Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

André Filipe Neves Farinha

Extracting Keywords from Tweets

Project Report – Final Master's Work

Supervisor: Ricardo Campos, Assistant Professor at the Polytechnic Institute of Tomar

Co-supervisor: Vítor Mangaravite, Researcher at INESC TEC

Project Report - Final Master's Work

Presented to the Polytechnic Institute of Tomar

to fulfill the requirements necessary

to obtain the degree of Master

in Computer Engineering- Internet of Things (MCE-IoT)

To my grandmother Maria who has had difficult times at this stage of my life.

To my parents Timóteo and Celeste for all the dedication and courage they gave me.

To my girlfriend Ana for all her unconditional love and support.

To the father of my girlfriend who always supported me with his good disposition and that reached the end of his life during this work.

To my friends and colleagues Ricardo, Pedro and João for the support and courage they gave me.

To my supervisors for their help, dedication and constant support.

Resumo

Nos últimos anos, uma enorme quantidade de informações foi disponibilizada na Internet. As redes sociais estão entre as que mais contribuem para esse aumento no volume de dados. O Twitter, em particular, abriu o caminho, enquanto plataforma social, para que pessoas e organizações possam interagir entre si, gerando grandes volumes de dados a partir dos quais é possível extrair informação útil. Uma tal quantidade de dados, permitirá por exemplo, revelar-se importante se e quando, vários indivíduos relatarem sintomas de doença ao mesmo tempo e no mesmo lugar. Processar automaticamente um tal volume de informações e obter a partir dele conhecimento útil, torna-se, no entanto, uma tarefa impossível para qualquer ser humano. Os extratores de palavras-chave surgem neste contexto como uma ferramenta valiosa que visa facilitar este trabalho, ao permitir, de uma forma rápida, ter acesso a um conjunto de termos caracterizadores do documento.

Neste trabalho, tentamos contribuir para um melhor entendimento deste problema, avaliando a eficácia do YAKE (um algoritmo de extração de palavras-chave não supervisionado) em cima de um conjunto de tweets, um tipo de texto, caracterizado não só pelo seu reduzido tamanho, mas também pela sua natureza não estruturada. Embora os extratores de palavras-chave tenham sido amplamente aplicados a textos genéricos, como a relatórios, artigos, entre outros, a sua aplicabilidade em tweets é escassa e até ao momento não foi disponibilizado formalmente nenhum conjunto de dados. Neste trabalho e por forma a contornar esse problema optámos por desenvolver e tornar disponível uma nova coleção de dados, um importante contributo para que a comunidade científica promova novas soluções neste domínio. O KWTweet foi anotado por 15 anotadores e resultou em 7736 tweets anotados. Com base nesta informação, pudemos posteriormente avaliar a eficácia do YAKE! contra 9 baselines de extração de palavra-chave não supervisionados (TextRank, KP-Miner, SingleRank, PositionRank, TopicPageRank, MultipartiteRank, TopicRank, Rake e TF.IDF). Os resultados obtidos demonstram que o YAKE! tem um desempenho superior quando comparado com os seus competidores, provando-se assim a sua eficácia neste tipo de textos. Por fim, disponibilizamos uma demo que visa demonstrar o funcionamento do YAKE! Nesta plataforma web, os utilizadores têm a possibilidade de fazer uma pesquisa por utilizador ou hashtag e dessa forma obter as palavras chave mais relevantes através de uma nuvem de palavras.

Palavras-chave

Extrator de palavras-chave, Twitter, Extração de informação

Abstract

In recent years, an incredible amount of information has been made available on the Internet. Social networks are within the ones that most contribute to this realm. Twitter, in particular, has paved the way as a social platform where people and organizations alike, interact between them generating huge portions of content ready to be explored. This information, if left unprocessed, turns out to be just a simple text. Instead, digesting its contents can lead to useful information. For example, reports that an individual has an illness, may not be at first instance, much relevant, but it can reveal an important information as for an outbreak is concerned, if several individuals are found to report the very same symptoms at the same time and place. In large quantities however, reading and digesting one such volume of data, turns out to be an impossible task for any human interested in obtaining knowledge in a timely manner. Keyword Extractors have emerged in this context as a valuable tool that aims to facilitate this work, by extracting a set of terms that are able to describe the subject of a document in a glimpse.

In this work, we try to give a contribute to this problem by testing the effectiveness of YAKE (an unsupervised keyword extraction algorithm) on top of tweets, a kind of text that is characterized not only by its short length nature, but also by unstructured and sometimes noisy text. Although keyword extractors have been widely applied to generic texts, such as reports, news articles or web documents, to name but a few, their applicability to tweets is scarce and no formal dataset has been made available. With this limitation in mind, we developed a new publicly available collection. KWTweet dataset was annotated by 15 human volunteer editors and resulted in 7736 annotated tweets. We believe that making this dataset available is an important contribution to the research community which will foster research in this particular domain. Based on this, we were then able to evaluate the effectiveness of YAKE! against 9 unsupervised keyword extractor baselines (TextRank, KP-Miner, SingleRank, PositionRank, TopicPageRank, MultipartiteRank, TopicRank, Rake and TF.IDF). The results obtained demonstrate that YAKE! performs better than any of its competitors, thus proving its validity and usefulness when tackling this kind of documents. Finally, we provide a demonstration to show the results of YAKE! when applied to tweets. Users will be able to query Twitter via a username or hashtag and to get the relevant keywords in a word cloud fashion.

Keywords

Keyword Extraction, Twitter, Information Extraction

Acknowledgements

As time goes by, I realize how much this project taught me and made me grow, not only personally, but also professionally. I began to study an area that was truly unknown for me. Conducting this project alone would have been a difficult task, if not impossible, which leads me to thank several people.

First of all, I would like to acknowledge my supervisor Ricardo Campos and co-supervisor Vítor Mangaravite for the strength, precious time they spent with me, for their ideas, knowledge and the way they showed me how research could be interesting.

I would also like to thank all of those who contributed with their time in the process of constructing the dataset. In particular Mickael Ferreira, Isabel Nunes, Tiago Fernandes, Letícia Lima, Carla Campos, Ana Mendes, Vítor Mangaravite, Vasco Fernandes, Cátia Serrano, Paulo Simões, Filipe Morais and Behrooz Mansouri. Without them this would not have been possible.

I cannot forget my parents, Timóteo and Celeste, for all the help, support and trust they have given me through life, which has made it possible to get here. I would also like to thank my girlfriend, Ana Mendes, for her support and help, not only in my personal life but also throughout my school journey.

I am also very grateful to my colleagues and friends Pedro Ferreira, Ricardo Anacleto, Nélson Gomes, João Faria, Letícia Lima, Tiago Fernandes and Ricardo Raimundo for all the help, support and strength they gave me, especially in the most difficult moments.

Finally, I am very grateful to both the IPT – Instituto Politécnico de Tomar, for the conditions they have given me to get here, and to INESC TEC for giving me all the needed conditions whenever I went to Porto.

Table of Contents

Dedication **III**

Resumo **VII**

Abstract **IX**

Acknowledgements **XI**

Table of Contents **XIII**

List of Figures **XV**

List of Tables **XVII**

1. Introduction **1**

1.1. Motivation Goals 1

1.2. Challenges 2

1.3. Contribution 2

1.4. Outline 3

2. Social Networks: Literature Review **5**

2.1. Introduction 5

2.2. Facebook 7

2.3. Google+ 8

2.4. Twitter 9

3. Keyword Extraction: Architecture and Literature Review **13**

3.1. Applications of Keyword Extractors 13

3.2. Architecture 14

3.2.1. Pre-Processing 15

3.2.2. Candidate Terms List 16

3.2.3. Feature Extraction 16

3.2.4. Scoring 17

3.2.5. Ranking 18

3.3. Keyword Extraction Approaches 18

3.3.1. Unsupervised Methodologies 18

3.3.2. Supervised Methodologies 22

4. KWTweet Dataset - A Data Collection for Keyword Extraction with Tweets **25**

4.1. The Task of Evaluation 25

4.2. Keyword Extractor Reference Datasets 26

4.2.1. Generic Keyword Extractor Datasets 27
4.2.2. Twitter Datasets 27
4.3. Data Collection and Labelling 29
4.3.1. Data Collecting 30
4.3.2. Annotation Task 32
4.3.3. GitHub 37
4.4. KWTweet Dataset Analysis 38
5. Detecting Keywords on Twitter **47**
5.1. Problem Definition 47
5.2. YAKE! Architecture 48
5.2.1. Text pre-processing and Candidate Term Identification 48
5.2.2. Single Term Weight 49
5.2.3. *n*-gram Generation and Keyword Weight Assignment 50
5.2.4. Data Deduplication and Ranking 50
6. Evaluation **53**
6.1. Evaluation Metrics 53
6.1.1. Precision at k 53
6.1.2. Recall at k 54
6.1.3. F1-Measure at k 54
6.1.4. Mean Average Precision at k 54
6.2. Results and Discussion 54
6.2.1. n-Gram Parameter 55
6.2.2. Feature Importance 55
6.2.3. YAKE! vs Baselines 59
7. Demo **63**
7.1. Individual Exploration of the Results 63
7.2. Aggregated Exploration of the Results 65
8. Conclusion and Future Work **67**
References **69**

