UNED
Escuela Internacional de Doctorado
EIDUNED

TESIS DOCTORAL

2019

AUTHORITY AND PRIORITY SIGNALS IN ONLINE REPUTATION MONITORING

JAVIER RODRÍGUEZ VIDAL
M.Sc. in Computer Science Research
(Universidad Complutense de Madrid)

PROGRAMA DE DOCTORADO EN SISTEMAS INTELIGENTES

Academic Advisors

Dr. JULIO GONZALO ARROYO Professor in the Natural Language Processing and Information Retrieval Group at Universidad Nacional de Educación a Distancia.

Dra. LAURA PLAZA MORALES Lecturer in the Natural Language Processing and Information Retrieval Group at Universidad Nacional de Educación a Distancia.

UNED Escuela Internacional de Doctorado EIDUNED

TESIS DOCTORAL

2019

AUTHORITY AND PRIORITY SIGNALS IN ONLINE REPUTATION MONITORING

JAVIER RODRÍGUEZ VIDAL
M.Sc. in Computer Science Research
(Universidad Complutense de Madrid)

PROGRAMA DE DOCTORADO EN SISTEMAS INTELIGENTES

Academic Advisors

Dr. JULIO GONZALO ARROYO Professor in the Natural Language Processing and Information Retrieval Group at Universidad Nacional de Educación a Distancia.

Dra. LAURA PLAZA MORALES Lecturer in the Natural Language Processing and Information Retrieval Group at Universidad Nacional de Educación a Distancia.

A mi familia

AGRADECIMIENTOS

Con esta Tesis se pone punto y final a una bonita etapa de mi vida que ha durado cuatro años. Estas pocas líneas son para agradecer a todas aquellas personas que han ayudado a hacer posible todo esto.

En primer lugar, me gustaría agradecer a Julio Gonzalo y a Laura Plaza el haberme formado tanto en los campos de PLN e IR, como para ser un buen investigador. También agradecerles la libertad que me han dado durante estos años y su ayuda.

Agradecer también a los revisores externos: Arkaitz Zubiaga, Eric SanJuan y Miriam Fernández, que generosamente han aceptado corregir esta Tesis, sus comentarios han sido de gran valor. También quisiera agradecer a Henry Anaya por su colaboración e inestimable ayuda a la hora de realizar esta Tesis y a Damiano Spina que, aunque nunca hemos llegado a conocernos en persona, siempre me ha ofrecido su ayuda cuando la he necesitado.

I would like to thank to professor Maarten de Rijke and Dr. Stevan Rudinac and the Information and Language Processing (ILP) group for hosting me during my 3-month research stay at the University of Amsterdam.

A mis compañeros del grupo de Lenguajes y Sistemas Informáticos (LSI) de la UNED, tanto aquellos que se fueron como de las nuevas incorporaciones. En especial a Agustín, Andrés, Ángel y Bernardo por compartir ideas y por encima de todo, muchas risas entre cafés y cervezas en el Lizarrán.

A mis amigos, en especial a Paco y Dani, por todos los buenos momentos que hemos vivido y viviremos juntos y por estar ahí siempre apoyándome, ellos saben cuánto los aprecio.

Sobre todo agradecer a los míos. A mis padres, Julián y María Teresa, por todos los sacrificios que han hecho para ayudarme a alcanzar todas mis metas. A mi hermano Marcos que, como buen hermano mayor, siempre ha cuidado de mí y por todos esos "asuntillos dirimidos" a golpe de mando de consola. A su mujer Mònika y a la pequeña estrella de la casa, Martel.

Por último, pero no por ello menos importante, a Ana. Ella es uno de los pilares fundamentales en mi vida, esta Tesis no habría sido posible sin su apoyo, amor y cariño, sobre todo, en los días menos buenos. Muchas gracias de corazón.

Javier Rodríguez Vidal
Madrid, Junio de 2019

ABSTRACT

Online Reputation Monitoring (ORM) comprises a collection of techniques which help to monitor and improve the image of a certain entity (company, organisation, individual) on the Internet. The ORM experts try to minimize the negative impact of the information about the image of the entity, while maximizing the positive material. In order to do this, the reputation experts need to track down all the information (good or bad) related to the entity in Social Networks, blogs, specialize sites, news sites, etc. to produce reputation reports, which summarizes the most important issues related to the entity and to timely detect potential reputation alerts, and instantly reply to controversial issues or dispel malicious rumours.

Before the existence of the Internet, the access to the information was controlled, and only the opinions of people with a great reputation (journalists, scientists, etc.) were taken into account by the rest of the population. Today, a new figure has emerged, specially in Social Networks, that, without accrediting any kind of authority or specialized knowledge, manages to change the opinion of other users within a community, the so-called influencers. These influencers have a legion of followers behind them, so that any negative message about the entities may be spread immediately among hundreds of thousands of users, who in turn transmit it to their followers, etc. causing very serious reputational crisis. The detection of influencers is essential for the ORM process carried out by the reputational experts but, it is not simple to identify influencers since, it not only depends on the number of followers but also, on the kind of followers that the profile has, the domain, etc. In this Thesis, we focus on the detection of influencers on Twitter. To this end, we present an exhaustive study of signals and we

also introduce two different ways to tackle the detection of influencers: (i) using the information originated from those profiles we want to discover their identity; (ii) performing a bottom-up search in which we use the information regarding the followers to characterize the followed profiles.

On the other hand, the information that has been extracted about the entities must be included in a reputational report. These reports collect and summarize all the topics that may affect the entities' reputation, that are discussed in the Social Networks. Since the amount of data about an entity that can be extracted from the Internet is enormous, it is impossible for a human to read it all in a reasonable amount of time. Besides, there may be repeated information, so that ORM experts must select those opinions that are more useful and show them without repetitions to the client. Given that influencer's opinions are potential threats to entities' reputation, they are good candidates to appear in the reputation report. In this Thesis, we study the task of automatically generating reputation reports using an extractive summarization approach. The relevance of the information is calculated from the signals that measure the authority and the domain knowledge of a user along with other state-of-the-art signals from the automatic summarization field (such as centrality, polarity, etc.).

RESUMEN

La Monitorización de la Reputación Online (ORM) es un conjunto de técnicas que ayudan a monitorizar y mejorar la imagen de una determinada entidad (empresa, organización, individuo) en Internet. Los expertos en ORM tratan de minimizar el impacto negativo de cierta información sobre la imagen de la entidad, maximizando al mismo tiempo el material positivo. Para ello, los expertos en reputación necesitan localizar toda la información (buena o mala) relacionada con la entidad en redes sociales, blogs, sitios especializados, sitios de noticias, etc. para producir informes de reputación, que resumen los temas más importantes relacionados con la entidad detectando oportunamente posibles alertas de reputación, y respondiendo instantáneamente a cuestiones controvertidas o para disipar rumores maliciosos.

Antes de la existencia de Internet, el acceso a la información estaba controlado, y sólo las opiniones de personas de gran reputación (periodistas, científicos, etc.) eran tenidas en cuenta por el resto de la población. Hoy en día, ha surgido una nueva figura, especialmente en las redes sociales, que, sin acreditar ningún tipo de autoridad o conocimiento especializado, consigue cambiar la opinión de otros usuarios dentro de una comunidad, los llamados influencers. Estos influencers tienen una legión de seguidores detrás de ellos, de modo que cualquier mensaje negativo sobre las entidades puede ser difundido inmediatamente entre cientos de miles de usuarios, que a su vez lo transmiten a sus seguidores, etc. pudiendo causar una crisis de reputación muy grave. La detección de influencers es esencial para el proceso de ORM que llevan a cabo los expertos en reputación, pero no es fácil identificar a los influencers, ya que no sólo depende del número de seguidores, sino también del tipo de seguidores que tenga el perfil, el dominio, etc. En esta

Tesis, nos centramos en la detección de influencers, específicamente, en Twitter. Para ello, presentamos un estudio exhaustivo de las señales extraídas e introducimos dos formas diferentes de abordar la tarea de detección de influencers: (i) utilizando la información procedente de los perfiles de los que queremos descubrir su identidad; (ii) realizando una búsqueda ascendente en la que utilizamos la información relativa a los seguidores para caracterizar los perfiles que siguen.

Por otro lado, la información que se ha extraído sobre las entidades debe incluirse en un informe de reputación. Estos informes recogen y resumen todos los temas que pueden afectar a la reputación de las entidades y que se debaten en las redes sociales. Dado que la cantidad de datos sobre las entidades que se pueden extraer de Internet es enorme, es imposible que un ser humano pueda leerlos todos en un tiempo razonable. Además, puede haber información repetida, por lo que los expertos de ORM deben seleccionar aquellas opiniones que sean más útiles y mostrarlas sin repeticiones al cliente. Como las opiniones de los influencers son amenazas potenciales para la reputación de las entidades, ya que tienen la capacidad de atraer a muchas personas, son buenos candidatos para aparecer en el informe de reputación. En esta tesis, estudiamos la generación automática de informes a partir de resúmenes extractivos de aquellos tweets que son más relevantes para las entidades. Esta relevancia se calcula a partir de las señales que miden la autoridad y el conocimiento del dominio de un usuario junto con otras señales conocidas en el estado-del-arte de la tarea de generación automática de resúmenes (como son la centralidad, la polaridad, etc.).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Wer immer strenden sich
bemüht/den könen wir erlösen

Johann W. Goethe-Faustus

The first chapter of this Thesis is devoted to introduce the problem of the automatic generation of reputation reports and its relation with a special type of user in Social Networks: the so-called influencers. Entities (companies, individuals, etc.) need to monitor what is being said about them in Social Networks and how it may affect their reputation. What is said about an entity may harm its reputation causing loss of credibility, and as a consequence, loss of money. On the other hand, there is a special kind of Social Networks' users known as influencers or opinion makers. These influencers are well known in certain communities and their opinions may reach a lot of people that accept those opinions as their own and help spreading them, so it is important for the entities to detect this kind of users and their opinions, in order to prevent reputational crisis.

This chapter is divided in the following sections: first, we provide a broad motivation of the task carried out. Second, we state the problem by giving the general picture of the core tasks that Online Reputation Management experts must address in their daily work relating them to the work developed in our Thesis. Third, once the problem is defined, we state our main research objective and, to achieve this objective, we propose different research questions that will be answered throughout our research. Finally, to accomplish the research objective, we define the methodology followed during this Thesis.

1.1 Motivation

With the emergence of the Internet and Social Networks in recent years, customers have more information about entities (products, companies, etc.) that allow them to differentiate between those that are trusted from those that are not. A good reputation is difficult to gain, and it can be ruined very fast. It is very important to keep always a good online reputation by managing it constantly. **Online Reputation Management** (ORM) comprises the collection of techniques that help monitoring and improving the public image of an entity (company, organisation, individual) on the Internet. The ORM experts try to minimize the negative impact of the information in the Internet while maximizing the positive material for being more trustworthy to the customers. In order to do that, ORM experts need to track down all the information (good or bad) related to the client in Social Networks, blogs, specialized sites, news sites, etc. to produce reputation reports, which summarize the most important issues about the client. A quick identification of the problems that affect a client, may avoid a reputational crisis, when the perception of the client to the eyes of costumers or investors is negative and therefore, translates into money loss for the client. According to Igniyte (2018), failing in monitoring negative content costed, in the *UK* in 2018, between $100,000£$ and $500,000£$ for the 5% of the companies. But, every minute a huge amount of information is published in the Internet; only Google offered over 3.4 million answers per minute in 2018[1], so that for human operators it is impossible to track down all these data. The following example shows how things can get out of control when a reputation problem is not addressed properly:

[1] http://marketingactual.es/internet/tecnologia/internet/big-data-cuantos-datos-se-generan-cada-minuto-en-el-mundo

Example: Apple's "Batterygate"

In 2017 a 17-year-old named Tyler Barney unveiled what was called the "Batterygate". When he updated his iPhone 6s to iOS11, the performance of the mobile phone fell drastically. Blaming the new operating system and waiting for a new update to solve the problem, the boy used his brother's iPhone 6 and realized that, despite being an older model than his, was better. Reviewing in the Internet, he found the solution: change the battery of the mobile and effectively, this solved the problem but he published in Reddit his adventure and here begins the Apple's Odyssey.

Seeing that Reddit's post was gaining strength among users, Apple decided to try to cut the bleeding and offer discounts on replacing their batteries during 2018 for iPhones 6 and newer. In March 2018, an operating system update gave users information about the health of the battery and demands began to arrive but these actions did not calm the things down between Apple users. In mid-March, the annual *Harris Poll Reputation Quotient*[a] poll came out, taking out Apple from the list of the 10 most reputable companies in the U.S. down to number 29. But the bad news for Apple didn't end there because, at the end of 2018, the Italian government's antitrust agency fined Apple €10 million for, among other things, not making it easy and transparent to replace their mobile batteries.

[a] https://theharrispoll.com/reputation-quotient/

From the example above we can draw several conclusions: there will always be someone who can discover a malpractice and the whole world will find out in a matter of minutes. And, for that, being capable to detect the problem, as quickly as possible, and give an adequate response will avoid a fall in the confidence of your customers (which may not be recoverable) and loss of money.

Before the existence of the Internet, the big lobbies controlled the flows of access to information, and only the opinions of a handful of people (media owners, journalists, scientists, intellectuals, recognized experts, etc.) were the ones taken into account by the rest of the population. Today a new figure has emerged, especially in Social Networks, that, without accrediting any kind of authority or specialized knowledge, manages to change the opinion of other users within a community, the so-called *influencers*. Influencers have a legion of followers behind them, so that any negative message about the client may be spread immediately among hundreds or thousands of users who, in turn, transmit it to their followers, etc., potentially causing serious reputational damage (Tucker and Melewar, 2005).

However, it is not simple to identify influencers, since it does not only depend on the number of followers, but also on the kind of followers that the profile has, the domain of the users (e.g. specialist in finance, music, etc.), etc. For example, in the cinema and music domains, the influencers have millions of followers while, in other domains such as education, the most popular influencers have, barely, tens of thousands followers Joyce (2018), for instance, states that the level of influence may be measure according to the engagement that the profile is capable to generate, so that the number of followers has an influence but also the number of retweets or answers have something to say. Also, unknown users for the general public could be *influencers* within a *domain*: for example, in the banking domain a Wall Street broker or in motor sport domain, a F1 mechanic are influencers because they are experts in their respective domains.

The automatic detection of influencers is, therefore, essential for the ORM process carried out by the reputational experts. This task is the first objective of this Thesis.

On the other hand, the information that has been extracted about the client must be included in a reputation report prepared by the human experts in ORM and that is delivered to the client. A reputation report collects and summarizes all the topics that may affect to the client which are discussed in Social Networks. This summarization is necessary, since the amount of data about a client that can be extracted from the Internet is enormous, it is impossible for a human to read it all in a reasonable amount of time. Given that influencers' opinions are potential threats to clients' reputation, since they have the capacity to engage many people, they are good candidates to appear in the reputation report.

In this Thesis, our second objective is the automatic reputation report generation by exploiting the information about the authority and domain knowledge of the users that produce and spread the information.

1.2 Problem statement

The purpose of this Thesis is to help reputational experts in their work of managing the reputation of entities (such as companies, people, etc.) in Social Networks through the automatic generation of reputation reports. These reports condense the information that is shared in the Internet about the entities. In particular, we focus on Social Networks, and specifically on Twitter, for two reasons: the availability of annotated datasets relevant to our problem, and the immediacy and open nature of Twitter content, which makes it particularly relevant and viable to prevent reputational crisis.

According to (Amigó et al., 2013), ORM experts address the following core tasks:

- **Filtering:** firstly, ORM experts determine which tweets are related to the entity and which are not. For example, if we study the reputation of Apple company, we have to distinguish between those tweets that refer to Apple's items like iPad, iCloud, etc. from those referring to the fruit as in the following tweet:

  #Apples and nuts are good ingredients for a simple salad for kids. #healthy

- **Reputation polarity classification:** then, ORM experts distinguish the implications that relevant facts or opinions have for the entities' reputation. From the point of view of the reputation management, facts and opinions could improve (positive), harm (negative) or do not have any effect (neutral) over the entities' reputation. Continuing with the Apple's example, we have the following polarity classification of tweets:

  positive fact: *Apple employs more people than a good sized city*
  neutral fact: *Apple originally had three co-founders*
  negative opinion: *Apple really was a great company! But creativity died with Steve Jobs.*

- **Topic detection:** next, ORM experts identify topics or conversations about the entity, grouping texts accordingly. For example, the following tweets talk about the same topic about the Apple's company:

  Topic: *BatteryGate*
  Tweets:*John Poole de GeekBench y Apple dando explicaciones en una comisión parlamentaria en Canada acerca del #timo #batteryGate #iPhoneSlow*
  @NPRone eat up 40% off my battery life in one hour. That's completely unacceptable. Is anyone else experiencing this? #batteryGate

- **Priority ranking:** fourth, ORM experts order the topics detected according to their importance for the entities' reputation, from reputational alerts to unimportant issues. It is expected that the most important topics will appear in the top positions of the ranking, while the less important ones will appear at the bottom. For example, the following tweets are ranked by reputational importance:

(1) *John Poole de GeekBench y Apple dando explicaciones en una comisión parlamentaria en Canada acerca del #timo #batteryGate #iPhoneSlow*
(2) *@NPRone eat up 40% off my battery life in one hour. That's completely unacceptable. Is anyone else experiencing this? #batteryGate*
(3) *Apple originally had three co-founders*

The problems that affect more negatively to the entity's reputation and must be addressed as soon as possible by the reputational experts, are called reputation *alerts*. The *mildly-important* issues affect negatively the entity's reputation but are less important than *alerts*. Finally, there are *unimportant* issues that should not have reputational implications.

Priority is given by different factors (Carrillo-de Albornoz et al., 2016) such as, for instance:

- **Authority:** if there are influencers involved in the conversation. The idea here is that influencers are capable to change other people's opinion so a bad review about an entity could fire a reputational alarm.
- **Polarity:** if the message has positive or negative implications for the entity's reputation.
- **Novelty:** if it is a new problem or is a recurring one. The idea here is that old issues are less likely to trigger a reputational alert.
- **Centrality:** the entity is the focus of the conversation. If the entity is not the main focus in a conversation, it is less likely to be a reputational alert.

- **Report generation:** finally, the ORM experts summarize all previous information in a report to be delivered to the client and/or handled from a Public Relations perspective. This report presents all information in a way so that humans are capable to read and understand it in a reasonable time thus making the decision process both possible and useful. The topics of conversations appear in the report according to the priority that they have for the client, being the highest relevant topics those that appear in the first positions, while the unimportant topics are excluded or appended to the main summary. Reputation reports, can also propose and outline different strategies to overcome reputational crises.

Figure 1.1 shows the main steps carried out during the annotation in online reputation monitoring process performed by the reputational experts:

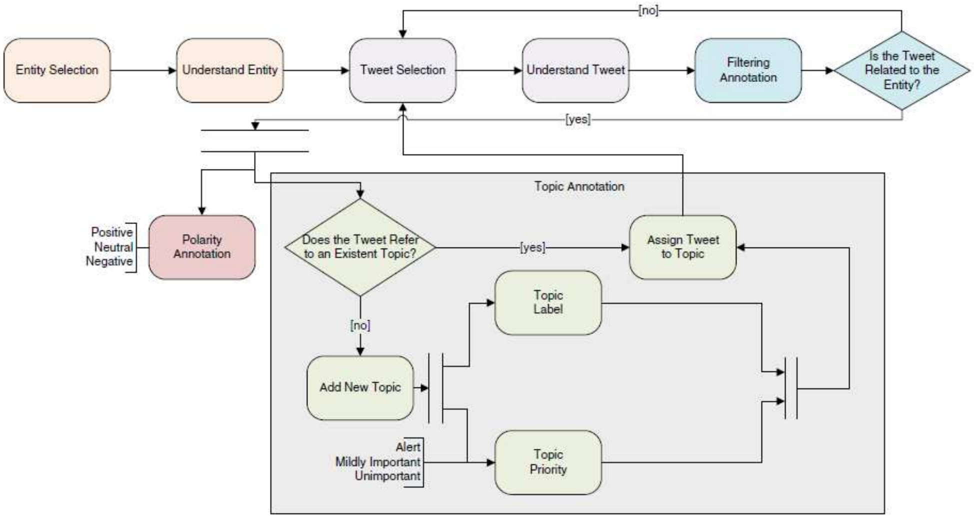


Figure 1.1: Workflow carried out during the online reputation monitoring process

In this Thesis, we address the report generation step, because the other tasks have been already tackled by our lab in the context of the RepLab initiatives (Amigó et al., 2013, 2014). As a previous step, we also address the problem of identifying influencers for a given activity domain, as such information should help creating better reputational reports.

1.3 Research questions and objectives

Our main research objective is to generate automatic reputation reports by given more priority to those texts written by the influencers. We can formalize this objective in the following statement:

RESEARCH OBJECTIVE: Studying the problem of automatic generation of online reputation reports and investigating the role of reputation priority signals, with special attention to the identification of Social Network influencers

In order to achieve the previous objective, we propose the following set of research questions:

- **Detection of influencers on Twitter**
  Our first objective is to develop a method to characterize and identify automatically influencers in Social Networks. Regarding this task; we state the following research questions:

  - **Research Question 1:** *What is the relative importance of authority signals versus domain expertise signals?* In order to be influential in a given domain, two major types of signals are involved: signals of authority (for instance, *does the user have many followers? Are her statements frequently retweeted? Are her posts similar to other influencers' texts?*) and signals of relevance to the domain (an influential voice in macro economics may have no authority at all when talking about music, for instance). We want to find out what is the relative role that each of these two types of signals play when detecting influencers and how to characterize them. We explore signals extracted from the Twitter network structure (i.e. number of followers) and from the content of the tweets.

  - **Research Question 2:** *How best to combine signals?* We will explore several ways for combining signals to discover influencers: supervised classification, unsupervised signal voting, supervised learning to rank, and supervised classification combined with signal voting. We will also combine all these approaches with language models that learn both the domain language and the language of influencers, using them as the primary textual signals.

  - **Research Question 3:** *How well followers characterize a Twitter profile? Can we establish authority using only information from followers? What is the best way of using followers' information to establish authority?* We want to know if the language used by followers is an important factor for establishing whether or not a user is an *influencer*. Moreover, *can information from followers enrich profile characterizations?* In other words, *is it complementary or redundant with the information from the profile itself?* Here we want to aggregate the information related to the profile with the information of her followers and verify if this aggregation improves the profile characterization.

- **Automatic generation of reputation reports**
  Our second objective is to retrieve from the Social Networks the texts with the greatest impact on an entity and generate reputational reports from them, exploring information regarding to the influencers, such as their knowledge about a domain and their degree of authority with other users. In this context, the following research questions are asked:

  – **Research Question 4:** *Can authority and domain signals be effectively exploited in order to create reputation summaries?* As we said before, we want to collect and summarize those messages in Social Networks that have greater impact in the reputation of an entity (e.g. a company or a product). Also, as we have said, influencers are a special kind of users in Social Networks that are well-known inside a community and whose opinions are followed by a huge amount of people. Because authority and domain signals have been used to characterize and identify influencers, we want to use these signals to identify opinions expressed by this kind of users and give them priority when generating reputation summaries.

  – **Research Question 5:** *What role do priority, polarity and centrality play in the generation of reputation summaries?* Beyond the signals that characterize influencers, we want to incorporate other signals which have been widely used in the state-of-the-art of automatic summarization (priority of the topic, polarity of the comments or centrality to the topic), in the creation of our reputational summaries and to study the value they incorporate to this task.

  – **Research Question 6:** *What is the performance of different similarity functions for avoiding redundancy?* Since a summary should not present repeated information, it is essential to have a mechanism to detect redundancy. For this reason, we want to study the effect of different ways of measuring redundancy in short texts (tweets in our case) and the effect this has on the creation of reputation summaries.

  – **Research Question 7:** *How can we use topic information to create reputation summaries?* Topics give information about the different subjects of conversation in Social Networks. These topics group different opinions, about the same issue, that may affect to the entity's reputation; therefore, this information must be included in the report. In order to include the information regarding to the topic, we test different ways of using topic information.

1.4 Research methodology

In this chapter we describe the methodology we have followed to accomplish the research objective presented in section 1.3. It consists of seven main steps:

1. *Review of the state-of-the-art*: Analysing the state-of-the-art has two main purposes: (i) knowing the big picture of the task to find new research opportunities; (ii) knowing the partial problems involved in it and how they have been addressed in the literature.

2. *Detection of influencers on Twitter*: First, developing a method for detecting automatically influencers by taking into account the different aspects that model the Twitter profiles (such as the characteristics of their followers, the texts in the tweets, etc.). Second, presenting the results obtained in a ranking so that it is easy, for the ORM experts, to distinguish between influencers and non-influencers.

3. *Creation of reputation reports*: Developing a method for creating reputational reports by extracting the most relevant tweets published about an entity, by exploiting information regarding the authority and domain knowledge of the authors and other priority signals from the state-of-the-art.

4. *Data collection*: Collecting and labelling data to verify the hypothesis and test the methods.

5. *Evaluation and analysis of results*: Exploring different evaluation metrics and selecting those that are best suited for each task. Performing an exhaustive discussion of the results obtained (both positive and negative) and comparing them with the results provided by the state-of-the-art systems.

6. *Proposition of future research lines*: Drawing some future lines of work consistent with the results of the Thesis.

7. *Results dissemination*: Publishing the partial results in impact-factor journals. In addition, summarizing all the research and conclusions in a Thesis, as a final contribution to the scientific community.

1.5 Publications of the author

The following papers correspond to research carried out during the Ph.D. period. Papers are presented chronologically.

Journals:

I Jorge Carrillo-de-Albornoz, Javier Rodríguez Vidal and Laura Plaza. 2018, November. Feature Engineering for Sentiment Analysis in e-Health Forums. PLoS ONE 13(11): e0207996.
doi: 10.1371/journal.pone.0207996, URL:
https://doi.org/10.1371/journal.pone.0207996 JCR Q1.

II Javier Rodríguez-Vidal, Julio Gonzalo, Henry Anaya Sánchez and Laura Plaza. 2019, January. Automatic Detection of Influencers in Social Networks: Authority vs Domain signals. Journal of the Association for Information Science and Technology, volume 70, pages 675-684.
doi:10.1002/asi.24156, URL:
https://onlinelibrary.wiley.com/doi/full/10.1002/asi.24156 JCR Q2.

III Javier Rodríguez-Vidal, Jorge Carrillo-de-Albornoz, Enrique Amigó, Laura Plaza, Julio Gonzalo and Felisa Verdejo. 2019, February. Automatic Generation of Entity-Oriented Summaries for Reputation Management. Ambient Intelligence & Humanized Computing.
doi:10.1007/s12652-019-01255-9, URL:
https://link.springer.com/article/10.1007%2Fs12652-019-01255-9 JCR Q3.

Peer-reviewed conferences:

IV Enrique Amigó, Jorge Carrillo-de-Albornoz, Mario Almagro-Cádiz, Julio Gonzalo, Javier Rodríguez-Vidal and Felisa Verdejo. 2017, August. EvALL: Open access evaluation for information access systems. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computer Machinery, New York, NY, USA, SIGIR'17, pages 1301-1304. Core A*.

Papers II and III are the two main published contributions of this thesis work. Papers I and II present work which is not directly related with the Thesis, in which I had the opportunity to collaborate.

1.6 Structure of the Thesis

The structure of this dissertation is the following:

Chapter 1

Introduction

We provide the motivation to create online reputation reports for the ORM field and define the problem of ORM, in general, and reputation report generation, in particular. We state the scope and the research goals of this Thesis and present the research methodology.

Chapter 2

Background and related work

We give an overview of the state-of-the-art prior to this dissertation (2019) in the fields of influencers detection and automatic summarization. We provide some background about Twitter and we contextualize our work covering the main techniques used for the different ORM tasks addressed.

Chapter 3

Detection of influencers on Twitter

We propose different approaches for tackling the detection and characterization of influencers. In particular, we study the detection of influencers using their profiles and the detection of influencers using information in their followers' posts.

Chapter 4

Authority & priority signals in automatic report generation for ORM

In this chapter we address the task of the automatic generation of summaries of reputational information. Here, we design and implement a system for the creation of extractive summaries and we study different signals in order to select the best ones for accomplishing this task.

Chapter 5

Conclusions and Future Work

We discuss and summarize the main conclusions and contributions of our work. We summarize the answers obtained to the research questions and the open issues for future work.

Additionally, the Thesis contains at the end three appendices with complementary results.

CHAPTER 2

BACKGROUND AND RELATED WORK

Per uarios usus artem experientia
fecit: exemplo monstrante uiam

Marcus Manilius

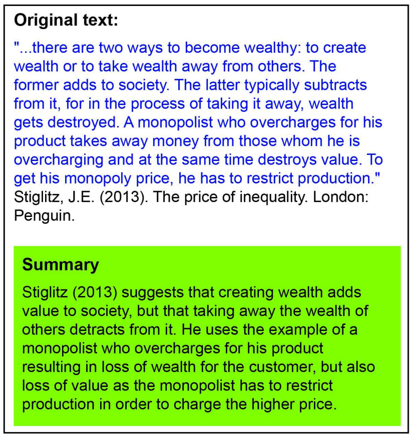
In this chapter, we provide the background and related work of our main tasks: the identification of influencers in Social Networks and the generation of summaries from online content, which are two of the main tasks in the management of online reputation information. The first task, the detection and characterization of influencers, owes its importance to the fact that these users are able, through their opinions, to make an entity (e.g. products, firms, etc.) win or lose money. Therefore, ORM experts must have them located to avoid possible reputation crises. Table 2.1 show an example of influencers on YouTube according to the number of subscribers they have.

Youtuber	Number of Subscribers
PewDiePie	59.5 million
HolaSoyGerman	33 million
elrubiusOMG	27 million
Whindersson Nunes	26.1 million
Fernanfloo	25.8 million

Table 2.1: The five YouTubers with most subscribers in 2018[2]

[2] https://www.bbc.com/mundo/noticias-42657099

The second task, the generation of reputation reports from online content, owes its relevance to the ability to present important content to the user (ORM experts in our scenario) but shorter than the original text(s). The original text(s) usually contains a large amount of information (including redundant and irrelevant information) being impossible, in a reasonable period of time, to read, understand and extract the most important content by a human operator. For this reason, one solution is to create summaries automatically. In figure 2.1 we observe a text with its corresponding summary that includes the main idea of the original text but written in a condensed form.



Original text:

"...there are two ways to become wealthy: to create wealth or to take wealth away from others. The former adds to society. The latter typically subtracts from it, for in the process of taking it away, wealth gets destroyed. A monopolist who overcharges for his product takes away money from those whom he is overcharging and at the same time destroys value. To get his monopoly price, he has to restrict production." Stiglitz, J.E. (2013). The price of inequality. London: Penguin.

Summary

Stiglitz (2013) suggests that creating wealth adds value to society, but that taking away the wealth of others detracts from it. He uses the example of a monopolist who overcharges for his product resulting in loss of wealth for the customer, but also loss of value as the monopolist has to restrict production in order to charge the higher price.

Figure 2.1: Summary example extracted from the University of Newcastle Library guides[3]

In this chapter, we discuss previous works on how to automatically detect and characterize users and how to create automatic summaries from online published content. The chapter is structured in the following sections: first, section 2.1 provides background knowledge about Twitter, the Social Network that will be the scenario where we will develop the experimentation. Then, section 2.2 shows how to detect and characterize users in Social Networks. In section 2.3, we study the problem of automatic summarization and the different ways of generating summaries: by *extraction* and *abstraction* and from a single or multiple input documents. We also introduce one of the application of automatic summaries to real life: the generation of reputation reports. Finally, in Section 2.4 we show our conclusions.

[3] University of Newcastle Library guides: https://libguides.newcastle.edu.au/paraphrasing-summarising/example-of-summarising

2.1 Twitter: a news and social networking site

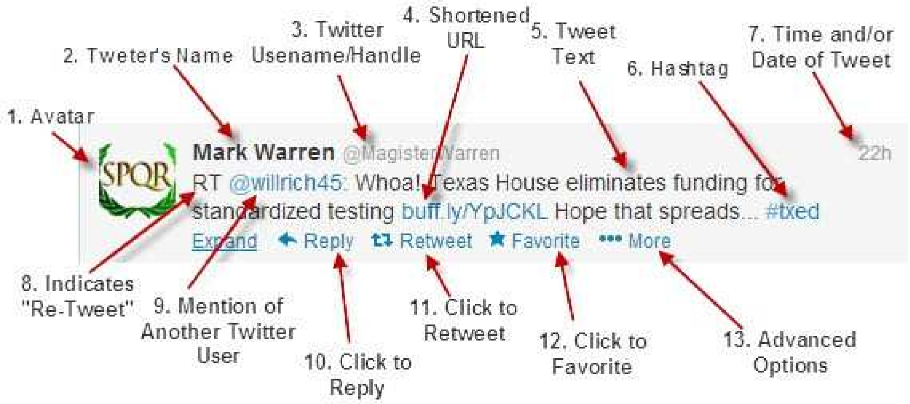
In this section, we provide some background about Twitter: its history and the definition of the most important Twitter signals that will be used in this Thesis.

Twitter is a microbbloging service that was founded in 2006 by Jack Dorsey, Noah Glass, Bliz Stone and Evan Williams. It has over 326 millions of active users per month[4] and its most important features, from the point of view of *Online Reputation Management* (ORM), are: (i) it is global, Twitter is available in different languages and it is accessible in the whole globe. From the point of view of ORM is crucial since a client may know its reputation not only in her region but across the world; (ii) it is asymmetric, the consent to add other account is not required. From the perspective of ORM, this characteristic is important because the entity does not need to add user by user to reach them, it only needs to engage them and let the users follow its novelties and spread its words across other users; (iii) it is immediate, communication is faster and the breaking news appear here first than in other Social Networks. This characteristic is essential to the ORM since reputation crisis will appear here first and they will spread faster than in other Social Networks; (iv) it is concise, messages are limited to 280 characters. For the reputational experts is easier to process a text and know the intention with which the author wrote the message. And finally, (v) it is intuitive, it does not require a broad knowledge about technology to use it. For ORM this is important since the clients may have dissimilar social and technological backgrounds.

Twitter has a specialized terminology that we will use in this Thesis. In Twitter, users can express their ideas, share their daily experiences, etc. openly or privately to others, through short messages called **tweets**, whose maximum length is 280 characters (originally 140) and may contain: labels about a topic called **hashtags** (these keywords are preceded by the character #), **mentions** to other users (by using the character @ before the name of the user), and **links** to external sources of contents or images, mainly. Users can also express their opinions about other people ideas by **replying** tweets, i.e. writing another tweet that discusses the main comment or other comments that arise in the **conversation**. Also, the users can express their agreement with the message written by clicking in the **favorite** button or sharing the message with their own audience

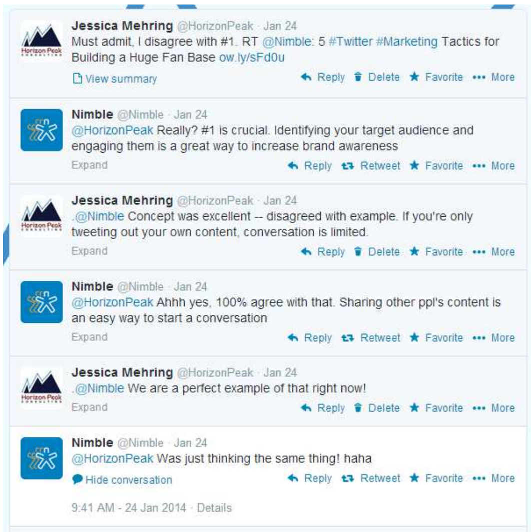
[4] Digital 2019 Global Digital Overview: https://www.slideshare.net/DataReportal/digital-2019-global-digital-overview-january-2019-v01

by **retweeting** it. Figure 2.2 shows the anatomy of the Tweets and figure 2.3 shows a *conversation*.



Mark Warren @MagisterWarren 22h
RT @willrich45: Whoa! Texas House eliminates funding for standardized testing buff.ly/YpJCKL Hope that spreads... #txed

Figure 2.2: Anatomy of a published Tweet[5]



Jessica Mehring @HorizonPeak · Jan 24
Must admit, I disagree with #1. RT @Nimble: 5 #Twitter #Marketing Tactics for Building a Huge Fan Base ow.ly/sFd0u

Nimble @Nimble · Jan 24
@HorizonPeak Really? #1 is crucial. Identifying your target audience and engaging them is a great way to increase brand awareness

Jessica Mehring @HorizonPeak · Jan 24
.@Nimble Concept was excellent -- disagreed with example. If you're only tweeting out your own content, conversation is limited.

Nimble @Nimble · Jan 24
@HorizonPeak Ahhh yes, 100% agree with that. Sharing other ppl's content is an easy way to start a conversation

Jessica Mehring @HorizonPeak · Jan 24
.@Nimble We are a perfect example of that right now!

Nimble @Nimble · Jan 24
@HorizonPeak Was just thinking the same thing! haha

9:41 AM - 24 Jan 2014 · Details

Figure 2.3: Example of a Twitter conversation[6]

Network users can subscribe to other profiles to follow the content they publish, these users are called **followers** and the person or profile being followed is called **followee**. There also exist **friendship** relationships in Twitter when two users follow each other. Figure 2.4 shows a diagram with *followers* and *followee*.

When a person accesses Twitter, it shows the tweets that have been written by the users she follows chronologically in its **timeline**, however, from 2016 Twitter

[5] https://www.smore.com/3evvx-twitter-cheat-sheet

[6] https://www.horizonpeakconsulting.com/are-you-killing-conversation-on-social-media/

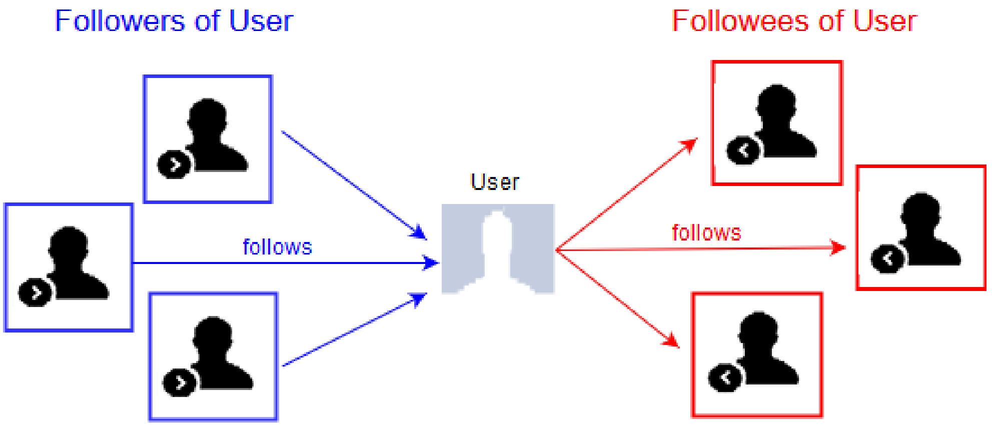
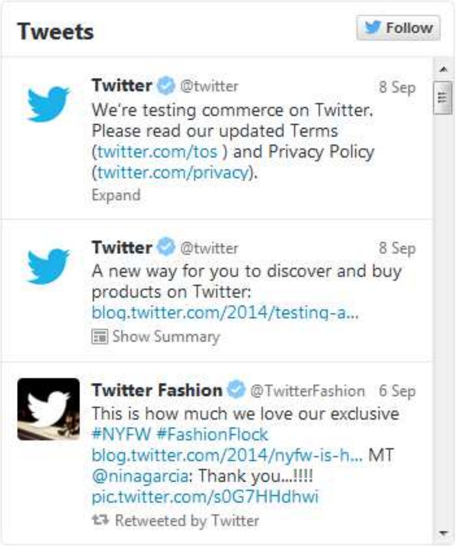


Figure 2.4: Blue squared users are the followers and the red squared users are the followees

has an alternative way of showing these tweets, through the use of an algorithm that shows first the tweets that are considered most relevant to the user, accounts or tweets with which the user has most frequently interacted. The figure 2.5 shows an example of a *timeline*:



Tweets

Twitter @twitter 8 Sep
We're testing commerce on Twitter. Please read our updated Terms (twitter.com/tos) and Privacy Policy (twitter.com/privacy).
Expand

Twitter @twitter 8 Sep
A new way for you to discover and buy products on Twitter: blog.twitter.com/2014/testing-a...
Show Summary

Twitter Fashion @TwitterFashion 6 Sep
This is how much we love our exclusive #NYFW #FashionFlock blog.twitter.com/2014/nyfw-is-h... MT @ninagarcia: Thank you...!!!! pic.twitter.com/s0G7HHdhwi
Retweeted by Twitter

Figure 2.5: Example of a timeline[7]

Table 2.2 summarizes the main concepts explained and its definitions:

[7] https://versatil.net.ve/twitter-example/

Concept	Definition
Tweet	Short messages published by Twitter users.
Hashtag	Metadata label that identifies a topic.
Mention	Metadata label that identifies a user.
Reply	Answer to a published tweet.
Conversation	The set of one main tweet and its replies.
Favorite	Feature that lets the original author know that users liked their tweet.
Retweet	Repost of a tweet published to show to other users' followers.
Follower	Users that follow other profile.
Followee	Profile followed by other users.
Friendship	Profiles that follow each other.
Timeline	Displays a stream of Tweets from followed accounts.

