando las incrustaciones de la oración E_{wi} en la red seq2seq. La puntuación de tópico de oración (STS) se obtiene ingresando las codificaciones de oración de distribución de tópico E_{Ti} a otra red seq2seq diseñada para codificar vectores de tópico de oración.

$$SCS_i = P(y_i = 1|E_{wi}) \tag{7}$$

$$STS_i = P(y_i = 1|E_{Ti}) \tag{8}$$

4.1.6 Extractor de novedad de oraciones

En esta subsección explicamos la nueva puntuación de la novedad de la frase propuesta, que escanea de una en una, progresivamente, todas las frases del documento, y las asigna una puntuación en función de la novedad de la frase con respecto a todas las anteriores.

La novedad de cada oración se calcula en función de las incrustaciones de oración E_{wi} y las codificaciones de oración de distribución de tópico E_{Ti}.

$$SNS_i = \begin{cases} 1 & \text{if } i = 1 \\ \frac{1}{i-1} \sum_{j=1}^{i-1} \frac{\left(1-(\text{Sim}(E_{wi},E_{wj})+\text{Sim}(E_{Ti},E_{Tj}))\right)}{2} & \text{otherwise} \end{cases} \tag{9}$$

Para obtener la novedad de la oración, se utilizan tanto el contenido de la oración como las representaciones de los tópicos, tal como se genera en la sección 1.4.1.4 . A través de incrustaciones de contenido de oraciones, podemos encontrar las oraciones que son semánticamente similares entre sí y, por lo tanto, podemos eliminar la redundancia en resumen. Al enriquecer el cálculo de la novedad con incrustaciones de tópicos de oraciones, las oraciones del resumen que mencionan tópicos similares se podrían descartar.

4.1.7 Extractor de posición de oración

En los documentos de noticias, las oraciones que aparecen en las primeras posiciones del documento se consideran más significativas en comparación con otras oraciones en el documento (Luhn, 1958; Edmundson, 1969). Por lo tanto, se propone una puntuación de posición de oración (SPS) que asigna a cada oración una puntuación relativa basada en su posición relativa en el documento. El SPS asignará puntuaciones más altas a las frases que se encuentran al principio del documento en comparación con las que aparecen en la parte posterior del documento.

$$SPS_i = \frac{N - P_i}{N} \tag{10}$$

4.1.8 Fusión de puntuaciones y generación de resumen

Como paso previo a la generación del resumen, se realiza una fusión de las puntuaciones SCS, STS, SNS y SPS para obtener una puntuación de oración final ponderada (FSS) para una oración i, con los parámetros α, β, γ, δ

$$FSS_i = \alpha \cdot SCS_i + \beta \cdot STS_i + \gamma \cdot SNS_i + \delta \cdot SPS_i \tag{11}$$

En FSS_i, la oración con la puntuación más alta se considera la más significativa para ser incluida en el resumen. Los valores de α, β, γ, y δ se determinan empíricamente. Finalmente, las oraciones de un documento se organizan en orden descendente con respecto

a su FSS y se seleccionan las oraciones o palabras k superiores de la lista para formar el resumen de texto extractivo del documento.

4.2 Experimentos y Resultados

4.2.1 Conjuntos de datos

Para entrenar nuestro método, hemos usado el conjunto de datos CNN/DailyMail (Hermann et al., 2015), que contienen 197,000 y 90,000 noticias, respectivamente. Como los resúmenes extractivos de los documentos de noticias no están disponibles, utilizamos los titulares (del inglés, highlights), que en realidad son resúmenes abstractivos, para producir sus resúmenes extractivos. En este trabajo se utiliza la división estándar de entrenamiento, prueba y validación para los conjuntos de datos como se indica en la Tabla 1. Para validar nuestro enfoque en un conjunto de datos diferente utilizamos el conjunto de datos DUC2002 [2].