List of Figures

Fig. 1 - An example of a sociogram. Source: futures.armyscitech.com 6
Fig. 2 - Structure of a Facebook post 7
Fig. 3 - Example of google+ post. Source: searchwilderness.com 8
Fig. 4 - Number of worldwide active Twitter users. Source: statista.com 9
Fig. 5 - Tweet post interface 10
Fig. 6 - Donald Trump post on 14/01/2018 11
Fig. 7 - Keyword extraction architecture. 15
Fig. 8 - Workflow of collecting and storing the tweets for the top-100 twitter user's 30
Fig. 9 - Annotation task process 33
Fig. 10 - Annotation application login screen 35
Fig. 11 - Annotation task interface 35
Fig. 12 - Number of gold keyword per tweet 38
Fig. 13 - Number of terms per keyword 39
Fig. 14 - Publications of tweets per day 39
Fig. 15 - Total Google tweets per day 40
Fig. 16 - Total FC Barcelona tweets per day 40
Fig. 17- Total ESPN tweets per day 41
Fig. 18 - Relationship between the #Followers and the #Tweets. 1e7 means 1*10^7 number of tweets 42
Fig. 19 - Word cloud of the 25 Twitter user's keywords 43
Fig. 20 - Word cloud of the hashtags from tweets text 45
Fig. 21 - Number of hashtags per day 46
Fig. 22 - YAKE! Architecture. Obtained from Campos et al. [1] 48
Fig. 23 - YAKE! MAP@10 Effectiveness on top of the KWTweet dataset when $1 \leq n \leq 5$ 55
Fig. 24 - YAKE! Feature Importance - MAP@10 effectiveness of YAKE – (KFkw, TCase, TFNorm, TPos, TRel and TSent) features. bl means baseline. 56
Fig. 25 - Individual keyword cloud for the "RealDonaldTrump" username 64
Fig. 26 - Individual keyword cloud for the "websummit18" hashtag 64
Fig. 27 - Aggregated keyword cloud for the "RealDonaldTrump" user 65
Fig. 28 - Aggregated keyword cloud for the "websummit18" hashtag 66

List of Tables

Table 1 - Keyword Extraction Approaches Summary. NA – Not Available 24
Table 2 - YAKE (left) and IBM (right) top-5 keywords for document 8 of Inspec collection. Keywords identified by both are printed in boldface 26
Table 3 – KWTweet Dataset Stats..... 32
Table 4 - Total Number of tweets per group of twitter users 33
Table 5 - Total number of tweets per twitter user and associated group..... 34
Table 6 - Number of tweets labelled per annotator 36
Table 7 - Example of the data used to calculate the inter-agreement of 3 annotators..... 37
Table 8 - YAKE! Feature Importance - MAP@10 effectiveness of YAKE – (KFkw, TCase, TFNorm, TPos, TRel, TSent, TFNormTSent, TFNormTSentTRel, TFNormTPos, TSenTPos, TSenTRel, TFNormTRel, TFNormTSenTPos) features. bl means baseline. ... 57
Table 9 - YAKE! Feature Importance - MAP@10 effectiveness of YAKE – (TCase, TCaseTFNorm, TCaseTSent, TCaseTrel, TCaseTPos, TCaseTFNormTPos, TCaseTFNormTRel, TCaseTSentTpos, TCaseTsentTPos) features. bl means baseline. ... 58
Table 10 - YAKE! Feature Importance - MAP@10 effectiveness of YAKE considering that S(t) = TCase, S(t) = TRel, S(t) = TSent, S(t) = TPos, S(t) = TFNorm. bl means baseline. 58
Table 11 - YAKE! effectiveness vs Baselines methods. P@10, R@10, F1@10 and MAP@10. Results are shown in descending order of the MAP@10 score. 60

Chapter 1

Introduction

The exponential growth of the information over the last few years has made it impossible for any user to handle and manipulate one such volume of data. A lot of this information comes from social networks, like Facebook, Google +, Twitter or LinkedIn to name just a few. Processing this information may reveal interesting patterns and useful knowledge. One way to handle this information is through keyword extractors. Their goal is to select the best relevant keywords in a way that texts get easily summarized. While keyword extractors have been widely applied to generic texts, their applicability to texts of short nature, such as tweets, is scarce. In this work, we aim to overcome this shortcoming by evaluating the effectiveness of YAKE! [1] (an unsupervised keyword extractor algorithm) on top of a collections of tweets, a different kind of texts that is characterized not only by its short length nature, but also by its unstructured and sometimes noisy text. Our aim is to understand whether this algorithm may be applied to this kind of texts, when compared to similar baselines. In the following we give an outline of this work. Section 1.1 details the motivation goals. Section 1.2 describes its main challenges. Section 1.3. presents its main contributions, Finally, Section 1.4 outlines the rest of this work.

1.1.Motivation Goals

With so much information available on the web, automatically processing texts turned out to be a core step for several tasks, including, text summarization, text clustering or information retrieval to name but a few. This requires having access to the most important terms of a text. In this work, we aim to evaluate the effectiveness of YAKE! [1] algorithm (Best Short Paper of ECIR'18 – 40th European Conference on Information Retrieval) on top of a dataset consisting of short texts, namely tweets, and verify its effectiveness when compared to state-of-the art algorithms. This will give us insights into the appropriateness of YAKE! when dealing with this kind of texts.

1.2. Challenges

Social networks, such as Twitter, incorporate a huge amount of information that may convey useful knowledge. Exploring such sources may be done with resort to a number of techniques, including the use of keyword extractors. While keyword extractors have been widely used for a myriad of documents, their applicability to small texts, such as tweet posts, is very scarce. One such type of documents, poses some challenges mostly due to their inherent characteristics, such as its small size, informal language or noisy text. Extracting relevant keywords from this kind of texts, is as such, a difficult and challenging task that motivates this work. Our aim is to understand whether YAKE! [1], which has been applied to generic texts such as reports, news, scientific papers, etc, can also be applied with satisfiable results to this kind of texts. In order to conduct this analysis, we need an appropriate dataset made of tweets. While keyword extractors have been applied to generic texts, there is a lack of resources when it comes to evaluate their effectiveness on top of tweets. This may be faced as an additional challenge, and the reason behind the development of a formal dataset that we ended up making available to the research community.

1.3. Contribution

Our research produced some scientific contributions as well as a dataset and a demo that we made available for the general public and the research community. In this section we outline the main ones.

(1) We provide an extensive literature review, including a summary comparison of similar keyword extractor approaches [Chapter 3];

(2) We publicly provide a dataset [https://github.com/LIAAD/KeywordExtractor-Datasets#KWTweet] of labelled tweets to the scientific community giving insights into its main characteristics [Chapter 4]. Part of this work has been submitted to the short paper track of the 41st European Conference on Information Retrieval (ECIR'19).

(3) We compare the effectiveness of YAKE! against 9 state-of-the-art algorithms (TextRank [2], KP-Miner [3], SingleRank [4], PositionRank [5], TopicPageRank [6], MultipartiteRank [7], TopicRank [8], Rake [9] and TF.IDF [10]) including three additional approaches where hashtags are considered keywords, users are

considered keywords and a combination of both (hashtags and users) are considered keywords [Chapter 6];

(4) We make available a demo [http://bit.ly/TwitterYAKE] to showcase the results of YAKE! when applied to tweets. Users will be given the possibility to query Twitter through a hashtag or username and to be given the best relevant corresponding keywords [Chapter 7].

1.4. Outline

The remainder of this work is structured as follow. Section 2 provides information about the evolution of social networks and their characteristics, with particular emphasis to Facebook, Google+ and Twitter. Section 3 begins by describing the overall architecture that is behind a keyword extractor system. Afterwards, we introduce the related research work differentiating between supervised and unsupervised algorithms. Section 4 details the process behind the development of the KWTweet dataset. A detailed descriptive analysis of the characteristics of this dataset has also been made available here. Section 5 is dedicated to stating the problem definition of our work. Once the problem is defined, we will describe the architecture behind YAKE! in a simple manner as this has been already described in Campos et al. [1]. Section 6 evaluates the effectiveness of YAKE! and of state-of-the-art algorithms on top of the KWTweet Dataset. Section 7, showcase a demo of Twitter-YAKE!, where we offer the user the chance to query Twitter via a username or hashtag and to get the relevant keywords in a word cloud fashion. Finally, Section 8 provides the conclusions and future work.

Chapter 2

Social Networks: Literature Review

Over the last few years, an incredible amount of information has been published on the Internet. Much of this information has been published in the form of news articles, documents, or multimedia files. The arisen of social networks, however, has changed the paradigm leading users to publish a whole new set of documents, made of Facebook posts, blogs, LinkedIn publications or even tweets. In this chapter we aim to present figures and facts of social networks before detailing keyword extraction approaches (in chapter 3), as social networks play an important role in this work. In particular, we will focus on their contribution to the increase, sharing and dissemination of information. More to the point, we will describe some social networks, differences between them and their underlying characteristics. The rest of this chapter is organized as follows. Section 2.1 gives a brief introduction on social networks and their history. Relevant research carried out having as a basis Facebook, Google+ and Twitter will then be described in Section 2.2, Section 2.3 and Section 2.4 respectively.

2.1. Introduction

Social networks are well established virtual communities, usually composed of hundreds, millions or even billions of different entities, mostly people and organizations [11] looking to connect with others (possibly unknown entities), to share their contents and realizations, to search for breaking news and events or even to make business. They can be represented by sociograms, visual representations of the network conveying a huge amount of valuable information which would be impossible or at least difficult to perceive in any other case. Fig. 1 represents a sociogram where the entities/nodes are identified by figures and the relationships between them are identified by lines

.