Table 2.2: Twitter main concepts and its definitions

2.2 Author profiling

The author profiling task distinguishes different characteristics of the authors, e.g. age, gender, etc., through the study of their texts. During the last 10 years this field of study has experienced a great growth both in the number of publications and in its practical applications in several research fields. Figure 2.6 and figure 2.7 show the evolution of the number of publications, according to the Web of Science, for the author profiling field.

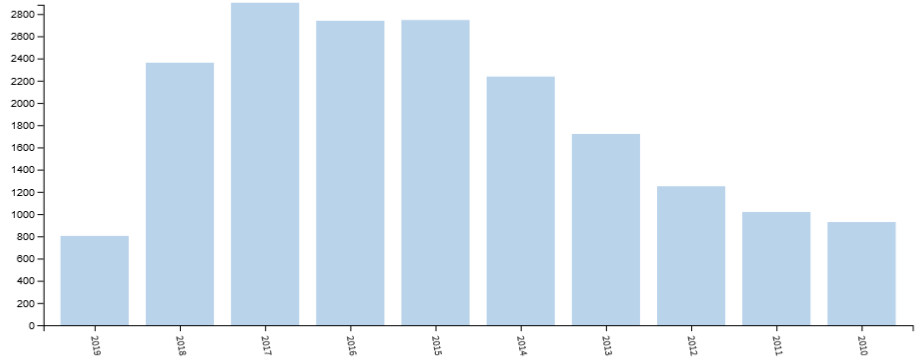


Figure 2.6: Evolution of the number of publications in author profiling field during last 10 years

As we can see in figure 2.6, during the last 10 years the author profiling field has increased its publications year after year reaching its top in 2017 with more than 2,800 papers published in 2017. In the first half of 2019, the number of papers published is 785. Figure 2.7 shows different research fields where author profiling is used, as we can see, these research fields are very diverse: physics,

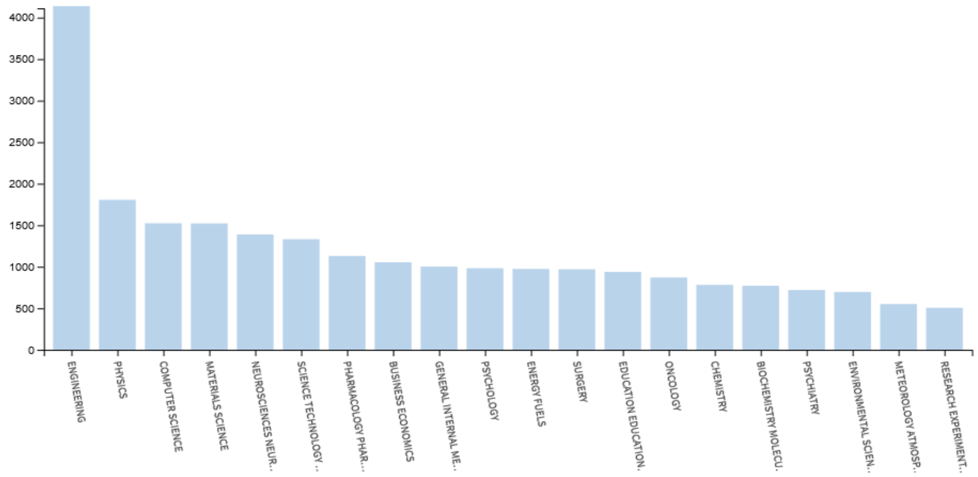


Figure 2.7: Research fields interested in author profiling

engineering, materials science, business economics, etc.

2.2.1 Recent advances in author profiling

Many recent works have addressed the task of author profiling. For example, Squicciarini et al. (2015) tackle the task of identifying cyberbullies in social networks. To determine whether a user is a cyberbully or not, it is important to analyse both their social interactions (e.g. the user writes posts in threads where other users are being bullied) as well as the language used. Another application of author profiling techniques is the detection of spammers in Social Networks. This kind of users utilize Social Networks to target certain demographic segments, to send content from fraudulent accounts: Xu et al. (2016) discuss different word signals and users' characteristics, such as the use of words like "https", "money" or "win" in the posts, for this task. Ultimately, spam detection is related with the credibility of contents. Castillo et al. (2011) focus on automatic methods for reviewing the credibility of a set of tweets. They analyse tweets related to the trending topics, and classify them as credible or not credible, using signals (based on the text of the posts, the citation to external sources and posting and re-posting users' behaviour) extracted from the tweets. The author profiling task in Social Networks is in continuous development, and it is complicated due to the Social Networks heterogeneity; often, useful signals are specific to some network. At the same time, there are some signals that most Social Networks share (such as sending a message or searching for another user (Benevenuto et al., 2009)).

Users' profiling is frequently given by the interactions between users and their environment; for example, Foursquare[8], connects the users with new places thanks to their position and gives a score according to their check-ins. Jin et al. (2016) characterize users from their weekly scores and analyse the correlations between their patterns and the Social Network characteristics (e.g. users' activities may indicate that Foursquare must give more attractive rewards to engage them) of user clusters. Srikanth Reddy et al. (2019) use the terms that appear in hotel reviews to locate the country in which users are located. Other study that take advantage of the environment to perform author profiling is Song et al. (2016), where volunteers can be identified from the content they have been posting on their Social Networks by using linguistic signals, the topics that users are talking about, posting behaviour, etc. They hypothesize that users with more volunteer friends have higher probability to become a volunteer. For this reason, they use a graph-based learning method to better capture the relations between users where the graph represents users' social environments. YouTube[9] is another Social Network of interest for the researchers. Here, the users can upload their videos and share them with the general public. These videos can be scored by the community by pressing Like or Dislike buttons and/or publishing a comment. In Maia et al. (2008), users in the YouTube network are characterized using nine signals: the number of videos uploaded, the number of videos and channels viewed, the date of registration, the age, a clustering coefficient (which measures the interconnection between a user and her neighbours), a reciprocity value (probability of mutual interconnections), an out-degree (number of subscriptions made), and an in-degree (number of subscriptions received). In Ortega-Mendoza et al. (2016), the authors want to know the role played by the personal sentences in the task of Author Profiling. They compare the classification performance when only personal phrases (i.e., sentences containing first person pronouns) and the entire documents are used. Their main discover is that personal phrases have high impact for predicting age and gender of the users in Social Media. Wanner et al. (2017) perform an extensive feature engineering based on the relevance of syntax and discourse, using signals such as character-based (number of exclamations in the text, colon ratio, etc.), word-based (number of characters per word, ratio of tokens in the text that are acronyms, etc.), sentence-based (words per sentence, range of words per

[8] Foursquare: https://es.foursquare.com/
[9] YouTube: https://www.youtube.com/

sentence, etc.), dictionary-based (polarity dictionaries to measure the expressiveness of the authors) and syntactic-based (part of speech, syntactic dependency trees, etc.), and uses Random Forest for age and gender identification using blog posts.

Twitter information is also used to generate profiles of the users. Raghuram et al. (2016) use a supervised learning method which categorizes Twitter users based on three main types of signals: tweet-based, which calculate the term weights according to the number of users that use the term and the number of total users; user-based, which represent the proportion of followers that users have between their followers and friends; and finally, time-series based, statistical signals of the user's time series like average, standard deviation, etc. of the time series. The work also proposes a real time method for author profiling on Twitter. This real time algorithm collects, periodically, a fixed number of random users using the Twitter API[10], extracts the previous signals and classifies them by using an existing machine learning model, Support Vector Machine (SVM) with Principal Component Analysis (PCA) to reduce dimensionality of the signal space, and finally, the users are incorporated to the model by adding them (if they are new) or replacing them (if they are already inserted in the model). When a request of classification is proportioned to the system, the algorithm uses the existing model to classify the new user. Pennacchiotti and Popescu (2011) observed that linguistic signals (e.g. prototypal words, typical lexical expressions and hashtags for people with similar interests, generic LDA, domain-specific LDA and sentiment words) are reliable in order to distinguish political affiliation. This conclusion is aligned with Conover et al. (2011), where using Twitter signals as retweets also provides competitive accuracy.

One of the main applications of author profiling on Twitter is related to predicting the users' demographic characteristics, such as the gender, the age or the profession. Ikeda et al. (2013) propose algorithms to estimate the demographic segment, essential for marketing, of Twitter's users based on their tweets' story and their community constructed from the follower/followee relationships. This type of segmentation takes into account variables such as the age, nationality, gender, religion, etc. The work proposes a hybrid method that first extracts the community to which the user belongs to. Then, it creates three different clusters based on friendship, co-worker and hobbies relationships between the members

[10] https://developer.twitter.com/en/docs.html

of the community and the user. Finally, based on the texts of these clusters and the user, it estimates which segment the user belongs to. Hussein et al. (2019) identify the gender of the Egyptian speakers on Twitter. The authors use a series of text-based signals such as emoticon-based, feminine suffixes, the use of words representing topics, and embeddings to be part of the Mixed Feature Vector (MFV) that, in addition to a N-Gram Feature Vector (NFV), are the input to a Random Forest (RF) and a Logistic Regression (LR) classifier respectively. The output of the system is the combination of both classifiers using ensemble weighting. In Marquardt et al. (2014), the authors want to identify gender and age in Social Media, their study is framed into the PAN 2014 competition[11]. In order to infer age and gender, they use several signals extracted from English and Spanish texts classified as: content-based (e.g. signals that express the sentiment of a sentence), stylistic (e.g. signals that measure the readability of a document) and, in addition, they employ a system of heuristics to predict the gender using a customized lexicon. They conclude that signals that work well across many genres of online textual media may not necessarily perform well on others. Stylistic signals are also used by Patra et al. (2013) for profiling authors by gender and age. Also, the authors of Palomino-Garibay et al. (2015) use lexical, statistical and word-specific signals in order to detect the age and the gender of a tweet writer. Some years later, within the PAN 2017 conference[12], took place another task consisted in identifying both the gender and the specific variation of the native language (such as British English, Spanish from Spain, Portuguese from Portugal, etc.) used by the different profiles in Twitter (Rangel et al., 2017). The systems presented in the competition used a wide range of signals to deal with the problem. The signals employed can be classified according to whether they use the content of the posts (bag of words, the 100 most discriminatory words per class from a list of 500 topic words, LSA, etc.), the linguistic style (ratio of links, hashtags or published mentions, emoticons and/or expressions of laughter), signals that denote emotion (emojis, positive words, etc.), signals that represent documents (word and character embeddings) and traditional signals (tf-idf). According to the results provided, the best signals were the result of mixing emotional, content and style signals.

[11] https://pan.webis.de/clef14/pan14-web/

[12] https://pan.webis.de/clef17/pan17-web/

2.2.2 Recent advances in influencer profiling

Influencers are a special kind of users in Social Networks. They are trustworthy to the members of their communities and their ideas are capable to change other people's mind about an entity, even jeopardizing the entity's reputation. Aral and Walker (2012) defend that, in order to predict the propagation of actions, it is important to use jointly the influence, the susceptibility and the likelihood of spontaneous adoption in the local network around individuals. But, as the authors point out, it is not clear whether influence and susceptibility are general signals or depend on the domain. Sharma et al. (2013) deal with the task of locating *influencers* which are helpful to spread brands' image among potential consumers. Their study is based on the principle of word-to-mouth where some types of clients can be potential brand ambassadors and attract new customers. The authors also discuss the word-to-mouth marketing concept, where the brand that we are scouting is connected to certain subscribers which are grouped (all or a subset of them) as *influencers*, and their friends are treated as potential consumers. One of the implications of this model is that we have to identify the right kind of consumers; the fact that two people are connected does not mean that they have the same tastes.

The problem of influencer profiling can be modeled as a classification or as a ranking task. The first approach is an intuitive choice because we have different items (authors), and we want to put them in different groups (influencers versus non-influencers). Classification approaches for author profiling typically use signals such as the number of followers, the number of published content, etc. in order to learn to predict which users are relevant and which are not. Maleewong (2016) studies how to predict the popularity of news tweets using linear regression. To do this, they study the impact of two main types of retweeters (i.e. people who retweet post): active users and popular users. To model these users, the authors propose two signals: the activity rate, which measures the participation of the users by counting the number of tweets posted plus the number of retweets shared throughout the users Twitter's lifetime; and the popularity score signal which measures the number of Twitter lists, a Twitter list groups profiles related to a specific topic, where the users appear. Pope III et al. (2015) introduce an approach to find *influencers* where the authors used classifiers based on fuzzy logic and linguistic signals such as the part of speech (POS) (noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction, and interjection) for the identification of *influencers*. Ramírez-de-la Rosa et al. (2014) describe a method for classify authors in a Social Network community into influencers or non-influencers. Their hypothesis here is that influential users would have similar writing styles as well as similar posting behaviours. To carry out their experimentation, the authors propose some signals (such as number of URLs used, vocabulary richness, etc.) which capture the stylistic and behavioural characteristics of the users that are the input to several classifiers (i.e. Naïve Bayes, Support Vector Machine, etc.) to distinguish the nature of the profiles. Danisch et al. (2014) measure the influence of users on Twitter by differentiating between social capitalists and regular users. Social capitalists are profiles that gain followers as soon as possible by promising to other users that if they follow their profile, they will follow back. To measure their influence on Twitter they proposed different signals (such as the number of friends, the number of characters per tweet, etc.) and these signals are the input to a logistic regression algorithm that allows them to obtain the probability of being a social capitalist. Using this probability, the authors balance the influence scores of their profiles.

Detecting *influencers* in ORM has two distinguishing signals: first, the number of influencers is orders of magnitude lower than the number of non-influencers. Second, potential influencers are usually scanned by reputation experts, which use automatic filters as a preliminary step. Both signals are characteristic of search problems, where ranking is the most natural way of presenting results to the users (in this case, the reputation experts). This approach is followed in many works such as Katsimpras et al. (2015) where the authors used supervised random walks in order to rank users, using all the textual information available, according to their topic-sensitive influence. The creation of rankings is also applied in Tidke et al. (2018) where the authors focus on localize properties, from known influential users, in terms of links that evolve from online Social Network data. In order to do this, they make two proposals: in the first one, a ranking algorithm called *Weight based Evolving Friends Follower Ranking (WEFFR)* assigns weights capturing the adaptive degree of the relationship; in the second one, which combines the first method with PageRank *(WEEFRPR)*, they measure the reciprocal influence between nodes. In Tsugawa and Kimura (2018), the authors investigate the effect of sampling on the identification of influencers. To do this, they sample a fraction of the Social Network (Twitter and Facebook). They then calculate the influence of each node through its degree of centrality, its proximity to centrality, etc. and generate a ranking of the nodes for each influence measure and take the first nodes of the ranking as influencers that will be evaluated according to their capacity to disseminate information. The sampling techniques used are: *sample edge count* (SEC) that obtains the node with the most expected measure until having the desired nodes (Maiya and Berger-Wolf, 2011), *breadth-first search* (BFS) that selects a random node and visits the neighbours of the node visited previously until having the desired nodes (Maiya and Berger-Wolf, 2011), *depth-first search* (DFS) which is a variation of the BFS that only visits the neighbours of the node visited most recently until having the desired nodes (Maiya and Berger-Wolf, 2011), and *random sampling* which selects nodes randomly until the desired number of nodes are extracted. The authors conclude that, for large-scale social networks, sampling is useful to locate influencers, while in collaborative social networks it is not a good mechanism. The authority of Twitter users can vary over time as well and future rankings will depend on the evolution of the influence. In this article (Simmie et al., 2014), the authors propose a model to capture both the invariant influence over time and the temporal influence to generate a ranking system of the most influential users that allows to give predictions on future states using current evidence. Aligned with the evolution of profiles over time, the authors of Liang et al. (2018) study the task of users dynamic profiling. They infer both the user and the dynamic representation of the words over the time and identify, dynamically, the K-top more relevant users and diversify keywords for every user profile. They propose a word embedding and dynamic user model called *DUWE*, which models simultaneously both models in a temporal way.

2.2.3 RepLab 2014 author profiling task

RepLab[13] was an evaluation campaign for reputation monitoring systems which, in its 2014 edition[14], included two author profiling tasks on Twitter: *author categorization* and *author ranking*. The first of these tasks, author categorization, consisted in classifying Twitter profiles into different types of authors (e.g. journalist, activist, investor, etc.). The second task, author ranking, aimed to distinguishing the users with the most reputational influence from the less influential users. In our work, we address the author ranking task and use for our experiments the RepLab 2014 dataset, which is described in section 3.2.2.1. In the

[13] RepLab: http://www.clef-initiative.eu/track/replab

[14] http://nlp.uned.es/replab2014/

following lines, we describe systems that participated in the competition and other methods that have appeared over the years and have also contributed to progress in the task.

The best system in the competition was AleAhmad et al. (2014), which implemented the idea that people who are opinion makers will talk more about hot topics. The algorithm distinguishes different topics by using simple keywords (the hashtags that appear in the tweets), but it does not perform well in domains with more diverse tweets. To solve this problem, outside the official running, the authors created two different categories of topics: grouping hashtags, so that the extracted keywords are grouped to form a number of representative topics in each domain; and using topic modeling, LDA, which is used to create some other topics for each domain. The combination of both topic sets is used instead of the simple distinction of keywords improving their previous results. Another approach proposed in the RepLab competition (Cossu et al., 2014b) assumed that influencers tend to produce more opinionated content in tweets. In their approach, they combine Poisson models (Bahl et al., 1988) and Hidden Markov Models (HMM) for both English and Spanish tweets. The system behaves as a binary classifier where each tweet in the author bag of tweets is opinionated or not. The system considers the majority label and decides whether or not the user is "opinion maker". The output of the system, a ranking of users, is created using the probability of being "opinion maker" and, in case of parity, they add the probability of the HMM. Vilares et al. (2014) proposed a system based on classifiers that uses the bag of words extracted from Twitter users' profile descriptions as signals. The authors presented two runs to the competition: in the first one, they feed a LibLinear classifier (Fan et al., 2008) with Twitter profile descriptions. The authors hypothesise that, since the corpus is domain-dependant, the biography of the users may be an indicator of influence. The second run uses meta-information provided by Twitter such as if the profile is a verified account, is geo-localization enabled, its number of followers, its number of friends, etc. Villatoro-Tello et al. (2014) used techniques for signal extraction and collected the most representative signals from each user's activity domain. The authors propose a two-step chaining method where the first step is a supervised approach and the second phase is an unsupervised approach that uses a Markov Random Field (MRF) (Kindermann, 1980). The supervised approach uses two different set of signals which capture stylistic and behavioural characteristics from each user that are classified as: self-description (words in the user's profile, mentions, number of hashtags, etc.) and statistics of use (number of tweets, number of followers, number of followees, etc.) signals. Once the signals are extracted, the authors tackled the ranking problem as a binary classification task where a SVM classifies the users as opinion makers or non-opinion makers and the confidence of the classifier is used to rank the users. The unsupervised step uses a MRF to refine the ranking created by the supervised approach. The MRF is a fully connected graph where each node is a user, which is represented as 2 random variables: opinion maker and non-opinion maker, also, the authors, defined a neighbourhood scheme in which each variable is adjacent to all the others. For estimate the similarities the authors define stylistic and behavioural signals extracted from the users' tweets called Style-Behavior signals (number of URLs, hashtags, user mentions, etc.). Finally, the initial configuration of the MRF defines as true opinion makers, the subset of users that were classified as opinion maker by the supervised phase and all other users as non-opinion makers. Then, the MRF configuration of minimum energy (Mean Average Precision, MAP) is obtained using the Iterated Conditional Modes (ICM) algorithm (Besag, 1986). At the end of this optimization process, a new re-ordered list is produced. The last participant (Lomena and Ostenero, 2014) proposed a method where the tweets of an author are the input of 4 different classifiers, the signals to feed the classifiers are: the number of followers and the average retweet speed (the authors examine the last retweet performed by the users, the main idea is that users with high speed are more active and they tend to create content). The final prediction of the user's class (opinion maker or non-opinion maker) is done by the majority selection of the classifiers and the ranking is created by sorting the users using the following formula:

$$weight = \frac{NumberofFollowers}{averageRTSpeed} \tag{2.1}$$

The hypothesis here is that it is more relevant a profile with a smaller number of followers and higher speed, than a profile with a bigger number of followers and lower speed. Using the same dataset, Cossu et al. (2016) tested a wide range of signals (classified in different categories such as user profile, publishing activity, local connections, etc.), using different Machine Learning (ML) algorithms (kernelized SVMs and also, Random Forest). The authors concluded that users from particular domains behave and write in their own specific way and using only text-based signals is enough to detect domain opinion makers. Aligned with the use of text signals, Nebot et al. (2018) obtained the best results, as far as we know, using the same dataset. Here the authors represent each document as an embedding of six signals extracted from the published texts: (i) average value of the document term weights; (ii) standard deviation of the document term weights; (iii) minimum value of the weights in the document; (iv) maximum value of the weights in the document; (v) overall weight of a document as the sum of weights divided by the total number of terms of the document; and (vi) proportion between the number of vocabulary terms of the document and the total number of terms of the document, which weights are learned by different machine learning algorithms: Naive Bayes and Bayes nets.

2.3 Automatic summarization

Since the advent of the Internet in the 20th century, the publication of contents has grown day by day making impossible to process, manually, all published information in a reasonable time. There is a need, therefore, to automatically extract and summarize all relevant (important to the users) information and put it in a more readable way in other words, to include essential information but in a shorter way. There are two main types of summary generation: *extractive* and *abstractive* (Das and Martins, 2007). Extractive summarization methods extract word sequences (phrases, sentences or paragraphs) from the original documents and copy them into the summary directly. This technique has the problem of the lack of coherence between sentences of the summarized document but stands out for its computational simplicity. Abstractive summaries are more difficult to create, because they involve paraphrasing the text in the source documents and generating text by using Natural Language Generation techniques, but they address the problem of cohesion between sentences in the summary (Das and Martins, 2007). Automatic summarization approaches may be also classified depending on the number of input documents that the system receives: *single-document* and *multi-document* (Nenkova and McKeown, 2012). Whereas the first approach creates automatically the summary from the information within a single document (Litvak and Last, 2008), the second approach uses the information obtained from different sources, talking about the same topic, to generate the summary (Lin and Hovy, 2002). This last approach may introduce redundant information (content that is expressed more than once) to the summary and, therefore, some mechanisms are necessary to avoid this problem (Inouye and Kalita, 2011; Takamura et al., 2011).

During the last 10 years automatic summarization field has been applied in different research fields such as computer science, mathematics, linguistics etc. Figure 2.8 show these fields, according to the data in the Web of Science.

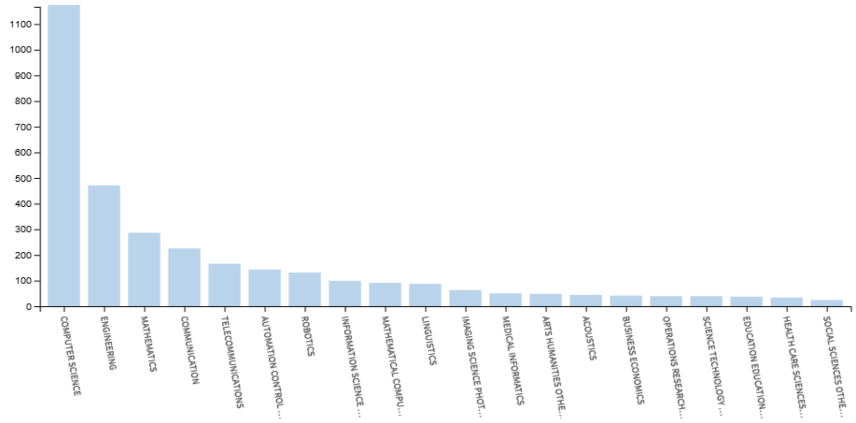


Figure 2.8: Research fields interested in the automatic generation of summaries

In the following sections we review full proposals for extractive and abstractive summarization and we also provide a review for a special case inside the automatic generation of summaries: reputation reports.

2.3.1 Extractive summarization overview

As already told, an extractive summary is composed of word sequences (sentences, or paragraphs) that are directly extracted from the original document and copied into the summary. This approach is simple but it has the problem of the coherence between sentences in the summarized document. The resulting summary condenses information that allows human operators to keep up to date with information contained in large sets of documents. Systems receive as input a collection of documents, and produce a condensed summary that avoids repeated or *redundant* information that does not provide new data to the summary. Figure 2.9 shows a basic architecture of an extractive summarization system.
The architecture of the system is formed for three main or basic modules: the first one extracts signals from the input document(s). Then, the second module ranks the sentences within the document(s) according to the previous extracted signals. Finally, if we are in the multi-document scenario, the third module removes

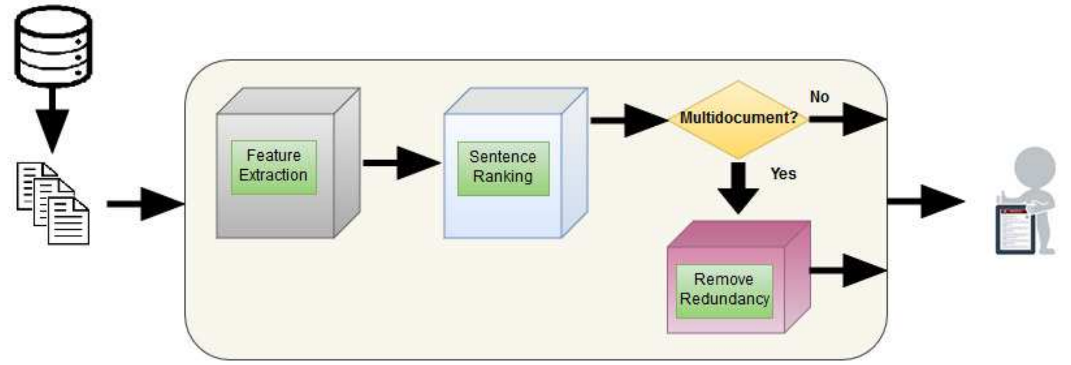


Figure 2.9: Architecture of an extractive summarization system

redundancy from the previous ranking (i.e. avoids including in the summary sentences that contain the same information). As the system output, a summary is retrieved.

In the next section, we review previous works that have tackled the task of automatic generation of extractive summaries by the type of techniques used and the type of the signals that are exploited. The problem of summarizing tweets on a company's reputation has been, to the best of our knowledge, never tackled before and presents additional challenges derived from the less massive availability of data and the greater diversity of issues involved.

2.3.1.1 Sentence-based signals for extractive summarization

The use of simple signals was the first method to generate automatically extractive summaries. The first automatic summarizer was created by Luhn in 1958 (Luhn, 1958). The system used the word frequency and the phrase frequency to build a ranking of the sentences in a document and the top ranked sentences were selected to be part of the summary. The main idea is that the word frequency is a useful measure for significant factor of a sentence but not all words are important, stop words (words without any semantic information, such "a", "the", etc.) are not considered for calculating the term frequency. Despite its long history, this technique is still used today for the generation of summaries (Abuobieda et al., 2012; Gupta et al., 2011; Shardan and Kulkarni, 2010). Term frequency is not enough if we want to calculate the importance of a sentence according to its more important terms. Terms are important if they are not very frequent in the whole collection. The *tf-idf* method compares the term frequency (tf) (Luhn, 1958) in a document with the times that the term appears along the document collection (idf) (Sparck Jones, 1972). This technique is still used today in extractive summarization works (Christian et al., 2016). Here, the authors create a method to generate extractive summaries from a single document using the tf-idf of the words. Each sentence in the document is scored by summing the tf-idf of the names and verbs that appear in the sentence and they are sorted in descending order according to the score obtained. Those ones with highest scores are selected to generate the summary until a compression rate is reached. Other main signals extracted from phrases that have been used to generate extractive summaries are: sentence length (Kupiec et al., 1999), use of proper nouns (Neto et al., 2002), etc.

There are some other helpful signals that indicate salient parts of the document, i.e. the position that the sentences occupy in the document. The hypothesis here is that texts that belong to a genre generally have a predictable structure, and the sentences with higher relevance tend to occur in specific locations (i.e. the first sentences in a news) (Lin and Hovy, 1997). In their study, the authors of this idea show an approach called Optimal Position Policy (OPP) which determines these positions in the texts. More recently, the authors of Hu et al. (2017) propose a summarization technique for identifying the most informative sentences of hotel reviews. The authors define the importance of a sentence in the review as a combination of three signals being one of them the sentence position. Other signals used in the state-of-the-art related to the sentences are: the length of the sentence (that is used to penalize sentences that are too short (Fattah and Ren, 2009) or too long (Abuobieda et al., 2012)) and the similarity of the sentences with the document title (Gupta et al., 2011).

2.3.1.2 Cluster or topic-based extractive summarization

Cluster or topic-based approaches have been widely used in text summarization. Here, the techniques used basically group the data that contains similar textual information (i.e. news that talk about the same fact, conversations about a football match, etc.) and select one or more representative texts of the cluster, avoiding to add duplicate (redundant) information to the summary. In Roul et al. (2019) for example, the authors address this problem by designing a method that extracts the different topics found within the documents, using the *Latent Dirichlet Allocation* (LDA) technique along with a heuristic that finds the optimal number of independent topics within the dataset. For each of these topics, the most important sentences are identified, using word and sentence level signals, and the summaries are created in such a way that the sentences within it are coherent, this coherence is achieved by rearranging sentences based on their similarity. Saini et al. (2019) proposal consists of generating clusters, using three different multi-objective optimization techniques: self-organized multi-objective differential evolution which is an integration of self-organizing map (SOM) and multi-objective differential evolution (MODE); multi-objective grey wolf optimization (Mirjalili et al., 2016) and multi-objective water cycle algorithm (Sadollah et al., 2015). Thereafter, the sentences of each cluster are ranked according to different signals: centrality, similarity with the title, length, etc. and finally, top ranked sentences are selected and are included in the summary. Alsaedi et al. (2016) modify a traditional centroid approach by including the time dimension of tweets, so that tweets that have been centroid of the clusters for the longest time on average over a time-window are selected for the summary. Zhuang et al. (2016) create a model, called S-model, which takes advantage of two social contexts that are important for topic generation and dissemination: the impact of experts and majority users, as as well as the content diversity based on entropy measures. Concerning the subject of the input tweets, most works have focused on sport and celebrity events (Inouye and Kalita, 2011; Sharifi et al., 2010). These events are massively reported in Social Networks, so that the number of tweets to summarize is huge. In this context, simple frequency-based summarizers perform well and even better than summarizers that use more complex information (Inouye and Kalita, 2011). The most closely related work is that of Louis and Newman (2012), which present a method for summarizing collections of tweets related to a business. To this end, they first learn groups of related words from business news articles that describe relevant business concepts. Next, tweets related to each company are clustered using a method that combine the sentiment of a tweet and the entropy of word distribution in the cluster, so that clusters discussing common issues are ranked higher than clusters with diverse content. Finally, the clusters are ranked using information such as influential subtopic and sentiment.

According to Mubarak (2016) cluster-based techniques have the main advantage of generating significant sentences from source text but their main disadvantage is that they do not consider the semantic aspects such as synonymy and polysemy.

2.3.1.3 Graph-based extractive summarization

Graph-based ranking algorithms have been widely used for text summarization. This technique, basically, represents textual information (i.e. words, sentences or even paragraphs) in the nodes of the graph and the edges between nodes are relations between the textual entities (e.g. semantic relations such as hyponymy, synonymy, polysemy, etc.). The main idea is that the graph's shape indicates properties that the salient elements have. A well-known summarization algorithm that uses a graph approach is LexRank (Erkan and Radev, 2004a) which is a multi-document summarization system. The idea in this work is that the importance of the sentence is given by the eigenvector centrality in a graph representation of sentences. This graph is calculated as an adjacency matrix that is modeled as a connectivity matrix based on intra-sentence cosine similarity. Then, PageRank (Page et al., 1999) is applied to the resulting graph and the sentences are ranked according to the scores obtained. The final summary is formed by extracting the top-k sentences, where k is the size of the summary. Mihalcea and Tarau (2004a) introduce a graph-based ranking model for text processing. This algorithm is used in multiple natural language applications, in particular, for sentence extraction. In this task, the authors create a graph where the nodes are the sentences and the edges measure the similarity between them. The resulting graph indicates the strength of the connections between pairs of sentences within the text. Next, they build a ranking of sentences by reversing the order of their score and the top ranked sentences are included in the final ranking. Litvak et al. (2013) present an unsupervised graph-based language-independent extractor. Here the nodes represent textual information meanwhile the edges are the order relation between nodes. The most connected nodes are assumed to represent the most salient textual information and every node is ranked according to its connectedness with other nodes. The top ranked nodes are extracted and included in the summary. Another example of the use of graphs applied for extractive summarization is Nguyen and Nguyen (2016). In this work, its authors propose a framework that uses Twitter information to generate summaries from a single document (a tweet). The document's sentences are linked to tweets by recognizing textual entailment (TER). They are then modeled using Dual Wing Entailment Graph (DWEG) which captures the entailment relation in order to calculate the sentence similarity. Finally, the most important sentences and the most representative tweets are ranked and added to the summary. Regarding to the use of graphs approaches applied to Twitter, the works here usually adapt traditional summarization systems (such as LexRank (Erkan and Radev, 2004b), DegExt (Litvak et al., 2013) and TextRank (Mihalcea and Tarau, 2004b)) to take into consideration the particularities of Twitter posts (Inouye and Kalita, 2011; Liu et al., 2012; Sharifi et al., 2010). These approaches usually include both content-based and network-based information into the text graph. Liu et al. (2012) propose a graph-based multi-tweet summarizer that leverages social network signals, readability and user diversity for selecting representative tweets. As Social Network signals, they consider the re-tweeted times and follower number of the Twitter account that produces the tweet. Diversity is introduced by preferring tweets from different user's accounts. However, the fact that one user may post from different accounts is not addressed. Finally, readability is assessed using traditional criteria such as the sentence length, the word length in syllables, the number of abnormal symbols and the number of out-of-vocabulary words. Similarly, Duan et al. (2012) develop a method that implements a graph-based ranking algorithm that takes into consideration both social influence of users and content quality of tweets. The most popular algorithm for microblog summarization is presented in Sharifi et al. (2010). The authors propose a topic-oriented summarization system for Twitter posts that automatically summarizes a collection of posts related to a same topic into a short, one-line summary. The system is based on the sentence reinforcement algorithm that iteratively constructs a graph for the set of posts where the nodes are overlapping sequences of words that occur both before and after the topic sentence. Nodes are labeled with the number of times each sequence of words occurs in the collection. Most overlapping phrases are selected to generate the summary.

2.3.1.4 Machine Learning extractive summarization

The approaches that we describe below are based on machine learning algorithms to produce their summaries. The machine learning methods are used to model the problem of extractive summarization as a binary classification task dividing all sentences into summary and non-summary sentences. Kupiec et al. (1999) proposes a Naïve Bayes classifier that estimates the probability of a sentence to be part of a summary. To do this, the authors extract some signals (sentence length cut off, uppercase word, etc.) from the training data to train the classifier. Once the classifier is trained, the test sentences are evaluated and ranked according to their score. The top-k sentences are included in the final summary. Following this line of work, Neto et al. (2002) test two different machine learning algorithms, Naïve Bayes and C4.5, and different signals (such as the occurrence of proper names, the occurrence of non-essential information ,etc.) to produce summaries. The authors show that depending on the classifier used, the performance of the summarizer may vary.

Regarding to extractive summarization using machine learning algorithms on Twitter, we highlight the work of El-Fishawy et al. (2014), which summarizes Arabic posts in Twitter. The authors propose different signals (such as term-frequency, number of followers, tweet length, etc.) that model the tweets, and then, these signals are filtered to keep those signals that better fit with the problem by using a Correlation Feature Selection (CFS) method (which gives high scores to subsets that include signals that are highly correlated to the class attribute but have low correlation to each other). Finally, each tweet receives a score according to the model tree algorithm used to calculate the weights of the important signals. In this work, the authors formulated the task as a regression problem instead of a binary classification. One of the reasons to change the approach of the problem is that formulating extractive summarization as a binary classification task may not provide the best possible summary. This is due, to the fact that the classification task provides individual sentence scores which it is not equivalent to find the best summary, which consists of several sentences (Nenkova and McKeown, 2012).

To carried out the previous studies, the authors need labelled data to train their models. This is a problem because labelling data requires a huge effort and sometimes the number of instances labelled are not enough to train a classifier. Wong et al. (2008) used a semi-supervised approach to reduce the labelling cost by combining labelled and unlabelled data. The authors used a co-trained probabilistic Support Vector Machine (SVM) and a Naïve Bayes classifier to exploit the unlabelled information. The co-trained approach assumes that each example is described by two different sets of signals that provide complementary information about the instance. The co-trained algorithm learns a separate classifier using any labelled example. Then, the classifier with the most confidence value over the unlabelled data is used to iteratively construct additional labelled information (Blum and Mitchell, 1998). The signals involved in this experimentation are grouped into different types (surface, content, relevance and event signals).

As we have seen, one of the main characteristics of the machine learning approaches are, besides the methods used, the high number and the different nature of the signals used to carried out the experimentations. These machine learning algorithms have the particularity that they can test the performance of high number of signals but, otherwise, they need a big corpus to extract conclusive results (Lloret and Palomar, 2012). On the other hand, they provide more generalized summaries but at the cost of the lack of semantic analysis of source text (Mubarak, 2016).

2.3.1.5 Priority versus centrality-based summarization

Since the pioneering works in automatic summarization, centrality has been one of the most widely used criteria for content selection. Centrality refers to the idea of how much a fragment of text (usually a sentence) covers the main topic of the input text (a document or set of documents). Centrality of a sentence is often defined in terms of the centrality of the words that it contains. Given a cluster of sentences that represents a document topic, the sentences that contain more words from the centroid of the cluster are considered as central (i.e. most representative of the document topic).