Tabla 1: Información de bases de datos

Conjunto de datos	Tipo	Uso	# Documentos	# Categorías
CNN/DailyMail	News	Entrenamiento	287,227	-
		Validación	13,368	-
		Pruebas	11,490	-
DUC 2002	Noticias y pruebas	567	59	

4.2.2 Evaluación

Se han usado las métricas de ROUGE-1, ROUGE-2 y ROUGE-L (Lin, 2004) para evaluar y comparar nuestro enfoque con otros métodos del estado de la técnica. Hemos comparado DeepSumm contra los siguientes métodos: NN-SE (Cheng and Lapata, 2016), SummaRuNNer (Nallapati et al., 2017a), HSSAS (Al-Sabahi et al., 2018). Para el conjunto de datos de CNN/DailyMail, informamos los resultados mediante REFRESH (Narayan et al., 2018a), Bi-AES (Feng et al., 2018), RNES (Wu and Hu, 2018), NeuSum (Zhou et al., 2020), BERTSum (Liu, 2019),PACSum (Zheng and Lapata, 2019) y JECS (Xu and Durrett, 2019). Para DUC 2002, tomamos en consideración - ILP Woodsend and Lapata (2012)Egraph (Parveen and Strube, 2015), Tgraph (Parveen et al., 2015), URANK (Wan, 2010) CoRank (Fang et al., 2017) y SummCoder (Joshi et al., 2019).

4.2.3 Resultados

Como se muestra en Tabla 2, DeepSumm logra la puntuación ROUGE más alta para el resumen de textos extractivos de un solo documento en el conjunto de datos DUC 2002, con puntuaciones ROUGE-1, ROUGE-2 y ROUGE-L de 53,2, 28,7 y 49,2 respectivamente.

[2] https://duc.nist.gov/data.html

Ninguno de los enfoques de resumen basados en Redes Neuronales Recurrentes, como NN-SE, SummaRuNNer, SummCoder, HSSAS, utiliza información de tópicos latentes en el documento, lo que hace que DeepSumm obtenga resultados superiores al generar resúmenes extractivos de mayor calidad. Esto respalda la eficacia de nuestro marco propuesto que utiliza vectores de distribución de tópicos y modelos de lenguaje para derivar resúmenes extractos del documento.

Tabla 2: Análisis comparativo de DeepSumm con algoritmos del estado de la técnica en DUC 2002

Métodos	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
Lead	43.6	21.0	40.2
ILP	45.4	21.3	42.8
NN-SE	47.4	23.0	-
SummaRuNNer	47.4	24.0	14.7
Egraph + coh	47.9	23.8	-
Tgraph + coh	48.1	24.3	
URANK	48.5	21.5	-
SummCoder	51.7	27.5	44.6
HSSAS	52.1	24.5	48.8
CoRank	52.6	25.8	-
DeepSumm	**53.2**	**28.7**	**49.2**

En el conjunto de datos de CNN/ DailyMail, nuestro método obtuvo la puntuación ROUGE-1 más alta, 43,3, y puntuaciones ROUGE-2 y ROUGE-L, 19,0 y 38,9, comparables a los mejores enfoques de resumen extractivo de la literatura, como se puede ver en la Tabla 3. El notable aumento en la precisión en comparación con la mayoría de los enfoques recientes demostró que nuestro método es bastante sólido para producir buenos resúmenes. El método DeepSumm propuesto es capaz de condensar la información destacada del documento, que de otro modo no se captura solo utilizando modelos de lenguaje y, por lo tanto, aumenta la precisión general del resumen extractivo.

Tabla 3: Análisis comparativo de DeepSumm con algoritmos de última generación en CNN/DailyMail

Métodos	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
NN-SE	35.5	14.7	32.2
Bi-AES	38.8	12.6	33.85
LEAD	39.2	15.7	35.5
SummaRuNNer	39.6	16.2	35.3
REFERESH	40.0	18.2	36.6
PACSUM (BERT)	40.7	17.8	36.9
RNES w/o coherence	41.2	18.8	37.7
NeuSum	41.5	19.0	37.9
JECS	41.7	18.5	37.9
HSSAS	42.3	17.8	37.6
BertSum	43.2	**20.2**	**39.6**
DeepSumm	**43.3**	19.0	38.9

4.3 Conclusiones

En este capítulo de la tesis hemos presentado DeepSumm, un método novedoso para el resumen extractivo que produce representaciones compactas de un solo documento.

El método propuesto genera resúmenes a través de la selección de las oraciones del documento original usando la puntuación obtenida por las mismas tras la combinación de cuatro puntuaciones propuestas: puntuación de contenido, de tópico, de novedad y de posición de las frases.

Para seleccionar las oraciones más relevantes del documento y a utilizar para generar el resumen final, se realizó una fusión ponderada de las cuatro métricas propuestas.