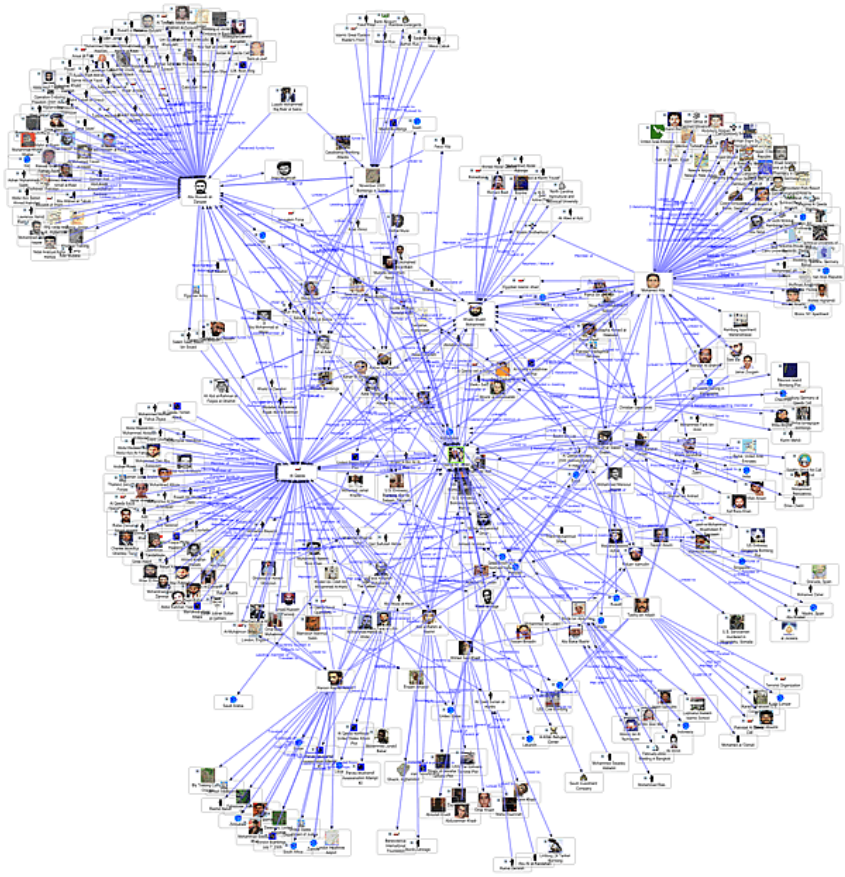

Fig. 1 - An example of a sociogram. Source: futures.armyscitech.com[1]

Sociograms like these, are the result of a massive interest and participation of people and organizations on social networks. Although most of us can only remember Facebook, Twitter or LinkedIn, there is an all realm back to 1997 when [SixDegress.com] first appeared [12], later followed by [LiveJournal.com], AsianAvenue [asianave.com/] and [BlackPlanet.com] in 1999. Eventually, SixDegress.com came to an end and it took four years to the point of no return when [LinkedIn.com], [Couchsurfing.com], [MySpace.com] and [Hi5.com] emerged. These were later followed by [SecondLife.com] in 2003, [YouTube.com] in 2005, [Facebook.com] in 2006, [Twitter.com] in 2006, and [Google+] in 2011. A detailed chronology of the history of social networks can be found on Ellison et al [12] and Miller et al [13]. In the following, we describe Facebook, Google+ and Twitter in more detail. We will specifically focus on Twitter as this will be the basis of this work, though introducing relevant research concerning Facebook whenever appropriate. This contrasts with Google+ for which relevant research is very scarce.

[1] https://futures.armyscitech.com/ex5/marketplace/automate-cultural-and-social-network-analysis/ [accessed on 23/01/2018]

2.2. Facebook

Facebook was born in February of 2004 and has 2072 million[2] active users by the end of the year 2017. It was founded by Mark Zuckerberg, Eduardo Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes, and it is probably one of the most impactful social networks [14] in terms of human society including changes and effects on human behaviour. It enables to share multimedia content and text, and applications such as games and live videos, being a huge source of information for people and companies to make use of. Fig. 2 shows two different ways to post on Facebook, i.e., by text (up to 55k characters according to the Facebook Community[3] - may or may not include multimedia content) or by live video. In both cases, it is possible for anyone to attach his/her feelings or mood.

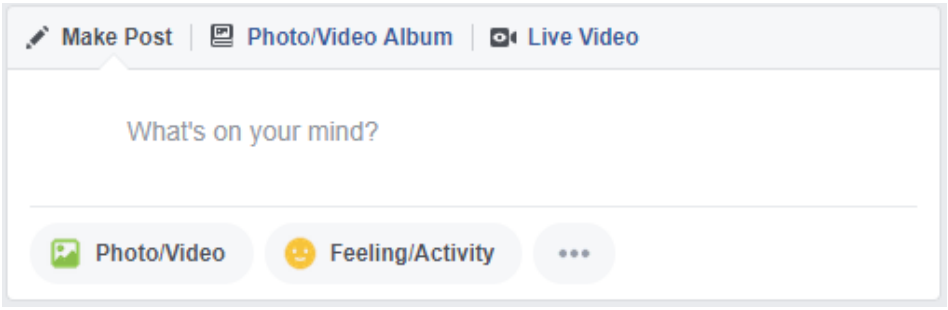

Fig. 2 - Structure of a Facebook post

The way people publish and share their contents poses however some challenges mostly concerning privacy and security issues. Young generations for example, are all equipped with smartphones, cameras, smartwatches or a bunch other equipment's such that sharing a photo, commenting on a post or publishing a video may be done on-the-fly. In most cases, however, people are not even aware of what they are publishing, nor of the risks they are running. A cross-cutting issue that affects not only young people, but also older generations for which Facebook is the entry point to the Internet. The way people publish and share their contents, and the privacy policies of Facebook have been the target of study over the last few years [15], [16], [17] with several works attesting that the big majority of people simply do not care or are not aware about privacy issues, leading them to disclose information without any restriction or concern.

The policies of Facebook with regard to the procedures of this social network as for the death of its users is concerned, has also been the subject of study [18]. While

[2] https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/ [accessed on 27-01-2018]
[3] https://www.facebook.com/help/community/question/?id=1473679909539541 [accessed on 16-01-2018]

Facebook's first option was to remove the user's account, today it transforms it in a memorial as a way to preserve and maintain people's sharing's alive, beyond death.

The potential of Facebook as a way to influence health and social behavior has also been the subject of several research studies. Fu et al. [19] for example, studies how social networks such as Facebook influence the tobacco use and its cessation. Fardouly et al. [20] in turn, studies how the act of sharing photos by users on Facebook social network, can cause dissatisfaction and sorrow on fellow other users who will be making comparisons at the body level or lifestyle. The way people express their emotion and their relationship with Facebook has also been studied over the last few years [21] [22]. What all these studies convey is that social networks such as Facebook have turned into a huge resource of knowledge readily to be used not only by Facebook itself, but also by users and companies alike, as a way to understand how people act, react and get affected and or influenced.

2.3. Google+

Google+ was formally launched in June 2011 by Google. By the end of October 2013, it had around 540 million active users[4], yet, and as of news of October 2018, it has apparently come to an end, which is notorious in the number of reduced papers related to it [Anderson & Still [23], Landerweerd et al. [24], Osborne & Dredze [25]]. Fig. 3 represents an example of a post in Google+ where a random user publishes a text concerning JavaScript.

Fig. 3 - Example of google+ post. Source: searchwilderness.com[5]

[4] https://www.thesocialmediahat.com/active-users [accessed on 10-02-2018]
[5] https://searchwilderness.com/new-google-plus-interactive/#gref [accessed on 27-01-2018]

Differently from social networks, but still under the umbrella of providing communication facilities, Google also has an instant messaging and video platform called Hangouts, which is available on both mobile (android and iOS) and web platforms. It allows one to send short messages of up to 150 members using the internet, thus directly competing with the SMS (Short Message Service) which are tendentially less used.

2.4. Twitter

Unlike Google+, Twitter has been gaining an increasing importance over the last few years from an "inexpressive" 30 million of users in 2010 to an incredible 330 million at the end of 2017 (see Fig. 4). On October 2018 it was ranked in the 11th position among social networks with more active users[6], only surpassed by short messaging applications (WhatsApp, Facebook Messenger, and WeChat) and social networks such as YouTube, [Instagram.com], and [Tumblr.com], which only reflects its increasing importance.

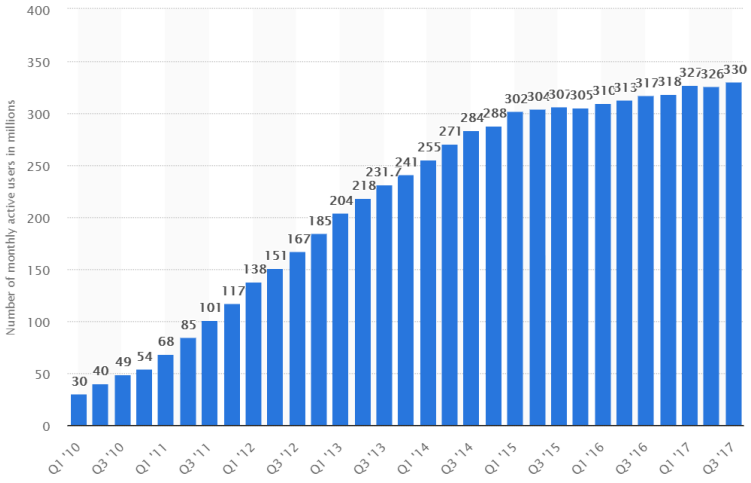

Fig. 4 - Number of worldwide active Twitter users. Source: statista.com[7]

Personalities like Donald Trump, Rihanna, Shakira, NASA, CNN and even other social networks such as Facebook, Instagram and YouTube share their content on this social network. Fig. 5 portraits the interface of a Twitter post.