A great number of summarization systems use centrality to identify relevant sentences for the summary, along with an algorithm to avoid redundancy (Erkan and Radev, 2004b; Litvak and Last, 2008; Mihalcea and Tarau, 2004b; Zhang et al., 2011). Concerning more recent approaches, Cho and Kim (2015) propose a Social Network-inspired method for the extraction of key sentences from a document. To this end, sentences are represented by their TF-IDF scores and connected according to the co-occurrence of keywords among them. Sentences are then scored based on their centrality in the co-occurrence network. Marujo et al. (2015) use a multi-document approach based on KP-centrality (i.e. the centrality of key phrases found within the text). KP are extracted from the documents using supervised machine learning on a bag of words model, and then are used to build a pseudo-passage that represents the central topic of each document (centroid). Most representative passages from each document are then extracted based on their closeness to the centroid, and then merged to build the multi-document summary. Sarkar et al. (2015) improve the computation of the similarity between sentences to produce a single summary from a set of related documents. They build a graph were nodes represent sentences and edges are added between nodes representing similar sentences. Centrality of sentences is then computed as the degree of the nodes, and next ranked according to such centrality and extracted to generate the summary.

However, the information need of users frequently goes far beyond centrality and should take into account other selection criteria such as diversity, novelty, priority, authority and belonging to a specific domain. This is specially true in the reputational scenario. Although the importance of enhancing diversity and novelty in various NLP tasks has been widely studied (Clarke et al., 2008; Mei et al., 2010), reputational priority is a domain-dependent concept that has not been considered before. Other priority criteria have been previously considered in some domains and scenarios: Plaza and Carrillo-de Albornoz (2013), for instance, showed that it is possible to improve summarization of scientific articles by prioritizing sentences covering the methods and results of the experiments reported in the articles. Similarly, Meena and Gopalani (2015) used the location of the sentence in a general-domain text as the main indicator of its priority, along with the presence of named entities and proper nouns. In (Fiszman et al., 2009), concepts related to treatments and disorders are given higher importance than other clinical concepts when producing automatic summaries of MEDLINE citations. In opinion summarization, positive and negative statements are given priority over neutral ones. Moreover, different aspects of the product/service (e.g., technical performance, customer service, etc) are ranked according to their importance to the user (Pang et al., 2008). This is sometimes referred as to aspect-based summarization, and has been recently tackled using convolutional neural networks (Wu et al., 2016). Priority is also tackled in query (or topic)-driven summarization, where terms from the user query are given more weight under the assumption that they reflect the user relevance criteria (Litvak and Vanetik, 2017; Nastase, 2008).

In the ORM scenario, priority refers to the importance of comments and opinions made by users for the company being analyzed. The priority detection problem in ORM was addressed at RepLab 2013 contest (Amigó et al., 2013). The systems participating showed that priority depends on a set of relevance criteria including the centrality of the topic, the influence of users that discuss on the topic, the sentiment of the comments (Cossu et al., 2014a), to name a few. However, the results of RepLab 2013 prove that priority classification for ORM is still an open problem and that further investigation on relevant priority signals must be done.

2.3.2 Abstractive summarization overview

According to Das and Martins (2007): "abstractive summarization puts strong emphasis on the form, aiming to produce a grammatical summary, which usually requires advanced language generation techniques". In the next sections, we review previous works that have been tackled the task of abstractive summarization by the type of techniques used.

2.3.2.1 Natural Language generation

There are many research papers that have developed abstractive summarizers. Riya Jhalani (2017) use a method, restricted to news articles on disasters or accidents, that employs sentence generation patterns on domain knowledge and dependency relations to generate summaries. The NLG techniques demand a lot of effort in terms of defining schemas as well as using deeper natural language analysis. One way to solve this problem is to generate templates (Oya et al., 2014; Wang and Cardie, 2013) but in an environment where topics are very variable, such as in the generation of news summaries, it is not a very effective technique. Gerani et al. (2014) generate a natural language summary by using a template-based NLG framework from an aspect tree. To do this, they rely on the information contained in the discourse structure of the product reviews. They first apply a discourse parser to each review to get a representation of the discourse tree. Then, they modify each leaf of the trees so that they contain only aspect words. They add the aspects and generate a graph from the discursive trees and select the sub-paragraph that represents the most important aspects, by using PageRank (Page et al., 1999), and transforming the sub-paragraph into an aspect tree. Banerjee et al. (2015), on the contrary, generate summaries in an unsupervised manner by combining information from several sentences on the same topic. Cheung and Penn (2014) join and re-assemble dependency analysis trees to produce abstractive summaries. The abstractive-approach discussed in Bing et al. (2015) takes advantage of small semantic units, named as noun/verb phrases, by merging them in order to maximize the salience of phrases extracted from the original documents.

Recent researches employ neural models for the generation of abstractive summaries. See et al. (2017) introduce an architecture which augments the standard sequence-to-sequence attentional model in two ways: using a hybrid pointer-generator which helps to reproduce more accurately information and produce novel words, and utilizing coverage to avoid repetitions in the final summary. Paulus et al. (2017) present a neural network model with intra-attention that deals separately with the input and output, which is generated continuously. Standard word prediction is combined with Reinforcement Learning's global sequence prediction training to generate more readable summaries. Li et al. (2017) base their abstract generation method on a sequence-to-sequence oriented encoder-decoder model equipped with a deep recurrent generative decoder (DRGN). Learning is based on a random recurrent latent model that improves the quality of the summary and the summary is based on both generative latent variables and discriminative deterministic states. In Zhou et al. (2017), the authors imitate human's summarization process: human selects the highlighted information before making the summary. To do this, they create a selective encoding model that extends the standard sequence-to-sequence used to generate abstractive summaries with a selective gate network. It builds a second level of sentence representation by controlling the flow of information between the encoder and decoder. Lately, several works have combined the extractive and abstractive generation of the summaries using neuronal models. Tan et al. (2017) for instance, use a hierarchical autoencoder with an attentional decoder, where the attention is calculated from a graph-based attention mechanism. This mechanism, is inspired in the graphs models widely used in the extractive scenario (LexRank, PageRank, etc.) and, it calculates the salient scores of the sentences that are part of the input documents. Chen et al. (2018) capture the semantic of the document by using a shared hierarchical encoder, an attention-based decoder for abstractive summarization and an extractor for sentence-level extractive summarization. With this approach, they obtain a better capture of the semantics and higher consistency.

2.3.2.2 Integer Linear Programming

Integer Linear Programming (ILP) (Schrijver, 1998) is the name given to *Linear Programming* (LP) (Nering and Tucker, 1993) with the additional constraint that some or all the variables have to be integers. There are different research works that have used this paradigm to create abstractive summary systems. The proposal of Berg-Kirkpatrick et al. (2011) uses an ILP formulation that extracts and compresses sentences in order to create the summaries. Due to the supervised nature of the method, it requires an extra effort to define signals for compressing the sentences. The signals used in this work are classified as bigram and subtree deletion signals. Banerjee et al. (2015) formulate the summary generation as an ILP problem, where the problem is to maximize all K shortest paths between the start and the end node of a word-graph structure. Similar sentences in the input documents are grouped under the same cluster, and each group, is used to build the corresponding word-graph. Each path is represented by the ILP problem as a binary variable. The solution to the optimization decides the paths to be included in the final abstractive summary which maximizes the information content and linguistic quality. Woodsend and Lapata (2012) use an ILP framework which allows them to combine the decisions made by expert learners and to select and rewrite input content using a mixture of objective setting, soft and hard constraints. Bing et al. (2015) use ILP for selecting phrases and merges them for obtaining an optimal global solution for their summaries. In Nayeem et al. (2018), the authors have designed an abstractive phrase-level fusion generation model that performs the fusion between sentences and paraphrasing. For the sentence selection, they rely on ILP, which allows extracting the sentences that maximize coverage and also ensures that the length of the abstract is within the pre-established parameters. In Li et al. (2015a), the authors add new signals to the ILP method of summary generation. For that, they use the syntactic information selected from the most important bigrams and estimate this importance by adding to the internal signals of the training documents (frequency of the document, positions of the bigram, signals extracted from external resources such as Dbpedia, Wikipedia, etc.). In this paper (Luo et al., 2018), the authors propose to augment the ILP-based summary framework to summarize documents that have been written by several authors and that have a great lexical variety. To do this, they use a low ranking approach of the co-occurrence matrix, and use lexically diverse data to further evaluate. Rudra et al. (2016) generate abstractive summaries from texts (tweets) that talk about alert or hazard situations using a two-phase approach. In the first of these phases, the ILP optimization technique is used to extract the most important tweets from the entire data set they have in order to generate a readable and informative summary. In the second phase, a graph of words and concepts of the events is used to produce the final abstractive summary. In line with the generation of event-oriented summaries, Li et al. (2016) propose the generation of concise and coherent summaries by extracting fine-grained events and building an event semantic link network. To do that, they propose a reduction of the network based on the ILP algorithm to obtain semantic information from the source texts, more condensed, meaningful and coherent.

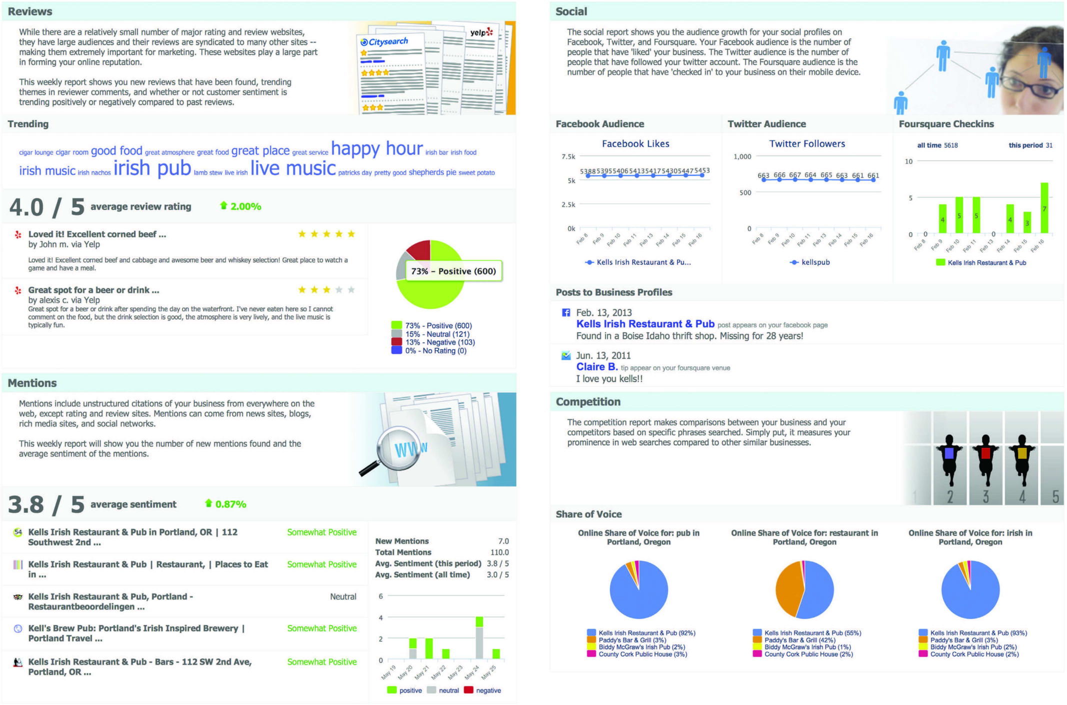
2.3.2.3 Graph-based abstractive summarization

This technique that is widely used in extractive summarization has become very popular in the abstractive summarization task. Ganesan et al. (2010) generate concise abstractive summaries, from highly redundant opinions, using graph-based approach. The algorithm represents textual information as a graph and finds the appropriate path that corresponds to a meaningful sentence in the graph to generate the abstractive summary. Mehdad et al. (2013) propose a supervised approach for summarization in which they generate a linked graph of sentences. The non-redundant and informative sentences are located thanks to the graph edges, which are the relations between linked sentences (the nodes of the graph). Their fusion approach used *Multi-sentence compressio*n (MSC) (Filippova, 2010), which generates an informative sentence by combining several sentences in a word-graph structure. Liu et al. (2018) use *Abstract Meaning Representation* (AMR) graphs to parse the source text. They use a graph-to-graph transformation to generate a summary graph which is created through the reduction of the source semantic graph. With this summary graph, the final text of the abstractive summary is generated. Moawad and Aref (2012) also use a graph-approach to tackle the abstractive summaries generation task. They summarize the input document by generating a semantic graph for it. This semantic graph enriches the traditional one by associating attributes to the graph nodes. After that, the approach reduces the semantic graph to a more abstracted one, and then it generates the abstractive summary from the final graph. In Yasunaga et al. (2017) the authors propose a system of multidocumental summaries in two steps: salience estimation of the sentences within documents and the selection of the sentences for the summary generation. To perform the first of the tasks, the system generates an embedding per sentence. They generate a sentence relation graph whose nodes are the sentences connected through edges. In their article (Niu et al., 2017), the authors propose an abstractive multi-document summarization system based on chunk-graph (CG) and recurrent neural network language model (RNNLM). The CG is based on word-graph and is used to organize all the information related to the sentences clusters. This approach allows to reduce the size of the graph and maintains more semantic information than the word-graph. They use each CG sentence cluster along with a beam search and a character-level RNNLM to generate the summaries. In the article Olariu (2014) the authors create a system for generating abstractive summaries from tweet flows by generating a word-graph. The summary is generated by finding the path with the highest score in the word-graph. To do this, the search begins by selecting the node with the highest weight and the node is expanded in order to maximize the score function that is favourable to the bigrams with higher frequency. The authors of this article (Bhargava et al., 2016) use graph-based techniques to generate abstractive summaries of redundant opinions and use sentiment analysis to combine statements. Sentences in a document are represented by a directed graph, the nodes are the words and also, contains information regarding to the word position in the sentence, POS tag information of the word in that node and the position of the sentence in the document. The edges between nodes represent the adjacency of the words in the sentences. Once the graphs are built, the algorithm scores the paths obtained based on the redundancy of the overlapping sentences. Then, it fuses two sentences if they share a verb. If the candidate sentences to be fused have the same sentiment, the algorithm uses an “and”, “or”, etc. connectors but, if they have contradictory sentiments, it uses “but” as a connector between them. Once the scores are obtained and the sentences have been fused, the algorithm ranks the sentences in descending order and removes similar sentences using Jaccard similarity and finally, selects the top-k sentences to be part of the final summary.

2.3.3 Automatic generation of reports

A case study of the application of automatic summaries is the generation of reputation reports. These reports not only collect and summarize the topics mentioned in the Social Networks concerning an entity (company, product, person, etc.) but also may include various statistics taken from the information flows, for example, the number of alerts on the total number of tweets collected, positive mentions, etc. Figure 2.10 shows an example of reputation report.

As far as we know, there are no works about the automatic report generation in ORM, but we review the automatic report generation in other domains. There are two main ways to generate reports. In the first of them a *template* is used, there is a predefined text that will be filled with the necessary data (Duboue, 2016; Yang et al., 2013). The second variant generates natural language and adapts



Reviews

While there are a relatively small number of major rating and review websites, they have large audiences and their reviews are syndicated to many other sites -- making them extremely important for marketing. These websites play a large part in forming your online reputation.

This weekly report shows you new reviews that have been found, trending themes in reviewer comments, and whether or not customer sentiment is trending positively or negatively compared to past reviews.

Trending

cigar lounge cigar room good food great atmosphere great food great place great service happy hour irish bar irish food irish music irish nachos irish pub lamb stew live irish live music patricks day pretty good shepherds pie sweet potato

4.0 / 5 average review rating ⬆ 2.00%

Loved it! Excellent corned beef ... ★★★★★
by John m. via Yelp
Loved it! Excellent corned beef and cabbage and awesome beer and whiskey selection! Great place to watch a game and have a meal.

Great spot for a beer or drink ... ★★★☆☆
by alexis c. via Yelp
Great spot for a beer or drink after spending the day on the waterfront. I've never eaten here so I cannot comment on the food, but the drink selection is good, the atmosphere is very lively, and the live music is typically fun.

Mentions

Mentions include unstructured citations of your business from everywhere on the web, except rating and review sites. Mentions can come from news sites, blogs, rich media sites, and social networks.

This weekly report will show you the number of new mentions found and the average sentiment of the mentions.

3.8 / 5 average sentiment ⬆ 0.87%

Kells Irish Restaurant & Pub in Portland, OR | 112 Southwest 2nd ...	Somewhat Positive
Kells Irish Restaurant & Pub | Restaurant, | Places to Eat in ...	Somewhat Positive
Kells Irish Restaurant & Pub, Portland - Restaurantbeoordelingen ...	Neutral
Kell's Brew Pub: Portland's Irish Inspired Brewery | Portland Travel ...	Somewhat Positive
Kells Irish Restaurant & Pub - Bars - 112 SW 2nd Ave, Portland, OR ...	Somewhat Positive

New Mentions	7.0
Total Mentions	110.0
Avg. Sentiment (this period)	3.8 / 5
Avg. Sentiment (all time)	3.0 / 5

Social

The social report shows you the audience growth for your social profiles on Facebook, Twitter, and Foursquare. Your Facebook audience is the number of people that have 'liked' your business. The Twitter audience is the number of people that have followed your twitter account. The Foursquare audience is the number of people that have 'checked in' to your business on their mobile device.

Facebook Audience | Twitter Audience | Foursquare Checkins

Posts to Business Profiles

Feb. 13, 2013
Kells Irish Restaurant & Pub post appears on your facebook page
Found in a Boise Idaho thrift shop. Missing for 28 years!

Jun. 13, 2011
Claire B. tip appear on your foursquare venue
I love you kells!!

Competition

The competition report makes comparisons between your business and your competitors based on specific phrases searched. Simply put, it measures your prominence in web searches compared to other similar businesses.

Share of Voice

Citysearch
yelp
73% – Positive (600)
73% - Positive (600)
15% - Neutral (121)
13% - Negative (103)
0% - No Rating (0)
WWW
6
4
2
0
May 19
May 20
May 21
May 22
May 23
May 24
May 25
positive
neutral
negative
Facebook Likes
7.5k
5k
2.5k
0k
5388 5395 5406 5413 5417 5430 5447 5453
Feb 8
Feb 9
Feb 10
Feb 11
Feb 13
Feb 14
Feb 15
Feb 16
Kells Irish Restaurant & Pu...
Twitter Followers
1,000
500
0
663 666 667 664 665 663 661 661
kellspub
all time 5618
this period 31
10
5
0
0 4 5 5 0 4 3 7
Kells Irish Restaurant & Pub
1
2
3
4
5
Online Share of Voice for: pub in Portland, Oregon
Kells Irish Restaurant & Pub (92%)
Paddy's Bar & Grill (3%)
Biddy McGraw's Irish Pub (2%)
County Cork Public House (3%)
Online Share of Voice for: restaurant in Portland, Oregon
Kells Irish Restaurant & Pub (55%)
Paddy's Bar & Grill (42%)
Biddy McGraw's Irish Pub (1%)
County Cork Public House (2%)
Online Share of Voice for: irish in Portland, Oregon
Kells Irish Restaurant & Pub (93%)
Paddy's Bar & Grill (3%)
Biddy McGraw's Irish Pub (2%)
County Cork Public House (2%)

Figure 2.10: An example of Reputation Report[15]

the content of the report according to the data needed (Bontcheva and Wilks, 2004; Schneider et al., 2013). In the following sections we review works that use templates and natural language generation techniques to generate automatic reports.

2.3.3.1 Generating reports using templates

The use of templates has become the most widespread solution for the automatic report generation. The automatic generation of reports have been extensively studied in the biomedical domain. In Liu et al. (2017), radiological reports are generated from images. The system discards automatically those images that are not relevant (those ones that present negative results for the clinical domain) using learned models. From the relevant images, some signals are extracted corresponding to the clinical findings and a natural language template is generated to create the final report. This report includes an explanation of the clinical findings by adding the information previously extracted. In (Hicks et al., 2018), the authors present a tool for generating medical reports, from templates, composed

[15] http://www.diversifiedsem.com/reputation-management/

of text and images taken in medical procedures. The analysis of medical images is done through a neural network that extracts the most relevant characteristics to add them to the final report. Other field of study, and the one that is tackled in this Thesis, is the creation of reputational reports. Unlike other domains, where report generation has been more studied, the creation of reputational reports is currently an emerging area, therefore, it has very little studies developed. Gonen (2012) introduces a method which collects information about accounts associated to different telephone numbers and creates a positive or negative reputation report calculated in real-time based on the financial information and non-financial information such as criminal history background, about the owner of the telephone number. One of the techniques used for generating reports through templates is *Question Answering*. For instance, in (Han et al., 2015) this technique was applied by the authors. The question-answering system receives a sequence of keyword and determines if these keywords are linked to entities and properties extracted from DBpedia[16]. Then, the system generates queries to extract all information that shows a relation to report. Finally, the system generates a report from the result using a Natural Language Generation (NLG) template database to return a report from the extracted information. In (Duboue, 2016) a demo is shown in which, given some input data, the system analyses the information and shows the results in a report created from a template, summarising relevant facts along with descriptions and graphical information.

The emergence of Social Networks in recent years has created an abundant data flow being interesting generate reports and summaries through the data extracted from them. In Zhang et al. (2017) information is used about events that have been crawled from websites or official news, being extended with useful information retrieved from different Social Networks (YouTube, Twitter, etc.), to generate captions of related images and extract latent topics to create reports about those events. Jeong et al. (2014) propose a system for generating analysis reports based on social big data mining. To do this, the results of the analysis of social data are dynamically selected and, according to the format of the report, a natural language summary is created from the analysis and these summaries will fill the gaps that appear in the template to generate the final report.

[16] DBpedia: https://wiki.dbpedia.org/

2.3.3.2 Generating reports using Natural Language generation

The use of *Natural Language Generation* (NLG) techniques provide more human-like reports. However, it is much more complicated to do than using templates since the state-of-the-art of NLG techniques are not mature enough to achieve competitive results. Nonetheless, some works deserve mention. For instance, Bontcheva and Wilks (2004) introduce a NLG approach for the automatic generation of reports from domain ontologies. The system prevents repeated information from appearing by creating a more fluid and readable report in the medical domain. Jordan et al. (2014) use an ontology for the HDFT domain (High Definition Fibre Tracking), which is used for generating reports. For this, the system makes use of an external judge who describes the symptoms found, the ontology analyzes these texts and identifies important information to be included in the final report and finally, the report is generated taking into account certain rules for ordering the content and including the data that have been extracted previously. Generating good and understandable medical reports is not only positive for regular doctors' appointments, but also for pre-hospital situations where paramedics need to have an overview of the scene as quickly as possible. Schneider et al. (2013) present a system composed of two main modules: *document planning*, the one in charge of selecting the most important events to include in the report, and *micro-planning*, where the structure of the document is defined and phrases are compiled through coordination and aggregation. The texts included are generated using the SimpleNLG tool (Gatt and Reiter, 2009) and the final report is showed as a XML document. But not only is important to generate reports automatically in the medical domain. Other domains like simulation systems need automatic reporting in order to help workers to make better decisions. In (Curry et al., 2013) the authors introduce an approach to generate generic reports that show the results obtained by simulation systems by including, automatically, relevant information to the decision maker. Texts, for these reports, are generated using two NLG tools: R (Team et al., 2013) and SimpleNLG. Other domain of study is the educational one where creating better reports could improve the skills of the students. Reiter et al. (2006) show a NLG system that produces short reports that give feedback to people who are taking online tests. Two NLG modules, *Microplanner* and *The Realiser*, are used to create such documentation. The first of them is in charge of expressing the content and structure, while the second of the modules generates the texts themselves based on decisions taken previously.

2.4 Conclusions

In this chapter we have reviewed two important areas for marketing online: detection of influencers in Social Networks and automatic summarization. Although these two topics have seen a large boost in the past years there is still a long way until the detection of influencers and creation of summaries tasks are solved plenty due to the need to adapt the state-of-the-art to the new scenario of social networks communication.

As a summary of the chapter, we can extract the following conclusions:

1. One of the main challenges of *Author Profiling* in Social Networks is having to deal with an unbalanced number of instances; usually the number of influencers is sensibly lower than the number of non-influencers. Another of the challenges raised is the approach taken: based on classification or through the use of rankings. For both, classification and ranking approaches, the use of signals is necessary but there are a large number of them. Some of the signals are common to all Social Networks (e.g. the number of followers) but others are specific to the network, therefore, specialized methods in one Social Network may not be valid for another because there are not a direct relationship between network signals.

2. The identification and characterization of users in Social Networks is based mainly on the social-demographic study of the users that make up these networks. One of the keys for finding influencers (or another type of user of interest) is to know the social interactions with other users, to check the extent of the diffusion of ideas among other users, to know the writing styles, etc. In the case of Twitter, one of the main challenges is the short length of the texts (280 characters maximum) and the different rules of writing that exist, two equal words can be written in very different ways, for example: “what” or “whaaaaaat”, so it is sometimes necessary to apply a regularization step before working with the texts.

3. There are two ways to *generate summaries*, by using complete sequences of the original texts (*extractive*), or by producing a new text as a human does (*abstractive*).

   - For generating *extractive* summaries (from a single document or from multiple documents) a ranking of sentences is created based on signals that indicate the priority or centrality of each sentence to the topic. The final summary is created by selecting the top-scored sentences in the ranking. The main inconvenience of this approach is the lack of cohesion between sentences that form part of the summary.

- *Abstractive* summarization is a less explored field because it requires an in-depth study of how different language structures are generated, in the case of generating natural language. There are other techniques to generate abstractive summaries, such as the use of ILP restrictions to improve the content and linguistic quality of the summary or the use of templates. The main inconveniences in this approach are: (i) the natural language generated is very poor; and (ii) template systems only work under restricted domains.

After the study of the state-of-the-art, we can identify some problems that remain open both in the identification of influencers and in the generation of extractive summaries:

- To create a characterization and identification of influencers independent of the Social Network used so that the model is as generic as possible.
- To explore how the domain language model may help in the identification of influencers.
- To know the role that followers of a profile play in characterizing and identifying her authority, beyond the use of the number of followers signal provided by Social Networks.
- To exploit the information obtained by characterizing and identifying influencers in the automatic generation of reputational reports.

To address the open lines of work previously identified, we have elaborated different strategies that will be developed in the next chapters of this Thesis:

- To make our algorithm Social Network independent. Although our dataset contains data on Twitter users only, we want to explore signals that have a direct correspondence in other Social Networks and, therefore, that have a minimum adaptation cost.

- To exploit the discourse of influencers by creating language models, that allow new profiles to be characterized in the absence of other information apart from that in their posts.

- To consider the importance of domain signals in the characterization of influencers. The hypothesis is that a profile may have no global authority but may be an influencer in a narrow, specialized domain.

- To investigate the role of the followers in the characterization of a profile according to its authority: our hypothesis is that if among the followers of a user are several influencers, it is possible that the message written by the main profile, is retweeted by several influencers and therefore, to spread quicker.

- To employ the information learned from the detection of influencers to the automatic generation of reputational reports. Our hypothesis is that those texts/tweets that are written by people of great authority are serious candidates to appear in the final summary, since the comments of these kind of users can lead, with higher probability, to the entity to suffer a reputational crisis.

CHAPTER 3

DETECTION OF INFLUENCERS ON TWITTER

Haec neque affirmare, neque refelle operae pretium est: famae rerum standum est

Titus Livius-V

In this chapter, we focus on the detection of influencers in Social Networks, specifically on Twitter. To this end, we present an exhaustive study of signals extracted in two different ways (i) using the metadata provided by Twitter (e.g. number of followers, number of published tweets, etc.); (ii) using the texts published by the users which allow us to check if the vocabulary in the discourse shows some degree of expertise about the domain and/or this vocabulary is similar to that employed by other profiles that we know that are authorities or influencers. We also introduce two different ways to tackle the detection of influencers: the first one uses the information originating from those profiles of which we want to discover their identity; while in the second approach, we perform a bottom-up search in which we use the information regarding the followers to characterize the followed profiles. Furthermore, we combine the information related to the profiles and their followers to better distinguish between influencer and non-influencer profiles.

This chapter is divided into the following sections: first, in section 3.1 we provide the motivation of the problem. Second, in section 3.2 we describe the identification and characterization of influencers using information from the main profiles. We describe how to calculate the signals and the methods used (section 3.2.1), the experimental framework (section 3.2.2), including the dataset used, the experiments performed, the metrics for evaluation and the baselines. We show the results and discuss them in section 3.2.3. Finally, in section 3.2.4 we analyse the errors found. In section 3.3 the identification and characterization of influencers using their followers is explained. In section 3.3.1, we describe the methods used in order to calculate signals and the algorithms used for ranking. In section 3.3.2 we describe the dataset used, the evaluation metrics and the baselines. Then, in section 3.3.3 we show the results and discuss them. Finally, in section 3.3.4 we analyse the errors found in our method. To conclude this chapter, in section 3.5 we draw our conclusions.

3.1 Motivation of the task

In traditional marketing it is imperative to know the types of users who share information about an entity. Opinions of anonymous people do not have the same impact as opinions of special users, well-known people within communities, and who have the power to change the opinions of other users. These kind of users are known as *influencers* or *opinion-makers*.

Before the advent of Social Media, people with the capacity of influencing the public opinion in a given domain were few and easy to identify: journalists from mass media, authorities with academic degrees and proved expertise, politicians, media owners, celebrities, etc. In practice, editorial boards and lobbies could effectively decide what information and what opinions reached the masses, and how. Public Relations (PR) for organizations and individuals were, then, a matter of addressing a few opinion makers to shape their reputation, i.e., how their image was projected to the public opinion. Social Media has significantly complicated matters for organizations from the point of view of Public Relations. Monitoring and managing social media brings unprecedented opportunities to know and interact with clients and stakeholders, but it renders previous PR methodologies obsolete. One of the key aspects of Online Media, and of Social Media in particular, is that any citizen is a candidate to become influential: it is no longer possible to narrow the filter to media owners, journalists, academic experts and other standard profiles. In this context, one of the key aspects of Online Reputation Monitoring (ORM) is to detect which social media profiles have the capacity of influencing the public opinion and, therefore, creating opinion and shaping the reputation of organizations, companies, brands and individuals (Madden and Smith, 2010). As we said, influencers have a great impact in ORM since they may cause a product or company to increase or suffer a serious loss of reputation (Burn-Callander, 2015), resulting in an impact on the benefits they may have.

Just as there are profiles of influencers, there are also other kind of users in Social Networks that support them and serve as a loudspeaker for the propagation of the ideas of influencers, they are called *followers*. As in the case of influencers, before the arrival of Social Media, supporters were restricted to be a number: the number of people who vote for a political party, the number of subscribers to a newspaper or the number of people who go to a football stadium to cheer on their team, for example. With the emergence of the Internet in general and Social Media in particular, followers have become more than just a number. As we have commented before, followers are in charge of spreading the ideas of an influencer, either by retweeting a post or by generating new texts from influencers' ideas. But what if the follower of one influencer is also another influencer? In this case, the message that the influencer-follower spreads will do nothing but corroborate the ideas of the first influencer and, in turn, give legitimacy to that message among the followers of the second influencer. That is, the message propagated will have more impact since it has been verified by another relevant profile, becoming a major reputational threat for the entity.

For these reasons, in this chapter we propose two different methods to identify and characterize influencers:

1. **Using the main profiles:** we directly classify each user profile as influencer or not, using information extracted from the profile itself (number of followers, number of published twees, language models, etc.). Figure 3.1 shows this idea.

2. **Through its followers:** we classify a profile as influencer or not taking into account the network of people with whom it is connected. The idea is that if a profile is followed by many influencers, it will probably be an influencer. Figure 3.2 illustrates this idea.

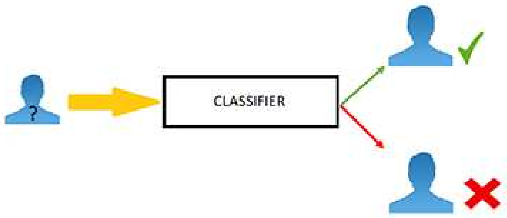


Figure 3.1: Finding influencers using information from the user's profile

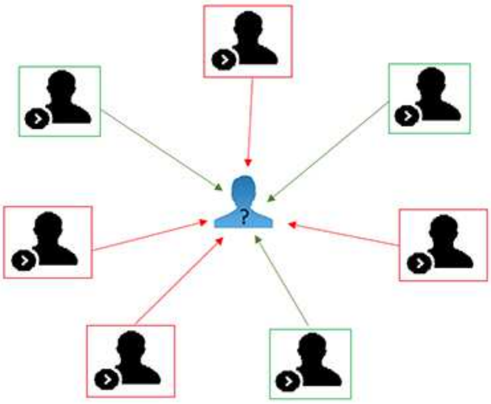


Figure 3.2: Finding influencers using information about their followers

As we said before, in this chapter we want to identify and characterize influencers. We distinguish here two different types of influencers: (i) people whose authority is restricted to a certain domain because they possess knowledge about that domain (i.e. brokers in economy, mechanics in automotive, etc.) or (ii) people whose authority transcends to other domains, for example in the case of celebrities, sportsmen, etc. (authorities). To help us to face this task, we state the following research questions:

> **Research Question 1:** *What is the relative importance of authority signals versus domain expertise signals?* In order to be influential in a given domain, two major types of signals are involved: signals of authority (for instance, *does the user have many followers? Are her statements frequently retweeted? Are her posts similar to other influencers' texts?*) and signals of relevance to the domain (an influential voice in macro economics may have no authority at all when talking about music, for instance). We want to find out what is the relative role that each of these two types of signals play when detecting influencers and how to characterize them. We explore

signals extracted from the Twitter network (i.e. number of followers) and from the content of the tweets.

Research Question 2: *How best to combine signals?* We will explore several ways for combining signals to discover influencers: supervised classification, unsupervised signal voting, supervised learning to rank, and supervised classification combined with signal voting. We will also combine all these approaches with language models that learn both the domain language and the language of influencers, using them as the primary textual signals.

Research Question 3: *How well followers characterize a Twitter profile? Can we establish authority using only information from followers? What is the best way of using followers' information to establish authority?* We want to know if the language used by followers is an important factor for establishing whether or not a user is an *influencer*. Moreover, *Can information from followers enrich profile characterizations?* In other words, *is it complementary or redundant with the information from the profile itself?* Here we want to aggregate the information related to the profile with the information of her followers and verify if this aggregation improves the profile characterization.

In the following sections we introduce the signals and the algorithms used, as well as the experimentation carried out to answer these research questions.

3.2 Using information from the main profiles to discover influencers

In this section, we focus on finding influencers using information from their posts and from the structure of their networks, without taking into account their followers. The structure of the section is the following: first, we explain the different signals extracted and how we use them in order to identify influencers. Then, we provide details about the experimental framework, which includes the dataset used, the design of the experiments, the metrics used and the baselines for comparing our results. Finally, we show and discuss the results obtained and the errors performed by our algorithm.

3.2.1 Methods

In this section we present the signals and algorithms used for the automatic detection of influencers in Twitter.

3.2.1.1 Signals

One of our goals is to extract some signals capable to identify and characterize users in Social Networks. In this section, we introduce signals used to perform our experiments; they are classified as: *Twitter signals* (signals extracted from the metadata provided by Twitter) and *textual signals* (signals extracted from the text in the posts).

Twitter signals
We have used the following signals that are extracted from Twitter data from the user's profile:

- **Tweets**: is the number of tweets published by the user.
- **RTs**: is the number of *retweets* received by the user's posts.
- **FAVs**: is the number of *favorites* received by the user's posts.
- **Foll**: is the number of *followers* of a profile.
- **Follees**: is the number of *followees* (people followed but the author).
- **DivFoll**: is the ratio of followers to followees of the profile. This should be a better indication of authority that simply the number of followers, because when the number of followees is very large, a significant fraction of followers may come out of polite reciprocity, rather than true interest in the user's posts. In other words, a user with 10,000 followers and only 100 followees should be more influential than another user with 10,000 followers and 10,000 followees.
- **DivFollees**: is the inverse of *DivFoll*.
- **DivRTFoll**: is the average number of retweets per follower.

- **DivFAVFoll**: is the average number of favorites per follower. Both *DivRTFoll* and *DivFAVFoll* represent the intensity of the interactions of the followers with the content published by the user.

- **DivRTFAVFoll**: is the average number of total interactions (retweets + favorites) per follower.

- **DivRTFollees**: replicates *DivRTFoll* but with respect to followees instead of followers.

- **DivFAVFollees**: replicates *DivFAVFoll* but with respect to followees instead of followers.

- **DivRTFAVFollees**: replicates *DivRTFAVFoll* but with respect to followees instead of followers.

- **RVR**: is the viral rate for retweets. Viral rates are well-known signals in the marketing field (Ramón and López, 2016) that measure how well the message spreads through the audience. The retweet viral rate is the number of tweets with at least one retweet divided by the total number of retweets; it is higher if all tweets receive similar attention (instead of one tweet accumulating most retweets and the other being ignored, for instance).

- **FVR**: is the viral rate for favorites. The favorite viral rate is the number of tweets with at least one favorite divided by the total number of favorites.

- **TVR**: is the viral rate for the total interactions (retweets+favorites).

- **Borda**: is a combination of the previous signals applying the Borda voting algorithm (Saari, 1999).

These signals are summarized in Figure 3.3.

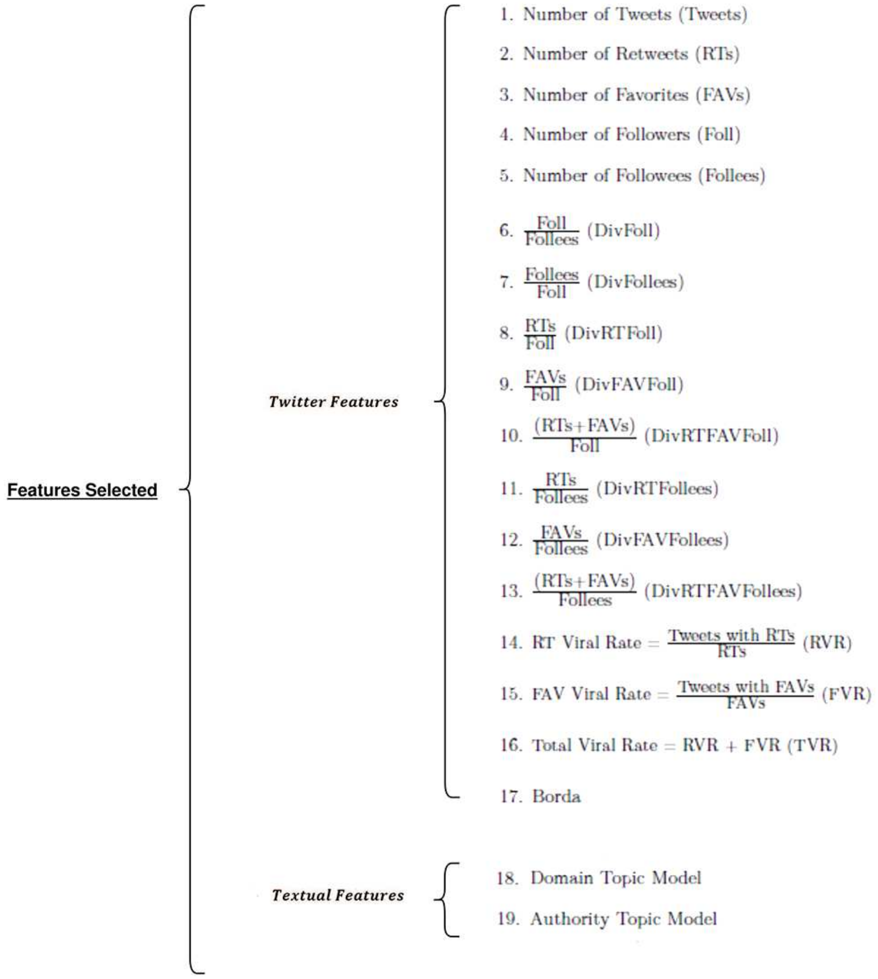

Features Selected
Twitter Features
1. Number of Tweets (Tweets)
2. Number of Retweets (RTs)
3. Number of Favorites (FAVs)
4. Number of Followers (Foll)
5. Number of Followees (Follees)
6. $\frac{\text{Foll}}{\text{Follees}}$ (DivFoll)
7. $\frac{\text{Follees}}{\text{Foll}}$ (DivFollees)
8. $\frac{\text{RTs}}{\text{Foll}}$ (DivRTFoll)
9. $\frac{\text{FAVs}}{\text{Foll}}$ (DivFAVFoll)
10. $\frac{(\text{RTs}+\text{FAVs})}{\text{Foll}}$ (DivRTFAVFoll)
11. $\frac{\text{RTs}}{\text{Follees}}$ (DivRTFollees)
12. $\frac{\text{FAVs}}{\text{Follees}}$ (DivFAVFollees)
13. $\frac{(\text{RTs}+\text{FAVs})}{\text{Follees}}$ (DivRTFAVFollees)
14. RT Viral Rate = $\frac{\text{Tweets with RTs}}{\text{RTs}}$ (RVR)
15. FAV Viral Rate = $\frac{\text{Tweets with FAVs}}{\text{FAVs}}$ (FVR)
16. Total Viral Rate = RVR + FVR (TVR)
17. Borda
Textual Features
18. Domain Topic Model
19. Authority Topic Model

Figure 3.3: Signals Selected

Textual signals

The textual content of a user's posts is a powerful signal for detecting influencers. Obviously, it gives useful domain information (active users in the banking domain, for instance, will use the distinctive vocabulary of the domain). But it may also provide evidence for authority, under the hypothesis that authorities have distinctive commonalities in the way they express their opinions or transmit information. For instance, an authority in the automotive domain may use technical words such as crankshaft, valves, etc. more often than regular users. Therefore, we have experimented with textual signals to characterize, both domain and authority traits.