Los resultados experimentales demostraron que DeepSumm obtuvo los mejores resultados con respecto a los métodos contra los que se comparó en el conjunto de datos DUC 2002, y logró un rendimiento cercano a los métodos del estado de la técnica en el conjunto de datos de CNN/DailyMail. En el futuro, exploraremos métodos para derivar otras características abstractas que contribuyan al resumen y también haremos uso de distribuciones temáticas probabilísticas para el resumen de texto abstractivo.

5 RankSum: resumen de textos extractivo no supervisado basado en la fusión de rangos

En este capítulo de la tesis se revisará RankSum, un enfoque no supervisado para obtener resúmenes de textos extractivos utilizando oraciones seleccionadas del documento original, seleccionando aquellas con mayor puntuación, obtenida tras fusionar las puntuaciones de cada oración individual

5.1 RankSum

Un documento D consiste en N oraciones (S_1, S_2, ..., S_N) y M palabras ($w_1, w_2, ..., w_M$). El objetivo de un algoritmo o método de resúmenes de texto automáticos es la extracción de un conjunto ordenado de las L frases más importantes que formarían el resumen de texto final.

El método de resumen propuesto, RankSum, utiliza cuatro características extraídas de las oraciones del documento original (tópicos, palabras clave, semántica y posición) para clasificar las oraciones en un documento. Primero se genera una ordenación de cada oración del documento de acuerdo con cada característica. A continuación se realiza una fusión de las puntuaciones ponderadas derivada de las cuatro ordenaciones generadas. Planteamos la hipótesis de que diferentes características se complementan entre sí y pueden producir una representación más significativa de un documento.

Figura 2) muestra un resumen gráfico del método propuesto.En las subsecciones siguientes, se resume brevemente cada una de las características utilizadas para clasificar las oraciones y la metodología de fusión aplicada.

5.1.1 Extractor de rango del tópico

En esta subsección se presenta un método nuevo para clasificar las oraciones en función de sus vectores de tópicos. La información del tópico puede preservar el significado global de un documento, lo que es útil en el resumen para comprender la información semántica de largo alcance en el texto. Empleamos la asignación de Dirichlet latente (LDA) (Blei et al., 2003) para encontrar los tópicos en el documento.

Al aplicar LDA, inicialmente se calculan los vectores de tópico T_D para cada documento en el corpus y los vectores de tópico T_w para cada palabra del documento. Después, obtenemos el vector de tópico de la oración, $T_S i$ promediando los vectores de tópico de cada palabra presente en la oración. Para clasificar las oraciones en el documento, calculamos la distancia euclídea, $ED_i(T_{S_i}, T_D)$ entre el vector de tópico de cada oración, T_{S_i} y el vector de tópico del documento T_D. Las oraciones que son más importantes se ubicarán cerca del vector de tópico del documento y se clasificarán en consecuencia. El vector de tópico de rango generado para un documento quedaría representado como R_T.