[6] https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/ [accessed on 10-02-2018]
[7] https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/ [accessed on 20-01-2018]

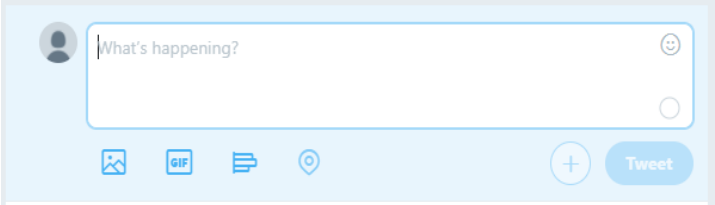

Fig. 5 - Tweet post interface

Unlike Facebook, posts on Twitter are short pieces of text consisting of up to 280 characters. The idea of disseminating information limited to just a few characters is not new, and may have the roots on short message texts, an old technology, which during several years was the preferred way to communicate with others. While 280 characters still seems too short, it is twice the double of 140, which was the leading rule of Twitter since the beginning of this social network until last September (2017), when Twitter decided to double the size[8]. One such difference, enables people to not only write in a more detailed manner and towards a particular subject, but also to publish more information. In a society, where publishing and consuming information turned into a way of life, this can make a huge difference for both users, publishers and consumers of this information (organizations included). Along with this, Twitter has two other peculiarities that distinguish it from other social networks, that is, the hashtag (#) and the username (@). Hashtags (#) are keywords related to an information, topic, or discussion taking place on Twitter. These are defined by the hashtag (#) followed by the keyword, thus allowing users to easily label and classify the data that is being discussed. An example of a hashtag is "#hurricaneirma" or "#irma" both related to the hurricane that occurred in August/September 2017 in the United States of America. Beyond being a crucial feature for the identification of the shared contents, hashtags also allow users to search for information in a quick manner. For example, users interested on information about Donald J. Trump, could simply search for the hashtag "#DonaldTrump", which, would retrieve, in return to this search, all the tweets marked by this hashtag.

Instead, the username (@), describes an entity or group defined in twitter, with the purpose of associating a unique username to an entity. In this case, the entity or group username is used after the symbol "@". For example, "@realDonaldTrump" is the username of Donald J. Trump, "@BarackObama" is Barack Obama's, and so on and so

[8] https://blog.twitter.com/official/en_us/topics/product/2017/Giving-you-more-characters-to-express-yourself.html [accessed 20-01-2018]

forth. Another important aspect is the retweet (RT) functionality which enables to simply re-post another user's tweet on the user's own profile. Retweet's are indicated in the posts themselves and exist to allow people to share discussions more easily. Fig. 6 portraits an example of a Donald Trump tweet published on January 14, 2018. In this tweet, Donald J. Trump sends a note for people aiming to move to the United States of America. By looking at the picture one can observe both hashtag (#AMERICA), username (@realDonaldTrump) and retweet (denoted by "Donald J. Trump Retweeted").

Fig. 6 - Donald Trump post on 14/01/2018

The study of this kind of messages may be very interesting and revealing, not only for political purposes (as is the case of predicting results elections [26]), but also for natural disasters response [27, 28], outbreak detection [29, 30, 31], sentiment analysis [32, 33] or crime prediction [34]. It may also be used to terrorist activity detection [35] or to cover a live (even if deadly event). This was the case in 2011, when the news of the death of Osama bin Laden first emerged on Twitter [36] before being communicated to the public or/and to the media by Barack Obama (president of the USA at that time).

Along with hashtags and usernames, there are also two other important features which may be part of a tweet: one is geolocation, the other is temporality. The first enables to tag a tweet with a geographic context. The later allows to anchor it on the timeline. Both have led to the emergence of a number of research articles within this context in the last few years. In particular, Han et al. [37] tries to predict the location of the twitter users at the city level based on the content of the text they share. Another research is the work of Chandra et al. [38] who also aims to predict the location of Twitter users, based on a probability distribution model. 500K tweets from 10,584 users of USA were used for training the model, while 600K tweets from 540 users were used for testing. The results obtained show an accuracy of 58.88% with a distance error of 300 miles, but it lowers to a range 10% and 22% (for each of the two models introduced) for distances up to 100 miles from the original location. Other works [39, 40, 41, 42, 43, 44, 45, 46] refer to the

detection of spatiotemporal events (e.g. crimes, protests, disasters, diseases) lay based on geotagged tweets. Sakaki et al. [39] for instance, makes use of tweets to detect earthquakes faster than the JMA (Japan Meteorological Agency). Lee et al. [41] for example, uses the tweets' location, time and text information for a surveillance flu and cancer system that may assist both doctors and patients on their decisions. Cheng & Wicks [42] in turn, aims to obtain spatiotemporal information about popular events on twitter, by applying statistical methods. A particular analysis to the London helicopter crash in 2013, based on 1.8M tweets collected from the UK between the dates of January 7 - 18, 2013, shows that most of the tweets were generated in large urban agglomerations such as London, Birmingham and Manchester and that tweets were mostly posted on main routes and train lines.

In all presented cases we can find slang and jargon vocabulary. The slang vocabulary is used in informal situations when people know each other well and may be particular or developed within a certain group. Some of these slang words stay in the language for a long time (e.g., "bae" which is a term of affection that is used in romantic relationships but also by intimate friends or "woke"[9] which means "awakening" to social injustices. "If you're so awake, why did not you vote?") but many of them disappear and others are invented. Another characteristic of slang is that it is often associated with speech rather than to writing.

Jargon vocabulary in turn, is more specific to a certain profession or activity. Terms such as "Yoda Conditions" (when two parts of an expression are reversed from the normal order of a conditional statement "if tall is the man") or "Pokémon Exception Handling" (the goal is to catch an error in an exception) for example, may be easily understood by programmers[10] but quite strange for people outside the area. In the following we describe the process of extracting keywords from text, in general and in the particular case of social networks.

[9] http://examples.yourdictionary.com/20-examples-of-slang-language.html#fDFgPrPb5KII7ukT.99 [accessed on 20-01-2018]
[10] https://blog.codinghorror.com/new-programming-jargon/ [accessed on 20-01-2018]

Chapter 3

Keyword Extraction: Architecture and Literature Review

The advent and intensive use of social networks has led to an incredible sheer volume of data to digest, making it impossible for any human to manually process it in a timely manner. To overcome this problem, researchers, professionals and common people alike have resorted to keyword extractors as a way to automatically digest and process the information. The goal of a keyword extractor is to extract keywords from text documents. Although keyword extractors have been extensively applied to generic texts, their applicability to texts of short nature is scarce and only a few works have tackled this problem. In this chapter, we aim to detail the architecture of the keyword extraction process and present the state-of-the-art of keyword extraction approaches. The rest of this chapter is organized as follows. Section 3.1 states the importance of using keyword extractors within the context of social networks and other important core areas. Section 3.2 describes the overall architecture of a keyword extractor. Finally, Section 3.3. presents a detailed analysis of the relevant literature within the keyword extractor domain. In particular, we divide our analysis between unsupervised and supervised approaches.

3.1. Applications of Keyword Extractors

Social networks are perhaps the most important platform nowadays for people and organizations to share their contents. They are supported by an increasing number of users, who constantly publish new documents. While this growth has been stabilizing over the last few years an increasing number of users can still be observed in most of the social networks. Much of these users are not only consumers but also producers of huge amounts of data, thus giving rise to a plethora of information ready to be explored. Analysing this information, makes it possible to know a lot about people and/or organizations, including the relationships people have, the places they visit, their likes, musical tastes, opinions, interests, and so on and so forth. One such amount of data may be of interest to anyone looking for relevant information, yet it would not be possible to digest without an automatic procedure.

The process of automatically extracting keywords appears in this context as an attempt to understand texts and to extract relevant information from documents. According to Beliga [47], it can be described as the "task that automatically identifies a set of terms that best describes the subject of a document". Basically, it is an automatic task where keywords are extracted from texts, enabling people to have a quick glimpse of it without even reading it. They are usually formed by single terms (e.g., in the expression "Car sales" we can have two keywords "Car" and "Sales") but may also consist of a set of joint words also known as *n*-grams (e.g., in the previous example we can have the bi-gram "Car sales" as keyword). Regardless the case, both have the same purpose, i.e., to characterize a text. For instance, this can be used by organizations to follow a well-known competitor, by politicians to infer their popularity, by lawyers and librarians to automatically label their texts, by jobseekers to summarize resumes, by marketeers to preview trending topics or by media outlets seeking for information about what famous personalities say or do in their daily life.

While the process of extracting keywords is an important step in the context of Text Mining (TM), it also plays an important role in several other core areas, including Information Retrieval [48], [49], Text Summarization [50], Text Categorization [51], Opinion Mining [52], Clustering [53], and Text Analytics and Visualization [54], [55], [56], to name just a few. In the following, we detail the architecture of the keyword extraction process, which consists of 5 steps.

3.2. Architecture

In this section, we describe the architecture that guides the extraction of relevant keywords. The overall idea of the process is to identify keywords which are relevant for a given text on five different steps depicted in Fig. 7 and explained in the remainder of this section: (1) the pre-processing stage; (2) the generation of the candidate keywords list; (3) the feature extraction process; (4) the scoring stage, where the candidate keywords are given a score; and (5) the ranking of the results. This will build the foundations for the application developed in this work: keyword extractor from tweet texts.

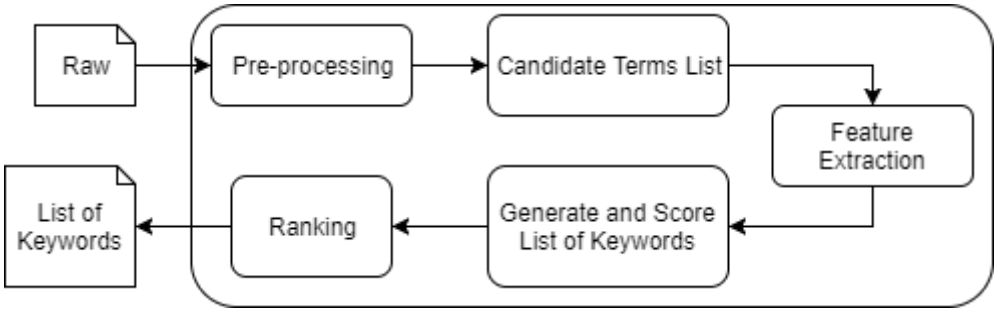

Fig. 7 - Keyword extraction architecture.