In our work, we represent textual information using the Topic Modeling approach described in Sánchez (2016). When modeling authority and domain, we use training tweets to build a language model of all profiles from a training set manually labeled with domain and authority information. Then, for each profile in the test set we estimate how compatible is her language with the language model learned from the training set, and use one single signal to store such compatibility (see section 3.2.2.1 for a description of the dataset).

Here we summarize the language modeling technique. It learns a model of the language underlying a domain of authors, D (which could be influencers or not or have some degree of expertise of the domain), in the context of a reference collection of documents, $C = d_1, .., d_{|C|}$, in our case C are the training texts written by the influencers and the regular users, with vocabulary $V = w_1, .., w_{|V|}$.

The aim of the model is to obtain a probability distribution of words, $p'(w)$, in which words likely to be included in an author message in the domain of authors D are assigned high probability values; whereas other words, including those that are very ambiguous or not domain-specific but occur in D, receive marginalized values.

Our methodology learns $p'(w)$ as a refinement of the posterior distribution of words $p(w|D, L)$, which we define proportional to $p(L|w) * p(D|w) * p(w)$, L and D being the background (the vocabulary used in all other domains except the one we are in) and the target domain, respectively. The aim of the refinement is twofold: (i) to boost the likelihood of words that accurately describe the most important signals to identify the authority context of the reference collection C and (ii) to decrease the likelihood of very common or ambiguous signals than can be close to random contents in C. Notice that some very frequent words occurring in the domain D can be actually relevant to distinguish the domain D from other domains, whereas some others don't because they can be found co-occurring with other words likely to model other domains. Thus, by representing the context of the reference collection C with a probability distribution of words $p(w)_{w \in C}$, our method learns the distribution of words $p'(w)$ as the one that minimizes the cross entropy value, as expressed in Eq.3.1:

$$H_s = -\sum_{w \in V} p(w|L, D) log((1 - \lambda)p'(w) + \lambda p(w)) \tag{3.1}$$

where the argument of the logarithm is a mixture in which λ is the weight that accounts for the proportion of "context noise" in $p(w|L, D)_{w \in V}$, and $p(w)$ is the probability of word w under the reference model (i.e., the prior of w in

C). The lower the value of λ ($0 \leq \lambda \leq 1$), the more refined the model $p(w|L, D)$ (i.e., the more content words related to both L and D are weighted higher, while contentless or off-topic words will be weighted lower). In our experiments, we have used $\lambda = 0.2$, which properly marginalizes the scoring of general domain (frequent) contentless words (such as prepositions and broad/ambiguous verbs) in the model.

This way of optimizing the language model underlying a lexical signature resembles the one employed in Zhou et al. (2007) to learn the so called *Topic Signature Language Models* (TSLM). However, instead of relying on a source word distribution (i.e., $p(w|L, D)$) and then considering cross-entropy to learn the target model, TSLM learns a model from a topic signature by relying on a set of documents $C_k(C_k \subseteq C)$ deemed to be relevant for the contents behind the set of words under modelling, which is unfeasible in our case.

Therefore, the main idea behind is that we want to know the probability distribution of the words used by each author preferring those words that minimize the entropy value, in other words, that maximize the quantity of information.

From the above optimization equation, we learn the distribution $p'(w)$ using an Expectation Maximization procedure that starts from initial (uniform) values for $p'(w)_{w \in V}$, and iteratively approximates the values in $p'(w)_{w \in V}$ until convergence by means of the following updates in the r-th iteration:

$p'^{(r)}(w)$ is the Maximization-Step and is defined as:

$$p'^{(r)}(w) = \frac{p(w|L, D) * Z(w)}{(\sum_{w' \in V} p(w'|L, D) Z(w'))} \tag{3.2}$$

where $Z(w)$ is the Expectation-Step and is defined as:

$$Z(w) = \frac{(1 - \lambda) p'^{(r-1)}(w)}{((1 - \lambda) p'^{(r-1)}(w) + \lambda p(w))} \tag{3.3}$$

In our work, we define both $p(w|L)$ and $p(w|D)$ as follows:

$$p(w|L) = \frac{tf(w, L)}{\sum_{w' \in L}(tf(w'))} \tag{3.4}$$

$$p(w|D) = \frac{tf(w, D)}{\sum_{w' \in D}(tf(w'))} \tag{3.5}$$

The probability of an author a belonging to the language model D is finally computed as:

$$p(D|a) = \sum_{w}(p(D|w) * p(w|a)) \tag{3.6}$$

where

$$p(D|w) = Z(w)$$

$$p(w|a) \propto tf(w, Y)$$

being Y the set of tweets of the author a.

Using this method, we calculate two signals: the first one reflects the discourse similarity between a user to be characterized and the influencers (authority topic model signal); the second one reflects the degree of expertise about the domain that a given user have (domain topic model signal).

3.2.1.2 Algorithms for ranking influencers

The detection and characterization of influencers is not tackled here as a traditional classification problem. Instead we address the problem as a ranking task because is the most natural way of presenting results to the reputation experts; influencers are expected to be in the first positions of the ranking while regular users occupy the lowest places in the ranking. We have compared four algorithms that aim to generate a ranking of users according to their probability to be influencers. The first of them is very simple and is based on ordering the users' profiles by one or more chosen signals. In the second approach, we uses the classifier confidence for ordering the users' profiles. The third one is more sophisticated and uses Learning to Rank. The last one combines the first and second approaches. The different approaches are explained below:

- **Direct Signal Rank Strategy (DSR):** Each extracted signal (see Section 3.2.1.1) generates a ranking of users. For instance, we can rank users by the number of followers they have: the number of followers is the (primary) audience of the user, and therefore is one basic indicator of the capacity of a user to influence the state of opinion. This is the simplest possible mechanism to produce a profile ranking out of authority signals. When we use two or more signals to produce a single rank, we apply Borda voting (Saari, 1999) to combine the ranks produced by each individual signal. If we have n elements to rank, the Borda voting lies in an ordination of the elements to consider for each signal individually in descending order, assigning the higher value, in our case n, to the first element of the ranking, the $n-1$ value to the second element and so on. The combined ranking is produced by adding the values assigned to each element by every rank, and using this number to produce the final ranking.

Note that this is an unsupervised approach unless one or more signals are obtained using the training data. In our experiments, the only supervised signals are the language models, which compare the language of each user with the language models of authorities (authority model) or with the language models of tweets belonging to the domain (domain model).

- **Classifier Rank Strategy (CR):** The problem that we want to solve is usually addressed as a binary classification problem (influencer or not). In fact we can use Machine Learning algorithms for binary classification, and use their confidence scores to rank items. We have experimented with several algorithms: SVM (Cortes and Vapnik, 1995), Bayes Net and Naïve Bayes (Neapolitan et al., 2004; Russell and Norvig, 2016), Decision Trees (C4.5, also known as J48 (Quinlan, 1993)) and AdaBoost (Freund et al., 1999).

  Using classifiers is the most widely employed technique for our problem (Ramírez-de-la Rosa et al., 2014; Vilares et al., 2014). For each instance, the classifier makes a binary choice (influencer or not) and provides a level of confidence for its choice. We take the confidence as the ranking signal, and we generate a ranking by ordering the instances in decreasing confidence score. This experimentation has been carried out using the Weka tool (Hall et al., 2009) and the parameters used in each classifier were the default values provided by Weka.

- **Learning to Rank Strategy (L2R):** Instead of learning to classify in order to rank, we can directly learn to rank using Learning to Rank algorithms from the Information Retrieval field (Liu, 2009). These ranking models seek to optimize a chosen evaluation measure on the training data (in our case *Mean Average Precision* (MAP) (Manning et al., 2008)). We have focused on pairwise approaches as our classification problem is binary, and we have experimented with three algorithms:

  - **MART:** *Multiple Additive Regression Trees. MART* uses gradient boosted decision trees for prediction tasks (Friedman, 2001).
  - **LambdaMart:** this algorithm is the combination of two methods: *MART* and *LambdaRank* (Wu et al., 2008). As opposed to *MART*, *LambdaMART* uses gradient boosted decision trees with a cost function derived from *LambdaRank* for solving a ranking task.
  - **RankBoost:** the algorithm combines the benefits of boosted tree classification and LambdaRank making it faster in both train and test (Freund et al., 2003). Training method wants to classify a set of data against each other by their associating a classification rank. The selection of positive and negative examples want to maximize the obtained positive score to the negative score.

  This experimentation was carried out using the RankLib tool (Dang, 2012).

- **Direct Signal with Classification Filter Rank Strategy (DSCFR):** This strategy was first proposed in (Lomeña, 2014), and combines classification with signal rank. The output of the classifier is used to divide the profiles in two groups. Profiles classified as influencers all go first in the ranking, and non-influencers after them. Inside each group, profiles are ranked according to the values of ranking signals (instead of using confidence scores from the classifiers), combined via Borda voting as in our first ranking strategy. The idea is that the classification step provides useful information to rank profiles, but the confidence measure might not be as useful as the information that authority signals provide directly.

  The steps are as follows:

  - **Classifier step:** we have used the same classifiers described above, using authority signals (see section 3.2.1.1) and we use them to divide items in two groups (influencer vs regular users).
  - **Re-grouping and re-ranking by signal step:** Each group is ordered, individually, according to the signals used by the classifier in the previous step. In the first positions of the final rank will appear the ordered influencers' group and right behind them, the ordered regular users' group.

3.2.2 Experimental framework

The primary focus of our experiments is to determine how our method for detecting influencers, which relies on signals extracted from the users' Twitter profile and the posts published by the users, compares to other methods that work under the same conditions. To do so, we perform experiments on the RepLab 2014 dataset (Amigó et al., 2014), which is the largest expert annotated dataset for the Author Ranking task. As we mentioned in the previous section, we select some signals to determine whether a user is an *influencer* or not. Not all signals have the same strength when it comes to detect this type of users, so, we will filter those that are not useful for us and then, we will generate the different types of rankings (see section 3.2.1.2).

3.2.2.1 Dataset

We have followed the guidelines of the RepLab 2014 competition (Amigó et al., 2014) and used the author ranking dataset for all our experiments. The RepLab 2014 author profiling task, systems are expected to "find out which authors have more reputational influence" for a given domain (automotive and banking). The systems' output is a ranking of Twitter profiles according to their probability of being influencers with respect to the domain.

The RepLab 2014 dataset (Amigó et al., 2014) consists of 7,622 Twitter profiles (all with at least 1,000 followers) related to the automotive and banking domains. The profiles are divided in: 2502 training profiles, 4862 test profiles (for automotive and banking) and 132 additional test profiles which are domain-independent. Each profile consists of (i) author name; (ii) profile URL and (iii) the last 600 tweets published by the author at crawling time. Reputation experts manually assessed each profile as *influencer* or *non-influencer*. The dataset contains tweets in two languages: English and Spanish.

Figure 3.4 shows the distribution of users (influencers and non-influencers) presented in the dataset.

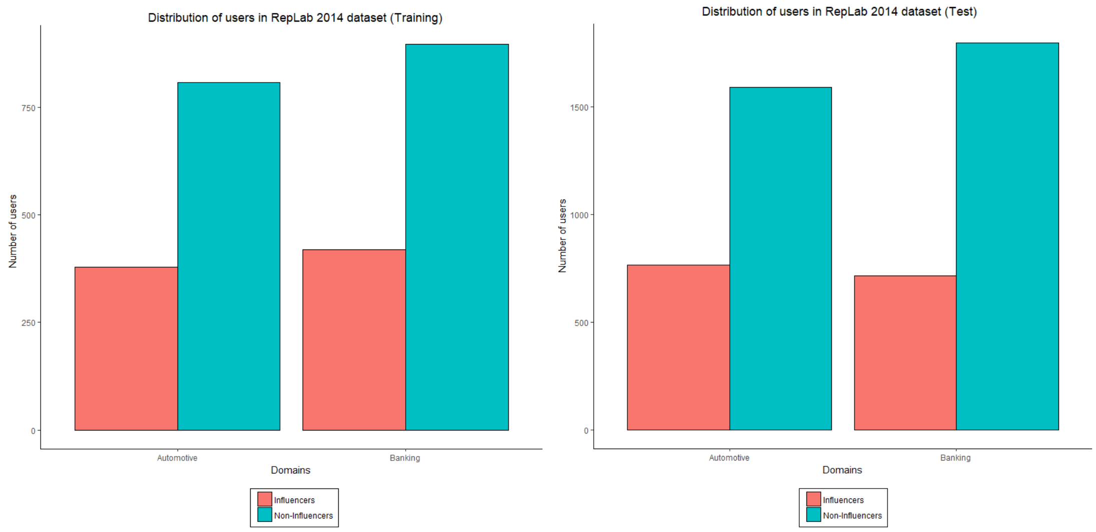


Figure 3.4: Distribution of users in RepLab 2014 dataset

3.2.2.2 Experiments

In this section we describe the signal selection process (using as input the signals listed in Figure 3.3), and the algorithmic approach for ranking generation.

Signal selection

Due to the large amount of signals (see Figure 3.3) it is advisable to apply a previous narrow down step in order to select and study the signals that better fit to our task. For this purpose, we apply feature engineering, the process of using knowledge of the data in order to select signals that make our machine learning algorithms work, as described below:

- First, we perform a division of the RepLab dataset according to its different languages (English and Spanish) and domains (Automotive and Banking). We have four different ways to divide it: **separated by domain and language** (i.e., Automotive and English, Automotive and Spanish, Banking and English and Banking and Spanish), **separated by domai**n (i.e., Automotive and Banking), **separated by language** (i.e., English and Spanish) and **not separated**, without any distinction between domain or language.

- Once we have the input signals of our algorithm and the different division strategies, we perform a *classifier step* using the training data. The different signals are divided, according to the division strategies, and they are used as input to the classifiers presented in section 3.2.1.2, and the best classifier is chosen (in our case, the Naive Bayes classifier).

- With the predictions, we generate intermediate rankings for each different signal, and we evaluate each of them using *Mean Average Precision* (MAP), which is the official measure used in the RepLab evaluation campaign.
- With the results for each signal and ranking strategy, we compute the average between the best and the worst signal result, discarding all signals below the average, and retaining the signals which perform above the average for all ranking strategies.

For example, let's suppose that we want to select the best signals from the set {Foll, RTs and FAVs}. After training and selecting the best classifier for each of these signals, the results given for {Automotive, English} are $Foll = 0.3, RTs = 0.1, FAVs = 0.2$. The average between Foll (best) and RTs (worst) is 0.2 so we select in this case is {Foll,FAVs}. Suppose that for {Automotive, Spanish} the results are $Foll = 0.3, RTs = 0.2, FAVs = 0.1$. The average between Foll (best) and FAVs (worst) is 0.2 and for this case we will select {Foll,RTs} as an output. When comparing the divisions {Automotive, English} and {Automotive, Spanish} we see that they have a common signal (Foll) which is finally selected.

Note that bag-of-words signals are not sorting signals since they do not provide a clear way to rank. This means that methods that use signals to rank directly such as *DSCFR* and *DSR* cannot use them.

The signals selected after the feature engineering process are shown in figure 3.5.

Rank generator

Once we have selected the best signals, we generate and evaluate the final ranks produced by each approach. First of all, we have combined the filtered signals in all possible ways in order to test if there exists a combination which gives the best result. Then we have split the dataset, in the way explained in the previous section.

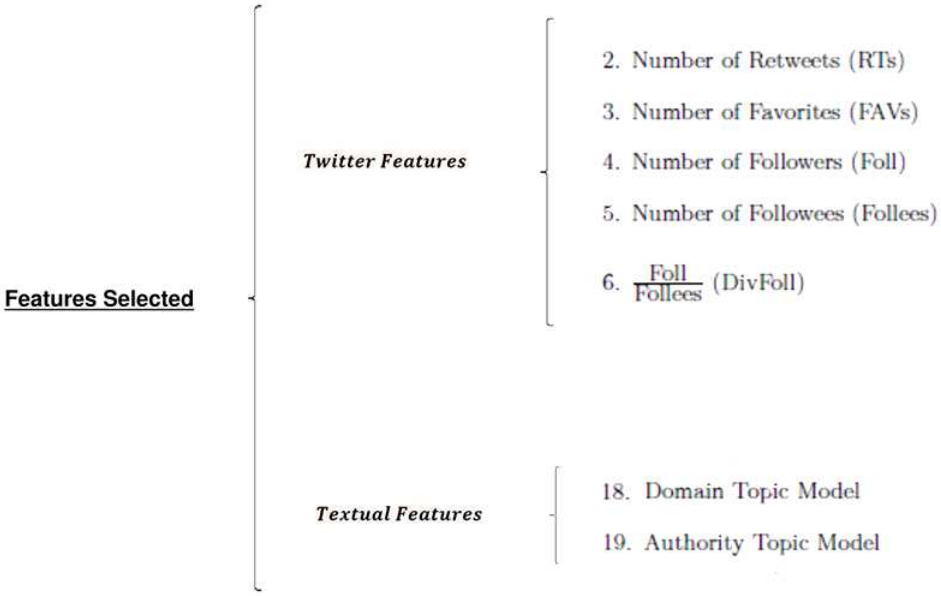

Features Selected
Twitter Features
2. Number of Retweets (RTs)
3. Number of Favorites (FAVs)
4. Number of Followers (Foll)
5. Number of Followees (Follees)
6. $\frac{\text{Foll}}{\text{Follees}}$ (DivFoll)
Textual Features
18. Domain Topic Model
19. Authority Topic Model

Figure 3.5: Signals selected after the feature engineering process

The different signal combinations are the input to the classifiers of section 3.2.1.2. Finally, we have generated the final rankings and we have evaluated them using MAP. Note that some re-ranking strategies use signals for this purpose and if we have a combination of two or more sortable signals, we have to apply a previous Borda voting step in order to obtain a final re-ranking signal.

3.2.2.3 Metrics

Mean Average Precision (MAP) (Manning et al., 2008) is the official metric for RepLab 2014 competition so that, in order to compare us with the state-of-the-art systems, we have also used it. This metric measures the average precision obtained for the top k documents after each relevant document is retrieved, then this value is averaged over the information needs. This metric is mathematically expressed in Eq.3.7.

$$\text{MAP}(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j}\text{Precision}(R_{jk}) \tag{3.7}$$

where: m_j is the number of relevant documents at position j in the ranking and R_{jk} are the retrieved documents from the top of the ranking until the document k is reached. According to (Amigó et al., 2014) two main reasons exist to adopt MAP for this task: it is well-known in information retrieval, and it is recall-oriented, and therefore lower ranking author's relevance is considered, which is well suited for the task: the goal is not to find just a few influencers, but as many as possible.

3.2.2.4 Baselines

As reference, we have considered two naive baselines and three state-of-the-art results:

1. **Followers:** The ranking of authors according to their descending number of followers is the baseline of the RepLab 2014 competition. The number of followers is a basic indication of the author's authority potential.

2. **Bag of Words:** a naive approach that uses the bag-of-words representation, where a text is represented as the bag (multiset) of its words, disregarding grammar and even word order. Bags of words are obtained from the tweets associated to the profiles in the RepLab 2014 datasets (600 tweets per profile). As a result, the size of the BoW vocabulary is over 850,000 different words in the training set.

3. **Best RepLab 2014** result (AleAhmad et al., 2014). The main idea in this study is that influencers or opinion makers talk more about hot topics. This method extracts hot topics from each domain and a time-sensitive voting algorithm is used to rank each author on their respective topic.

4. **Best result published using feature engineering** to date that uses the RepLab 2014 dataset. Cossu et al. (2016) review some state-of-the-art signals, develops new textual signals and propose different ways to group them. Their system combines signals corresponding to the following categories: user activity, profile fields, stylistic aspects and external data and the score obtained with this combination is used to rank the users. From now on we will call Cossu'16 to this baseline.

5. **Best result published** to date that uses the RepLab 2014 dataset (Nebot et al., 2018). The authors of this work, evaluate two different ways to classify profiles without using a previous feature engineering step: deep learning classifiers and traditional classifiers with embeddings. Their best result is using Bayes Net & Naive Bayes (for automotive and banking domains

respectively) along with *Low Dimensionality Statistical Embedding* (LDSE), which represents documents as a probability distribution of their words in the different classes (influencer vs non-influencer in our case). From now on we will call Nebot'18 to this baseline.

3.2.3 Results and discussion

Table 3.1 summarizes the results of all experiments using a simple approach to handle textual content (textual signals are bag-of-words extracted from the author posts). Table 3.2 summarizes the same set of experiments, but replacing bag-of-words signals for a single signal that estimates the probability of the textual content belonging to the language model of the authorities, and a similar signal that estimates the probability of the textual content belonging to the language model of both domains (automotive and banking). Both tables average the results obtained in the two RepLab domains: banking and automotive. Note that although results slightly vary across domains, the relative difference between strategies is relatively stable, with no major discrepancies. Apart from the MAP metric, we have computed precision at 10, precision at 50 and precision at 100. Results for these metrics are found in appendix B. However, since the results for all metrics are consistent, we focus on discussing MAP results.

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	**0.38**	0.26	**0.54**	0.27
BoW with RTs	0.33	0.42	0.52	0.33
BoW with FAVs	0.32	0.42	0.51	0.33
BoW with DivFoll	0.34	**0.44**	0.44	0.32
BoW with FAVs & Foll	0.36	0.42	0.48	0.45
BoW with Foll & DivFoll	**0.38**	0.38	0.43	0.54
BoW with Follees, Foll & DivFoll	**0.38**	0.38	0.48	0.58
BoW with RTs, FAVs, Foll, Follees & DivFoll	0.36	0.38	0.44	**0.59**
Baseline - Nebot'18	0.842			
Baseline - Cossu'16	0.714			
Baseline - Best RepLab 2014	0.57			
Baseline - Followers	0.38	0.34	0.38	0.30

Table 3.1: Overall MAP results using BoW and Twitter-based signals

	DSR	**CR**	**DSCFR**	**L2R**
RTs, FAVs, Foll, Follees and DivFoll (Twitter Auth.)	0.38	0.34	0.38	0.32
Domain Vocabulary	0.43	0.45	0.56	0.60
Twitter Auth. + Domain Voc.	0.53	**0.54**	0.57	0.63
Authority Vocabulary	**0.73**	0.40	**0.73**	0.70
Twitter Auth. + Authority Voc.	0.65	0.46	0.71	0.72
Domain Voc. + Authority Voc.	0.68	0.47	0.72	**0.74**
Baseline - Nebot'18	0.842			
Baseline - Cossu'16	0.714			
Baseline - Best RepLab 2014	0.57			
Baseline - Best BoW	0.36	0.38	0.44	0.59
Baseline - Followers	0.38	0.34	0.38	0.30

Table 3.2: Overall MAP Results using our authority and domain textual signals

First of all, according to the results in Table 3.1, we conclude the following: (i) the best results obtained for every ranking approach equals or improves the baseline results and L2R approach also improves the best RepLab 2014 system; (ii) the results obtained using BoW and twitter-based signals are far from the rest of the baselines.

Secondly, a direct comparison of Tables 3.1 and 3.2 reveals that our proposed topic model provides a substantial boost in performance with respect to the bag-of-words approach: learning topic models for the textual content behaves better than applying any other learning algorithm using words as signals, which means that there is a language model that characterizes influential people. Therefore, we focus on the results using language models (Table 3.2) for the rest of the discussion.

According to the results in Table 3.2, we can conclude the following:

1. The vocabulary in the tweets is enough to learn domain and authority signals. Our best result is obtained with *L2R* over domain and authority signals learned from the training data (0.74), which also outperforms the best published result on the dataset using a feature engineering approach (0.714 (Cossu et al., 2016)), but it can not overcome the best system published (0.842 (Nebot et al., 2018)) that use an embedding representation of the terms. The role of the domain signal, however, is only marginally relevant: the authority signal alone provides 0.73 (−1.3%) without using any

ML algorithm. A possible explanation is that modeling the vocabulary of authorities in a given domain also, implicitly, includes domain information (we are comparing authorities in the domain with all other profiles in the dataset, which includes people in and out of the domain).

2. Our exhaustive test of alternative ways of using signals indicates that no algorithm (classification, learning to rank or combined approach) is able to improve significantly the raw use of signals to rank candidates. Surprisingly, ranking candidates according to how well their vocabulary fits into the models built with the training data provides results almost as good as any ML algorithm, even *L2R* which seems particularly well suited for the task.

   With suboptimal signals, however, L2R and the combined strategy are sometimes able to boost performance (e.g. *L2R* improves 0.43 → 0.60 using a domain vocabulary signal; and the combined method improves 0.43 → 0.56 with the same signal).

   Classification performs poorer than L2R, which is not surprising given that the latter's objective optimization function is oriented to rank elements while classification's optimization functions are adjusted to classify.

3. When comparing unsupervised versus supervised approaches, we extract the main following conclusions:

   (a) The best unsupervised method (0.53) is a Borda combination of the rankings provided by three unsupervised signals: number of followers, followers/followees, and the domain signal. Note that this result is only 7% worse than the best system in the competition (0.57). But using the same signals, the best supervised method (*L2R*) provides a 19% relative improvement (0.63).

   (b) The best supervised method is L2R over the vocabulary signals modeling domain and authority (0.74). Note that the authority signal is itself supervised, as it models the vocabulary of Twitter profiles annotated as authorities in the training dataset. This represents a 40% improvement over the best unsupervised method.

4. The study of alternative evaluation measures (P@10) (see Tables 3.3 and 3.4) corroborates the results given by MAP. Given a ranking, precision at k

($P@k$) is the proportion of the top-k documents that are relevant (Craswell, 2009). It is defined in Eq.3.8 as:

$$\text{P@k} = \frac{r}{k} \tag{3.8}$$

where: r is the number of relevant documents that have been retrieved at rank k. As already told, in appendix B there are detailed results for alternative evaluation metrics (P@10, P@50, P@100) and for each of the RepLab domains (automotive and banking) separately.

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	**0.60**	0.40	0.35	0.25
BoW with RTs	0.10	**0.50**	0.45	0.28
BoW with FAVs	0.25	**0.50**	0.35	0.15
BoW with DivFoll	0.45	0.35	0.25	0.25
BoW with FAVs & Foll	0.35	**0.50**	0.45	0
BoW with Foll & DivFoll	0.40	0.45	**0.60**	0.10
BoW with Follees, Foll & DivFoll	0.20	0.20	0.50	0
BoW with RTs, FAVs, Foll, Follees & DivFoll	0.30	0.45	0.40	**1**
Best Combination	**0.60**	**0.50**	**0.60**	**1**

Table 3.3: P@10 average results between automotive and banking domains using BoW (* not use BoW for ranking because it is not a sortable.)

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.35	0.38	0.50	0.40
Domain Vocabulary	0.70	0.70	0.70	0.85
Twitter Auth. + Domain Voc.	0.90	0.40	0.65	0.95
Authority Vocabulary	0.90	0.45	0.90	**1**
Twitter Auth. + Authority Voc.	**1**	**0.80**	**1**	0.95
Domain Voc. + Authority Voc.	0.85	0.35	0.90	**1**
Best Combination	**1**	**0.80**	**1**	**1**

Table 3.4: P@10 average results between modeled automotive and modeled banking domains

5. Even though results per domain are not presented here (for more details see appendix B), the evaluation has also shown that in the automotive domain is easier to identify influencers than in the banking domain. This happens

because the vocabulary concerning to banking is specific (mortgages, stock values, etc.) and therefore both influential and anonymous people use these words since no other less formal synonyms exist. On the other hand, in the automotive domain there are informal words that are mostly used by non-experts users (e.g. wheel vs pneumatic tyre).

3.2.4 Error analysis

In this section we perform a qualitative analysis of error cases showed by our approaches. The results showed in Tables 3.5 and 3.6 tell us that an important part of the errors found in the automotive domain have a common base: well-known users (regional like chihohoho or global like BrooklynNets) are used by the car brands in order to advertise their products, this partnership does not, usually, happen in the banking sector so well-known users are selected as influencers in the automotive domain, due to the advertising relationships they have, even though they do not have knowledge about cars. In the banking domain, errors are contingent. Our algorithm has found, also, annotation errors like *4t7ford* which is a regular user selected as an *influencer*, it talks about its day-by-day experiences, and *wallstCS* which does not have any activity in its account.

User Name	Aut.Model	Dom.Model	Foll	Follees	RT	FAV	DivFoll
wallstCS	0	0	7	0	0	0	–
4t7ford	-0.04	0.02	108	390	329	193	0.28
chihohoho	-0.10	-0.01	1862	502	647	274	3.71
antonello	0.04	-0.02	19456	1719	468	537	11.32

Table 3.5: Example of influencers selected as non-influencer

User Name	Aut.Model	Dom.Model	Foll	Follees	RT	FAV	DivFoll
BrooklynNets	-0.14	0.07	57091	8632	1903	1725	6.61
MercedesUpdate	0.40	0.28	10135	1422	144	642	7.13
LokerDir	-0.55	-0.12	659979	71	1950	1814	9295.48
PressRoom_BBVA	0.13	-0.09	12467	610	975	460	20.44

Table 3.6: Example of non-influencers selected as influencer

3.3 Using information from the followers' profiles to discover influencers

In this section we focus on finding influencers by exploiting information about their followers, beyond the use of the number of followers signal provided by Twitter. The structure of the section is the following: first, we explain the different signals extracted and how we use them in order detect influencers. Then, we provide details about the experimental framework, which includes the dataset used, the design of the experiments, the metrics and the baselines used for comparison. Finally, we show the results obtained and discuss them.

3.3.1 Methods

In this section we introduce the signals and the algorithms employed for the automatic detection of influencers in Twitter by analysing their followers.

3.3.1.1 Signals

We first obtain the language models for authority and domain, as explained in section 3.2.1.1, for the followers' posts instead of using the main profiles' posts. Next, we extract from them some signals in order to explore the role played by followers in the detection of influencers. One of our main goals is to compare the utility of these signals with those extracted from the main's profiles.

Authority signals
As in the case of the main profile, we have experimented with language models. We have studied how the language model used by the followers of a profile helps to determine if that profile is an influencer or not. To this end, we have first calculated some signals that indicate how similar the language of these followers is to the language of authorities, as we did in the section 3.2.1.1 for the language of the main profile. The name, description and formula of these signals, are described below:

1. **Auth:** probability of being an authority. In order to compute this signal, we first obtain the language model for the set of followers and the language model of the authorities. We calculate this signal just as we did in section

3.2.1.1. We denote this signal as $P_{auth}(f)$, where $f \in F$ and F is the set of followers of a given profile.

2. **Not_Auth:** probability of being non-authority. The signal value is computed by comparing the language models for the set of followers and the language model of the non-authorities. We calculate this signal following the procedure described in section 3.2.1.1. Note that, due to the fact that the language used by the authorities and the non-authorities may contain common words, the probability of being authority is not the complementary of the non-authority probability. We denote this signal as $P_{\neg auth}(f)$, where $f \in F$ and F is the set of followers of a given profile.

3. **#_Foll_Auth:** number of followers being authorities. This signal indicates the quality of the connections made for the main profiles. It is expected that connections with the right people lead to a high probability of being influencer. In order to compute this signal, we have to estimate the probabilities of being authority and non-authority for each follower. We count the followers as influencers if they fulfil the following condition: $P_{auth}(f) - P_{\neg auth}(f) > 0$, where $f \in F$ and F is the set of followers of a given profile.

4. **Mod_Foll_Auth:** similar to the previous one, it computes if a main profile is well connected. In other words, if a main profile is followed by a high number of influencers. Like the previous signal, we have to compute, previously, the probability of being authority and not being authority. The final signal is calculated as: $\sum_{f \in F} P_{auth}(f) - P_{\neg auth}(f)$, where F is the set of followers of a given profile.

5. **Avg_Mod_Foll_Auth:** it calculates, on average, the degree of authority of the followers of each main profile. This signal is computed as $\frac{Mod\_Foll\_Auth}{num\_foll}$, where *num_foll* is the number of followers of the main profile.

6. **Prop_Foll_Auth:** ratio of followers being authorities. This signal is computed as $\frac{\#\_Foll\_Auth}{num\_foll}$, where *num_foll* is the number of followers of the main profile.

7. **Avg_Prob_Auth:** is the ratio of followers' influence. Higher values indicate that the main profile messages are validated and disseminated by several experts. This signal is calculated as: $\frac{P_{auth}(f)}{num\_foll}$, where *num_foll* is the number of followers of the main profile and f are the followers of a given profile.

8. **Sum_Foll_Auth:** is the sum of the probabilities of the followers of a main profile being influencers. This signal is calculated as $\sum_{f \in F} P_{auth}(f)$ where $f \in F$ and F is the set of followers of a given profile.

Domain signals

As in the case of the main profile, we have experimented with language models extracted from different domains: automotive and banking. We studied how the language model used by the followers of a profile helps to determine if that profile is an influencer or not within the domain. To this end, we have first calculated some signals that indicate how similar the language of these followers is to the language used for users in both domains (automotive and banking), as we did in the section 3.2.1.1. The name, description and formula of these signals, are described below:

1. **Dom:** it measures how well the discourse of the followers fits in a domain. In order to compute this signal, we first model the language of the follower set and the domains. We calculate this signal as we did in section 3.2.1.1. We denote this signal as $P_{dom}(f)$, where $f \in F$ and F is the set of followers of a given profile.

2. **#_Foll_Dom:** number of followers, of a main profile, that belong to a domain. A follower f fits in a domain if she fulfils the following requirement: $P_{dom}(f) - P_{\neg dom}(f) > 0$. Note that $P_{\neg dom}(f)$ is the probability of not belonging to the domain. Due to that words can belong to different domains, the probability of belonging to a domain may not be the complementary to the probability of not belonging to it.

3. **Mod_Foll_Dom:** it computes the connections of a main profile inside a domain, in other words, whether or not a main profile is followed by people with some knowledge about a domain. It is calculated as: $\sum_{f \in F} P_{dom}(f) - P_{\neg dom}(f)$, where $f \in F$ and F is the set of followers of a given profile.

4. **Avg_Mod_Foll_Dom:** it calculates, on average, the knowledge about a domain that the followers of a main profile have in other words, whether or not a profile is followed by experts in a domain. This signal is computed as $\frac{Mod\_Foll\_Dom}{num\_foll}$, where *num_foll* is the number of followers of given profile.

5. **Prop_Foll_Dom:** is the ratio of followers which belong to a certain domain. This signal is computed as $\frac{\#\_Foll\_Dom}{num\_foll}$, where *num_foll* is the number of followers of the main profile.

6. **Avg_Prob_Dom:** is the ratio of followers which belong to a certain domain. Higher values indicate that the posts published by the main profile can be viewed and confirmed by domain experts. This signal is calculated as $\frac{P_{dom}(f)}{num\_foll}$, where *num_foll* is the number of followers of the main profile and f are the followers of a given profile.

7. **Sum_Foll_Dom:** is the sum of the probabilities of the followers of a main profile of belonging to a domain. This signal is calculated as $\sum_{f \in F} P_{dom}(f)$ where $f \in F$ and F is the set of followers of a given profile.

3.3.1.2 Algorithms

As we explained in section 3.2.1.2, the detection and characterization of influencers is covered here as a ranking problem because it is the most natural way of presenting results to the reputation experts. We have compared the two approaches that obtained the best results for the identification of influencers using the main profiles (see section 3.2.3) to generate a ranking of users' profiles:

- **Direct Signal Rank Strategy (DSR):** each extracted signal (see section 3.3.1.1) or a combination of them generate a ranking of users.

- **Learning to Rank Strategy (L2R):** each main profile that L2R receives is represented as a 1-hot vector, whose length is the total number of followers that exist for all main profiles without repetition. Those positions corresponding to a real follower of a profile, is filled with its respective value (probability of being authority or belonging to the domain), the other positions contain 0 as a value, which indicates that is not a follower of the profile. To combine these vectors, we only concatenate them.

  This experimentation was carried out using the RankLib tool (Dang, 2012).

3.3.2 Experimental framework

One of our primary objectives is to determine how our method, which is based on signals extracted from posts published by the followers of a Twitter user, behaves for identifying influencers, and compare it with other approaches that only use information from the main profile. To do so, our experiments are performed using as main profiles the ones in the RepLab 2014 dataset. As we mentioned in the previous section, we select some signals from the followers' posts to estimate whether a user is an influencer or not (see section 3.3.1.2).

3.3.2.1 Dataset

Since the RepLab 2014 dataset does not supply information related to the followers of its profiles (beyond the number of followers they have), we had to collect the names and tweets of the followers of each profile in the RepLab dataset. The followers retrieved are those whose profiles were created by the extraction time of RepLab (1st June 2012-31st December 2012) and have some post published during that period of time. We gathered 600 tweets per follower, but due to the time elapsed since the dataset was built, some tweets have been lost causing some of the followers to have less than 600 tweets. Despite the main profiles were manually assessed (as influencers or not influencers) by reputation experts, we do not have such information for the followers' profiles.
This extraction was carried out using GetOldTweets-java tool (Jefferson-Henrique, 2016).

3.3.2.2 Metrics

To evaluate our system, we use *Mean Average Precision* (MAP) as in section 3.2.2.3 and also, Precision at 10, 50 and 100 (see appendix B).

3.3.2.3 Baselines

As reference, we have considered the same baselines than in the previous experiments (see section 3.2.2.4) but with one modification:

- **Best result using Feature Engineering** to date that uses the RepLab 2014 dataset are the results of section 3.2.3 that are published in (Rodríguez-Vidal et al., 2019). Here we applied language models of authorities and domain (automotive and banking) knowledge to identify and characterize influencers. From now on we will call Rodríguez'19 to this baseline.

3.3.3 Results and discussion

Table 3.7 summarizes the results (in terms of MAP) of all experiments for each signal explained in section 3.3.1.1. Note that, for those experiments that only use a single signal, their results are presented for the *DSR* strategy only, since *L2R* behaves like *DSR* with one signal.