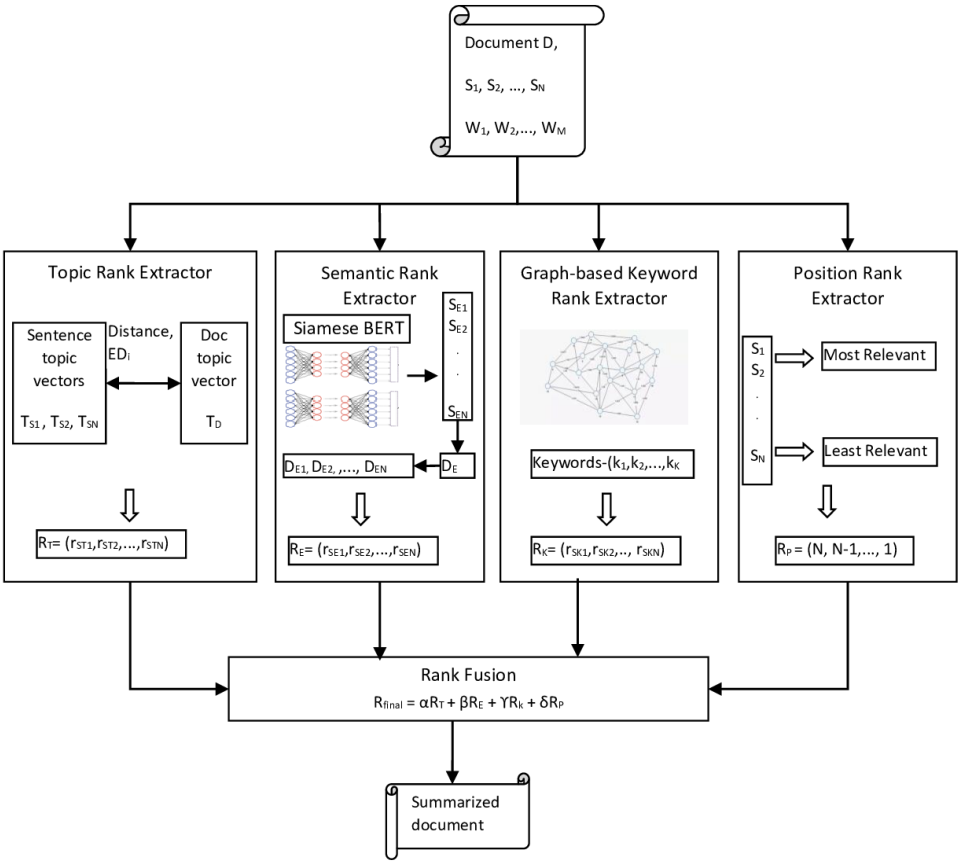


Figura 2: Esquema de la arquitectura RankSum

$$\mathbf{R_T} = (r_{ST_1}, r_{ST_2}, \ldots, r_{ST_N}), \tag{12}$$

5.1.2 Extractor de rango semántico basado en incrustaciones

Para obtener las incrustaciones de oraciones para cada oración en el documento usamos SBERT (Reimers and Gurevych, 2019), arquitectura basada en BERT que usa redes siamesas y triples para obtener incrustaciones semánticamente significativas. SBERT ha demostrado superar otras incrustaciones del estado de la técnica (Devlin et al., 2019; Conneau et al., 2017; Liu et al., 2019) en siete tareas de Similitud Textual Semántica (STS). SBERT también es computacionalmente eficiente en comparación con otras incrustaciones de oraciones.

Desarrollamos un algoritmo novedoso para encontrar la clasificación de oraciones

en función de sus respectivas incrustaciones. S_{E_i} representa las incrustaciones obtenidas usando la arquitectura SBERT para cada oración S_i en el documento. Calculamos el documento incrustado D_E promediando las incrustaciones de todas las oraciones en el documento. Para identificar la relevancia de cada oración en el documento, eliminamos esa oración del documento y nuevamente obtenemos un nuevo documento que incluye D_{E_i}.

A continuación, para medir la importancia de la oración S_i en el documento, calculamos la distancia euclídea d_E entre los vectores del documento, D_E y D_{E_i}. Una oración importante generará un valor alto de d_{E_i} en comparación con las oraciones que no expresan el significado del documento. Por lo tanto, producimos el vector de rango $\mathbf{R_E}$ para todas las oraciones del documento en función de sus puntuaciones d_{E_i}.

5.1.3 Extractor de rango de palabras clave

Las palabras clave capturan la información del contenido estructural en un documento. Las oraciones que tienen palabras clave contendrían información más significativa en comparación con otras oraciones del documento con menos palabras clave. Para calcular el conjunto de palabras clave $K = (k_1, k_2, ...k_K)$ en un documento, primero eliminamos las palabras vacías y aplicamos lematización. Después, seguimos una estrategia basada en grafos (Brin and Page, 1998) para identificar los términos clave en un documento.

A continuación, generamos el rango $\mathbf{R_K}$ con respecto a las palabras clave dentro de cada oración en el documento. Elegimos dar un rango más alto a las oraciones que constan de más palabras clave. Si algunas oraciones contienen la misma cantidad de palabras clave, las clasificamos de acuerdo con sus posiciones. Suponemos que las oraciones importantes contienen más palabras clave en comparación con las oraciones que tienen menos palabras importantes.

5.1.4 Extractor de rango de posición

La posición relativa de la oración en un documento indica la importancia de la oración para la generación de resumen (Luhn, 1958; Edmundson, 1969). Por lo tanto, usamos la posición como un atributo relevante para clasificar las oraciones en el documento.

Las oraciones que aparecen al inicio de un documento son más relevantes en comparación con las oraciones que aparecen más adelante en el documento (Gupta et al., 2011). Generamos el vector de rango de posición $\mathbf{R_P}$ asignando un rango a las oraciones dependiendo de su posición. La oración en la primera posición recibe el rango más alto y la última oración, recibe el rango más bajo.