3.2.1. Pre-Processing

The first step, known as the pre-processing stage, receives a raw document and applies Natural Language Processing (NLP) techniques to generate a list of candidate keywords. NLP is a core area in the text mining domain, which is used by computers to translate the human language behind any unstructured text, to a more readable parsed format. A typical pre-processing procedure usually involves cleaning the text, sentence splitting, tokenization and text annotation. The process of cleaning the text is rather simple and involves the elimination of unnecessary symbols or strange characters. Sentence splitting, in turn, concerns splitting a sentence according to the defined punctuation and it may be used as a way to maintain and keep track of the list of sentences. The next step is the tokenization process, which takes place before the text annotation stage. Its aim is to transform a sentence into individual units, called tokens. For example, the string "My conduct is always professional" is implicitly segmented into tokens based on spaces. While both processes seem fairly simply, they can be quite challenging as different languages offer different problems. Thai for example, does not use a period as a sentence delimiter. The use of space is also unreliable for languages such as Chinese or Japanese, and even languages with well-known punctuation marks may present surprising problems. A period, for example, is usually associated to the end of a sentence, but it may also be used as a decimal point or even in abbreviations (e.g., Mr. Smith, would be segmented into two sentences, should a simply rule-based model be used). The issues and challenges of tokenization and segmentation are greatly described in the works of Palmer [57] and of Read et al. [58]. Finally, the text annotation procedure involves a linguistic methodology that may or may not include part-of-speech (POS), Named Entity Recognition (NER), Stemming and the use of stopwords. Each on these will be described in more detailed in the coming lines:

- PoS, for example, is the process that aims to assign grammatical categories to the text, such as nouns, verbs, adjectives, adverbs, pronouns, prepositions, conjunctions and interjections, to parts of the text. For example, in the sentence "My conduct is always professional" the word "conduct" is a noun. Yet the same word is a verb in the sentence "I conduct myself in a professional manner".
- NER, in turn, is an information extraction subtask that seeks to locate and identify entities within a text, including, names of people (e.g. "Donald Trump"), organizations (e.g., "IBM") and locations (e.g., "New York"), among other categories.
- Stemming is the process of reducing inflected (or derived) related words to their base or root form. For example, if the word ends with "ed" (e.g., "Played") the "ed" of the word is removed (thus resulting in "play").
- Finally, stopwords are extremely common words that can be considered irrelevant in terms of describing the content of a text. These words are filtered before or after processing natural language data (text). Some examples include the word "about", "after" or "before", but there are many more[11] (for instance, for the text "My conduct is always professional" the stop words would be "My", "always" and "is").

Each of these steps is characterized by being language dependent which means that depending on the language, a different tool may be required to further process them.

3.2.2. Candidate Terms List

After the pre-processing stage, the second step compiles the final list of candidate terms. This can be done through PoS patterns or *n*-grams generation. For the first one, PoS-tag sequences are usually used to select the words as candidate terms, typically a sequence of nouns and adjectives. For the latter a sliding *n*-gram (sequence of sliding *n* items in a text) may be used. For instance, the phrase "I live in NY" can be represented by *uni*-grams (e.g., "I", "live", "NY"), *bi*-grams (e.g., "I live", "live in", "in NY"), *tri*-grams ("I live in", "live in NY"), and so on and so forth.

3.2.3. Feature Extraction

In the third step each candidate term is represented by numerical features, likely to capture the nature and the importance of the term. This may include, for example, term frequency

[11] http://xpo6.com/list-of-english-stop-words/ [accessed on 20-01-2018]

features, such as how frequent a term appears in the text; positional features, if the position of a term is an important clue; or even linguistic features, such as casing. Joorabchi & Mahdi [59] introduces a few of them, of which we highlight Term Frequency, First Occurrence, Last Occurrence, Occurrence Spread, Length and Lexical Diversity. A description of each one of them is given in the coming lines. **Term Frequency** (TF) describes how frequently a keyword occurs in a document. These values are divided by the highest TF value so as to normalize the value in a range between 0 and 1. The keywords being closer to 1 have more relevance. **First Occurrence** describes the distance between the beginning of the document and the first candidate keyword. This feature points to discover the candidate keywords that happen at the beginning of the documents, such as titles. **Last Occurrence** aims to compute the distance between the candidate keyword and the end of the document, as a way to find candidate keywords near the conclusions of the document. **Occurrence Spread** is translated by the distance between the first and the last occurrence of a candidate keyword. The underlying idea is to infer whether a given candidate keyword is or not spread throughout the document. **Length** corresponds to the size of the keyword, on the assumption that candidate keywords consisting of several words are more likely to be a keyword. **Lexical Diversity** refers to the fact that any word in a document can assume several lexical forms. In the case of Joorabchi & Mahdi [59], lexical diversity is calculated lay based on case-folding and stemming.

3.2.4. Scoring

In the fourth step, the characteristics defined in the feature extraction process will be used to generate a list of keywords and to assign them a corresponding weight. This weight defines the importance of the word and may be obtained by means of an unsupervised **graph-based approach** (TextRank [2], SingleRank [4], ExpandRank [4]), TopicPageRank [6], TopicRank [8], PositionRank [5], MultipartiteRank [7], **statistical methods** (Yake [1], Rake [9], TF.IDF (which is the product of two statistics, term frequency [60] and inverse document frequency [10]), KP-Miner [3] or **supervised machine learning methods** (KEA [61], MAUI [62]). More details of this scoring stage will be given in Section 3.3 when introducing each one of these works.

3.2.5. Ranking

The number of candidate terms could range from tens to thousands depending on a number of factors including the length of the document and the algorithm used. In this step, distance measures like Levenshtein [63] or Jaro-Winkler [64] are used as a way to deduplicate syntactic similar terms. In particular, the Levenshtein's measure is a method to compare two sequences where the results is the minimal number of operations that are necessary in order to transform one sequence into another. Instead, Jaro-Winkler's is based on the similarity between words and consists of the application of a formula that takes into account the number of correlations between units, the size of both sequences and the number of transpositions. Finally, the weights are then used to order the words producing the final list of ranked keywords.

3.3. Keyword Extraction Approaches

While the problem of extracting relevant keywords from documents is not new and has been tackled over the last few years, only a few works have considered to apply them to documents of short nature. In the following two sub-sections, we describe a number of solutions applied to both realms. Papers related to Twitter will also be presented. The remainder of this chapter is organized as follows. Section 3.3.1 presents approaches concerning unsupervised methodologies. Section 3.3.2 describes solutions related to supervised approaches.

3.3.1. Unsupervised Methodologies

Unsupervised methods are characterized by the inexistence of a training process thus avoiding the need to have labelled information. In this approach, result are usually obtained through a set of features capable of encoding the intrinsic properties of a keyword.

When talking about keyword extractors, TextRank [2], SingleRank [4], ExpandRank [4], KP-Miner [3], TopicPageRank [6], RAKE [9], TopicRank [8], PositionRank [5] and MultipartiteRank [7] are some of the most-well known unsupervised methodologies. In the following we describe each one of these works in more detail, together with YAKE! a newly research recently developed by Campos et al. [1], which is in the basis of this work. When talking about unsupervised methods, a fundamental division arises between statistical methods and graph-based ones. These will be introduced in Section 3.3.1.1 and

Section 3.3.1.2 respectively. In addition to this, unsupervised works that are under the umbrella of social media will also be introduced in Section 3.3.1.3.

3.3.1.1. Statistical Methods

The baseline method in unsupervised approaches is **TF.IDF** [10], which compares the frequency of a term in a document with regards to the whole collection. However, over the years, other approaches have been developed. Rafea & El-Beltagy [3] presents **KP-Miner**, an heuristic approach, which may be divided into 3 distinct stages: (1) Candidate keyword selection, (2) Candidate keyword weight calculation, (3) Final Candidate Phrase List Refinement. In the first step, candidate keywords are defined according to a number of pre-defined rules. In the second step, TF.IDF and two boosting factors (word length and position in the document) are taken into account in order to determine the importance of a keyword. Finally, KP-Miner sorts the list of candidate words and returns the keywords. Rose et al. [9], in turn, describes the use of **RAKE**, which is undoubtedly one of the most well-known approaches of this kind. Initially, candidate keywords are selected using text delimiters and stopwords. Each candidate keyword is then assigned a score based on a number of features extracted from the text. This includes the raw word frequency, word degree (i.e., the number of times a candidate word co-occurs with another candidate keyword within a window of 1) and a ratio between both measures. The final score of each word is then computed by applying an heuristic methodology which simply sums the scores of each of the three features. More recently a new statistical approach, named **YAKE!** was proposed by Campos et al. [1] to extract relevant keywords from single documents. In this work, the authors devise and combine a number of features to describe the nature of each term. The same authors have proposed and presented a demo [65] [http://yake.inesctec.pt/] where their methodology can be used and tested, either in the site or by means of a Python package [https://pypi.python.org/pypi/yake]. Both works, form the basis of this work and will be described in more detail on Section 5.