	DSR	**L2R**
#_Foll_Auth	0.39	-
Mod_Foll_Auth	0.44	-
Avg_Mod_Foll_Auth	0.47	-
Prop_Foll_Auth	0.44	-
Avg_Prob_Auth	0.41	-
Sum_Foll_Auth	0.42	-
#_Foll_Dom	0.35	-
Mod_Foll_Dom	0.41	-
Avg_Mod_Foll_Dom	0.42	-
Prop_Foll_Dom	0.44	-
Avg_Prob_Dom	0.42	-
Sum_Foll_Dom	0.45	-
Best Auth. & Domain Combined	0.35	0.59
All Previous Signals Combined	0.58	0.61
Authority of Followers	-	0.71
Domain of Followers	-	0.60
Authority of Foll. + Domain of Foll. (R1)	-	0.75
Baseline - Followers	0.38	

Table 3.7: Overall MAP results using followers' posts to locate influencers

Regarding the results shown in table 3.7, we can extract the following conclusions:

- The language used by the followers allows to characterize the authority of a profile much better than using only the number of followers provided by the Social Network (0.75 vs 0.38, an improvement of 97%).
- Regarding variants of the number of followers, the best results in our experiments comes from the average authority of followers (instead of their

aggregate authority). In other words, the number of followers is not as important as their average quality. This is relevant because the diffusion of a message through followers will be faster, and therefore will have a greater impact, if other influencers validate and spread that message.

- Followers information alone can reach a performance (0.75) that improves all results in the state-of-the-art except Nebot'18.

Table 3.8 summarizes the results of aggregating the information related to the main profile, with the information of her followers (this aggregation is done by concatenating the signal vector that defines the main profile and the signal vector of her followers). Here, we tested two different ways to add information: the first one (*R1+R2*) combines the scores provided by the L2R algorithm in the experiments marked as *R1* and *R2* using Borda voting. This combination balances the weight of both the main profile and the followers since the latter are defined by a single signal, the score, instead of a signal vector. Meanwhile, the second approach, incorporates directly the information regarding to the main profile as if it were another follower.

	DSR	**L2R**
R1 + R2	0.72	-
Main_Auth_Vocabulary + Main_Dom_Vocabulary + R1	0.74	0.75
Baseline - Nebot'18	0.842	
Baseline - Rodríguez'19 (R2)	0.68	0.74
Baseline - Best RepLab 2014	0.57	
Baseline - Followers	0.38	

Table 3.8: Overall MAP Results using followers' posts to locate influencers

From the results shown in table 3.8 we may conclude that our mechanisms do not combine properly the information regarding to the main profile and her followers, because they can not improve the results obtained by the followers themselves.

3.3.4 Error analysis

In this section we perform a qualitative analysis of error cases showed by our approaches. The results showed in Tables 3.9 and 3.10 reinforce the idea of section 3.2.4 that a number of the errors found in the automotive domain have a common base: well-known users (regional like chihohoho or antonello) are used by the car brands in order to advertise their products. In banking domain, errors appear related to newpapers or news profiles where, from time to time, advertising about banks appear in their pages and so they use a more related banking language. Our algorithm have found, also, annotation errors like *wallstCS*, which does not have any activity in its account.

User Name	#Foll_Auth	Avg_Mod_Foll_Auth	Prop_Foll_Auth	#Foll_Dom	Avg_Mod_Foll_Dom	Prop_Foll_Dom
wallstCS	1	-0.028	0.14	1	-0.013	0.14
MirrorFootball	177	0.022	0.0032	35	-0.027	0.0021
chihohoho	1	-0.136	$5 * 10^{-4}$	3	-0.016	0.0016
antonello	177	0.022	0.009	35	-0.028	0.0017

Table 3.9: Example of influencers selected as non-influencer

User Name	#Foll_Auth	Avg_Mod_Foll_Auth	Prop_Foll_Auth	#Foll_Dom	Avg_Mod_Foll_Dom	Prop_Foll_Dom
FerrariFanDaily	61	0.023	0.005	121	0.054	0.009
MercedesUpdate	34	-0.016	0.0033	67	0.014	0.0066
AlertaDeportes	13	-0.02	0.001	49	-0.038	0.0005
UKnewsV	4	-0.0047	0.003	23	-0.007	0.015

Table 3.10: Example of non-influencers selected as influencer

3.4 Comparing error analysis results

In this section we analyse the errors made by our first approach, in section 3.2.4, and show their evolution, if any, when the followers are involved. Table 3.11 shows four columns corresponding to: user name, approach used, automatic classification values and the goldstandard value.

Each domain ranking was constructed separately. Only two of the profiles selected belong to the Banking domain: *LokerDir* and *PressRoom_BBVA*, the rest users are from the automotive domain.

We can observe from Table 3.11, that some of the profiles that have been incorrectly classified in the first approximation have evolved favourably through the use of their followers: textit4t7ford, *BrooklynNets* and *LokerDir* or have a positive outcome depending on what ranking generator system we choose: *PressRoom_BBVA*. The case of textit4t7ford is special since is a regular user which

User Name	Main Profile	Follower's Profile	Real
wallstCS	Non-influencer	Non-influencer	Influencer
4t7ford	Non-influencer	Influencer	Influencer
chihohoho	Non-influencer	Non-influencer	Influencer
antonello	Non-influencer	Non-influencer	Influencer
Brooklyn Nets	Depends	Non-influencer	Non-influencer
Mercedes Update	Depends	Influencer	Non-influencer
LokerDir	Influencer	Non-influencer	Non-influencer
PressRoom_BBVA	Influencer	Depends	Non-influencer

Table 3.11: Comparison between using main profiles and using follower's profiles approaches

talks about its daily experiences and should be selected as non-influencer so that this is clearly an annotation error. Only in one case, the classification evolves in a negative way: *MercedesUpdate*. Some of its followers are completely related to the automotive domain and has high proportion of influencers between them. In all other cases, the output remains the same, for example, for *wallstCS*, which does not have any activity in its account. In general terms, characterizing followers beyond the use of the signal given by the Social Network provides useful information that helps to better characterize them.

3.5 Conclusions

In this chapter, we have presented a study about the characterization and detection of influencers in Social Networks (in our case Twitter), using information from the profiles of the users and from the profiles of their followers. For this purpose, we have explored different signals provided by Twitter and focused our efforts on the study of authority and domain signals. Authority signals indicate how influential a user is compared to the others while domain signals indicate the degree of knowledge about the domain. We explore two different approaches: the first one is an approach based on the profile of the users to be classified as influencers or not; and, in the second approach, we characterize the profiles based on their followers' information. We spread the authority of each of these followers to know the authority of the profile they follow.

Regarding the research questions that we introduced at the beginning of this chapter, the main conclusions are summarized below.

Research Question 1: *What is the relative importance of authority signals vs domain expertise signals?*
Regarding our first approach, our main objective was to investigate and to compare the role of domain and authority signals in the task of finding Twitter influencers for a given domain. From the experimentation performed we may conclude the following:

- Although it is a common practice to assume that influencers are simply users with a number of followers above a certain threshold, our results indicate that reality is much more complex. The number of followers might be an initial filtering criterion (no one can be influential without an audience), but for users with an important number of followers, the textual content signal (in particular, the domain and authority model) is significantly more powerful than the number of followers and any other Twitter authority indicator (number of retweets, favorites, ratio of followers/followees, etc.).
- Our best result is obtained with L2R using the output of our domain and authority topic models (0.74), which also outperforms the best result which uses feature engineering on the dataset (0.71 (Cossu et al., 2016)). Comparing the signals that model text (authority and domain), we see that the authority signal clearly improves, individually, the domain signal (except for the CR approach). This indicates that, in order to find influencers, it is better to know the degree of authority of a user, within the community that she belongs to, than her knowledge about the domain.

Research Question 2: *How best to combine signals?*
Also in the first part of this chapter (see section 3.2.1.1), we extracted some other signals from the users profiles (such as the number of published tweets, number of followers, etc.) in addition to the textual content signals. Due to the high amount of signals selected, we applied a previous feature engineering process to discard those signals that do not fit in our task and we explored how to combine the remaining signals to better locate and characterize influencers. According to our results in section 3.2.3, the best way to combine signals is:

- In the absence of training data, our best applicable method is a Borda voting combination of the rankings provided by three unsupervised signals: number of followers, followers/followees, and the domain signal (which is modeled using the training part of the dataset, but does not use the authority labels or any other hand-tagged data).

- Note that, in the representation of the tweets, each word is a signal and therefore, the number of textual signals is significantly larger than the rest of signals. That might be a problem for learning algorithms, because the effect of non-textual signals might be shadowed by the large textual signal.

Research Question 3: *How well followers characterize a Twitter profile? Can we establish authority using only information from followers? What is the best way of using followers' information to establish authority?? Moreover, Can information from followers enrich profile characterizations? In other words, is it complementary or redundant with the information from the profile itself? Here we want to aggregate the information related to the profile with the information of her followers and verify if this aggregation improves the profile characterization.* Regarding our second approach, our main objective was to reveal the role played by the environment, the followers, of a profile to detect influencers beyond the use of a numeric signal that represents this environment. From the previous experimentation, we extract the following conclusions:

- The followers of a user may provide essential information for characterizing her. The profiles followed by other influencers are more likely to be influencers. This indicates that influencers tend to be connected with other of their kind.

- This discovery leads us to an interesting discussion. Since influencers tend to be connected to each other, an idea written by one of them is accepted and validated (e.g. using a retweet) by other influencers, and this may cause more severe reputational crises since: (i) the diffusion of a message by other influencers adds a new audience to that idea; and (ii) if the opinions of one influencer are reliable for the users of a given community, the validation by another influencer(s) gives the message a greater veracity in the eyes of that community.

We have also answered the following question: *can information from followers enrich profile characterizations?* In other words, *is it complementary or redundant with the information from the profile itself?*

We have combined, using different techniques, the information related to the main profile with the data extracted from its followers. As we can see in the section 3.3.3, this combination does not provide an improvement with respect to the isolated use of the information of the followers, this may be because the information of the main profiles and their followers is not complementary.

In addition to this, we can draw another interesting conclusion: the fact that the textual content is the key signal in our experiments, and makes Twitter signals unnecessary, is a positive result in terms of applicability of our findings, as we may expect to have competitive results in any other Social Network that includes textual content. In fact, in most other Social Networks there is no limit on the size of the posts (as there is in Twitter), and with longer posts and more textual content results might be even better.

Our study has, however, some limitations. Most importantly, the RepLab dataset samples data from two domains, and we have seen that there is a substantial variability across domains. Our results should be confirmed in a larger and more diverse range of domains. Second, we have focused on supervised approaches (our unsupervised approaches are all Borda voting of primitive signal ranks). The difference between unsupervised and supervised approaches might stretch with more sophisticated unsupervised approaches to the problem. Third, the study of the characterization of influencers using the information of their followers, in spite of giving us good results, is not a complete study since, as we have commented in the chapter, it has not been possible to recover information from some of the followers due to the time elapsed between the crawling time of the RepLab dataset and our study.

Despite our work does not have the best results detecting influencers on Twitter, we believe that our system performs a better detection of influencers than (Nebot et al., 2018). This is due to the fact that the system presented in (Nebot et al., 2018) check whether or not the terms belong to the influencer category. This may provoke that profiles with knowledge about a domain, for instance automotive, but that use terms not employed by the influencers is detected as non-influencer. On the contrary, our system detects as influencers those profiles who have a great knowledge about the domain even though their language may be different from the best-known influencers.

CHAPTER 4

AUTHORITY & PRIORITY SIGNALS IN AUTOMATIC REPORT GENERATION FOR ORM

> This business is well ended. My liege and madam, to expostulate What majesty should be, what duty is, Why day is day, night night, and time is time, Were nothing but to waste night, day, and time. Therefore, since brevity is the soul of wit And tediousness the limbs and outward flourishes, I will be brief.
>
> Polonius-Hamlet, Act II, Scene II

In this chapter, we study the role of the signals seen in the previous chapter, authority and domain knowledge, when they are applied to generate, automatically, reputation reports. This study is completed by exploring other state-of-the-art signals, typically used in the generation of summaries, such as: priority, centrality, polarity and the use of information regarding the topics of a conversation. To know the utility of the previous signals, different experiments have been performed.

Reputation reports are created from extractive summaries generated from tweets that talk about the same entity. Two main types of extractive summaries are explored (i) ranking tweets sorted by the use of the studied signals, from which the top-k tweets are selected to be part of the summary (avoiding redundant information); (ii) clustering tweets that talk about the same entity in different topics. Then, the tweets with higher priority, from each cluster, are chosen and they are included into the summary. In this second approach, we remove redundant information by using the same mechanisms that we used in the first approach.

This chapter is divided into the following sections: first, in section 4.1 we present the task. Second, in section 4.2 we motivate the need of using authority and domain signals and the computation of such signals. Third, in section 4.3, we introduce the different priority signals explored and the final set of priority signals selected to perform the experiments and how to improve the reputational polarity estimators. Fourth, in section 4.4 we exploit the centrality information by using embeddings. Fifth, in section 4.5 we explain the use of topic information to generate reputation reports. Then, in section 4.6 we introduce the dataset developed for this task, the experimental framework, the metrics and the baselines used. In section 4.7 we show the results of the experimentation. Finally, in section 4.8 we summarize the conclusions of this chapter.

4.1 Motivation

The reputation of an entity is given by its appearances in traditional mass media such as television and newspapers, and by the opinions that such media express about it. To monitor the reputation of an entity, the reputational experts had to track all these media, digest and summarize the information, this technique is known as *press clipping*. This work is extremely important because it gives a global vision of the entity's positioning in society: its strengths and weaknesses, the state of its competitors, and it may help to foresee possible communication crises. The data collected in this process must be studied by experts who separate important information from non-important one. Likewise, they carry out a task of information synthesis in order to present it, in an understandable way, to the clients who have hired the clipping service. As it can be observed, this task is both important and very expensive, since it is necessary to monitor and classify the data flows that come from the mass media. Figure 4.1 shows a diagram of the clipping tasks performed by a human operator.

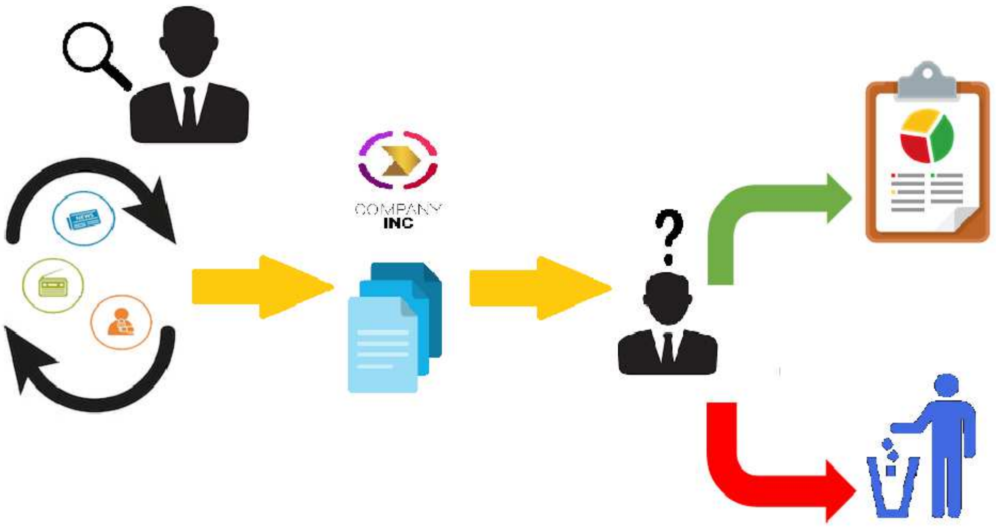


Figure 4.1: The press clipping diagram

With the advent of the Internet in the 20th century and the emergence of Social Networks such as Facebook, Twitter or Flickr, blogs, YouTube, etc. in the 21st century, it is becoming increasingly difficult for humans to monitor all possible sources of information and control the data flow. As already discussed in chapter 3, all sources of information in Social Networks are not equal, as there are some profiles that have more impact in a community than others, the so-called *influencers*. Reputation experts should, therefore, identify these profiles and collect their points of view in order to avoid possible reputational crisis. Figure 4.2 shows a photogram of a Kellogg's advertising campaign with the collaboration of three well-known youtubers (Elrubius, Alexby11 and Mangel). The video which was broadcasted in the three accounts, got more than four million views[17]. This is an example of how just three people, with few technical resources, were able to propagate the advertising message in a more effective way than traditional mass media (TV, radio, newspapers, etc.).

Being able to collect information only solves part of the problem. The other part of the problem is knowing how to present such information so that humans are capable to read and understand it in a reasonable time thus making the decision process both possible and useful. There is, therefore, a need to summarize the entire flow of information to obtain only those data that are relevant and that

[17] https://www.marketingdirecto.com/marketing-general/publicidad/6-incursiones-marketeras-elrubius-que-dejan-claro-marcas-confian-el

Figure 4.2: Photogram of Kellogg's campaign with three well-known youtubers.

provide new information, since duplicate information does not add value to the summary.

But what is a summary? In general, a summary is a document that contains the same main concepts than the original texts in a condensed way. As mentioned in chapter 2, in the NLP research field, there are two main ways of generating summaries automatically: *extractive* and *abstractive* (Das and Martins, 2007). In the extractive approach, word sequences (phrases, sentences or paragraphs) are chosen from the original document and copied into the summary directly. This technique has the problem of the lack of coherence between sentences in the summary but stands out for its computational simplicity. *Abstractive* summaries are more difficult to create, because they involve paraphrasing the text in the source document and generating text by using Natural Language Generation techniques, but ideally, they solve the problem of cohesion between sentences in the summary (Das and Martins, 2007).

Table 4.1 shows an example of *extractive* and *abstractive* summary for a same source document. It may be seen that the *extractive* summary retrieves the most important sentences from the text but there is no coherence between them: the first sentence talks about the number of people climbing in the U.S. and the second one tells us that many methods and devices can increase the climber's safety. However, in the *abstractive* summary the text is perfectly cohesive and readable.

Original Text	An estimated nine million people rock climb in the United States. Millions more take part in the activity around the world. Some do it just for personal satisfaction. Others compete. Rock climbing can be dangerous. But there are many methods and protective devices that can increase a climber's safety.
Extractive Summary	An estimated nine million people rock climb in the United States. But there are many methods and protective devices that can increase a climber's safety.
Abstractive Summary	Millions of people practice rock climbing around the globe. Despite the risk of this activity, exist many methods and protective devices that increase a climber's safety.

Table 4.1: Example of extractive and abstractive summaries

Automatic summarization may classified according to the number of input documents that the system receives: *single-document* and *multi-document* (Nenkova and McKeown, 2012). Whereas the first approach creates automatically the summary from the information within one single document (Litvak and Last, 2008), the second approach uses the information obtained from different sources, written on the same topic, to generate automatically the summary (Lin and Hovy, 2002). This last approach may introduce redundant information (content that is expressed more than once) to the summary, therefore, some mechanisms are necessary to avoid this problem (Inouye and Kalita, 2011; Takamura et al., 2011).

In this section, we deal with the problem of creating *reputational summaries*, which show condensed information about the opinions that people express about companies, products, etc. in Social Networks, such as Twitter. *Reputational summaries* are part of a wider structure called *reputation reports*. These reports systematize different *topics*, i.e. for an automotive company, the topics that could appear in a reputational report vary between the different models of cars they manufacture, the durability of car components, etc., that affect or may concern to the client and in a near future produce, a reputational crisis. The topics of conversations appear in the report according to the relevance or *priority* that they have for the client, being the highest priority topics those that appear in the first positions, while the unimportant topics occupy the last positions in the report. Reputation reports can also propose and outline different strategies to overcome reputational crises. The reputation summaries that form part of the reputation reports, summarize the collected opinions within the different topics of the conversations about the entities (an example is provided in appendix C). In our case, because we focus on the management of reputational crises, the reports only contain alerts and mildly-important topics. In figure 4.1, we can see an example of different topics about a client that could appear in a conversation, ordered by their priority.

```
Entity: Wells Fargo
  Topic: Robberies in Bank Branches, priority: alert
  Topic: Analysis From Bank Economists, priority: mildly_important
  Topic: Situations Vacant, priority: mildly_important
  Topic: Credit Cards Overdrafts, priority: mildly_important
```

Figure 4.1: Example of topics about Wells Fargo

As mentioned before, each topic has different relevance or *priority* for a client. Those issues that affect most seriously to the entity's reputation have greater *priority* than those less important problems. Priority depends on many different factors (Amigó et al., 2013), including the *authority* and the *domain knowledge* of the users (are influencers engaged in the conversation?), *polarity* (does it has positive or negative implications for the client?), *centrality* (is the client central to the conversation?), etc. Figure 4.2 shows a synthesis of a reputation report where the topics are sorted by their priority and each topic contains two summaries: abstractive (labelled as "abstract") and extractive (labelled as "extract").

```
Entity: Wells Fargo
  Topic: Disability Organizations Protest in San Francisco, priority: alert
    abstract: San Francisco senior and disability organizations protest
        outside of Wells Fargo Bank
    extract: San Francisco senior and disability organizations protest outside
         of Wells Fargo Bank - SLIDESHOW: Two San Franc... http://bit.ly/Yj3COZ

  Topic: Bob Ryan Hiring, priority: mildly_important
    abstract: The top housing adviser in the Obama administration is leaving
        to join Wells Fargo
    extract: @politicsnation ARGH!!! Obama Housing Official to Join Wells
        Fargo - Developments http://on.wsj.com/X24gLd  @msnbc @cnnbrk @edshow
        @maddow @dccc
```

Figure 4.2: Example of reputation report regarding Wells Fargo

One of our main goals is to discover the role played by the signals that model *influencer* in the generation of *reputation summaries*, and combine these signals with other signals, widely used in the state-of-the-art of the automatic summarization task. In this research, we used *extractive* summaries over *abstractives* ones because many of the tweets are written without following any orthographic or cohesive rules, so here generating, automatically, well-written and cohesive summaries from them is a task that requires an extensive research in NLG methods.

In summary, the research questions tackled in this chapter are:

> **Research Question 4:** *Can authority and domain signals be effectively exploited in order to create reputation summaries?* As we said before, we want to collect and summarize those messages in Social Networks that have greater impact in the reputation of an entity (e.g. a company or a product). Also, as we have said, influencers are a special kind of users in Social Networks that are well-known inside a community and whose opinions are followed by a huge amount of people. Because authority and domain signals have been used to characterize and identify influencers, we want to use these signals to rescue opinions expressed by this kind of users and give them priority when generating reputation summaries.
>
> **Research Question 5:** *What role do priority, polarity and centrality play in the generation of reputation summaries?* Beyond the signals that characterize influencers, we want to incorporate other signals which have been widely used in the state-of-the-art of automatic summarization (priority of the topic, polarity of the comments or centrality to the topic), in the creation of our reputational summaries and to study the value they incorporate to this task.
>
> **Research Question 6:** *What is the performance of different similarity functions for avoiding redundancy?* Since a summary should not present repeated information, it is essential to have a mechanism to detect redundancy. For this reason, we want to study the effect of different ways of measuring redundancy in short texts (tweets in our case) and the effect this has on the creation of reputation summaries.
>
> **Research Question 7:** *How can we use topic information to create reputation summaries?* Topics give information about the different subjects of conversation in Social Networks. These topics group different opinions, about the same issue, that may affect to the entity's reputation; therefore, this information must be included in the report. In order to include the information regarding to the topic, we test two different ways to use topic

information: (i) using the topic division provided by the dataset; (ii) using learning similarity functions.

4.2 Exploiting authority and domain information to generate automatic summaries

Just as in human societies where there have always existed figures (heroes in legends, political leaders, scientists, etc.) whose ideas have been respected by their peers and their next generations, in Social Networks there are also users whose opinions influence the rest of users in the community, these users are the influencers or authorities. The authority can be circumscribed only to a certain domain, for example: banking, music, cars, etc. or it can transcend to other domains, for example, in the case of celebrities, sportsmen, etc.

The study of the global authority and the domain authority has not been previously exploited in the context of reputation summary generation. Given that the purpose of this type of summaries is to tackle possible reputational damage of an entity and since messages from authorities can reach thousands of people, it seems reasonable to give more relevance to messages from authorities than to those from regular users (although potentially these users can also trigger reputational crises, their impact rate is lower). The textual information generated by these messages is incorporated to our experimentation through the use of *Language Models* (LM) that model the discourses of global and domain authorities, as we did in chapter 3.

- *Authority model.* When modeling global authorities, we build a language model of all profiles in the training set manually labelled as authorities (see section 3.2.1.1). Our hypothesis is that authorities will employ a distinct way of expressing their opinions. Then, for each profile in the test set we estimate how compatible is her language with the language model learned from the training set, and use one single signal to store such compatibility.

- *Domain model.* We build language models to estimate the domain knowledge, using the same procedure as described in section 3.2.1.1. The hypothesis is that the language used, for example, to talk about football is not the same as that the language used to talk about fashion. Training is carried out with texts of the domain under consideration. Therefore, the domain signal is an unsupervised process with respect to the task, but it requires labels to assign the domain.

In our experimentation, we combine these *authority* and *domain* signals with other state-of-the-art signals that have proven its effectiveness in summarization, such as priority, polarity, centrality and topic-related signals.

4.3 Exploiting priority information to generate automatic summaries

In a field as changing as online reputation management, where it is crucial to know what is said in order to cut a possible reputation crisis, it is necessary to estimate the priority of the different contents to know the order in which to respond to the problems encountered and to detect reputational alerts. Thus, when generating a reputation summary, the highest priority issues should appear in the first positions of the summary. It is not the same to attend first a reputational alert, where a bad comment can make the companies to lose a lot of money, than a commentary talking about a subject of minor importance for the entity. Carrillo-de Albornoz et al. (2016) state that priority signals depend on several factors: popularity (if there are many people commenting on a fact), polarity for reputation (if the message has positive or negative implications for the client), novelty (is a new problem or is a recurring one), authority (there are opinion makers in the conversation), centrality (the client is the focus of the conversation), etc.

Following the guidelines of (Carrillo-de Albornoz et al., 2016), our summarization proposal is modeled as a search for diversity problem (Yang et al., 2010). In this task, a system provides a ranked list of documents that maximizes both relevance (documents are worthwhile to the query) and diversity (documents reflect the different query intents, when the query is ambiguous, or the different facets in the results when the query is not ambiguous). The production of an extractive summary is similar: the texts chosen to be part of the final summary must maximize both the relevance (sentences express essential information from documents) and the diversity (coverage of topics related to the entity). To do this, we generate a ranking of tweets and then choose the texts from the ranking that maximize relevance (priority of each of the topics related to the entity) and minimize redundancy. Table 4.2 shows the explored signals and their definition. Note that polarity signals have been extracted using three affective lexicons: the General Inquirer (Stone et al., 1966), SentiSense (de Albornoz et al., 2012) and SentiStrength (Mike et al., 2010).

Signal	**Definition**
Author_Num_Tweets	Number of tweets published by the author
Author_Num_Followers	Number of Followers that a profile has
Author_Num_Followees	Number of people followed by the profile
Long_Tweet	Tweet length
Date	When the tweet was written
Mentions_Count	Number of Twitter users mentioned
Hashtags_Count	Number of hashtags used
URLS_Count	Number of URLs in a tweet
Words_Count	Number of words in a tweet
TFIDF_Word	Sum of the tf/idf of each word in a Tweet
Words_in_SpellChecker	Number of well written words
Num_Pos_Words	Number of words with positive sentiment
Num_Neg_Words	Number of words with negative sentiment
Num_Pos_Emoticons	Number of emoticons associated with positive sentiment
Num_Neg_Emoticons	Number of emoticons associated with negative sentiment
Similar tweets in 24h	Number of similar tweets produced in a time span of 24 hours
Similar_Tweets	Number of similar tweets

Table 4.2: Signals explored

The explored signals can be classified according to the information they represent: characteristics extracted from users profiles (Author_Num_Tweets, Author_Num_Followers and Author_Num_Followees), signals collected from tweets (Long_Tweet, Date, Mentions_Count, Hashtags_Count, URLS_Count, Words_Count, TFIDF_Word and Words_in_SpellChecker), polarity signals (Num_Pos_Words, Num_Neg_Words, Num_Pos_Emoticons and Num_Neg_Emoticons) and signals related to conversation topics (Similar tweets in 24h, Similar_Tweets).

In order to select the signals, we computed two estimators of the quality of each signal: the ratio between average values within priority values (if priority tweets receive higher values than unimportant ones, the signal is useful) and the Pearson correlation between the signal values and the manual priority values. We select those signals with a Pearson correlation above a certain threshold, in our case we chose 0.02 as threshold, and with a ratio of averages above 10%. Hence, these estimators reduce the set of signals showed in table 4.2 to the following ones, showed in table 4.3:

Signal
Author_Num_Followers
Author_Num_Followees
Mentions_Count
URLS_Count
Num_Neg_Words
Num_Pos_Emoticons
Similar tweets in 24h

Table 4.3: Set of selected signals

These signals are calculated for each profile and tweet. These signals will be used to rank the tweets, as explained in section 4.6.2.1.

4.3.1 Improving reputational polarity estimators to generate automatic summaries

As already explained, we extract two indirect indicators of tweet polarity: the number of words that denote negative sentiments and the number of positive emoticons. Since some cases could not been properly detected with both previous indicators (for example, in the case of detecting the polarity of an ironic tweet, where the writer's intention may be contrary to what she writes) an improvement has been made in the tweet polarity calculation.

The improvement of the polarity estimators is based on the study of (Giachanou et al., 2017), who also work on the same scenario as ours, online reputation management, and uses a similar dataset with short texts obtained from Twitter. Her main task is to determine whether or not factual texts are positive, negative or neutral spreading the sentiment found in other texts that talk about the same fact, to the factual ones. To separate polar facts from others, the authors build a binary classifier, based on an SVM, which monitors whether an input tweet is a polar fact or not, without differentiating between positive and negative tweets. The signals used are classified in three different groups:

- **n-grams:** with $n \in [1, 4]$
- **stylistic:** where the authors explore the writing style of the users: number of capitalized words, number of exclamations used, etc.

- **lexicons:**
  - **Manual lexicon:** use a list of words with their polarity annotated (positive or negative). The use of opinionated words indicates the polarity of the whole tweet. One of the main problems of Twitter texts is that many words are misspelled and therefore, are not found in the lexicon.
  - **Augmented lexicon:** in order to solve previous issues, the lexicon is expanded with new words that have been learned from the training data and, once increased, the occurrence number is used to predict the polarity of the document (using Pointwise Mutual Information (PMI) (Church and Hanks, 1990)).

The authors explore the effectiveness of the classifier using three different training scenarios: *domain and entity-independent*, *domain-dependent* and *entity-dependent*.

Once the polar facts are classified, their polarity is calculated based on the polarity of similiar tweets. In order to propagate the sentiment between tweets, the authors proposed two different methods based on the *maximum sentiment of similar tweets* and based on the *similarity between each tweet and each one of the polarity classes*.

- **Maximum sentiment of similar tweets:** in this approach, the sentiment of the tweet is the maximum number of similar tweets that are positives, negatives or neutral.

- **Similarity between each tweet and each one of the polarity classes:** in this case, the polarity of the tweet is determined by the class whose average polarity, between the tweet and the class (positive, negative and neutral), has the maximum value.

In order to determine the group of tweets for which we already know the sentiment, the authors used two approaches: the first one (Spina et al., 2014), trains a classifier to predict if two tweets belong to the same cluster using hierarchical agglomerative clustering, and the second approach, considers cosine similarity over a bag of terms representation.

In this work, we predict the polarity (positive, neutral or negative) of each tweet calculating PMI score of each term of the tweet using the previous *Augmented Lexicon*, the final polarity of the tweet corresponds to the highest PMI obtained for its terms. Then, we rank these tweets and the position of in the ranking is used as a signal that models the priority.

4.4 Exploiting centrality information using embeddings to generate reputation reports

As we said in chapter 2, centrality has been one of the most widely used techniques for content selection. It is related to the idea of how much a fragment of text (usually a sentence) covers the main topic of the input text (a document or set of documents). Centrality of a sentence is often defined in terms of the centrality of the words that it contains. Different ways exist to represent a word but, lately, vectorial representation of words has gained strength. From all the techniques to represent a word as a vector, word embeddings is the most used in the state-of-the-art because it captures the context of a word in a document, semantic and syntactic similarities, etc. In this section, we explore the use of embeddings to represent the whole tweet, not only each word separately, to calculate the centrality of the tweet to the topic by two different ways: using a sequence to sequence model and document to vector.

4.4.1 Creating tweet vectors using standard sequence to sequence model

In this approach, we create a sequence to sequence model (seq2seq) (Sutskever et al., 2014) for generating vectors which represent tweets. Typically, the seq2seq models are constructed using two RNNs (Recurrent Neural Networks), functioning as encoders and decoders respectively. We build the standard seq2seq model following the guidelines provided by (Li et al., 2015b). The seq2seq input will be a matrix of vectors, each one representing a tweet, and each position in the vector will be the word vector representation. We use the specific Twitter pre-trained word vectors from GloVe project[18].

[18] Glove: https://nlp.stanford.edu/projects/glove/

With this input, we aim to compress the huge amount of information contained in the matrix of vectors into a single vector. Each vector represents the entire tweet and contains a mixture of the semantic information of the words appearing in the tweet. For implementing the seq2seq model, we have used two LSTMs (Long Short-Term Memory) (Hochreiter and Schmidhuber, 1997), one for the encoder layer and the other for the decoder layer. We use LSTMs instead RNNs for avoiding the problem of long term dependencies because they remember information for long periods of time. The encoder layer receives the input matrix and the output generates the vector representation of each tweet. This content will be used as input of the decoder layer which rebuilds the input by predicting the tokens inside the document, in our case tweets. The following image illustrates the model:

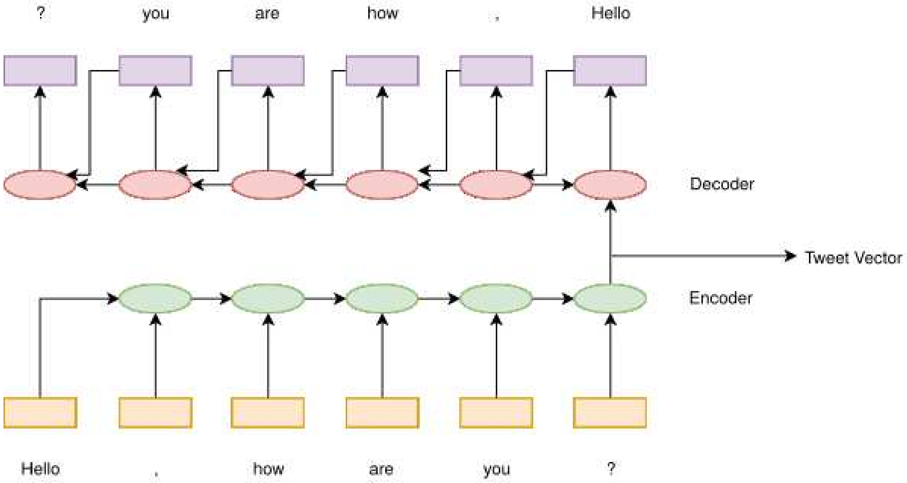


Figure 4.3: Standard Sequence to Sequence schema

Each LSTM consists in 1000 neurons in the hidden layer, for this reason, the dimension of the vectorial representation of each tweet is 1000 as well. For the parametrization of the network, we use the same values suggested by (Li et al., 2015b) in their article (we used this parametrization because, in this work, the model shown was trained with relatively short texts and it had a satisfactory performance):

- The input documents are reversed.
- The word embedding size is 200 because the pre-trained embeddings used have this length. This length is less than the vector generated because

the tweet vector includes the information regarding the whole tweet meanwhile, the word embedding only provides information regarding the words separately.

- The parameters of the network are initialized with a uniform distribution between [-0.08,0.08].
- The size of the batch is 32.
- The number of epochs is 7.
- The learning rate is fixed in 0.1 and, from the 5th epoch, this value is halving.
- The optimizer used is a stochastic gradient descent without using momentum and using clipping when the norm exceed the value of 5.
- The loss function used is categorical crossentropy given that the words in the vocabulary belong to its own category and we have several different words.
- The activation function used is softmax in the output layer because we have more than two classes, one for each different word in the vocabulary.

4.4.2 Creating tweet vectors using document to vector model

In this approach, we have used a well-known method called document to vector (doc2vec) (Le and Mikolov, 2014) which is a version of word2vec (Mikolov et al., 2013) that creates a numerical representation of the document regardless of its unsupervised length. Unlike word2vec, where each vector represents the concept of a single word, doc2vec represents the concept of a complete document and saves the internal context of that document. Given a set of training documents, doc2vec can be used in the following way: a W vector is generated that represents each of the words and a D vector that represents the document. The weights of the internal layer are adjusted with the training and recalculated when a new instance is learned.

Doc2vec has two different ways to be trained: the first one called *distributed memory model of paragraph vector (PV-DM)*, in which the order of the words in the document is preserved, and the second training approach, called *distributed bag of words model of paragraph vector (PV-DBOW)*, in which the order is not preserved. Figure 4.4, illustrates these two training options.

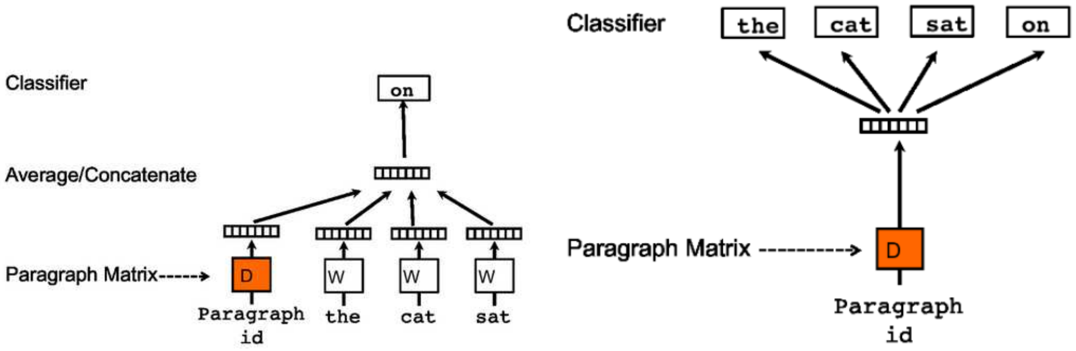


Figure 4.4: PV-DM (left image) and PV-DBOW (right image) schemas[19]

Since our dataset has a relatively small training set, and tweets are not usually written following the orthographic rules (for example, we can find the word “hello” written as “heeeeellooooo”, “ellooooo”, etc.), and do not usually are syntactically correct, we use the model PV-DBOW because it is more effective with small quantities of data. As we mentioned before, since doc2vec does not have any length restrictions of the input documents, we use each tweet as an input document and the output vectors are the vectorial representations of the tweets. In order to generate the tweet vectors, we follow the following steps:

1. **Training data:** since our dataset is small, we use all available data to train and thus create the widest vocabulary possible. All words that are not in the vocabulary are assigned the value *Out of Vocabulary* (OOV).

2. **Tweet labelling:** in doc2vec, we need to specify the number of sentences (tweets in our case) that convey a semantic meaning, so that the algorithm could identify it as a single entity. We specify an unique label per tweet which it means that each tweet conveys a semantic meaning and they may or may not have similarity among them.

3. **Parametrization:** we use the following standard configuration of parameters in our doc2vec model:

[19] Images extracted from: https://upc-mai-dl.github.io/emb-space-theory/

- *vector size:* 1000.
- *number of epochs:* 7.
- *learning rate:* 0.1, halved from the 5th epoch.
- *discard words:* with a frequency below than 2.

4. **Training the model:** we train our model using the vocabulary build from the tweets labeled and the tweets labeled with the parameters specified previously.

5. **Infer vectors:** once the model is trained, vectors from the test data are found.