5.1.5 Extractor de novedad de oraciones

Para eliminar oraciones redundantes en resúmenes de texto extractivos, proponemos un nuevo extractor de novedad que usa las representaciones de oraciones S_{E_i} obtenidas

en la sección 1.5.1.2. Al encontrar el número de bigramas y trigramas, podemos predecir qué dos oraciones son similares entre sí, sin embargo, se estaría ignorando la semántica de las oraciones. Para superar este problema, complementamos nuestro extractor de novedades con incrustaciones de oraciones.

Las incrustaciones generadas con SBERT (Reimers and Gurevych, 2019) son bastante sólidas y funcionan bien al predecir oraciones similares durante la generación de resumen. Para calcular la novedad de la oración, S_{Nov_i}, utilizamos la medida de similitud del coseno entre las dos incrustaciones de oraciones y contamos el número de bigramas y trigramas similares presentes en las oraciones.

5.1.6 Fusión de rango y generación de resumen

Finalmente combinamos la información proporcionada por los módulos antes mencionados a nivel de rango. Fusionamos todos los vectores de rango, $\mathbf{R_T}$, $\mathbf{R_K}$, $\mathbf{R_E}$, $\mathbf{R_P}$ y generamos un rango final R_final para cada oración en el documento.

$$\mathbf{Rank_{final}} = \alpha \cdot \mathbf{R_T} + \beta \cdot \mathbf{R_K} + \gamma \cdot \mathbf{R_E} + \delta \cdot \mathbf{R_P}, \tag{13}$$

donde los valores de α, β, γ, y δ se determinan empíricamente.

Después de un proceso iterativo, se agrega una nueva oración al resumen si es distinta de las oraciones de resumen ya agregadas basadas en el extractor de novedades, S_{Nov_i} definido en la sección 1.5.1.5

5.2 Resultados y análisis experimentales

5.2.1 Conjuntos de datos

Usamos el conjunto de datos (Hermann et al., 2015) de CNN/DailyMail para entrenar a la red siamesa para la inserción de oraciones. Dividimos CNN/DailyMail en entrenamiento, validación y pruebas siguiendo la distribución indicada en la Tabla 4. Usamos el conjunto de pruebas de CNN/DailyMail y el conjunto de datos DUC 2002 [3] para evaluar nuestro enfoque propuesto, junto con otros algoritmos para el resumen de textos extractivo.

Tabla 4: Conjuntos de datos usados para entrenamiento y evaluación

Conjunto de datos	Tipo	Uso	# Documentos	# Categorías
CNN/DailyMail	Noticias	Entrenamiento	287,227	-
		Validación	13,368	-
		Pruebas	11,490	-
DUC 2002	Noticias y pruebas	567	59	

[3] https://duc.nist.gov/data.html

5.2.2 Evaluación

Comparamos RankSum con los siguientes métodos del estado de la técnica sobre DUC2002 y CNN/DailyMail: NN-SE (Cheng and Lapata, 2016), SummaRuNNEr (Nallapati et al., 2017a), HSSAS Al-Sabahi et al. (2018) y LEAD, que selecciona las tres primeras oraciones iniciales del documento para generar un resumen. Además, informamos la precisión en CNN/DailyMail usando Bi-AES (Feng et al., 2018), REFERESH (Narayan et al., 2018a), PACSUM (BERT) (Zheng and Lapata, 2019), RNES (Wu and Hu, 2018), JECS (Xu and Durrett, 2019), NeuSum (Zhou et al., 2020), HIBERTM (Zhang et al., 2019b) BERTSum (Liu, 2019), BERTSUM + Clasificador (Liu, 2019), BART (Lewis et al., 2020), PEGASUSLARGE (Zhang et al., 2019a) y MATCHSUM (Zhong et al., 2020). (ILP) (Woodsend and Lapata, 2012), Tgraph (Parveen et al., 2015), URANK (Woodsend and Lapata, 2012), SummCoder (Joshi et al., 2019) y CoRank Fang et al. (2017).