3.3.1.2. Graph-based Methods

TextRank [2] is certainly one of the most famous approaches of this kind. It assumes an unsupervised methodology based on graphs to extract relevant keywords, where words are represented as vertices. A co-occurrence relationship between vertices is established if words co-occur to a certain extent. After the graph is constructed, the score associated to

each vertice is set to an initial value of 1, and a ranking algorithm similar to Google's PageRank [66] is executed to determine the most relevant words. Based on TextRank, Wan & Xiao [4] propose the **SingleRank** algorithm which is another example of a graph-based approach. In this work, a graph is created based on the candidate words of a document and a classification algorithm is applied to define the value of each word and, the words with the highest value are considered keywords. Another extension of the TextRank algorithm is the **ExpandRank** proposal [4]. In this work a neighbourhood is initially built where the main document is divided into smaller documents called neighbouring documents. At the level of the neighbouring documents, a graph is then created based on all the candidate words and a classification algorithm is applied to define the score of each word. Then an evaluation is made to the candidate phrases (based on the score of the words previously classified) being choose the keywords with the highest score in the document. More recently other works have been proposed within the graph-based field. **TopicPageRank** [6] presents a graph-based approach to extract keywords through topics. In the first step, topics may be generated either through manually annotated knowledge bases or through unsupervised machine learning techniques. Once the topics are generated, a word graph is then constructed for each topic and the importance of each keyword is calculated for each topic. Finally, top scored topics are returned as keywords. **TopicRank** [8] emerges as an improved version of TextRank that uses graphs where vertices are not candidate keywords but topics. A topic is defined as a cluster consisting of words or sets of similar words. After the graph is created, TextRank is used to classify the topics. Finally, to select the gold keywords of the document, a candidate keyword is selected from each of the highest ranked clusters. In turn, **PositionRank** [5] uses the position information of the occurrence of words. First, nouns and adjectives as selected as candidate keywords. A word graph is then built on top of each candidate keywords and weights are assigned to these words based on PageRank. The main characteristic of PositionRank is to assign a higher probability to keywords that are found at the beginning of a document. The most recent approach however, is the work of Boudin [7] who developed **MultipartiteRank**. In this work, a topic graph is generated where candidate keywords are associated with each topic. After the generated graph, the TextRank is applied in order to classify the candidate keywords. The top-*n* scores are then selected as keywords.

3.3.1.3. Social Media

Another strand of research works lies within texts of short nature. In Timonen et al. [67], small texts of about 30 to 60 words consisting of product descriptions, movie descriptions and events are considered. In this work, an unsupervised keyword extractor model called Informativeness-based Keyword Extraction (**IKE**) is applied to extract relevant keywords based on a clustering algorithm. Another research work based on twitter was introduced by Zhao et al. [68] who proposes to extract keywords as a way to summarize Twitter posts. In particular, a context-sensitive topical PageRank (**cTPR**) method to identify keywords is applied, based on the number of co-occurrences of two words within a certain window size and a given context. This means that even if the word "apple" appears close to "juice" it will hardly be considered a relevant word in the context of "electronic products". Each keyword is then ranked according to a probabilistic ranking model which takes into account two features, relevance and interestingness. A twitter dataset from Singapore users, collected from December 2009 to April 2010, was used to evaluate the proposed system. However, neither the dataset nor an implementation of the proposed method is available, hindering a comparison against this work.

More recently, Marujo et al. [69] describes an extension to the MAUI algorithm [70] which extracts a list of candidate keywords from a document and trains a decision tree to predict relevant keywords based on a number of features. In their work, the authors extend the number of features assigned to a candidate keyword making use of two unsupervised methods: the brown clustering and the continuous word vectors. Brown clustering is used to group lexical variants, where for example the words "yes" and "yesss" are placed in the same cluster. In the continuous word vectors approach, a hidden layer that maps the words to a continuous vector is defined. Their approach is then evaluated on top of 1827 tweets. The authors claim an F1-M of 71.61% for the MAUI (Brown + Word Vectors) approach [70], yet, similar to the previous work nor an implementation neither a dataset is available, thus hindering the reproducibility of the experiments.

In contrast, the works of Wu et al. [71] and Wang et al. [72] resort to Twitter and to keyword extractors as a way to approach different purposes. Wu et al. [71], for example, introduces a system to automatically generate annotation tags to label Twitter user's interests and concerns, lay based on TF.IDF [10] and TextRank [2]. Wang et al. [72], in turn, proposed the **Double Ranking** approach, a methodology whose aim is to identify

search keywords to find tweets on Twitter, rather than identifying relevant keywords for a given tweet. In the following, we describe the approaches that fit within the supervised methodology.

3.3.2. Supervised Methodologies

A supervised methodology is a method that uses labelled data for training. The training data consists of a set of examples labelled with the corresponding output. Examples of well-known supervised methods are support-vector-machines (SVM) [73, 74], Decision trees [75] and Naïve Bayes classifiers [76]. The use of these methods can be widely found in bioinformatics [77], Pattern Recognition [78], or spam detection [79], to name but a few, but also in information retrieval [80] and keyword extraction, of which the work of Witten et al. [61] is certainly the most well-known and recognized research. In this work, the authors propose **KEA** as a way to identify relevant keywords. To this regard, they apply the Naïve Bayes classification algorithm. In the first step, a learning model is created using training documents where the authors identify the words. In the second step, the learning model is used to determine the best relevant keywords from a new document given as input. Kea was evaluated from a collection of 1800 documents, of which, 1300 were used for training and the remaining 500 for testing. Turney [81], in turn, uses the C4.5 decision tree induction algorithm [75] for the learning task and the **GenEx** algorithm for extracting keywords, based on features such as phrase frequency, common verbs or word frequency. For the evaluation stage, five different datasets going from news articles, web pages and emails were used. The obtained results show that the GenEx algorithm is able to obtain better results when compared to the C4.5 algorithm. Witten et al. [62] presents an approach called **MAUI** (automatic multi-topic indexing) which is an algorithm for keyword extraction based on the KEA algorithm [61]. The MAUI algorithm is composed of two stages, the (1) candidate selection; and the (2) machine learning based filtering step.

In the candidate selection stage, the text is initially split up into sentenses which are subsequently separated into tokens of up to *3*-grams not beginning or ending with stopwords. A number of features is then determined and calculated for each candidate keyword to be processed by a machine learning model, which will calculate the probability of a candidate keyword being a relevant keyword. MAUI uses the same features as KEA (TF.IDF, Position of the first occurrence, Keyphraseness, Phrase length, Node degree), plus a few other innovative features that are based on Wikipedia and that are computed by

the WikipediaMiner toolkit[12]. These are the Wikipedia-based keyphraseness, Node degree, Semantic relatedness and Inverse Wikipedia linkage. More recently, Meng et al. [82] proposes **CopyRNN**, which makes use of neural networks as a way to predict keywords from scientific texts. Along with a solution that extracts keywords from texts, the authors also propose a method to generate keywords that do not appear in a document, thus recognizing the problem of absent keywords, which deals with the fact that humans tend to choose as a descriptive term of a text, keywords that may not appear on it. Empirical analysis on six baselines on a broad range of datasets demonstrates that their proposed model significantly outperforms existing supervised (KEA [61], MAUI [62]) and unsupervised extraction methods (TF.IDF [10], TextRank [2], SingleRank [4], ExpandRank [4]) on extracting keywords that appear in the source text, but can also generate absent keywords (recalls up to 20%) based on the semantic meaning of the text.

In the context of social media Zhang et al. [83] proposes a novel algorithm, named **joint-layer Recurrent Neural Network**. This approach uses neural networks, computational models inspired by the central nervous system, which allows the learning and recognition of patterns. To evaluate the proposed method, a new dataset consisting of 41K tweets collected from Twitter was developed by the authors. Each hashtag presented in the tweet is considered a keyword, which enabled users to automatically label each one of the tweets. For instance, in the text "The Warriors take Game 1 of the #NBAFinals 104-89 behind a playoff career-high 20 from Shaun Livingston", the 2-gram "NBA Finals" would be considered a relevant and the only keyword of that particular tweet. Although this strategy enabled the authors to classify a huge portion of tweets it suffers from the fact that several other likely relevant keywords will simply not be gathered as they are not preceded by an hashtag. Besides, we may also argue that not all the hashtags are relevant to turn into a keyword and that as referred by Hu et al. [84] only 20% of the tweets, approximately, incorporate hashtags. Another shortcoming relates to the unavailability of the dataset, thus making it impossible to compare our approach against this proposal. More details on this and other keyword datasets will be given in the next chapter.

Table 1 presents a short summary of the methods approached in this chapter with regards to language dependency (stopwords list, PoS) and the type of algorithm (unsupervised or supervised). A more detailed analysis on the state-of-the art of keywords

[12] https://github.com/dnmilne/wikipediaminer [accessed on 13-03-2018]

extractor approaches can be found in the surveys of Lott [85] and of Hasan & Ng [86]. In the next chapter, we discuss the lack of available twitter datasets for keyword extraction purposes and elaborate on the need to develop a new dataset that fits our needs.

Table 1 - Keyword Extraction Approaches Summary. NA – Not Available

Methodology	**Method Name**	**Scope**	**Language Dependence**		
			Stopword List	**POS**	**Stemming**
Unsupervised	KP-Miner [3]	Generic	✓		
	RAKE [9]	Generic	✓		
	YAKE [1]	Generic	✓		
	TextRank [2]	Generic	✓	✓	
	SingleRank [4]	Generic	✓	✓	
	ExpandRank [4]	Generic		✓	
	TopicPageRank [6]	Generic	✓	✓	✓
	TopicRank [8]	Generic	✓	✓	✓
	PositionRank [5]	Generic	✓	✓	✓
	MultipartiteRank [7]	Generic	NA	✓	✓
	IKE [67]	Short Texts	✓	✓	NA
	cTPR [68]	Twitter	✓	NA	NA
	Double Ranking [72]	Twitter	NA	NA	NA
Supervised	KEA [62]	Generic	NA	✓	✓
	GenEx [75]	Generic	NA	✓	✓
	MAUI [63]	Generic	✓	✓	✓
	CopyRNN [82]	Generic	NA	✓	✓
	Joint-layer RNN [83]	Twitter	NA	NA	NA

Chapter 4

KWTweet Dataset - A Data Collection for Keyword Extraction with Tweets

In order to compare likely different approaches between them, the algorithms need to be evaluated on top of public datasets. While several reference collections already exist to evaluate keyword extractor methods, most of them consist of generic reports, papers or abstracts, and no dataset related to the problematic of extracting keywords from tweets has been develop so far. In this research, we aim to overcome this problem by developing a data collection suited to the particular characteristics of Twitter. This may be an added value to a wide number of researchers working on this kind of topics and the first main contribution of this work. The rest of this chapter is organized as follows. Section 4.1 refers to the importance and to the task of algorithm evaluation. Section 4.2 begins by describing generic keyword extractor datasets, before introducing a few collections related to twitter. The inexistence of a collection that suits our particular needs is also discussed. This motivates Section 4.3, which shows the construction process of the dataset used in this research. Finally, Section 4.4 provides some insightful analysis made on top of the data collected.