For carrying out these experiments, we have used Gensim[20] library, which is a robust set of modeling open-source tools to deal with vector spaces and it is widely used in other state-of-the-art works (Markov et al., 2017; Maslova and Potapov, 2017; Trieu et al., 2017).

4.4.3 Extracting signals form vectorial information

Once the vectorial representations of the tweets are obtained, using seq2seq (section 4.4.1) or doc2vec (section 4.4.2), we calculate a centroid vector from the tweets of each topic. Here we take advantage of the topic division provided by the dataset (see section 4.6.1), each component of the centroid is calculated as the average of each component of the vectors. Let's assume that we want to calculate the coordinates of the vector centroid $\vec{O}$ given three vectors that represent three different tweets: $Tw\vec{e}et1 = (7, 6)$, $Tw\vec{e}et2 = (1, 7)$ and $Tw\vec{e}et3 = (4, 2)$ as we can see in figure 4.5 (left image). For calculate the x,y coordinates of the centroid we average the value of the coordinates x,y of each vector:

$$\vec{O}_x = \frac{7+1+4}{3} = 4$$

$$\vec{O}_y = \frac{6+7+2}{3} = 5$$

In figure 4.5 (right image) we can see the centroid calculated.

With this centroid, we calculate the cosine distance (Huang, 2008) between each tweet and the centroid, these values are the signals that we use to represent

[20] https://radimrehurek.com/gensim/models/doc2vec.html

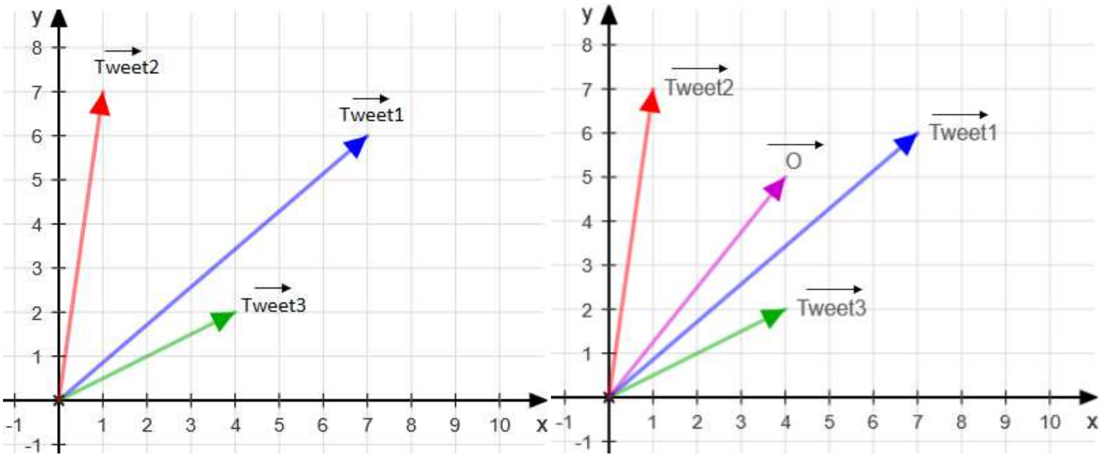


Figure 4.5: Initial scenario (left image) and centroid calculated (right image)

centrality. Figure 4.6 shows the graphical representation of the cosine distance. Note that we extract two sets of signals: the first one corresponds to the seq2seq representation and the second one to the doc2vec.

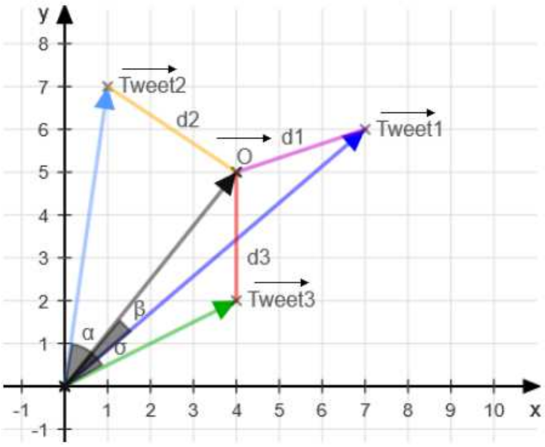


Figure 4.6: Cosine distance between every tweet vector and the centroid vector

Continuing with the previous example, we calculate the cosine distance between the centroid $\vec{O}$ and each of the tweet vectors: $T\vec{weet}1$, $T\vec{weet}2$ and $T\vec{weet}3$. The cosine distance is defined as:

$$cosine\_distance(\vec{u}, \vec{v}) = 1 - \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||_2 ||\vec{v}||_2}$$

where

$$\vec{u} \cdot \vec{v} = |\vec{u}||\vec{v}|cos(\sigma)$$

$$cos(\sigma) = \frac{\sum_{i=1}^{n} \vec{u}_i \vec{v}_i}{\sqrt{\sum_{i=1}^{n} \vec{u}_i^2}\sqrt{\sum_{i=1}^{n} \vec{v}_i^2}}$$

$$|\vec{u}| = ||\vec{u}||_2 = \sqrt{\sum_{i=1}^{n} \vec{u}_i^2}$$

therefore:

$$cosine\_distance(\vec{u}, \vec{v}) = 1 - \frac{|\vec{u}||\vec{v}|cos(\sigma)}{||\vec{u}||_2||\vec{v}||_2} = 1 - \frac{||\vec{u}||_2||\vec{v}||_2 cos(\sigma)}{||\vec{u}||_2||\vec{v}||_2} =$$

$$= 1 - cos(\sigma) = 1 - \frac{\sum_{i=1}^{n} \vec{u}_i \vec{v}_i}{\sqrt{\sum_{i=1}^{n} \vec{u}_i^2}\sqrt{\sum_{i=1}^{n} \vec{v}_i^2}}$$

in our example,

$$cosine\_distance(\vec{Tweet1}, \vec{O}) = 1 - \frac{7 * 4 + 6 * 5}{\sqrt{7^2 + 6^2}\sqrt{4^2 + 5^2}} \approx 0.02$$

$$cosine\_distance(\vec{Tweet2}, \vec{O}) = 1 - \frac{1 * 4 + 7 * 5}{\sqrt{1^2 + 7^2}\sqrt{4^2 + 5^2}} \approx 0.14$$

$$cosine\_distance(\vec{Tweet3}, \vec{O}) = 1 - \frac{4 * 4 + 2 * 5}{\sqrt{4^2 + 2^2}\sqrt{4^2 + 5^2}} \approx 0.09$$

The cosine distances between the centroid vector and the tweet vectors are used to create a ranking of tweets where the first k elements, that do not introduce redundant (see section 4.6.2.1) information, are chosen to be part of the summary.

4.5 Using topic information to generate reputation summaries

Topic detection is crucial for the generation of reputation summaries. Topics give information about the different subjects of conversations in Social Networks (i.e. the number of robberies suffer for a bank company, factory defects appearing on a car model, etc.), that may affect to the client and should be taken into account in order to avoid reputational crisis. Reputation reports must reflect these topics of conversations according to its reputational importance: first to appear in the report must be very important topics (alerts) while in the last positions of the report should appear unimportant ones. The use of topics is useful to generate summaries automatically because tweets grouped under the same topic contain similar information and therefore, it is easier to avoid redundancy and add diversity to the final summary.

In this section we explore the use of topics as an intermediate step to generate summaries. For this reason, first we take advantage of the topic division provided by the dataset to have an idea of the best result reachable for the topic extraction methods. Then, we use and approach for topic detection in ORM.

4.5.1 Using manually labeled topics

The *oracle* or *manual* strategy uses the division by topics of the entities given in the dataset. The main advantage is that is not necessary to implement any type of software to obtain the clusters and the elements of each one of them and besides, for a machine it is difficult to reach the level of precision of an expert annotator. The main drawback is that human annotators, no matter how expert they are, are not exempt from making mistakes and they can include in clusters elements that do not belong to it and another issue is that of course, these clusters or topic classification are not presented in real world. But, despise of the drawbacks, this strategy gives us an idea of the best result reachable for the topic extraction methods.

4.5.2 Topic detection learning similarity functions

This approach is based on the work presented in (Spina et al., 2014)[21] for topic detection in ORM. Unlike the *oracle* approach, here we do not use the labels of the topics as clusters but we make use of different signals of similarity between tweets to automatically build the clusters.

In order to automatically detect topics, the authors proposed two subtasks to address. In the first one, they learn a similarity function between tweets that allows them to know whether or not two tweets belong to the same cluster. Different similarity signals are studied to measure the degree of overlapping between two tweets, and are grouped as:

- **Term signals:** the overlapping of words is taken into account. The idea behind it is that if two tweets share a large number of words, then they

[21] The code is publicly available at https://github.com/damiano/learning-similarity-functions-ORM

have a high probability of talking about the same topic.

- **Semantic signals:** the authors use external data from Wikipedia, through the entity-linking (Meij et al., 2012) technique, to know which concepts are semantically related to the tweet. To do this, they calculate the commonness probability that a concept/entity is the target of an anchor text link in Wikipedia.
- **Metadata signals:** extracted from tweets, such as author (tweets published by the same author are more likely to talk about the same topic), number of shared mentions between two tweets, number of urls that co-occur between two tweets and number of hashtags shared.
- **Time-aware signals:** tweets written in short periods of time are more likely to indicate the same event, e.g. a music concert, football match, etc.

The second of the subtasks uses the similarity matrix for each pair of tweets calculated in the previous step as input to an Agglomerative Hierarchical Algorithm (HAC) (Schütze et al., 2008). In HAC, there is no need to specify the number of clusters a priori, it works in the following way: it first creates an individual cluster for each of the tweets; next, two clusters are agglutinated when the similarity between them exceeds a certain threshold, which acts as a condition for stopping the algorithm. According to the authors, the main drawback of this algorithm is that clusters may be merged due to single noisy elements being close to each other.

4.5.3 Adding topic information to the summary

Once the topics are created and the tweets are assigned to them, we assign a priority that defines the importance of the topic inside the summary. To do this, we take advantage of the influence, modeled by the authority and domain knowledge (see section 4.2) signals, that the people who post have. The main idea here is that the topics with high number of influencers are more likely to be a potential threat to the reputation of the entities and, therefore, must be included in the summary before other less important topics. We define the priority of a topic as the average authority of the tweets' authors of each topic. We use three different ways to model the priority: using the authority signal, using the domain knowledge signal and combining both (see section 4.6.2.2). Next, we sort the tweets that are inside each topic according to their priority, using the same signals used to assign the priority to the topics, and finally, we choose one or more tweets representative of each topic, avoiding to add redundant information to the summary (the similarity between the chosen tweets and those of the summary does not exceed a certain threshold, see section 4.6.2.1). Figure 4.7 shows this process:

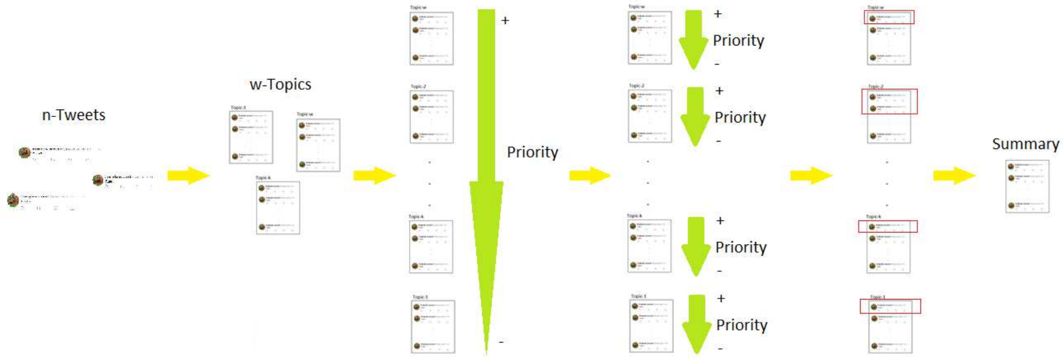


Figure 4.7: Topic localization and prioritization diagram

4.6 Experimental framework

The primary focus of our experiments is to determine the importance of the signals that model the authorities and the profiles that have a broad knowledge of a domain for the automatic generation of summaries. To do so, we perform experiments on the RepLab Summarization dataset, that has been developed during this work because, as far as we know, no similar resource is available for research. Another focus of our experiments is to investigate the role of other state-of-the-art summarization signals for the automatic generation of reputation summaries. For this purpose, we select some signals (explained in previous sections) that model the priority, polarity, centrality and topic information along with the authority and domain signals. We compare our results with state-of-the-art baselines in order to measure the adequateness of our signals.

4.6.1 The RepLab summarization dataset

As part of the present work, and because no similar resource is available for research, we have developed the RepLab Summarization Dataset[22]. The RepLab summarization dataset contains companies data from the RepLab 2013 dataset[23], where users from Twitter talk about different topics of a set of companies. Each topic consists of a different number of tweets posted by Twitter users.

RepLab 2013 dataset uses Twitter data in English and Spanish. This collection comprises tweets about 61 entities from four domains: *automotive*, *banking*, *universities* and *music*. The RepLab Summarization collection only comprises tweets from two of such domains (31 entities in total): *automotive* and *banking*. We only use tweets from these domains because they consist of large companies, i.e. Wells Fargo, Bank of America, Nissan, Fiat, etc., which are the standard subject of reputation monitoring as it is done by experts: the annotation of universities and music bands and artists is more exploratory and does not follow widely adopted conventions as in the case of companies. As a result, our subset of RepLab 2013 comprises 71,303 tweets in English and Spanish distributed as shown in Table 4.4. For each entity, tweets are grouped in topics and for each topic three different summaries are manually generated: abstractive English, abstractive Spanish and extractive.

	Automotive	**Banking**	**Total**
Entities	20	11	31
# Tweets (Training)	15,123	7,774	22,897
# Tweets (Test)	31,785	16,621	48,406
# Tweets (Total)	46,908	24,395	71,303

Table 4.4: Subset of RepLab 2013 used in the *RepLab Summarization dataset*

Tweets provided by the RepLab 2013 dataset were manually grouped by topics and, for each topic, a priority was manually assigned by reputational experts. To develop our summarization dataset, we presented to an annotator the tweets grouped by topic. Only “Alert” and “Mildly important” topics are considered: we discard “Unimportant” topics, as we consider them irrelevant for summarization purposes. For each tweet in a topic, the following information is available: the *ID* or unique identifier of the tweet, the *date* when the tweet was written, the number

[22] RepLab Summarization Dataset: `https://zenodo.org/record/2536801#.XDcq2lxKiUk`

[23] RepLab 2013 Dataset: `http://nlp.uned.es/replab2013/`

of *followers* of the author of the tweet, the *reputational polarity* (i.e. if the tweet has positive/neutral/negative implications for the reputation of the entity) of the tweet, and the *text* of the tweet.

For each topic, we asked the annotator to generate:

- An *extractive summary*, selecting the tweet or tweets that best summarize the content of the topic. The annotator was allowed to make no selections if she considered that no tweet is representative of the topic. We asked the annotators to be very careful not to include redundant tweets in the selection. If two tweets are equivalent for summarization purposes, the annotator was instructed to select the tweet whose author has more followers and, in case of a tie, to pick the one that was created first. In practice, the number of tweets selected as a representative summary ranges from 0 to 3.

- An *abstractive summary*, writing a paragraph that summarizes the content of the topic, both in English and in Spanish (note that the RepLab dataset contains tweets in both languages).

As a result, for each entity in the dataset we obtained (i) an **extractive summary** that consists of the list of tweets that summarize each of the topics for that entity, ordered by the priority of the topics the tweets come from; and (ii) two **abstractive summaries** (one in English and one in Spanish), which are the concatenation of the paragraphs that summarize each of the alerts and mildly important topics. The average number of words in a entity's abstract depends on the domain and the language. Spanish abstracts in the automotive domain have, on average, 391 words while in the banking domain the average number of words per abstract is 677. For English abstracts, average number of words is 323 for automotive and 553 for banking. Average sentence length is 4.4 words in Spanish abstracts and 3.7 in English ones. Figure 4.3 shows the manual summaries generated for a topic (cluster) from the RepLab Summarization dataset.

```
<cluster label="K-Pax" priority="mildly_important">
    <tweet id="237940080940023809" date="Tue Aug 21 17:51:50 CEST 2012"
        followers="1835" polarity="positive"> Volvo and K-Pax: Changing the
        definition of a race car: When we’re not feverishly pounding the
        keyboards here at... http://bit.ly/OVoFCJ </tweet>
    <tweet id="267133443899539456" date="Sat Nov 10 06:15:50 CET 2012"
        followers="83" polarity="positive"> Goodluck #RobertThorne at
        qualifying tomo! Lets go @KPAXracing @kpaxracingllc @volvocarsus
        @volvo_racing</tweet>
    <summary
      abstract_EN="Race car made by Volvo and K-Pax"
      abstract_ES="Coche de carreras de Volvo y K-Pax"
      extract="237940080940023809"/>
</cluster>
```

Figure 4.3: Example summaries for a RepLab topic referring to Volvo

4.6.2 Experiments

In this section we describe the summary generation process, and the different experimentation scenarios.

4.6.2.1 Summary generation system (SGS)

In this section we describe the steps followed by our system since it receives the input documents (tweets) and parameters until it generates the final summary. The summary generation system (from now on SGS) generates extractive summaries shaped like rankings of tweets. Figure 4.8 illustrates each step of the system.

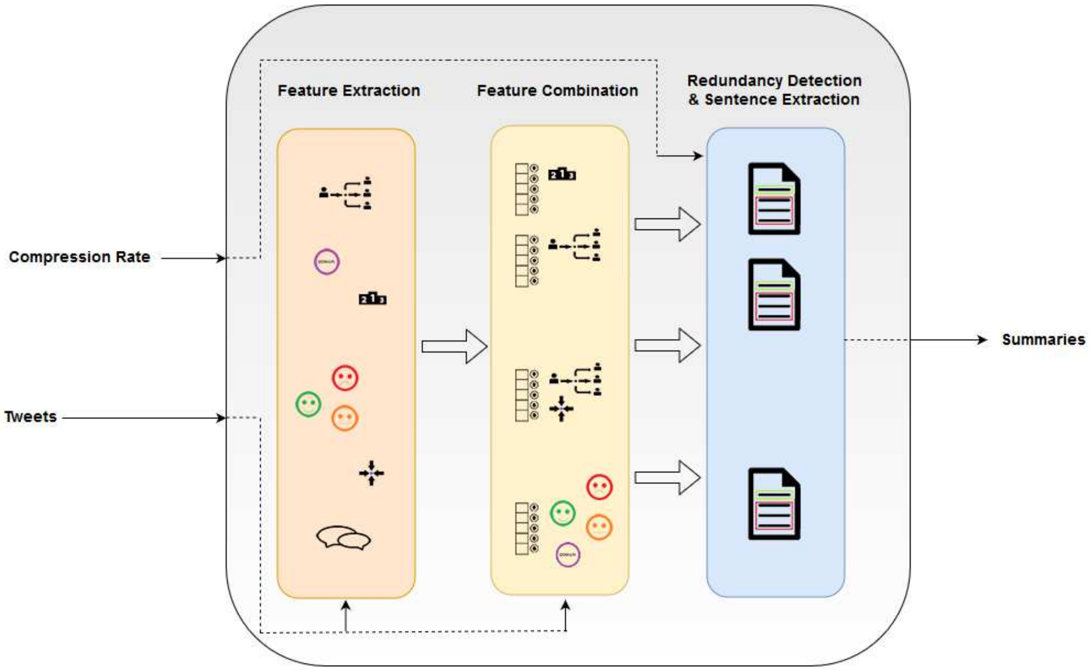


Figure 4.8: Summary Generation System schema

The architecture of the system is formed by three modules: the first one extracts signals from the input tweets. Then, the second module uses the previous signals, or a combination of them, to create different rankings of tweets. Finally, the third module extracts the top-k tweets from each ranking and creates summaries without redundancy until the input compression rate, which indicates the desired length of the summary, is achieved. As the system output, an extractive summary is retrieved. We provide, in detail, the description of the modules below:

Signal extraction. The first step in our system is to extract several signals of interest from a given set of tweets. Signals extracted are those presented in sections 4.2 to 4.5.

Signal combination. In this step, the system receives the signals generated previously in the *signal extraction* step along with the input tweets and generates, as output, different rankings, as many as combinations of signals the algorithm performs. If the system selects two or more signals to arrange tweets, it applies a previous Borda voting (Saari, 1999) step, this method finds an unique sorting signal which is a combination of the original ones. The output of this step is a ranking of tweets sorted in descending order of priority.

Redundancy detection and sentence extraction. Since the input information proceeds from multiple tweets, it is necessary to detect and remove the information that is already included in the summary, in other words, we need to detect and remove redundant information. The redundancy algorithm includes tweets from the rankings generated in the *signal combination* step, according to their position in it, only if the vocabulary overlap between the tweet selected and each one of the tweets already included in the summary is less than a similarity threshold. This threshold has been experimentally set to 0.02. We experiment with two different similarity functions that differ in how the terms are weighted:

- **Jaccard:** It computes the Jaccard similarity (Jaccard, 1901) between the set of (unweighted) terms.

$$Jaccard(Tw_1, Tw_2) = \frac{|Tw_1 \cap Tw_2|}{|Tw_1| + |Tw_2| - |Tw_1 \cap Tw_2|} \tag{4.1}$$

- **LIN:** It computes a variant of LIN similarity (Lin et al., 1998). Instead using the set of unweighted terms as *Jaccard*, it uses the tf-idf of each word (w) calculated using all texts of the dataset.

$$LIN(Tw_1, Tw_2) = \frac{\sum_{w \in (Tw_1 \cap Tw_2)} -log(tfidf(w))}{\sum_{w \in (Tw_1)} -log(tfidf(w)) + \sum_{w \in (Tw_2)} -log(tfidf(w))} \tag{4.2}$$

Next, the system extracts the top k tweets from the ranking to be included in the summary. This parameter, k, is calculated from the compression rate provided by the user and indicates the number of tweets, from the entire input set, that must be included in the final summary. Once the draft summary is created, the system checks if its length is shorter than desired; in this case, discarded tweets are reconsidered and included in the summary by recursively increasing the threshold in 0.02 until the desired compression rate is reached.

4.6.2.2 Experimentation scenarios

In this section, we explain the different experimental scenarios that we have built to generate extractive summaries automatically. Each of the following experiments takes its name based on the different signals that intervene in the generation of the ranking of tweets.

- *Authority and domain scenario:* uses the authors' authority and domain signals explained in section 4.2:

    1. **Authority:** we only use the *Authority* signal to generate the summary. This signal estimates the likelihood of a given profile to be an authority.
    2. **Domain:** we only use the *Domain* signal to generate the summary. This signal estimates the degree of domain knowledge that users have.
    3. **Authority+Domain:** we use the Learning to Rank (L2R) algorithm to combine *Authority* and *Domain* signals, the score obtained is used as a ranking signal.

- *Priority scenario:* incorporates the signals described in section 4.3 for priority detection to the different cases explained in the *authority and domain scenario*.

    1. **Authority+Priority:** *Authority* and *priority* signals are combined by using a previous Borda voting step (Saari, 1999). This voting mechanism generates a final signals which is a combination of the input ones and we use it to rank the tweets.
    2. **Domain+Priority:** *Domain* and *priority* signals are combined by using a previous Borda voting step, as previously explained.
    3. **Authority+Domain+Priority:** *Authority*, *Domain* and *priority* signals are combined by using Borda.

- *Polarity scenario:* here we want evaluate the effect of polarity on the summary generation. For doing this, we substitute the indirect polarity signals, described in section 4.3, with those calculated using the polarity classification algorithm, based on PMI, explained in section 4.3.1. Furthermore, we want to check the compatibility of both polarity calculation approaches.

    1. **Authority+Domain+Priority+PMIPolarity:** we use the L2R score of the *Authority* and *Domain* signals, the priority signals in section 4.3 without the indirect polarity indicators and the signal calculated in section 4.3.1. Signals are combined using a previous Borda voting step, as usual.

2. **Authority+Domain+Priority+BothPolarities:** we use the L2R score of the *Authority* and *Domain* signals, the priority signals in section 4.3 and the signal calculated in section 4.3.1. Signals are combined using Borda.

- *Centrality scenario:* here we use information related to the centrality for the automatic generation of summaries, and calculate centrality using word embeddings. We test two different solutions explained in section 4.4.

  1. **Seq2Seq:** we calculate the centrality signal using the *seq2seq* approach explained in section 4.4.1. This signal is used to produce a ranking of tweets.
  2. **Doc2Vec:** we calculate the centrality signal using the *doc2vec* approach explained in section 4.4.2. This signal is used to generate the ranking of tweets.

- *Topic scenario:* the objective here is to exploit topic information in order to generate summaries. We combine this information along with *authority* and *domain* signals using Borda, as we explained previously.

  1. **Authority+ManualTopics:** for each topic/cluster of each entity in the dataset, a priority is assigned which is given by the *authority* signal. The selection of the highest priority tweet of the cluster is done as explained in section 4.6.2.1.
  2. **Domain+ManualTopics:** for each topic/cluster of each entity in the dataset, a priority is assigned which is given by the *domain* signal. The selection of the highest priority tweet of the cluster is done as explained in section 4.6.2.1.
  3. **Authority+Domain+ManualTopics:** for each topic/cluster of each entity in the dataset a priority is assigned that is given by the L2R score of the *Authority* and *Domain* signals. The selection of the highest priority tweet of the cluster is done as in previous experiments.
  4. **Authority+Topic:** dataset tweets are grouped into clusters using the method seen in the section 4.5.2. For each cluster, a priority is assigned by the *authority* signal. The selection of the highest priority tweet of the cluster is done as previously.

5. **Domain+Topic:** dataset tweets are grouped into clusters using the method seen in the section 4.5.2. For each cluster, a priority is assigned by the *domain* signal. The selection of the highest priority tweet of the cluster is done as previously.
6. **Authority+Domain+Topic:** dataset tweets are grouped into clusters using the method seen in the section 4.5.2. For each cluster, a priority is assigned by the L2R score of the *Authority* and *Domain* signals. The selection of the highest priority tweet of the cluster is done as usual.

- *Best case scenario:* this strategy combines the best signals of the scenarios: *Authority*, *Domain*, *Priority* and *Seq2Seq* or *Doc2Vec* signals. The signal combination is done by using Borda, as usual.

4.6.3 Metrics

Since the purpose of a summary is to obtain the same main information as the input documents but in a shorter way, we must select a metric that is more recall-oriented than precision-oriented, for this reason we use **Recall-Oriented Understudy for Gisting Evaluation** (ROUGE) (Lin, 2004) metric to evaluate our task. ROUGE compares automatically generated summaries against a set of reference summaries (typically created by humans) and computes a series of metrics, including the following:

- **ROUGE-N:** it measures the overlapping of n-grams between the system and reference summaries. Its most widely used variations are:
  - **ROUGE-1:** measures the overlap of 1-gram (each word).
  - **ROUGE-2:** measures the overlap of 2-grams (bigrams).
- **ROUGE-L:** based on Longest Common Subsequence (LCS) (Lin and Och, 2004) statistics. LCS measures level structure similarity and identifies the longest co-ocurring n-gram sequences.
- **ROUGE-W:** based on weighted LCS statistics which favors consecutive LCSes (Longest Common Sequences).

- **ROUGE-S:** based on skip-bigram (Lin and Och, 2004) co-occurence statistics. Unlike bigrams, where the two words are consecutively found in the sentence, skip-bigrams contain gaps between the chosen words in the sentence, thus some words are omitted.

- **ROUGE-SU:** this final variation merges skip-bigram and unigram-based co-occurence statistics.

From all ROUGE variants, we selected ROUGE-2 due to its high correlation with human judges shown in many test collections. In our case, the evaluation is carried out by comparing our system outputs against both, extractive and abstractive manual summaries provided by the RepLab Summarization dataset (see section 4.6.1). This evaluation was done using ROUGE 2.0 tool (Ganesan, 2015).

4.6.4 Baselines

We have collected different baselines summaries, using different compression rates (5, 10, 20 and 30 %), for comparing our results.

- **Followers**: this signal is a basic indicator of priority because things said by people followed by a high number of users are more likely to be spread all over the network. In this baseline, tweets are ranked according to the number of followers of the author who wrote them. The baseline summary is built by choosing the top ranked tweets until the compression rate is reached.

- **LexRank**: this algorithm (Erkan and Radev, 2004b), is one of the most popular centrality-based methods for multi-document summarization, and for this reason, we use it as baseline. The algorithm uses a graph, where the nodes are the candidate sentences to be included in the summary, and two nodes are connected if the similarity between them is above of a given threshold. Once the graph is built, the system finds the most central sentences performing a random walk on the graph. Here we expect that the algorithm captures those tweets that are more relevant to the entity and use them as a summary.

- **SSV-priority**: this baseline system uses the signals showed in table 4.3 to produce a ranking of all tweets for a given test case (an entity), and we then combine the rankings using Borda algorithm. We refer to this baseline as SSV.

- **L2R-priority**: this baseline uses the same initial set of signals than *SSV*. L2R approach makes use of a machine learning (ML) algorithm (we have evaluated several ML algorithms and finally selected random forest (Breiman, 2001)) and an optimization function in order to generate several rankings with the aim of maximizing the optimization function (here we optimize nDCG metric due to its similarities with the evaluation of the proposed problem). We refer to this baseline as L2R.

4.7 Results and discussion

In this section we present the results of the experiments and discuss such results. The following tables show the results according to the summaries used as reference summaries (extractive or abstractive) and according to the similarity function used (Jaccard and LIN) for removing redundancy.

Table 4.5 shows the results of the different experiments when (i) extractive and abstractive manual summaries are used for evaluation (see description of the RepLab Summarization corpus in section 4.6.1), (ii) different compression rates are used, and (iii) the Jaccard coefficient is used to remove redundancy.

- If we analyse the results of the *authority and domain scenario*, we can see that the domain signal improves the authority signal for all compression rates and for both types of reference summaries. This seems to indicate that due to that we are in specialized domains, people with some knowledge about the domain concern more to the clients, because their specialized opinion is more valuable and more valued for the general public and, therefore, is more likely to cause reputational damages. One example of specialized domain is banking. Here the clients, i.e. financial institutions, are interested to know the opinion of economic gurus (such as, the President of the International Monetary Fund) because their messages could vary the global economy and affect their investments.

	ROUGE-2							
	Extractive				***Abstractive***			
	Compression Ratio				***Compression Ratio***			
	5%	*10%*	*20%*	*30%*	*5%*	*10%*	*20%*	*30%*
Authority	0.12	0.21	0.36	0.48	0.09	0.16	0.28	0.35
Domain	0.15	0.24	0.41	0.52	0.11	0.18	0.29	0.36
AuthorityDomain	0.14	0.24	0.40	0.53	0.10	0.18	0.29	0.38
AuthorityPriority	0.15	0.24	0.40	0.51	0.12	0.17	0.30	0.37
DomainPriority	0.16	0.25	0.42	0.53	0.12	0.18	0.30	0.37
AuthorityDomainPriority	0.18	0.26	0.43	0.54	0.12	0.18	0.30	0.38
AuthorityDomainPriorityPMIPolarity	0.17	0.26	0.41	0.52	0.11	0.18	0.30	0.38
AuthorityDomainPriorityBothPolarity	0.16	0.26	0.41	0.52	0.11	0.18	0.31	0.38
Seq2Seq	0.21	0.31	0.46	0.59	0.10	0.15	0.21	0.26
Doc2Vec	0.14	0.23	0.35	0.48	0.10	0.17	0.27	0.35
AuthorityManual	0.30	0.55	0.63	0.65	0.18	0.33	0.40	0.43
DomainManual	0.30	0.55	**0.72**	**0.73**	0.18	0.33	**0.41**	**0.44**
AuthorityDomainManual	0.30	0.55	0.62	0.65	0.18	0.33	0.39	0.42
AuthorityTopic	**0.36**	**0.64**	**0.72**	**0.73**	**0.20**	**0.35**	**0.41**	0.43
DomainTopic	**0.36**	**0.64**	**0.72**	**0.73**	**0.20**	**0.35**	**0.41**	**0.44**
AuthorityDomainTopic	**0.36**	**0.64**	**0.72**	**0.73**	**0.20**	**0.35**	**0.41**	0.43
BestCaseScenario	0.14	0.23	0.38	0.50	0.10	0.17	0.28	0.36
Baseline-LexRank	0.20	0.29	0.40	0.50	0.09	0.12	0.17	0.22
Baseline-Followers	0.19	0.31	0.49	0.60	0.09	0.15	0.23	0.28
Baseline-SSV	0.24	0.36	0.52	0.64	0.12	0.17	0.25	0.30
Baseline-L2R	0.18	0.28	0.45	0.57	0.09	0.14	0.22	0.27

Table 4.5: Jaccard Results

- This problem of the authority signal is hindered in the case of mixing it with the domain signal. In the case of high ratios, 30%, the combination of both signals offers a better summary.

- From the analysis of the *priority scenario*, we can see an improvement in the results with the inclusion of priority signals, and this is true for both types of evaluation (abstractive and extractive). This reinforces the idea of the importance of the content being written, the polarity of what is written, and the users joining the conversation, for example. In this scenario, the priority signals, make users belonging to the domain appear upper in the tweets ranking. In abstractive summaries results are quite similar. The combination of authority and domain with priority signals, remains the best choice when it comes to create summaries, for all compression rates.

- The results from *polarity scenario* shows that the inclusion of the new improved signal, based on the occurrences of opinionated terms, worsens the previous results that use less complex polarity signals. This situation may be due to the very nature of tweets. These texts present generally misspellings words, lack of the appropriate syntax and, sometimes, do not contain any sentiment word although they may impact the entity's reputation.

- The *centrality scenario* shows the results of the different text vectoring systems explained in section 4.4. It may be seen that while seq2seq is more favourable in the evaluation with extractive models, doc2vec behaves better in the evaluation against abstractive. This is due to the way that vectors are generated. Since seq2seq is generated sequentially, each word depends on its previous one and, therefore, the vocabulary represented by the vector is similar to that which appears in a tweet. On the contrary, doc2vec system generates the vectors without taking into account the previous words, therefore, it will promote the appearance, in the summary, of coherence tweets which increase the probability of find overlapping bigrams between the system and the reference summaries.

- The results of our final approach, the *topic information scenario*, show that it is crucial to know the topics that people with some knowledge about the domain talk about, because their specialized opinion is more valuable and valued for the general public and is more susceptible of creating reputation crisis, using both approaches: manual or learning similarity functions for topic detection. Note that the topic detection approach behaves better than the clusters provided by the manual annotation of the dataset, in general. This is in line with the results in (Spina et al., 2014), which demonstrate that their system is close to the inter-annotator agreement rate.

- The *BestCaseScenario* mixes the scenario with better results (in our case it is the second with the fusion of authority, domain and priority signals) along with the best way to vectorize texts (seq2seq for the evaluation against extractive, and doc2vec in the evaluation against abstractive). Including centrality information produces worse results than the best scenario alone, which leads us to think that centrality is not as crucial as knowing the authority of the user, the knowledge about the domain that certain profile has or the priority of the issues that may affect the clients when it comes to generate summaries automatically.

Figure 4.9 compares the results for the Jaccard similarity measure when summaries are evaluated against extractive and abstractive models, for a compression rate of 10%. Although higher rate summaries show better results, we consider 10% a reasonable compression rate as it includes enough information about the entity to be summarized and is also easier for a human operator to read. It is observed that the combination of the information about the authority and the domain knowledge of the users along with the topic, is crucial to create reputational summaries, because the opinions of the influencers are more valuable for the general public and, therefore, more susceptible to create reputational crisis.

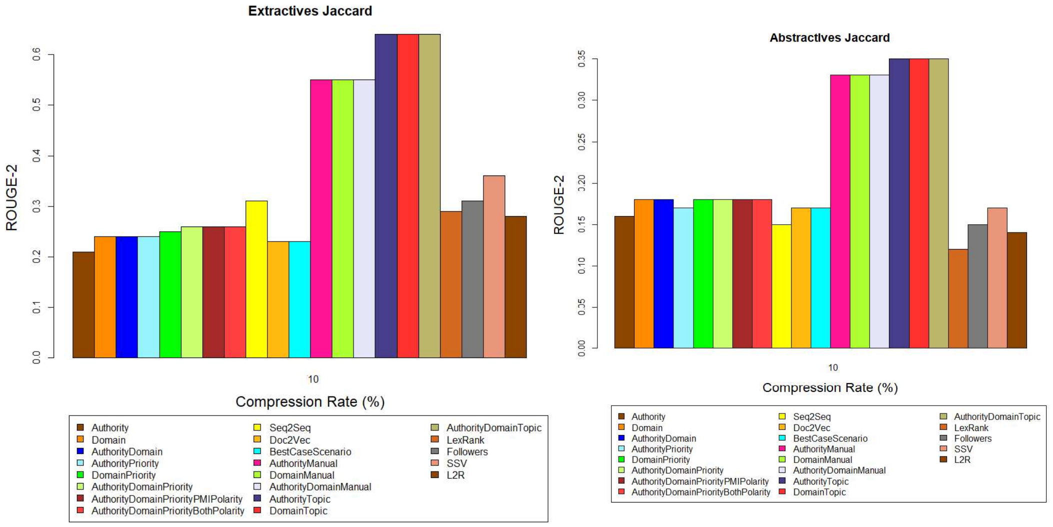


Figure 4.9: Jaccard similarity results at 10% for evaluation against extractive (left image) and abstractive (right image) models

Table 4.6 shows the results of the different experiments when (i) both extractive and abstractive manual summaries are used for evaluation (see description of the RepLab Summarization dataset 4.6.1), (ii) different compression rates are used, and (iii) the LIN coefficient is used to remove similarity.

- Analysing the results of *authority and domain scenario*, we can see that the domain signal improves the authority signal for all compression rates and for both types of reference summaries, as in Jaccard. This situation reinforces our hypothesis, that in specialized domains, people who have better background knowledge about the domain are more important for the clients than global authorities because the first kind of users have a more specialized opinion of the domain and, therefore, are more likely to cause reputational damages within a domain. The combination of the authority

	ROUGE-2							
	Extractive				***Abstractive***			
	Compression Ratio				***Compression Ratio***			
	5%	*10%*	*20%*	*30%*	*5%*	*10%*	*20%*	*30%*
Authority	0.13	0.23	0.40	0.51	0.10	0.18	0.30	0.38
Domain	0.15	0.27	0.44	0.54	0.10	0.19	0.32	0.40
AuthorityDomain	0.14	0.27	0.42	0.53	0.10	0.20	0.31	0.39
AuthorityPriority	0.15	0.27	0.44	0.54	0.11	0.19	0.30	0.38
DomainPriority	0.16	0.28	0.41	0.52	0.11	0.19	0.30	0.39
AuthorityDomainPriority	0.16	0.27	0.43	0.53	0.11	0.20	0.31	0.39
AuthorityDomainPriorityOnePolarity	0.15	0.28	0.42	0.55	0.10	0.20	0.31	0.39
AuthorityDomainPriorityBothPolarity	0.15	0.26	0.43	0.53	0.10	0.19	0.32	0.38
Seq2Seq	0.12	0.23	0.38	0.50	0.10	0.17	0.29	0.37
Doc2Vec	0.13	0.24	0.40	0.51	0.10	0.18	0.29	0.37
AuthorityManual	0.30	0.55	0.63	0.65	0.18	0.33	0.40	0.43
DomainManual	0.30	0.55	**0.72**	**0.73**	0.18	0.33	**0.41**	**0.44**
AuthorityDomainManual	0.30	0.55	0.62	0.65	0.18	0.33	0.40	0.42
AuthorityTopic	**0.36**	**0.64**	**0.72**	**0.73**	**0.20**	**0.35**	**0.41**	0.43
DomainTopic	**0.36**	**0.64**	**0.72**	**0.73**	**0.20**	**0.35**	**0.41**	**0.44**
AuthorityDomainTopic	**0.36**	**0.64**	**0.72**	**0.73**	**0.20**	**0.35**	**0.41**	0.43
BestCaseScenario	0.14	0.25	0.41	0.51	0.10	0.19	0.31	0.38
Baseline-LexRank	0.20	0.29	0.40	0.50	0.09	0.12	0.17	0.22
Baseline-Followers	0.19	0.31	0.49	0.60	0.09	0.15	0.23	0.28
Baseline-SSV	0.24	0.36	0.52	0.64	0.12	0.17	0.25	0.30
Baseline-L2R	0.18	0.28	0.45	0.57	0.09	0.14	0.22	0.27

Table 4.6: LIN Results

and domain signals do not improve the results compared with the use of the domain signal alone.