5.2.3 Resultados

La tabla 5 muestra los resultados obtenidos por RankSum y el resto de algoritmos utilizados en la comparación en el conjunto de datos DUC 2002 utilizando métricas ROUGE. RankSum obtiene puntuaciones de ROUGE-1, ROUGE-2 y ROUGE-L de 53,2, 27,9 y 49,3, respectivamente, superando al resto de métodos de la comparativaRankSum se basa en aprendizaje no supervisado, no requiere ningún dato etiquetado para entrenar el sistema, algo que sí necesitan otros métodos más que es el requisito de los métodos de resumen propuestos más recientemente, como SummaRuNNer, HSSAS y NN-SE.

Tabla 5: Análisis comparativo de RankSum con algoritmos del estado de la técnica sobre el conjunto de datos de DUC 2002

Método	ROUGE-1	ROUGE-2	ROUGE-L
Lead	43.6	21.0	40.2
ILP	45.4	21.3	42.8
NN-SE	47.4	23.0	-
SummaRuNNer	47.4	24.0	14.7
Egraph + coh	47.9	23.8	-
Tgraph + coh	48.1	24.3	-
URANK	48.5	21.5	-
SummCoder	51.7	27.5	44.6
HSSAS	52.1	24.5	48.8
CoRank	52.6	25.8	-
RankSum	**53.2**	**27.9**	**49.3**

Con respecto al conjunto de datos de CNN/DailyMail, observamos una mejora en la precisión con las puntuaciones ROUGE-1, ROUGE-2 y ROUGE-L de 44,5, 24,0 y 41,0. Como se puede ver en la Tabla 6, RankSum obtuvo las mejores puntuaciones ROUGE-1 y ROUGE-2 con respecto a los algoritmos evaluados, y una puntuación ROUGE-L 0,1 puntos inferior a PEGASULARGE.RankSum obtuvo resultados superiores a otros métodos basado en aprendizaje supervisado, como PACSUM, HIBERTM, HSSAS, BertSum, PE-

GASUSLARGES, JECS y BART, dándole la ventaja de no necesitar datos de entrenamiento para poder obtener resultados del estado de la técnica.

Tabla 6: Análisis comparativo de RankSum con algoritmos de última generación en CNN/DailyMail

Métodos	ROUGE-1	ROUGE-2	ROUGE-L
NN-SE	35.5	14.7	32.2
Bi-AES	38.8	12.6	33.8
Lead	39.2	15.7	35.5
SummaRuNNer	39.6	16.2	35.3
REFERESH	40.0	18.2	36.6
PACSUM (BERT)	40.7	17.8	36.9
RNES sin coherencia	41.2	18.8	37.7
JECS	41.7	18.5	37.9
NeuSum	41.5	19.0	37.9
HSSAS	42.3	17.8	37.6
HIBERTM	42.3	19.9	38.8
BertSum	43.2	20.2	39.6
BERTSUM + Clasificador	43.2	20.2	39.6
BART	44.1	21.2	40.9
PEGASUSLARGE	44.1	21.4	**41.1**
MATCHSUM	44.4	20.8	40.5
RankSum	**44.5**	**24.0**	41.0

5.3 Conclusiones

En este capítulo de tesis se ha presentado RankSum, un método de aprendizaje no supervisado para realizar resúmenes de texto extractivos de un solo documento. Partiendo de la base de que una persona, a la hora de realizar un resumen de texto, trata de combinar diferentes características del documento para resumirlo, RankSum se basa en combinar cuatro características estructurales y semánticas de un documento para resumirlo.

Nuestra propuesta captura información multidimensional del documento utilizando palabras clave, tópicos, incrustaciones de oraciones y la posición de una oración en el documento. Finalmente, RankSum realiza una fusión de los rangos obtenidos a través de dichas características para asignar un rango final a cada oración. El resumen extractivo del documento se formará seleccionado las frases del documento original en base al rango final calculado por el método.Experimentalmente, se ha demostrado que RankSum produce una descripción más completa de las frases del documento a la hora de realizar resúmenes de texto extractivos, superando a la mayoría de métodos del estado de la técnica evaluados en los conjuntos de datos CNN/Dailymail y DUC2002.En trabajos futuros, nos gustaría evaluar el rendimiento de RankSum en la tarea de resúmenes de texto abstractivos.

6 Conclusiones de la Tesis y Trabajo Futuro

En cumplimiento del punto 3º del artículo 19 del Reglamento de las enseñanzas oficiales de doctorado y del título de doctor de la Universidad de León, aprobado en Consejo de Gobierno el 25/9/2012, las conclusiones y líneas de trabajo futuro de esta tesis, han sido presentadas en el capítulo 6.