4.1. The Task of Evaluation

The evaluation of any algorithm as for effectiveness is concerned is one mandatory step of any new proposal. Although algorithm evaluation can be done by many different means, including user studies, it offers greater value when conducted on top of formal datasets. In the context of keyword extraction each dataset is usually composed of two components: (i) the corpus, i.e., the texts; and (ii) the set of corresponding relevant keywords (aka gold keywords or ground-truth). Once we have this defined, a comparison between the different algorithms is made possible by allowing the comparison of the results (keywords) retrieved by each different proposal with the gold keywords of the reference datasets. An illustrative example of this task is shown in Table 2 for the document #8 of the well-known Inspec collection. In the example, a comparison between YAKE [1, 65] and the IBM Natural

Language Understanding[13] system is given for the referred text. Gold keywords, i.e., those keywords considered relevant by the editors, are printed directly in the text in boldface. The result here presented shows that YAKE! would hypothetically retrieve 3 relevant keywords out of 5, while IBM would retrieve 2 out of 5. One interesting thing to note here is that contrary to other research tasks, the problem of extracting keywords is commonly associated to low precision values. Three reasons for this can be pointed at: (1) the need to have an exact match between the keywords of a system and the gold keywords; (2) the problem of absent keywords which only recently has been addressed [82] and (3) the definition of keyword, which may vary from person to person, thus hindering the retrieval of keywords satisfying all the users.

Table 2 - YAKE (left) and IBM (right) top-5 keywords for document 8 of Inspec collection. Keywords identified by both are printed in boldface

New investors get steal of a deal [**Global Crossing**] **Hutchison Telecommunications** and **Singapore Technologies** take control of **Global Crossing** for a lot less money than they originally offered. The deal leaves the **bankrupt** carrier intact, but doesn't put it in the clear just yet	
global crossing **hutchison telecommunications** telecommunications and singapore **singapore technologies** technologies take control	**global crossing** bankrupt carrier new investors **singapore technologies** deal

In the following we describe a few reference collections used in the context of the keyword extraction task. This anticipates Section 4.3 which will discuss the construction of our proposed dataset.

4.2. Keyword Extractor Reference Datasets

Over the years, a few reference collections have been set up in the context of keyword extraction, fostering the development and evaluation of new algorithms. Most of these standard collections, relate, however, with generic reports, papers or abstracts and none is devoted to the particular problem of extracting keywords from tweets. In the following we begin by introducing a brief overview of some of the most important collections used in the process of keyword extraction from (generic) texts. Collections related to the particular

[13] https://www.ibm.com/watson/services/natural-language-understanding/ [accessed on 13-03-2018]

case of Twitter, though not entirely related to our task, will be described in Section 4.2.2. This motivates the need to develop a new dataset that suits our task, which will be then discussed in Section 4.3.

4.2.1. Generic Keyword Extractor Datasets

The problem of extracting keywords from texts is longstanding and has been well studied over the last few years, much due to the existence of a considerable number of reference datasets devoted to this problem. In this section, we opt to describe some of the most well-known collections in this regard. In particular, we will detail Inspec [87], SemEval-2010 [88], Krapivin [89], Nguyen2007 collection [90] and KP20k [82].

The Inspec [87] collection consists of 2,000 abstracts of scientific journal papers from Computer Science collected between the years 1998 and 2002. SemEval-2010 [88] in turn, consists of 244 full scientific papers extracted from the ACM Digital Library (one of the most popular datasets which have been previously used for keyword extraction evaluation), each one ranging from 6 to 8 pages and belonging to 4 different computer science research areas (distributed systems; information search and retrieval; distributed artificial intelligence – multiagent systems; social and behavioral sciences – economics). The Krapivin dataset [89] consists of 2,304 full papers from the Computer Science domain, which were published by ACM in the period ranging from 2003 to 2005. In addition, the Nguyen [90] is a dataset composed of 211 scientific conference papers. Finally, the KP20k [82] dataset is composed by a large amount of high-quality scientific metadata, namely the titles, abstracts and the keywords of 20K scientific computer science articles randomly selected from 567K articles obtained from the ACM Digital Library, ScienceDirect, Wiley, and Web of Science.

While each one of these datasets are well established and well-known collections within the research community, none of them fulfils the needs related to tweets, a particular piece of text characterized by its small length. A few other collections related to this particular social network have been defined, yet they are either not available or do not suit our particular needs. A description of a few of them is given in the next section.

4.2.2. Twitter Datasets

The study of social networks is a recent topic which has boosted the emergence of a huge number of research papers, conferences and workshops devoted to the discussion of this

thematic (the BroDyn'18[14] and the NLP4SMA'18[15] are just two examples of a few recent workshops dedicated to this matter). In the following, we describe a few datasets related to Twitter, although not entirely related to our task. Several other publicly available datasets can be found on *docnow* website[16], a valuable resource which gathers an immense collection of Tweet datasets.

The PDI7IN_2016 [91] contains the tweet ids of approximately 280 million tweets related to the 2016 United States presidential election, collected between July 13, 2016 and November 10, 2016. More recently, Cram et al. [92] released the GE2017 which contains 34 million tweets collected between April 29, 2017 and June 4, 2017. A set of 56 keywords related to the British general election in 2017 (GE2017) was used to collect tweets on the topic. In the same line of research, Darwish et al. [93] developed the USPresElect2016 dataset, which consists of 3,450 labelled tweets (as: support/attack Trump/Clinton, or both, or neither (neutral)) representing the top 50 most retweeted tweets on the US presidential elections 2016 for every day during the period from 1 September 2016 to 8 November 2016 (the election day). The BTC (Broad Twitter Corpus) [94] is a large dataset of 9551 tweets collected from different regions (USA, UK, New Zealand, Ireland, Canada and Australia) and temporal periods (ranging between 2009 and 2014) developed to provide a representative sample of named entities (person; location; and organization). Finally, the Signal1m-tweetir dataset [95] combines two datasets of one million news articles [96] and 3.2 million tweets [97], to identify and rank tweets likely related to news article.

In addition to these collections, several other datasets related to the extraction of keywords from twitter can be found within the research works described in Section 3.3. Unfortunately, none has been made available. For instance, Zhao et al. [68] constructed a collection with 1.3 billion tweets from Singapore users collected during a timeframe of 20 weeks, from December 1, 2009 to April 18, 2010. Ten topics covering a diverse range of content in Twitter were then selected by applying Twitter-Latent Dirichlet Allocation (T-LDA [98]), a method to find "hidden" topics (e.g., "arts", "education") on tweets, without being affected by the noisy nature of this kind of texts. For each of the 10 selected topics, the authors then ran 4 methods (their proposal + 3 baselines), thus following a pooling approach, to determine the final list of gold keywords for each of the 10 topics. The quality

[14] https://sites.google.com/view/brodyn2018/home [accessed on 17/03/2018]
[15] http://setn2018.upatras.gr/index.php/nlp4sma-2018/ [accessed on 17/03/2018]
[16] http://www.docnow.io/catalog/ [accessed on 25/03/2018]

of each keyword was then evaluated by two judges having lived in Singapore and familiar with Twitter. Want et al. [72] collected 20,762 tweets distributed by 5 datasets, 3 of them related to health, and 2 related to tv programs. For the annotation keyword process, two specialists were recruited. One specialist in the health domain and another one in TV subjects. Another work, Marujo et al. [69] developed an annotated keyword dataset consisting of 1,827 tweets obtained from Gimpel et al. [99]. Unfortunately, no information is provided with regards to both the keyword annotation process and the availability of the collection. Finally, Zhang et al. [83] constructed a dataset with more than 41 million tweets[17]. Each tweet was then automatically assigned a keyword based on the tweets hashtags. For instance, the tweet "The Warriors take Game 1 of the #NBAFinals 104-89 behind a playoff career-high 20 from Shaun Livingston", would have "NBA Final" as a relevant keyword. Although automatic, this annotation process suffers from some problems, particularly the fact that only hashtags are considered relevant keywords. This implies that hashtags which may not be relevant, may be considered keywords, while, relevant keywords not tagged as hashtags may be simply disregarded.

Although several forums and research works have dedicated to this problem, with some of them actually developing datasets, none has made available a public collection in the context of extracting relevant keywords from tweets. In order to overcome this problem, we propose the KWTweet dataset. Our purpose is twofold. First, to foster the reproducibility of the experiments. Second, to boost research in this particular domain. Section 4.3 describes the construction and the characteristics of this new dataset, which may be understood as our first main contribution.

4.3. Data Collection and Labelling

One of the main challenges in evaluating algorithms in keyword extraction from tweet posts, is the absence of sizeable and annotated datasets. To overcome this problem, we decided to manually build our own collection of labelled tweets. In the following, we detail this process. Section 4.3.1 describes the methodology used to collect the tweets. Section 4.3.2 describes the annotation task. Finally, Section 4.3.3 refers to the repository where the collection was made available and explains how one can obtain the tweets that are part of the collection.