- Regarding the *priority scenario*, its results show an improvement in relation to the results of the *authority and domain scenario*, as with the Jaccard similarity. The combination of authority, domain and priority signals still remains as the better option when we compare the summaries produced by them against the abstractive reference models but, when we compare against the extractive manual summaries, sometimes the best choice is to combine just authority and priority signals.

- According to the relative minimal improvement of the results obtained for the *polarity scenario*, the development of a more complex algorithm to collect more refined polarity signals from tweets is not justified. With a simpler polarity estimators (e.g. the number of negative words) we obtain practically the same results. This idea is aligned with the conclusion extracted from analysing the Jaccard results.

- Analysing the results of the *centrality scenario* we can see that the doc2vec approach is the best choice using LIN function. This situation is due because doc2vec generates vectors without using the context that surrounds the words that are part of the tweet and, because the LIN function prioritizes the use of terms individually.

- The results given by the *topic information scenario* show that the combination of the information about people with some knowledge about the domain and with the topic of the conversations provides the best results. Note that, again, the topic detection approach behaves better than the clusters provided by the manual annotation of the dataset, in general.

- The results for the *BestCaseScenario*, which combines authority, domain, priority and centrality (calculated using doc2vec) signals, indicate that the comparison with extractive references does not improve the baselines but, on the contrary, we overcome the results of the baselines in the evaluation against abstractive manual summaries. In both cases, the performance of this scenario falls substantially if we compare it with the *authority and domain*, *priority* and *polarity* cases but it improves the *centrality* scenario. According to the results, our conclusion here is that centrality is, again, not as important as the signals of authority, domain and priority.

Figure 4.10 summarizes the results for the LIN similarity measure in the evaluation against extractive and abstractive models, for a compression rate of 10%.

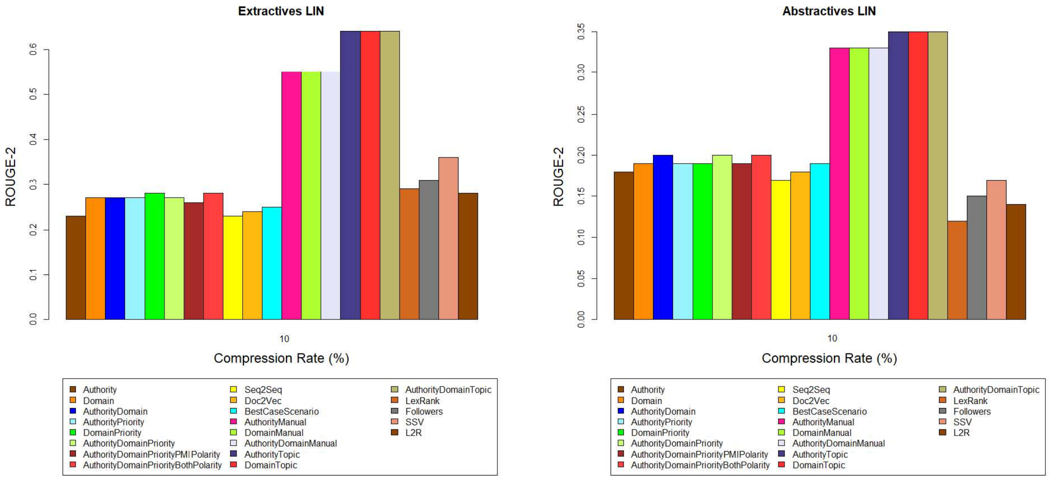


Figure 4.10: LIN similarity results at 10% for evaluation against extractive (left image) and abstractive (right image) models

It is observed in the images that the combination of the information about the authority of the users along with the topic, again, is crucial to create reputational summaries. Comparing the results obtained for our proposals with the baselines, we extract the following conclusion: it is important to include in the summary, those topics of conversation that influencers (global or domain) talk about because their opinions may contribute to change the general public's perception of the entities (companies, products, people, etc.) and cause important reputation damages.

Table 4.7 shows a comparison between the best results of Jaccard, for each scenario, and its comparison with LIN. As we can see, Jaccard overcomes LIN only in summaries with low compression rate and in isolate cases, but in general, LIN works better than Jaccard.

Figure 4.11 shows the best results for both the extractive and abstractive evaluations, for a compression rate of 10%, only for LIN and the best configuration of each family of experiments.

	ROUGE-2							
	Extractive				*Abstractive*			
	Compression Ratio				*Compression Ratio*			
	5%	*10%*	*20%*	*30%*	*5%*	*10%*	*20%*	*30%*
Domain Jaccard	0.15	0.24	0.41	0.52	0.11	0.18	0.29	0.36
Domain LIN	0.15	0.27	0.44	0.54	0.10	0.19	0.32	0.40
AuthorityDomainPriority Jaccard	0.18	0.26	0.43	0.54	0.12	0.18	0.30	0.38
AuthorityDomainPriority LIN	0.16	0.27	0.43	0.53	0.11	0.20	0.31	0.39
AuthorityDomainPriorityOnePolarity Jaccard	0.17	0.26	0.41	0.52	0.11	0.18	0.30	0.38
AuthorityDomainPriorityOnePolarity LIN	0.15	0.28	0.42	0.55	0.10	0.20	0.31	0.39
BestCaseScenario Jaccard	0.15	0.26	0.40	0.52	0.11	0.19	0.31	0.39
BestCaseScenario LIN	0.15	0.27	0.42	0.53	0.10	0.19	0.31	0.40
DomainTopic-Jaccard	0.36	0.64	0.72	0.73	0.20	0.35	0.41	0.44
DomainTopic-LIN	0.36	0.64	0.72	0.73	0.20	0.35	0.41	0.44

Table 4.7: Comparison between Jaccard and LIN results

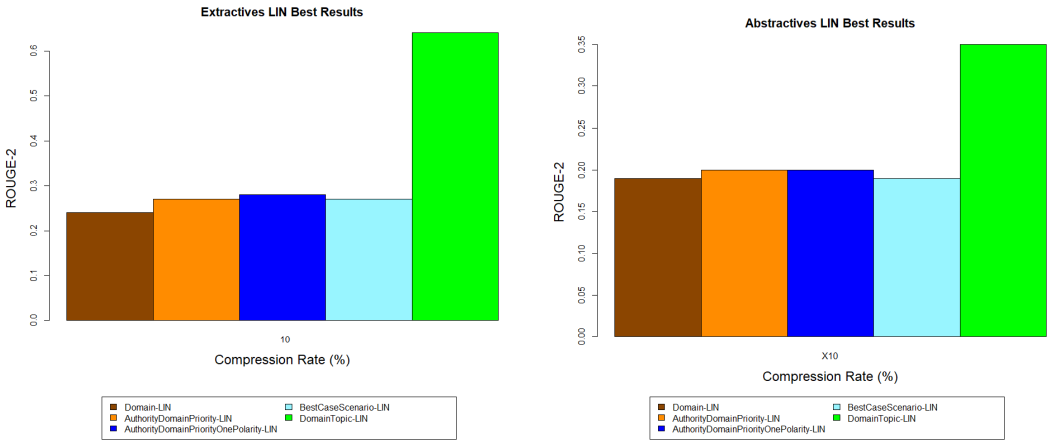


Figure 4.11: Best LIN results at 10% for evaluation against extractive (left image) and abstractive (right image) models for the best configuration of each family of experiments

The figures show that, once again, the combination of signals related to the authority and topic information obtains the best results to automatically generate reputation summaries.

Finally, the table 4.8, shows the best system for each scenario.

	ROUGE-2							
	Extractive				***Abstractive***			
	Compression Ratio				***Compression Ratio***			
	5%	*10%*	*20%*	*30%*	*5%*	*10%*	*20%*	*30%*
Domain	0.15	0.27	0.44	0.54	0.10	0.19	0.32	0.40
AuthorityDomainPriority	0.16	0.27	0.43	0.53	0.11	0.20	0.31	0.39
AuthorityDomainPriorityOnePolarity	0.15	0.28	0.42	0.55	0.10	0.20	0.31	0.39
BestCaseScenario	0.15	0.27	0.42	0.53	0.10	0.19	0.31	0.40
DomainTopic	**0.36**	**0.64**	**0.72**	**0.73**	**0.20**	**0.35**	**0.41**	**0.44**
Baseline-LexRank	0.20	0.29	0.40	0.50	0.09	0.12	0.17	0.22
Baseline-Followers	0.19	0.31	0.49	0.60	0.09	0.15	0.23	0.28
Baseline-SSV	0.24	0.36	0.52	0.64	0.12	0.17	0.25	0.30
Baseline-L2R	0.18	0.28	0.45	0.57	0.09	0.14	0.22	0.27

Table 4.8: Best Results for each case of study

For all the scenarios we show the results given by LIN similarity because it have a better performance than the Jaccard similarity. The results provided by combining the domain signal with topic detection using learning similarity functions overcome, systematically, baseline results for both the extractive and abstractive evaluations.

Figure 4.12 shows graphically the results in table 4.8 for a compression rate of 10%.

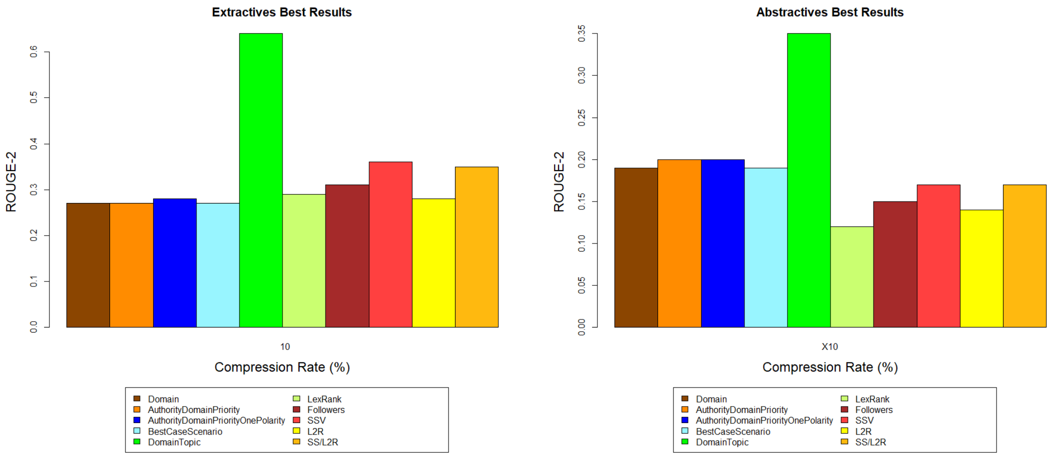


Figure 4.12: Best results at 10% for evaluation against extractive (left image) and abstractive (right image) models

4.8 Conclusions

In this chapter we presented a study about the signals that model, from published tweets, the authority and the knowledge about a certain domain of Twitter profiles and their use to generate reputation summaries. We have developed a Summary Generation System (SGS) to select the most representatives tweets about a given entity (company, person, etc.) exploiting information related to authority and knowledge about domains in conjunction with other information typically used in automatic summarization such as priority, centrality, polarity, etc.

Our experimental results have allowed us to answer the research questions raised previously in the section 4.1.

1. **Research Question 4:** *Can authority and domain signals be effectively exploited in order to create reputation summaries?*

   In this chapter we have shown how authority and domain signals may be used to generate summaries automatically. In the field of marketing, knowing the opinions of influential people is crucial, since they have the ability to convince others and therefore to influence opinions. With this idea in mind, we hypothesize that the signals that shape authority and domain are key to generate better reputation summaries.

Our experiments results indicate that such signals, are useful, mostly, combined with other signals. The results may be negatively affected by the fact that:

- As we indicate in section 4.6.1, due to the lack of textual information provided by the dataset, only one tweet per user, we had to expand it by collecting more tweets per profile during the same period of time, 1st June 2012 to 31st December 2012. Due to the elapsed time from the RepLab crawling time to ours, some profiles are no longer available, approximately a third of the dataset users, so we used the only tweet provided by the dataset to generate their authority and domain signals causing that the generation of these signals is not done in the most appropriate way. However, this cause is not entirely feasible due to the good results obtained when we compare our model against to abstractive manual summaries.

On the other hand, the improvement obtained for the comparison with abstractive manual summaries, indicates that the tweets written by influential users contain content of interest and should be included in the reputational summary.

2. **Research Question 5:** *What role do priority, polarity and centrality play in the generation of reputation summaries?*

   In this chapter we wanted to evaluate other characteristics that have been used traditionally in the task of automatic summarization and in other ORM tasks.

   In the field of online reputation management, it is essential to know which are the different topics that users talk about and how they affect to the entity's reputation: positively, negatively or neutral. Finally, it is also necessary to know the centrality of the different opinions, in other words, if these ideas are shared by the participants and are central within a topic of conversation.

   In our study we have defined a series of scenarios to include first the priority signals, second a refinement of the polarity signal and finally, the centrality signal based on vectorizing tweets and see how close each tweet is, within a topic, to its centroid. Based on the results presented in the section 4.7, we can see that knowing the priority of the topic of which the influencers speak is fundamental. This priority indicates the relevance of the topics to the entity. As we are generating reputational summaries, this priority measures the negative (reputational alerts) or midly-important impact that topics could have on the reputation to the entities which leads into a loss of prestige in the eyes of its customers.

Moreover, our results show that, with a not very refined polarity (such as the number of negative words or the number of positive emoticons used) we obtain better performance than using a more refined polarity, therefore, this refinement is not necessary. Finally, centrality does not show the desired impact with respect to other signals such as those that measure authority, domain knowledge and priority.

3. **Research Question 6:** *What is the performance of different similarity functions for avoiding redundancy?*

   In this chapter we studied two different similarity functions, Jaccard and LIN, for avoiding redundancy in the automatic summary. While to calculate the first of these options, Jaccard, we only need to check the number of words that are common and not common to the input sets, the second option, LIN, means to capture the specificity of the terms with respect to the entity of interest. By including this specificity in the similarity calculation, we take into account the relevance of the words with respect to the entity and, therefore, duplicate content of the entity will be removed from the summary. For example, in two tweets where many words irrelevant to the entity appear, common to both tweets, and some terms of interest to the entity that are not repeated, Jaccard will remove one of them for having redundant content while LIN will retain both for having few specific terms in common.

   In the case the of the reputation summaries, it is important that everything that is relevant to the entity appears in order to have, to a large extent, as much knowledge as possible about the issues that may affect it. This is reflected in the results presented in the section 4.7, where LIN similarity systematically improves Jaccard's results. For this reason, it seems better to use the LIN variant instead of Jaccard for removing redundancy.

4. **Research Question 7:** *How can we use topic information to create reputation summaries?*

   In this chapter we introduced two different ways of creating summaries. In the first method, we generate a ranking with the tweets that refer to an entity and, through a system of tweet selection, we choose the texts according to the order of appearance in the ranking, eliminating those that are redundant. In the second method, we generate clusters of tweets to which a priority is assigned, being the higher priority clusters those that have greater probability of appearing in the final summary, ranking the tweets within each cluster also according to their priority to the topics, and selecting one or more tweets from each cluster to be part of the final summary.

   The results obtained for both approaches show that the second method improves significantly the results of the manual clustering for any combination of authority signals used.

CHAPTER 5 CONCLUSIONS AND FUTURE WORK

He finished the picture yesterday noon. Now he looks at it detail by detail

C.P. Cavafis-Picture of a 23-year-old painted by his friend of the same age, an amateur

This final chapter concludes this doctoral Thesis. In this Thesis, we have investigated the process of creating automatic reputation reports and how it differs from conventional summarization, and we have investigated the problem of automatically detecting influencers as a preliminary enabling step to produce high-quality reputation reports. Therefore, we have addressed two main tasks: (i) the detection and characterization of influencers in Social Networks (specifically Twitter), and (ii) the automatic generation of reputation reports. In the first task, we have tested different signals extracted from the Twitter profiles and from their tweets and explored the information provided by a characterization of the followers. In the second task, we have created reputation reports by automatically extracting those tweets that are more relevant to the entities from a reputational point of view. This relevance is calculated from the signals that measure the authority and the domain knowledge of a user (calculated in the first task) along with other state-of-the-art signals from the automatic summarization task (such as centrality, polarity, etc.)

This chapter is divided into the following sections: first, in section 5.1 we summarize the main contributions of this Thesis. Second, in section 5.2 we present some limitations of our work. Third, in section 5.3 we discuss the main conclusions of this Thesis. Finally, in section 5.4 we summarize open issues and future research lines.

5.1 Main contributions

As part of this Thesis we have produced several useful contributions for the research community:

1. **An exhaustive study of the relative importance of authority and domain signals for the identification and characterization of users on Twitter along with other signals extracted from Twitter profiles:** we study the usefulness of signals of different nature, extracted from the users profiles (such as the number of published tweets, the number of followers, etc.) and from textual content, to locate and characterize influencers. We have analyzed the authority and domain knowledge of the users. This study is useful since the nature of the influencers is not unique, we have different types of influencers: (i) people whose authority is restricted to a certain domain because they posses knowledge about that domain or (ii) people whose authority transcends to other domains. Our experimental results indicate that both Twitter metadata and the user's textual content provide useful characterization signals, but ultimately the textual signal, modeled adequately, is enough to get optimal or near-optimal results. Text can be used to identify authority traits and also domain pertenence. Both are useful to characterize influencers, but ultimately the supervised authority signal implicitly contains also domain information and is enough to find influencers in a given domain.

2. **Different strategies to characterize and identify a profile on Twitter:** two different approaches have been studied to identify an influencer on Twitter: using the information coming from the profile and using the information given by the environment of the profile we want to identify. We have seen that: (i) for users with an important number of followers, what they and their followers say is significantly more powerful than the number

of followers and any other Twitter authority indicator; (ii) the followers of a user may provide useful information for her characterization.

3. **An exhaustive study of different ways to generate a ranking:** in this work we have experimented with four different ways to generate a ranking: by using signals directly (DSR), using the confidence of the classifier (CR), a mixture of both (DSCFR) and learning to rank (L2R). Each of these forms is of a different nature, e.g. unsupervised for DSR or supervised for L2R, and we have tested the ideal candidate for the detection of influencers. Our results indicate that the method used to generate rankings is less relevant than having powerful textual signals (a particular form of language models in our study).

4. **Creation of an annotated test collection for the task of producing entity-oriented reputation reports from Twitter data.** Starting from RepLab 2013 manual annotations, which provided topics, reputational polarity and reputational priority for companies in the banking and automotive domains, we have created a dataset with manually generated reputation reports for all those entities. The dataset includes extractive and abstractive summaries both in English and Spanish, and has been key to our experimental studies and is a valuable resource for future research on this topic.

5. **Study of the impact of using information about users' authority and domain knowledge in the automatic generation of reputation reports:** given that influencers' opinions are potential threats to the entities' reputation, since they have the capacity to engage many people, they are good candidates to appear in the reputation report. In our study, we have used those signals that model the influence of a user along with other well-known signals in the state-of-the-art of the automatic summarization task (centrality, polarity, etc.) to automatically generate reputation reports. We have seen that: (i) the authority and domain knowledge signals are useful in combination with other signals; (ii) knowing the priority of the topic of which the influencers speak is fundamental, and (iii) it is important to include in the summary those topics that influencers talk about, because their opinions may contribute to cause important reputational damage.

5.2 Limitations

Despite the results obtained in our experimentations, our work have some limitations. As we mentioned in chapter 3, the sample data used in the detection of influencers come from only two domains: automotive and banking. We have found that there is substantial variability across domains, and therefore the results must be confirmed in other domains. Second, the location of influencers using the information from followers is not complete, since it has not been possible to recover information from some of the followers due to the time elapsed between the crawling time of the RepLab dataset and our study. Third, we focused on studying supervised approaches (our only unsupervised approach is combining signal ranks via voting) so, the difference between unsupervised and supervised approaches might stretch with more sophisticated unsupervised mechanisms.

Regarding the generation of reputation reports, the main limitation is given by the lack of a user evaluation that allows us to assess the real usefulness of these reports. The extractive summaries generated in our experimentation may have two types of users: as a final report to be delivered to the entity of interest, the user is the entity itself (for instance, the PR department of a company); as an intermediate product, in can be used by reputation experts to produce manual quality reports.

The usefulness of these reports is also given by different factors such as the coverage of information, its presentation, etc. Moreover, it is an incomplete report that should be completed in addition to the extractive summary, with aggregated information such as statistics (e.g. number of negative/positive comments collected), graphs illustrating these statistics, etc.

5.3 Conclusions

Throughout this Thesis, we have worked with the notion of the authority and domain knowledge that users have on Social Networks. This authority and domain knowledge are important, from the point of view of the ORM process, because users with a great ability to convince a lot of people to adopt their ideas can harm an entity reputation if their opinions about her are negative since, as we have said, they are able to transmit their message among large number of people. For this reason, reputation experts must locate and characterize users who have the most authority and domain knowledge in order to avoid and deactivate, as far as possible, reputational alerts. In our work, we have carried out an extensive study of different signals that can help to identify influencers (see chapter 3) on Twitter, and we have determined a way to locate them through the texts they publish. The use of signals extracted from their profiles is not crucial when these profiles have high number of followers: in this case, knowing the content written is a great help to discriminate influencers from non-influencers users. On the other hand, we have determined the nature of the followers of the different profiles in order to identify these main profiles as influencers or not. From the results obtained in this approach, we observe that the information extracted from the followers play a essential role for identify influencers, those profiles followed by influencers, are more likely to be influencers. We have also combined the information regarding to the main profiles with the information extracted from the followers but, unfortunately, we have not been able to properly combine this information to perform a better identification of the influencers.

The study of the influence is also our guiding thread in our next task, **automatic generation of reputational reports** (section 4). Our hypothesis here is that the ideas expressed by influencers should take priority in the reputation reports, as they are more likely to provoke a reputational crisis. A reputational report collects and summarizes all the topics, that may affect to the entity, that are discussed in Social Networks. These reports include as much information as possible, in a condensed way, about each of the topics, concerning the entity, that users of Social Networks talk about. This task is essential for ORM since the large amount of information flowing through the Internet makes it impossible for a human expert to process it in a reasonable quantity of time. Along with the signals that determine the influence of users, we used other state-of-the-art signals (such as priority, polarity, centrality or topic information signals) in order to improve the summarization of the contents published and we tested two different ways to create the reports: (a) creating a ranking of tweets according to certain signal or a combination of signals (e.g. authority signal, authority+domain+priority signals, etc.) and selecting those tweets, in order of appearance in the ranking, that include new information to the report (i.e. ensuring that the selected tweet does not have redundant information with the text already included in the report) and (b) grouping the tweets of an entity, in topics according to their content and selecting one tweet per topic. From the results obtained, we have seen that the use of the signals that model the influence, may not be as effective, by themselves, as other state-of-the-art signals. When we combine the influence signals with the priority signals we obtain better results than the previous case, this indicate that is useful to know the relevance of what influencers are saying about the entity not only who is spreading the message. The aggregation of a more refined polarity and centrality signals do not improve previous results, which indicates that these signals are less useful when creating reputational reports. Finally, the use of information regarding the topics along with the influence and priority signals, is the best way to generate reports in an automatic way since the those topics of conversation that influencers talk about may contribute to cause important reputation damages. This reports give an idea to the human experts of the issues that may compromise the reputation of the entity.

5.4 Future lines of work

For each of the tasks carried out in this Thesis, we summarize other interesting research lines of work that we want to tackle in the future. Concerning the first of the tasks, the **identification and characterization of influencers**, we would like to test our method in a different Social Network, such as Facebook, where there is no limit in the length of the texts because, in view of the results obtained with short texts, this method should locate influencers with a better precision with longer texts. We are also interested in applying neural networks to the identification and characterization of influencers with this dataset, where the number of samples is relatively low. Finally, we are interested in predicting the future evolution of the influence of the different profiles. From the point of view of ORM this is crucial, since it could be possible to locate and monitor future influencers which could led to avoid potential threads to the entities before they even would happen.

Regarding the second of our tasks, the **automatic generation of reputation reports**, we want to address the following research opportunities. For the ranking algorithm we want to test a different similarity function that instead of using the common words written in different tweets, as Jaccard or LIN do, uses the meaning of the complete sentence (i.e. BIOSSES (Soğancıoğlu et al., 2017)) because this similarity will focus on what are tweets saying instead of what words compose the tweet and, therefore, it might give a more accurate way to discover similar tweets. For both algorithms showed, ranking and clustering, we want to set the thresholds (such as the overlap between tweets, the optimal compression rate, etc.) in an automatic way, in a similar way to the proposal made by the authors of (Delgado et al., 2017). We think that algorithms should be as human independent as possible, so that using a mechanism that adapts the thresholds automatically according to the input data is essential to reach this goal. For the clustering approach, we want to test the utility of graph methods since they are used in the state-of-the-art to extract the main information of the texts. Our idea here is that, instead of using the most priority tweet to be included in the report, use a sentence generated automatically that combines the information of the different tweets in the topic and including this sentence in the report. This final sentence will have some other information that the most priority tweet may not content and, otherwise, could be lost. Regarding the reputational reports, we want to improve them to include useful statistical information to the entity and to create automatic reports according to different aspects chosen by the users of the system. We also want to improve our reports by adding a recommendation about how to address the reputation crisis. For doing this, we want to create a recommendation system which advices the best strategy to follow, by selecting the most adequate approach from a variety of them. Finally, we want to test the usefulness of these reports by conducting user studies.

BIBLIOGRAPHY

Albaraa Abuobieda, Naomie Salim, Ameer Tawfik Albaham, Ahmed Hamza Osman, and Yogan Jaya Kumar. Text summarization features selection method using pseudo genetic-based model. In *2012 International Conference on Information Retrieval & Knowledge Management*, pages 193–197. Institute of Electrical and Electronics Engineers, 2012.

Abolfazl AleAhmad, Payam Karisani, Masoud Rahgozar, and Farhad Oroumchian. University of tehran at replab 2014. In *Conference and Labs of the Evaluation Forum (Working Notes)*, pages 1528–1536, 2014.

Nasser Alsaedi, Peter Burnap, and Omer Farooq Rana. Automatic summarization of real world events using twitter. 2016.

Enrique Amigó, Jorge Carrillo De Albornoz, Irina Chugur, Adolfo Corujo, Julio Gonzalo, Tamara Martín, Edgar Meij, Maarten De Rijke, and Damiano Spina. Overview of replab 2013: Evaluating online reputation monitoring systems. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 333–352. Springer, 2013.

Enrique Amigó, Jorge Carrillo-de Albornoz, Irina Chugur, Adolfo Corujo, Julio Gonzalo, Edgar Meij, Maarten de Rijke, and Damiano Spina. Overview of replab 2014: author profiling and reputation dimensions for online reputation management. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 307–322. Springer, 2014.

Sinan Aral and Dylan Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, 2012.

R. Lalit Bahl, Raimo Bakis, Peter V. De Souza, and Robert L. Mercer. Obtaining candidate words by polling in a large vocabulary speech recognition system. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 489–492. Institute of Electrical and Electronics Engineers, 1988.

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. Multi-document abstractive summarization using ilp based multi-sentence compression. In *International Joint Conference on Artificial Intelligence*, pages 1208–1214, 2015.

Fabrício Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virgílio Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th Association for Computing Machinery'sSpecial Interest Group on Data Communication conference on Internet measurement conference*, pages 49–62. Association for Computing Machinery, 2009.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 481–490. Association for Computational Linguistics, 2011.

Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(3):259–279, 1986.

Rupal Bhargava, Yashvardhan Sharma, and Gargi Sharma. Atssi: Abstractive text summarization using sentiment infusion. *Procedia Computer Science*, 89: 404–411, 2016.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J Passonneau. Abstractive multi-document summarization via phrase selection and merging. *arXiv preprint arXiv:1506.01597*, 2015.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. Association for Computing Machinery, 1998.

Kalina Bontcheva and Yorick Wilks. Automatic report generation from ontologies: the miakt approach. In *International conference on application of natural language to information systems*, pages 324–335. Springer, 2004.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Rebecca Burn-Callander. Bad reviews and online 'trolls' cost uk businesses up to $30,000 a year. Online Newspaper - The Telegraph, may 2015. http://www.telegraph.co.uk/finance/businessclub/11635195/Bad-reviews-and-online-trolls-cost-UK-businesses-up-to-30000-a-year.html .

Jorge Carrillo-de Albornoz, Enrique Amigó, Laura Plaza, and Julio Gonzalo. *Tweet Stream Summarization for Online Reputation Management*, pages 378–389. Springer International Publishing, Cham, 2016. ISBN 978-3-319-30671-1. doi: 10.1007/978-3-319-30671-1_28.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web*, World Wide Web '11, pages 675–684, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 978-1-4503-0632-4. doi: 10.1145/1963405.1963500. URL http://doi.acm.org/10.1145/1963405.1963500.

Yangbin Chen, Yun Ma, Xudong Mao, and Qing Li. Abstractive summarization with the aid of extractive summarization. In *Asia-Pacific Web and Web-Age Information Management Joint International Conference on Web and Big Data*, pages 3–15. Springer, 2018.

Jackie Chi Kit Cheung and Gerald Penn. Unsupervised sentence enhancement for automatic summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 775–786, 2014.

Su Gon Cho and Seoung Bum Kim. Summarization of documents by finding key sentences based on social network analysis. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 285–292. Springer, 2015.

Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). *ComTech: Computer, Mathematics and Engineering Applications*, 7(4):285–294, 2016.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international Association for Computing Machinery's Special Interest Group on Information Retrieval conference on Research and development in information retrieval*, pages 659–666. Association for Computing Machinery, 2008.

Michael D Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. Predicting the political alignment of twitter users. In *Institute of Electrical and Electronics Engineers Third Inernational Conference on Social Computing Privacy, Security, Risk and Trust*, pages 192–199. Institute of Electrical and Electronics Engineers, 2011.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Jean-Valère Cossu, Benjamin Bigot, Ludovic Bonnefoy, and Grégory Senay. Towards the improvement of topic priority assignment using various topic detection methods for e-reputation monitoring on twitter. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*, pages 154–159. Springer, 2014a.

Jean-Valère Cossu, Kilian Janod, Emmanuel Ferreira, Julien Gaillard, and Marc El-Bèze. Lia@ replab 2014: 10 methods for 3 tasks. In *4th International Conference of the Conference and Labs of the Evaluation Forum initiative.* Citeseer, 2014b.

Jean-Valère Cossu, Vincent Labatut, and Nicolas Dugué. A review of features for the discrimination of twitter users: application to the prediction of offline influence. *Social Network Analysis and Mining*, 6(1):1–23, 2016.

Nick Craswell. *Precision at n*, pages 2127–2128. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_484. URL https://doi.org/10.1007/978-0-387-39940-9_484.

James Curry, Weihang Zhu, Brian Craig, Lonnie Turpin, Majed Bokhari, and Pavan Mhasavekar. Using a natural language generation approach to document simulation results. In *Winter Simulation Conference*, pages 2116–2126. Institute of Electrical and Electronics Engineers, 2013.

Van Dang. The lemur project-wiki-ranklib. *Lemur Project,[Online]. Available: http://sourceforge.net/p/lemur/wiki/RankLib*, 2012.

Maximilien Danisch, Nicolas Dugué, and Anthony Perez. On the importance of considering social capitalism when measuring influence on Twitter. In *International Conference on Behavioral, Economic, and Socio-Cultural Computing*, pages 1–7, Shanghai, China, October 2014. Institute of Electrical and Electronics Engineers. doi: 10.1109/BESC.2014.7059501. URL https://hal.archives-ouvertes.fr/hal-01105133.

Dipanjan Das and André FT Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at Carnegie Mellon University*, 4:192–195, 2007.

Jorge Carrillo de Albornoz, Laura Plaza, and Pablo Gervás. Sentisense: An easily scalable concept-based affective lexicon for sentiment analysis. In *LREC*, pages 3562–3567, 2012.

Agustín D Delgado, Raquel Martínez, Soto Montalvo, and Víctor Fresno. Person name disambiguation in the web using adaptive threshold clustering. *Journal of the Association for Information Science and Technology*, 68(7):1751–1762, 2017.

Yajuan Duan, Zhumin Chen, Furu Wei, Ming Zhou, and Heung-Yeung Shum. Twitter topic summarization by ranking tweets using social influence and content quality. *Proceedings of International Conference on Computational Linguistics 2012*, pages 763–780, 2012.

Pablo Duboue. Automatic reports from spreadsheets: Data analysis for the rest of us. In *Proceedings of the 9th International Natural Language Generation conference*, pages 244–245, 2016.

Nawal El-Fishawy, Alaa Hamouda, Gamal M Attiya, and Mohammed Atef. Arabic summarization in twitter social network. *Ain Shams Engineering Journal*, 5(2):411–420, 2014.

Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December 2004a. ISSN 1076-9757. URL http://dl.acm.org/citation.cfm?id=1622487.1622501.

Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004b.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. URL https://dl.acm.org/citation.cfm?id=1442794.

Mohamed Abdel Fattah and Fuji Ren. Ga, mr, ffnn, pnn and gmm based models for automatic text summarization. *Computer Speech & Language*, 23(1):126 – 144, 2009. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2008.04.002. URL http://www.sciencedirect.com/science/article/pii/S0885230808000296.

Katja Filippova. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 322–330. Association for Computational Linguistics, 2010.

Marcelo Fiszman, Dina Demner-Fushman, Halil Kilicoglu, and Thomas C Rindflesch. Automatic summarization of medline citations for evidence-based medical treatment: A topic-oriented evaluation. *Journal of biomedical informatics*, 42(5):801–813, 2009.

Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969, 2003.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Kavita Ganesan. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. 2015.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics*, pages 340–348. Association for Computational Linguistics, 2010.

Albert Gatt and Ehud Reiter. Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 90–93. Association for Computational Linguistics, 2009.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1602–1613, 2014.

Anastasia Giachanou, Julio Gonzalo, Ida Mele, and Fabio Crestani. Sentiment propagation for predicting reputation polarity. In *European Conference on Information Retrieval*, pages 226–238. Springer, 2017.

Shlomo Gonen. Reputation scoring and reporting system, February 14 2012. US Patent 8,117,106.

Pankaj Gupta, Vijay Shankar Pendluri, and Ishant Vats. Summarizing text by ranking text units according to shallow linguistic features. In *13th International Conference on Advanced Communication Technology*, pages 1620–1625. Institute of Electrical and Electronics Engineers, 2011.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining explorations newsletter*, 11(1):10–18, 2009.

Sangdo Han, Hyosup Shim, Byungsoo Kim, Seonyeong Park, Seonghan Ryu, and Gary Geunbae Lee. Keyword question answering system with report generation for linked data. In *International Conference on Big Data and Smart Computing.*, pages 23–26. Institute of Electrical and Electronics Engineers, 2015.

Steven Alexander Hicks, Sigrun Eskeland, Mathias Lux, Thomas de Lange, Kristin Ranheim Randel, Mattis Jeppsson, Konstantin Pogorelov, Pål Halvorsen, and Michael Riegler. Mimir: An automatic reporting and reasoning system for deep learning based analysis in the medical domain. In *Proceedings of the 9th Association for Computing Machinery Multimedia Systems Conference*, Multimedia Systems Conference '18, pages 369–374, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 978-1-4503-5192-8. doi: 10.1145/3204949.3208129. URL http://doi.acm.org/10.1145/3204949.3208129.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Ya-Han Hu, Yen-Liang Chen, and Hui-Ling Chou. Opinion mining from online hotel reviews – a text summarization approach. *Information Processing & Management*, 53(2):436 – 449, 2017. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2016.12.002. URL http://www.sciencedirect.com/science/article/pii/S0306457316306781.

Anna Huang. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference, Christchurch, New Zealand*, pages 49–56, 2008.

Shereen Hussein, Mona Farouk, and ElSayed Hemayed. Gender identification of egyptian dialect in twitter. *Egyptian Informatics Journal*, 2019.

Igniyte. The reputation report 2018 - igniyte, 2018. https://www.igniyte.co.uk/-wp-content/uploads/2018/03/The-Reputation-Report-2018-Igniyte.pdf.

Kazushi Ikeda, Gen Hattori, Chihiro Ono, Hideki Asoh, and Teruo Higashino. Twitter user profiling based on text and community mining for market analysis. *Knowledge-Based Systems*, 51:35–47, 2013.

David Inouye and Jugal K Kalita. Comparing twitter summarization algorithms for multiple post summaries. In *Institute of Electrical and Electronics Engineers Third Inernational Conference on Social Computing Privacy, Security, Risk* *and Trust*, pages 298–306. Institute of Electrical and Electronics Engineers, 2011.

Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

Jefferson-Henrique. Getoldtweets-java @online, April 2016. URL https://github.com/Jefferson-Henrique/GetOldTweets-java.

Heo Jeong, Lee Chung Hee, Oh Hyo Jung, Yoon Yeo Chan, Kim Hyun Ki, Jo Yo Han, and Ock Cheol Young. Automatic generation of issue analysis report based on social big data mining. *Korea Information Processing Society Transactions on Software and Data Engineering*, 3(12):12, 2014.

Lei Jin, Xuelian Long, Ke Zhang, Yu-Ru Lin, and James Joshi. Characterizing users' check-in activities using their scores in a location-based social network. *Multimedia Systems*, 22(1):87–98, 2016.

Pamela Jordan, Nancy Green, Chistopher Thomas, and Susan Holm. Tbi-doc: Generating patient & clinician reports from brain imaging data. In *Proceedings of the 8th International Natural Language Generation Conference*, pages 143–146, 2014.

Gemma Joyce. Mujeres y hombres más influyentes en twitter en 2018, December 2018. URL https://www.brandwatch.com/es/blog/lista-influencers-twitter-2018/.

Georgios Katsimpras, Dimitrios Vogiatzis, and Georgios Paliouras. Determining influential users with supervised random walks. In *Proceedings of the 24th International Conference on World Wide Web*, pages 787–792. Association for Computing Machinery, 2015.

Ross Kindermann. Markov random fields and their applications. *American mathematical society*, 1980.

Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. *Advances in Automatic Summarization*, pages 55–60, 1999.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

Chen Li, Yang Liu, and Lin Zhao. Using external resources and joint learning for bigram weighting in ilp-based multi-document summarization. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 778–787, 2015a.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015b.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. Deep recurrent generative decoder for abstractive text summarization. *arXiv preprint arXiv:1708.00625*, 2017.

Wei Li, Lei He, and Hai Zhuge. Abstractive news summarization based on event semantic link network. In *Proceedings of International Conference on Computational Linguistics 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 236–246, 2016.

Shangsong Liang, Xiangliang Zhang, Zhaochun Ren, and Evangelos Kanoulas. Dynamic embeddings for user profiling in twitter. In *Proceedings of the 24th Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining International Conference on Knowledge Discovery & Data Mining*, pages 1764–1773. Association for Computing Machinery, 2018.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

Chin-Yew Lin and Eduard Hovy. Identifying topics by position. In *Fifth Conference on Applied Natural Language Processing*, 1997.

Chin-Yew Lin and Eduard Hovy. From single to multi-document summarization. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, 2002.

Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 605. Association for Computational Linguistics, 2004.

Dekang Lin et al. An information-theoretic definition of similarity. In *International Conference on Machine Learning*, volume 98, pages 296–304. Citeseer, 1998.

Marina Litvak and Mark Last. Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24. Association for Computational Linguistics, 2008.

Marina Litvak and Natalia Vanetik. Query-based summarization using mdl principle. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 22–31, 2017.

Marina Litvak, Mark Last, and Abraham Kandel. Degext: a language-independent keyphrase extractor. *Journal of Ambient Intelligence and Humanized Computing*, 4(3):377–387, Jun 2013. ISSN 1868-5145. doi: 10.1007/s12652-012-0109-z. URL `https://doi.org/10.1007/s12652-012-0109-z`.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. Toward abstractive summarization using semantic representations. *arXiv preprint arXiv:1805.10399*, 2018.

Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, March 2009. ISSN 1554-0669. doi: 10.1561/1500000016. URL `http://dx.doi.org/10.1561/1500000016`.

Wen P Liu, Bogdan Georgescu, Shaohua Kevin Zhou, and Dorin Comaniciu. Automatic generation of radiology reports from images and automatic rule out of images without findings, November 23 2017. United States Patent App. 15/158,375.

Xiaohua Liu, Yitong Li, Furu Wei, and Ming Zhou. Graph-based multi-tweet summarization using social signals. *Proceedings of International Conference on Computational Linguistics 2012*, pages 1699–1714, 2012.

Elena Lloret and Manuel Palomar. Text summarisation in progress: a literature review. *Artificial Intelligence Review*, 37(1):1–41, 2012.

Jacinto Jesús Mena Lomeña. Identificación y clasificación automática de creadores de opinión en Twitter. Master's thesis, Universidad Nacional de Educación a Distancia, 2014.

Jacinto Jesús Mena Lomena and Fernando López Ostenero. Uned at clef replab 2014: Author profiling. 2014.

Annie Louis and Todd Newman. Summarization of business-related tweets: A concept-based approach. *Proceedings of International Conference on Computational Linguistics 2012: Posters*, pages 765–774, 2012.

Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

Wencan Luo, Fei Liu, Zitao Liu, and Diane Litman. A novel ilp framework for summarizing content with high lexical variety. *arXiv preprint arXiv:1807.09671*, 2018.

Mary Madden and Aaron Smith. Reputation management and social media. 2010.

Marcelo Maia, Jussara Almeida, and Virgílio Almeida. Identifying user behavior in online social networks. In *Proceedings of the 1st workshop on Social network systems*, pages 1–6. Association for Computing Machinery, 2008.

Arun S Maiya and Tanya Y Berger-Wolf. Benefits of bias: Towards better characterization of network sampling. In *Proceedings of the 17th Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining international conference on Knowledge discovery and data mining*, pages 105–113. Association for Computing Machinery, 2011.

Krissada Maleewong. An analysis of influential users for predicting the popularity of news tweets. In *Pacific Rim International Conference on Artificial Intelligence*, pages 306–318. Springer, 2016.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

Ilia Markov, Helena Gómez-Adorno, Juan-Pablo Posadas-Durán, Grigori Sidorov, and Alexander Gelbukh. Author profiling with doc2vec neural network-based document embeddings. In Obdulia Pichardo-Lagunas and Sabino Miranda-Jiménez, editors, *Advances in Soft Computing*, pages 117–131, Cham, 2017. Springer International Publishing. ISBN 978-3-319-62428-0.

James Marquardt, Golnoosh Farnadi, Gayathri Vasudevan, Marie-Francine Moens, Sergio Davalos, Ankur Teredesai, and Martine De Cock. Age and gender identification in social media. In *Conference and Labs of the Evaluation Forum (Working Notes)*, pages 1129–1136, 2014.

Luís Marujo, Ricardo Ribeiro, David Martins de Matos, Joao P Neto, Anatole Gershman, and Jaime Carbonell. Extending a single-document summarizer to multi-document: a hierarchical approach. *arXiv preprint arXiv:1507.02907*, 2015.

Natalia Maslova and Vsevolod Potapov. Neural network doc2vec in automated sentiment analysis for short informal texts. In Alexey Karpov, Rodmonga Potapova, and Iosif Mporas, editors, *Speech and Computer*, pages 546–554, Cham, 2017. Springer International Publishing. ISBN 978-3-319-66429-3.

Yogesh Kumar Meena and Dinesh Gopalani. Feature priority based sentence filtering method for extractive automatic text summarization. *Procedia Computer Science*, 48:728–734, 2015.

Yashar Mehdad, Giuseppe Carenini, Frank Tompa, et al. Abstractive meeting summarization with entailment and fusion. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 136–146, 2013.

Qiaozhu Mei, Jian Guo, and Dragomir Radev. Divrank: the interplay of prestige and diversity in information networks. In *Proceedings of the 16th Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining international conference on Knowledge discovery and data mining*, pages 1009–1018. Association for Computing Machinery, 2010.

Edgar Meij, Wouter Weerkamp, and Maarten De Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth Association for Computing Machinery international conference on Web search and data mining*, pages 563–572. Association for Computing Machinery, 2012.

Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of Empirical Methods in Natural Language Processing 2004*, pages 404–411, Barcelona, Spain, July 2004a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W04-3252.

Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004b.

Thelwall Mike, Buckley Kevan, Paltoglou Georgios, and Cai Di. Sentiment in short strength detection informal text. *JASIST*, 61(12):2544–2558, 2010.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Seyedali Mirjalili, Shahrzad Saremi, Seyed Mohammad Mirjalili, and Leandro dos S Coelho. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, 47:106–119, 2016.

Ibrahim F Moawad and Mostafa Aref. Semantic graph reduction approach for abstractive text summarization. In *Seventh International Conference on Computer Engineering & Systems*, pages 132–138. Institute of Electrical and Electronics Engineers, 2012.

D Muhammad Noorul Mubarak. An overview of extractive based automatic text summarization systems. *International Journal of Computer Science & Information Technology*, 8(5):33–44, 2016.

Vivi Nastase. Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 763–772. Association for Computational Linguistics, 2008.

Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1191–1204, 2018.

Richard E Neapolitan et al. *Learning bayesian networks*, volume 38. Pearson Prentice Hall Upper Saddle River, NJ, 2004.

Victoria Nebot, Francisco Rangel, Rafael Berlanga, and Paolo Rosso. Identifying and classifying influencers in twitter only with textual information. In *International Conference on Applications of Natural Language to Information Systems*, pages 28–39. Springer, 2018.

Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer, 2012.

Evar D.. Nering and Albert William Tucker. *Linear Programs and Related Problems*. Academic Press, 1993.

Joel Larocca Neto, Alex A. Freitas, and Celso A. A. Kaestner. Automatic text summarization using a machine learning approach. In Guilherme Bittencourt and Geber L. Ramalho, editors, *Advances in Artificial Intelligence*, pages 205–215, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-36127-5.

Minh-Tien Nguyen and Minh-Le Nguyen. Sortesum: A social context framework for single-document summarization. In *European Conference on Information Retrieval*, pages 3–14. Springer, 2016.

Jianwei Niu, Huan Chen, Qingjuan Zhao, Limin Su, and Mohammed Atiquzzaman. Multi-document abstractive summarization using chunk-graph and recurrent neural network. In *Institute of Electrical and Electronics Engineers International Conference on Communications, 2017*, pages 1–6. Institute of Electrical and Electronics Engineers, 2017.

Andrei Olariu. Efficient online summarization of microblogging streams. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 236–240, 2014.

Rosa María Ortega-Mendoza, Anilú Franco-Arcega, Adrián Pastor López-Monroy, and Manuel Montes-y Gómez. I, me, mine: The role of personal phrases in author profiling. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 110–122. Springer, 2016.

Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *Proceedings of the 8th International Natural Language Generation Conference*, pages 45–53, 2014.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Alonso Palomino-Garibay, Adolfo T Camacho-Gonzalez, Ricardo A Fierro-Villaneda, Irazu Hernandez-Farias, Davide Buscaldi, Ivan V Meza-Ruiz, et al. A random forest approach for authorship profiling. In *Proceedings of Conference and Labs of the Evaluation Forum*, 2015.

Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

Braja Gopal Patra, Somnath Banerjee, Dipankar Das, Tanik Saikh, and Sivaji Bandyopadhyay. Automatic author profiling based on linguistic and stylistic features. *Notebook for PAN at Conference and Labs of the Evaluation Forum*, 1179, 2013.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

Marco Pennacchiotti and Ana-Maria Popescu. A machine learning approach to twitter user classification. *International Association for the Advancement of Artificial Intelligence Conference on Weblogs and Social Media*, 11(1):281–288, 2011.

Laura Plaza and Jorge Carrillo-de Albornoz. Evaluating the use of different positional strategies for sentence selection in biomedical literature summarization. *BMC bioinformatics*, 14(1):71, 2013.

French Pope III, Rouzbeh A Shirvani, Mugizi Robert Rwebangira, Mohamed Chouikha, Ayo Taylor, Andres Alarcon Ramirez, and Amirsina Torfi. Automatic detection of small groups of persons, influential members, relations and hierarchy in written conversations using fuzzy logic. In *Proceedings of the International Conference on Data Mining*, page 155. The Steering Committee of

The World Congress in Computer Science, Computer Engineering and Applied Computing, 2015.

John Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993. ISBN 1-55860-238-0.

Mandyam Annasamy Raghuram, K Akshay, and K Chandrasekaran. Efficient user profiling in twitter social network using traditional classifiers. In *Intelligent Systems Technologies and Applications*, pages 399–411. Springer, 2016.

Gabriela Ramírez-de-la Rosa, Esaú Villatoro-Tello, Héctor Jiménez-Salazar, and Christian Sánchez-Sánchez. Towards automatic detection of user influence in twitter by means of stylistic and behavioral features. In *Mexican International Conference on Artificial Intelligence*, pages 245–256. Springer, 2014.

Antonia Estrella Ramón and Cristina Segovia López. *Comunicación integrada de marketing.* Editorial Escuela Superior de Ingenieros Comerciales, 2016.

Francisco Rangel, Paolo Rosso, Martin Potthast, and Benno Stein. Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. *Working Notes Papers of the Conference and Labs of the Evaluation Forum*, 2017.

Ehud Reiter, Sandra Williams, and Lesley Crichton. Generating feedback reports for adults taking basic skills tests. In *Applications and Innovations in Intelligent Systems XIII*, pages 50–63. Springer, 2006.

Yogesh Kumar Meena Riya Jhalani. An abstractive approach for text summarization. *International Journal of Advance Computational Engineering and Networking*, (5), January 2017. ISSN 2320-2106.

Javier Rodríguez-Vidal, Julio Gonzalo, Laura Plaza, and Henry Anaya Sánchez. Automatic detection of influencers in social networks: Authority versus domain signals. *Journal of the Association for Information Science and Technology*, 2019.

Rajendra Kumar Roul, Samarth Mehrotra, Yash Pungaliya, and Jajati Keshari Sahoo. A new automatic multi-document text summarization using topic modeling. In *International Conference on Distributed Computing and Internet Technology*, pages 212–221. Springer, 2019.

Koustav Rudra, Siddhartha Banerjee, Niloy Ganguly, Pawan Goyal, Muhammad Imran, and Prasenjit Mitra. Summarizing situational and topical information during crises. *arXiv preprint arXiv:1610.01561*, 2016.

Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach.* Malaysia; Pearson Education Limited,, 2016.

Donald G. Saari. Explaining all three-alternative voting outcomes. *Journal of Economic Theory*, 87:313–355, 1999.

Ali Sadollah, Hadi Eskandar, Ardeshir Bahreininejad, and Joong Hoon Kim. Water cycle algorithm for solving multi-objective optimization problems. *Soft Computing*, 19(9):2587–2603, 2015.

Naveen Saini, Sriparna Saha, Anubhav Jangra, and Pushpak Bhattacharyya. Extractive single document summarization using multi-objective optimization: Exploring self-organized differential evolution, grey wolf optimizer and water cycle algorithm. *Knowledge-Based Systems*, 164:45–67, 2019.

Henry Anaya Sánchez. *Discovering and Describing Coherent and Meaningful Topics from Document Collections.* PhD thesis, Universitat Jaume I, 2016.

Kamal Sarkar, Khushbu Saraf, and Avishikta Ghosh. Improving graph based multidocument text summarization using an enhanced sentence similarity measure. In *Institute of Electrical and Electronics Engineers 2nd International Conference on Recent Trends in Information Systems*, pages 359–365. Institute of Electrical and Electronics Engineers, 2015.

Anne Schneider, Alasdair Mort, Chris Mellish, Ehud Reiter, Phil Wilson, and Pierre-Luc Vaudry. Mime-nlg in pre-hospital care. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 152–156, 2013.

Alexander Schrijver. *Theory of linear and integer programming.* John Wiley & Sons, 1998.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press, 2008.

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

Rajesh Shardan and Uday Kulkarni. Implementation and evaluation of evolutionary connectionist approaches to automated text summarization. 2010.

Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. Summarizing microblogs automatically. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*, pages 685–688. Association for Computational Linguistics, 2010.

Divya Sharma, Debashis Saha, and Parthasarathi Dasgupta. A graph-based scheme for brand promotion in social media platforms using influencer nodes. Technical report, IIM Calcutta, Working Paper Series, 2013.

Donal Simmie, Maria Grazia Vigliotti, and Chris Hankin. Ranking twitter influence by combining network centrality and influence observables in an evolutionary model. *Journal of Complex Networks*, 2(4):495–517, 2014. doi: 10.1093/comnet/cnu024. URL `http://dx.doi.org/10.1093/comnet/cnu024`.

Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. Biosses: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58, 2017.

Xuemeng Song, Zhao-Yan Ming, Liqiang Nie, Yi-Liang Zhao, and Tat-Seng Chua. Volunteerism tendency prediction via harvesting multiple social networks. *Association for Computing Machinery Transactions on Information Systems*, 34 (2):10, 2016.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

Damiano Spina, Julio Gonzalo, and Enrique Amigó. Learning similarity functions for topic detection in online reputation monitoring. In *Proceedings of the 37th international Association for Computing Machinery's Special Interest Group on Information Retrieval conference on Research & development in information retrieval*, pages 527–536. Association for Computing Machinery, 2014.

Anna Squicciarini, Sarah Rajtmajer, Y Liu, and Christopher Griffin. Identification and characterization of cyberbullying dynamics in an online social network. In *Proceedings of the 2015 Institute of Electrical and Electronics Engineers/Association for Computing Machinery International Conference on Advances in*

Social Networks Analysis and Mining 2015, pages 280–285. Association for Computing Machinery, 2015.

Gopu Srikanth Reddy, T. Murali Mohan, and T. Raghunadha Reddy. Author profiling approach for location prediction. In Raju Surampudi Bapi, Koppula Srinivas Rao, and Munaga V. N. K. Prasad, editors, *First International Conference on Artificial Intelligence and Cognitive Computing*, pages 389–395, Singapore, 2019. Springer Singapore. ISBN 978-981-13-1580-0.

Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The general inquirer: A computer approach to content analysis. 1966.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Hiroya Takamura, Hikaru Yokono, and Manabu Okumura. Summarizing a document stream. In *European conference on information retrieval*, pages 177–188. Springer, 2011.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1171–1181, 2017.

R Core Team et al. R: A language and environment for statistical computing. 2013.

Bharat Tidke, Rupa Mehta, and Jenish Dhanani. Sirif: Supervised influence ranking based on influential network. *Journal of Intelligent & Fuzzy Systems*, (Preprint):1–13, 2018.

Lap Q. Trieu, Huy Q. Tran, and Minh-Triet Tran. News classification from social media using twitter-based doc2vec model and automatic query expansion. In *Proceedings of the Eighth International Symposium on Information and Communication Technology*, Symposium on Information and Communication Technology 2017, pages 460–467, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 978-1-4503-5328-1. doi: 10.1145/3155133.3155206. URL http://doi.acm.org/10.1145/3155133.3155206.

Sho Tsugawa and Kazuma Kimura. Identifying influencers from sampled social networks. *Physica A: Statistical Mechanics and its Applications*, 507:294–303, 2018.

Laura Tucker and T C Melewar. Corporate reputation and crisis management: The threat and manageability of anti-corporatism. *Corporate Reputation Review*, 7(4):377–387, Jan 2005. ISSN 1479-1889. doi: 10.1057/palgrave.crr.1540233. URL `https://doi.org/10.1057/palgrave.crr.1540233`.

David Vilares, Miguel Hermo, Miguel A Alonso, Carlos Gómez-Rodríguez, and Jesús Vilares. Lys at clef replab 2014: Creating the state of the art in author influence ranking and reputation classification on twitter. In *Conference and Labs of the Evaluation Forum (Working Notes)*, pages 1468–1478, 2014.

Esaú Villatoro-Tello, Gabriela Ramírez-de-la Rosa, Christian Sánchez-Sánchez, Héctor Jiménez-Salazar, Wulfrano Arturo Luna-Ramírez, and Carlos Rodríguez-Lucatero. Uamclyr at replab 2014: Author profiling task. In *Conference and Labs of the Evaluation Forum (Working Notes)*, pages 1547–1558, 2014.

Lu Wang and Claire Cardie. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1395–1405, 2013.

Leo Wanner et al. *Feature engineering for author profiling and identification: on the relevance of syntax and discourse.* PhD thesis, Universitat Pompeu Fabra, 2017.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 985–992. Association for Computational Linguistics, 2008.

Kristian Woodsend and Mirella Lapata. Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural*

Language Learning, pages 233–243. Association for Computational Linguistics, 2012.

Haibing Wu, Yiwei Gu, Shangdi Sun, and Xiaodong Gu. Aspect-based opinion summarization with convolutional neural networks. In *International Joint Conference on Neural Networks*, pages 3157–3163. Institute of Electrical and Electronics Engineers, 2016.

Qiang Wu, Chris JC Burges, Krysta M Svore, and Jianfeng Gao. Ranking, boosting, and model adaptation. Technical report, Technical report, Microsoft Research, 2008.

Hailu Xu, Weiqing Sun, and Ahmad Javaid. Efficient spam detection across online social networks. In *Institute of Electrical and Electronics Engineers International Conference on Big Data Analysis*, pages 1–6. Institute of Electrical and Electronics Engineers, 2016.

Guoliang Yang, Kim Young, Su Huang, John Shim, and Wieslaw Lucjan Nowinski. Method for creating a report from radiological images using electronic report templates, September 26 2013. US Patent App. 13/989,774.

Kuiyuan Yang, Meng Wang, Xian-Sheng Hua, and Hong-Jiang Zhang. Social image search with diverse relevance ranking. In *International Conference on Multimedia Modeling*, pages 174–184. Springer, 2010.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. Graph-based neural multi-document summarization. *arXiv preprint arXiv:1706.06681*, 2017.

Han Zhang, Marcelo Fiszman, Dongwook Shin, Christopher M Miller, Graciela Rosemblat, and Thomas C Rindflesch. Degree centrality for semantic abstraction summarization of therapeutic studies. *Journal of biomedical informatics*, 44(5):830–838, 2011.

Jing Zhang, Xiaoxue Li, Weizhi Nie, and Yuting Su. Automatic report generation based on multi-modal information. *Multimedia Tools and Applications*, 76(9): 12005–12015, 2017.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. Selective encoding for abstractive sentence summarization. *arXiv preprint arXiv:1704.07073*, 2017.

Xiaohua Zhou, Xiaohua Hu, and Xiaodan Zhang. Topic signature language models for ad hoc retrieval. *Institute of Electrical and Electronics Engineers transactions on knowledge and data engineering*, 19(9):1276–1287, 2007.

Hao Zhuang, Rameez Rahman, Xia Hu, Tian Guo, Pan Hui, and Karl Aberer. Data summarization with social contexts. In *Proceedings of the 25th Association for Computing Machinery International on Conference on Information and Knowledge Management*, pages 397–406. Association for Computing Machinery, 2016.

APPENDIX A

OUTCOMES OF THE THESIS

In this appendix, we present the list of publications in journals and conferences conducted during the course of this Thesis, the released datasets and the research visits.

A.1 Datasets

As part of some of the previous publications, the following datasets have been developed and labelled:

- **eDiseases dataset:** this dataset contains patient data from the MedHelp[24] health site, where different communities share information and opinions about diseases. Each community consists of a number of conversations; a conversation being a sequence of comments posted by patients. This dataset is publicly available at: https://zenodo.org/record/1479354#.XHVdUOhKiUk.

- **RepLab summarization dataset:** this dataset contains companies data from the RepLab 2013[25] dataset, where users from Twitter talk about different topics of the companies. Each topic consists of a different number of tweets posted by Twitter users. Is publicly available at: https://zenodo.org/record/2536801#.XHVceOhKiUl.

[24] http://www.medhelp.org/

[25] http://nlp.uned.es/replab2013/

A.2 Research visits

During the development of this thesis, one research centre has been visited:

- Universitait van Amsterdam, Informatics Institute: This visit took place during the period of April-June 2017. The supervisor in charge was Maarten de Rijke which is the European reference for the automated processing of Social Media in the areas of Information Retrieval and Text Mining. During the stay, in conjunction with Stevan Rudinac, we developed a system to locate extreme far right influencers using their texts and discover topics by using entity linking methods. This research was framed inside the Vox-Pol[26] project

[26] https://www.voxpol.eu/

APPENDIX B
DETAILED RESULTS

Here we list detailed results for alternative evaluation metrics (P@10, P@50, P@100) and for each of the RepLab domains (automotive and banking) mentioning in chapter 3.

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	**0.37**	0.53	0.54	0.30
BoW with RTs	0.31	0.64	0.54	0.30
BoW with FAVs	0.32	0.64	0.53	0.29
BoW with DivFoll	0.36	**0.67**	0.47	0.29
BoW with FAVs and Foll	0.35	0.64	0.54	0.58
BoW with Foll and DivFoll	**0.37**	0.51	**0.59**	0.58
BoW with Followees, Foll and DivFoll	0.33	0.51	**0.59**	0.70
BoW with RTs, FAVs, Foll, Followees and DivFoll	0.35	0.51	**0.59**	**0.71**
Best Combination	**0.37**	**0.67**	**0.59**	**0.71**
Best Result Published	0.80	0.80	0.80	0.80
Best RepLab 2014	0.72	0.72	0.72	0.72
Baseline - Followers	0.37	0.35	0.37	0.34

Table B.1: Automotive domain with BoW results (* not use BoW for ranking because it is not a sortable.)

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	0.38	0.23	**0.54**	0.24
BoW with RTs	0.35	0.21	0.51	0.36
BoW with FAVs	0.31	0.21	0.49	0.36
BoW with DivFoll	0.38	0.22	0.40	0.32
BoW with FAVs and Foll	0.36	0.21	0.42	0.46
BoW with Foll and DivFoll	**0.39**	**0.24**	0.32	0.46
BoW with Follees, Foll and DivFoll	0.30	**0.24**	0.37	0.35
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.37	**0.24**	0.33	**0.50**
Best Combination	**0.39**	**0.24**	**0.54**	**0.50**
Best Result Published	0.63	0.63	0.63	0.63
Best RepLab 2014	0.41	0.41	0.41	0.41
Baseline - Followers	0.39	0.33	0.39	0.26

Table B.2: Banking domain with BoW results (* not use BoW for ranking because it is not a sortable.)

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	**0.50**	0.80	0.50	0.20
BoW with RTs	0.10	**1**	0.50	0.25
BoW with FAVs	0.40	**1**	0.40	0.20
BoW with DivFoll	0.20	0.40	**1**	0.20
BoW with FAVs and Foll	0.40	**1**	0.30	0
BoW with Foll and DivFoll	**0.50**	0.90	0.60	0.10
BoW with Follees, Foll and DivFoll	0.30	0.20	0.50	0
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.30	0.80	0.50	**1**
Best Combination	**0.50**	**1**	**1**	**1**

Table B.3: P@10 automotive domain with BoW (* not use BoW for ranking because it is not a sortable.)

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	**0.70**	0	0.20	0.30
BoW with RTs	0.10	0	0.40	0.30
BoW with FAVs	0.10	0	0.30	0.10
BoW with DivFoll	**0.70**	**0.30**	0	0.30
BoW with FAVs and Foll	0.30	0	**0.60**	0
BoW with Foll and DivFoll	0.30	0	**0.60**	0.10
BoW with Follees, Foll and DivFoll	0.10	0.20	0.50	0
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.30	0.10	0.30	**1**
Best Combination	**0.70**	**0.30**	**0.60**	**1**

Table B.4: P@10 banking domain with BoW (* not use BoW for ranking because it is not a sortable.)

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	0.46	0.62	0.48	0.42
BoW with RTs	0.26	0.94	0.56	0.42
BoW with FAVs	0.32	**0.98**	0.60	0.42
BoW with DivFoll	0.28	**0.98**	**1**	0.46
BoW with FAVs and Foll	**0.50**	0.96	0.66	0
BoW with Foll and DivFoll	0.40	0.94	0.74	0
BoW with Follees, Foll and DivFoll	0.24	0.90	0.66	0
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.32	0.86	0.68	**1**
Best Combination	**0.50**	**0.98**	**1**	**1**

Table B.5: P@50 automotive domain with BoW (* not use BoW for ranking because it is not a sortable.)

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	0.40	0.22	**0.62**	0.56
BoW with RTs	0.28	0.14	0.60	0.56
BoW with FAVs	0.32	0.14	0.58	0.56
BoW with DivFoll	0.34	**0.36**	0.4	0.56
BoW with FAVs and Foll	**0.50**	0.22	0.3	0
BoW with Foll and DivFoll	0.44	0.10	0.3	0.10
BoW with Follees, Foll and DivFoll	0.30	0.02	0.56	0
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.48	0.08	0.38	**1**
Best Combination	**0.50**	**0.36**	**0.62**	**1**

Table B.6: P@50 banking domain with BoW (* not use BoW for ranking because it is not a sortable.)

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	0.35	0.54	0.59	0.47
BoW with RTs	0.28	0.94	0.68	0.47
BoW with FAVs	0.31	**0.96**	0.62	0.47
BoW with DivFoll	0.37	0.95	**1**	0.50
BoW with FAVs and Foll	**0.43**	**0.96**	0.72	0
BoW with Foll and DivFoll	0.37	0.89	0.71	0.02
BoW with Follees, Foll and DivFoll	0.20	0.90	0.66	0
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.28	0.87	0.71	**1**
Best Combination	**0.43**	**0.96**	**1**	**1**

Table B.7: P@100 automotive domain with BoW (* not use BoW for ranking because it is not a sortable.)

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	**0.48**	0.22	**0.65**	0.51
BoW with RTs	0.36	0.10	**0.65**	0.51
BoW with FAVs	0.31	0.10	0.59	0.51
BoW with DivFoll	0.33	**0.27**	0.2	0.51
BoW with FAVs and Foll	0.43	0.15	0.36	0
BoW with Foll and DivFoll	0.46	0.11	0.31	0.08
BoW with Follees, Foll and DivFoll	0.31	0.07	0.57	0
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.44	0.09	0.33	**1**
Best Combination	**0.48**	**0.27**	**0.65**	**1**

Table B.8: P@100 banking domain with BoW (* not use BoW for ranking because it is not a sortable.)

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	**0.60**	0.40	0.35	0.25
BoW with RTs	0.10	**0.50**	0.45	0.28
BoW with FAVs	0.25	**0.50**	0.35	0.15
BoW with DivFoll	0.45	0.35	0.25	0.25
BoW with FAVs and Foll	0.35	**0.50**	0.45	0
BoW with Foll and DivFoll	0.40	0.45	**0.60**	0.10
BoW with Follees, Foll and DivFoll	0.20	0.20	0.50	0
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.30	0.45	0.40	**1**
Best Combination	**0.60**	**0.50**	**0.60**	**1**

Table B.9: P@10 average results between automotive and banking domains using BoW (* not use BoW for ranking because it is not a sortable.)

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	0.43	0.42	0.55	0.49
BoW with RTs	0.27	0.54	0.58	0.49
BoW with FAVs	0.32	0.56	0.59	0.49
BoW with DivFoll	0.31	**0.67**	**0.70**	0.51
BoW with FAVs and Foll	**0.50**	0.59	0.48	0
BoW with Foll and DivFoll	0.42	0.52	0.52	0.05
BoW with Follees, Foll and DivFoll	0.27	0.46	0.61	0
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.40	0.47	0.53	**1**
Best Combination	**0.50**	**0.67**	**0.70**	**1**

Table B.10: P@50 average results between automotive and banking domains using BoW (* not use BoW for ranking because it is not a sortable.)

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	0.42	0.38	0.62	0.49
BoW with RTs	0.32	0.52	**0.67**	0.49
BoW with FAVs	0.31	0.53	0.61	0.49
BoW with DivFoll	0.35	**0.61**	0.60	0.51
BoW with FAVs and Foll	0.43	0.56	0.54	0
BoW with Foll and DivFoll	**0.46**	0.50	0.51	0.05
BoW with Follees, Foll and DivFoll	0.26	0.49	0.62	0
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.36	0.48	0.52	**1**
Best Combination	**0.46**	**0.61**	**0.67**	**1**

Table B.11: P@100 average results between automotive and banking domains using BoW (* not use BoW for ranking because it is not a sortable.)

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.37	0.35	0.37	0.35
Domain Vocabulary	0.64	0.71	**0.76**	0.69
Twitter Auth. + Domain Voc.	**0.75**	0.74	0.75	0.70
Authority Vocabulary	**0.75**	0.63	**0.76**	0.73
Twitter Auth. + Authority Voc.	0.63	0.74	0.72	0.74
Domain Voc. + Authority Voc.	0.68	**0.75**	**0.76**	**0.79**
Best Combination	**0.75**	**0.75**	**0.76**	**0.79**
Best Result Published	0.80	0.80	0.80	0.80
Best RepLab 2014	0.72	0.72	0.72	0.72
Baseline - Followers	0.37	0.35	0.37	0.34

Table B.12: Modeled automotive domain results

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.39	0.33	0.39	0.30
Domain Vocabulary	0.22	0.19	0.36	0.52
Twitter Auth. + Domain Voc.	0.31	**0.35**	0.39	0.57
Authority Vocabulary	**0.71**	0.17	**0.70**	0.68
Twitter Auth. + Authority Voc.	0.67	0.18	**0.70**	**0.70**
Domain Voc. + Authority Voc.	0.68	0.18	0.68	0.68
Best Combination	**0.71**	**0.35**	**0.70**	**0.70**
Best Result Published	0.63	0.63	0.63	0.63
Best RepLab 2014	0.41	0.41	0.41	0.41
Baseline - Followers	0.39	0.33	0.39	0.26

Table B.13: Modeled banking domain results

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.40	0.38	0.50	0.10
Domain Vocabulary	0.80	**1**	0.80	0.90
Twitter Auth. + Domain Voc.	**1**	0.70	0.70	0.90
Authority Vocabulary	**1**	0.90	**1**	**1**
Twitter Auth. + Authority Voc.	**1**	**1**	**1**	0.90
Domain Voc. + Authority Voc.	0.90	0.70	**1**	**1**
Best Combination	**1**	**1**	**1**	**1**

Table B.14: P@10 modeled automotive domain results

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.30	0.37	0.50	0.70
Domain Vocabulary	0.60	0.40	0.60	0.80
Twitter Auth. + Domain Voc.	0.80	0.10	0.60	**1**
Authority Vocabulary	0.80	0	0.80	**1**
Twitter Auth. + Authority Voc.	**1**	**0.60**	**1**	**1**
Domain Voc. + Authority Voc.	0.80	0	0.80	**1**
Best Combination	**1**	**0.60**	**1**	**1**

Table B.15: P@10 modeled banking domain results

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.22	0.20	0.46	0.24
Domain Vocabulary	0.84	0.90	0.84	0.86
Twitter Auth. + Domain Voc.	0.84	0.84	0.92	0.88
Authority Vocabulary	**0.96**	0.78	**0.96**	0.94
Twitter Auth. + Authority Voc.	0.82	**0.96**	0.82	**0.96**
Domain Voc. + Authority Voc.	0.92	0.76	0.94	**0.96**
Best Combination	**0.96**	**0.96**	**0.96**	**0.96**

Table B.16: P@50 modeled automotive domain results

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.28	0.20	0.40	0.58
Domain Vocabulary	0.38	0.32	0.38	0.68
Twitter Auth. + Domain Voc.	0.38	**0.34**	0.38	0.86
Authority Vocabulary	0.90	0	0.90	**0.92**
Twitter Auth. + Authority Voc.	**0.92**	0.12	**0.92**	**0.92**
Domain Voc. + Authority Voc.	0.50	0	0.90	**0.92**
Best Combination	**0.92**	**0.34**	**0.92**	**0.92**

Table B.17: P@50 modeled banking domain results

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.23	0.19	0.35	0.30
Domain Vocabulary	0.88	0.88	0.88	0.81
Twitter Auth. + Domain Voc.	0.88	0.87	0.91	0.86
Authority Vocabulary	**0.95**	0.76	**0.95**	0.92
Twitter Auth. + Authority Voc.	0.80	**0.95**	0.84	**0.95**
Domain Voc. + Authority Voc.	0.93	0.73	0.92	0.92
Best Combination	**0.95**	**0.95**	**0.95**	**0.95**

Table B.18: P@100 modeled automotive domain results

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.25	0.17	0.48	0.51
Domain Vocabulary	0.26	0.34	0.48	0.63
Twitter Auth. + Domain Voc.	0.39	**0.42**	0.49	0.77
Authority Vocabulary	**0.88**	0	**0.88**	0.87
Twitter Auth. + Authority Voc.	0.85	0.07	0.86	**0.88**
Domain Voc. + Authority Voc.	0.38	0	0.84	0.87
Best Combination	**0.88**	**0.42**	**0.88**	**0.88**

Table B.19: P@100 modeled banking domain results

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.35	0.38	0.50	0.40
Domain Vocabulary	0.70	0.70	0.70	0.85
Twitter Auth. + Domain Voc.	0.90	0.40	0.65	0.95
Authority Vocabulary	0.90	0.45	0.90	**1**
Twitter Auth. + Authority Voc.	**1**	**0.80**	**1**	0.95
Domain Voc. + Authority Voc.	0.85	0.35	0.90	**1**
Best Combination	**1**	**0.80**	**1**	**1**

Table B.20: P@10 average results between modeled automotive and modeled banking domains

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.25	0.20	0.43	0.41
Domain Vocabulary	0.61	**0.61**	0.61	0.77
Twitter Auth. + Domain Voc.	0.61	0.59	0.65	0.87
Authority Vocabulary	**0.93**	0.39	**0.93**	0.93
Twitter Auth. + Authority Voc.	0.87	0.54	0.87	**0.94**
Domain Voc. + Authority Voc.	0.71	0.38	0.92	**0.94**
Best Combination	**0.93**	**0.61**	**0.93**	**0.94**

Table B.21: P@50 average results between modeled automotive and modeled banking domains

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.24	0.18	0.42	0.41
Domain Vocabulary	0.57	0.61	0.68	0.72
Twitter Auth. + Domain Voc.	0.64	**0.65**	0.70	0.82
Authority Vocabulary	**0.92**	0.38	**0.92**	0.90
Twitter Auth. + Authority Voc.	0.83	0.51	0.85	**0.92**
Domain Voc. + Authority Voc.	0.66	0.37	0.88	0.90
Best Combination	**0.92**	**0.65**	**0.92**	**0.92**

Table B.22: P@100 average results between modeled automotive and modeled banking domains

APPENDIX C

EXAMPLE OF A REPUTATION SUMMARY

```
1 "Santander may sell U.S. car finance arm to raise cash http://bit.ly/WCi6Za "
2 "Santander planea absorber Banesto http://www.telecinco.es/informativos/
    economia/Santander-absorber-Banesto-CNMV-cotizacion_0_1526175033.html "
3 "Sernac ofició al Banco Santander por nueva falla http://bit.ly/RfDthz "
4 "Inditex, Mercadona y Santander lideran el ranking de mejores empresas para
    trabajar en España #empleo #trabajo http://ow.ly/fo7Rh "
5 "Elmo: 6 de diciembre - 5.00 Santander aumenta las alarmas sobre Salfacorp:
    duda que pueda cumplir sus compromisos de http://goo.gl/bwn65 "
6 "Santander cerrará 700 oficinas tras la integración de las filiaes Banesto y
    Banif. http://bit.ly/U6ZCy7 #economia #finanzas #bolsa #forex"
7 "Banco Santander despide a 1.200 empleados de Brasil por el pinchazo ... http
    ://bit.ly/Unyo3l "
8 "El #SERNAC pidió antecedentes al Banco Santander por nuevo fallo en sus
    sistemas http://ow.ly/fViPT "
9 "¿Financieros?: compras en CaixaBank y Santander, ventas en Mapfre y Popular
    http://bit.ly/Tx780W #finanzas #economia"
10 "Concurso FotoTalentos´13 Fundación Banco Santander y Universia http://ow.ly/
    g1YEA "
11 "Santander y la burbuja: ""Algunas comunas de Santiago presentan alzas que no
    ... - Diario inmobilia... http://bit.ly/XjLdAI #inmobiliaria"
12 "Negative outlook for Santander UK says S&P: Santander UK has been taken off
    CreditWatch negative by Standard and... http://bit.ly/T5kdUT "
13 "Ingresa unos 11,9 millones Emilio Botín vuelve a cobrar todo el dividendo de
    Santander en efectivo http://www.cincodias.com/ "
14 "Anuncia Banco Santander en España cierre de 700 sucursales http://mile.io/
    Yb0DpB "
15 "Santander plans to invest in Spain's bad bank http://dlvr.it/2VSJ4K #forex"
```

```
"Santander y Aegon se alían para potenciar el negocio de bancaseguros en España | http://Diarioelaguijon.com http://www.diarioelaguijon.com/noticia/12280/ECONOMIA-Y-EMPRESAS/Santander-y-Aegon-se-alian-para-potenciar-el-negocio-de-bancaseguros-en-Espana.html "
"Segunda convocatoria del programa Becas Santander. http://buzz.mw/-SJp_y "
"Get a Car - Enter your zip code to find dealers near you that offer financing with one of Santander programs. http://bit.ly/pZGfh0 "
"Santander considers absorbing Banesto - http://FT.com - Financial Times http://tinyurl.com/d2ked9s "
"El Santander cerrará 700 sucursales al integrar Banesto en su estructura http://ow.ly/g9V8J Banesto se dispara en Bolsa"
"VIDEO Un grupo de jóvenes arremete contra una sucursal del Santander y revienta el escaparate con una valla http://www.youtube.com/watch?v=x2QevygFits #14N"
"Conveyancing Top solicitor pulls off Santander mortgage fraud - Bridging and Commerical: Top solicitor pulls off... http://bit.ly/W8WfYV "
"#Colombia Santander tiene un programa de tecnología para mujeres empresarias http://bit.ly/Wqorr5 "
"Santander says to close 700 bank branches after Banesto buyout: MADRID, Dec 17 (Reuters) - Spain's largest bank ... http://bit.ly/SDNW76 "
"#Spain's #Santander studying how to absorb #Banesto: http://bit.ly/Zci9x1 | #MADRID #Banco"
"Mirad gráfico al final del post y entenderéis como uno puede convertirse en banquero casi gratis #Santander # Banesto http://www.gurusblog.com/archives/banco-santander-absorber-banesto/17/12/2012/ "
"La absorción de Banesto por parte del Santander pone fin a 110 años de historia de la entidad: http://www.telecinco.es/informativos/economia/absorcion-Banesto-Santander-historia-entidad_0_1526175166.html "
"Santander México es reconocido como Banco del Año - http://bit.ly/SsTE9p "
"Santander invertirá 660 millones y Caixabank, 470 millones en la primera fase del banco malo: Santander y CaixaB... http://bit.ly/Scq4Hp "
```