[17] No reference is given regarding the source of the tweets

4.3.1. Data Collecting

Before collecting the tweets, we began by selecting the top-100 Twitter users most followed as of the date of 15th of October 2017 on TwitterCounter[18], a website that gathers the most popular/followed Twitter users. For each user, we then collected all the tweets published during the period of one complete month (September 1, 2017 to September 30, 2017). In order to collect the tweets, we resort to the Twitter API[19], which requires a Twitter account and registering the application on Twitter through the Twitter Apps website[20]. To pursue this research, we opt to subscribe the free version (standard), which enables one to make 180 requests, that is 180 calls to the Twitter API, every 15 minutes. Note that in each request, we can retrieve 200 tweets, thus on 15 minutes we may retrieve a total number of 36k (180*200) tweets. Each tweet retrieved consists of a maximum of 140 characters which was the maximum number allowed as of the date of September 2017 when the tweets were collected. A new version of the Twitter API was meanwhile released at the end of September 2017 allowing now users to access tweets with a maximum number of 280 characters. Each collected tweet was then stored on a SQLite database together with the Twitter user id. Fig. 8 represents the process of collecting and storing the tweets for each one of the top-100 twitter users.

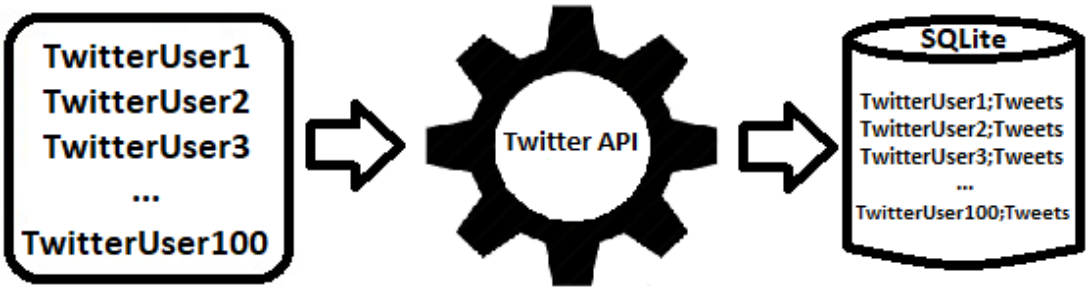

Fig. 8 - Workflow of collecting and storing the tweets for the top-100 twitter user's

Overall, we were able to collect a total number of 12,652 tweets corresponding to 100 Twitter users. In order to reduce the annotators effort in labelling these tweets, we decided to only select the top-25 twitter users who have tweeted at least, an average of 2 tweets per day, during the period above defined. Our filtered collection consists of 25 users and a total number of 8722 tweets. An overall analysis of the results enabled us to conclude that most of the users posted on a daily-basis with an average number of posts

[18] https://twittercounter.com/pages/100 [accessed on 02/03/2018]
[19] https://developer.twitter.com/ [accessed on 17/03/2018]
[20] https://apps.twitter.com/ [accessed on 17/03/2018]

varying in-between 2 to 33 tweets. The largest number of tweets collected refer to the Google account, with most of them corresponding to helpdesk tweet answers. In contrast, the smallest number is that of Rihanna with a total number of 63 tweets collected during the 30 days of September 2017. The set of 25 Twitter users considered, include tech companies (e.g. Twitter, YouTube, Instagram and Google), football news and sport clubs (e.g. FC Barcelona, ESPN, TeamOfTheYear - Champions League, Sports Center) as well as well-known personalities (e.g. Donald J. Trump[21], Rihanna, Kim Kardashian, Demi Lovato, Shakira) among others. Table 3 gives a more detailed information about the users/tweets data collected. In the following section we describe the annotation task process.

[21] Note that, within the 25 Twitter users, there are two similar usernames (Donald J. Trump and President Trump). The former is the private profile of Trump, and the later the profile of the President of the United States

Table 3 – KWTweet Dataset Stats

Name	Days Posted	Daily Average	Total Monthly
Google	25	33	1026
FC Barcelona	30	23	743
ESPN	30	23	739
★ #TeamOfTheYear ★	30	21	657
SportsCenter	30	20	645
CNN Breaking News	30	17	545
NASA	30	15	468
Real Madrid C.F.	30	14	464
National Geographic	30	14	438
Kim Kardashian West	28	11	355
Donald J. Trump	30	9	292
Ellen DeGeneres	29	8	253
Khloé	25	6	216
Alejandro Sanz	28	6	216
Twitter	26	6	215
BBC Breaking News	30	6	196
NICKI MINAJ	28	6	193
President Trump	26	6	192
YouTube	30	6	189
Kevin Hart	26	5	183
Demi Lovato	28	5	158
Niall Horan	26	3	109
Instagram	29	3	96
Shakira	25	2	71
Rihanna	25	2	63

4.3.2. Annotation Task

In this section, we describe the process of manually annotating the dataset. To conduct this task, we asked 15 volunteers to help us in the annotation process. In order to reduce the annotator's effort, we begun by dividing the 25 twitter users into 5 balanced groups of 5 twitter users each. Each group was then assigned 3 annotators. This annotation task is represented in Fig. 9.

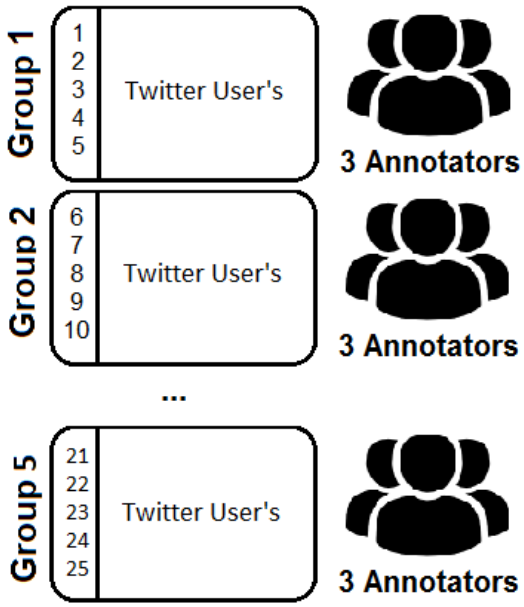

Fig. 9 - Annotation task process

On average, each annotator had to look at 1750 tweets. Table 4 lists the total number of tweets per group of Twitter users. A distribution of the tweets per twitter user is shown in Table 5.

Table 4 - Total Number of tweets per group of twitter users

Twiter User's Group	# Tweets
1	1730
2	1706
3	1774
4	1737
5	1775

Table 5 - Total number of tweets per twitter user and associated group

Twitter Users' Group	Username	Name	# Tweets
1	Twitter	Twitter	215
1	Google	Google	1026
1	KimKardashian	Kim Kardashian West	355
1	Shakira	Shakira	71
1	Rihanna	Rihanna	63
2	NASA	NASA	468
2	KevinHart4real	Kevin Hart	183
2	Khloekardashian	Khloé	216
2	FCBarcelona	FC Barcelona	743
2	Instagram	Instagram	96
3	Cnnbrk	CNN Breaking News	545
3	Espn	ESPN	739
3	YouTube	YouTube	189
3	NiallOfficial	Niall Horan	109
3	POTUS	President Trump	192
4	BBCBreaking	BBC Breaking News	196
4	TheEllenShow	Ellen DeGeneres	253
4	NatGeo	National Geographic	438
4	NICKIMINAJ	NICKI MINAJ	193
4	ChampionsLeague	★ #TeamOfTheYear ★	657
5	Realmadrid	Real Madrid C.F.	464
5	Ddlovato	Demi Lovato	158
5	realDonaldTrump	Donald J. Trump	292
5	SportsCenter	SportsCenter	645
5	AlejandroSanz	Alejandro Sanz	216

In order to collect the annotators answers, we decide to create a web application using Vue.JS, for the front-end, and Node.JS and SQLite for the backend. Each annotator

was required to register at the system so that we can keep track of the task evolution. Fig. 10 refers to the log-in screen presented to the annotator.

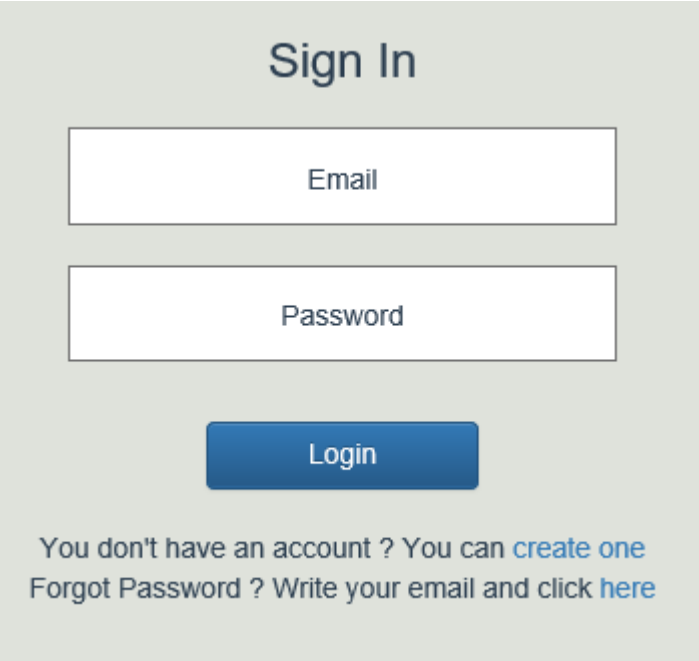

Fig. 10 - Annotation application login screen

Once logged into the system, annotators were given instructions to pursue their task. In a nutshell, each annotator had to look at each tweet and to choose as many keywords as possible, from 1 (minimum) to 5 (maximum) keywords. Annotators were especially instructed to the fact that a keyword may either be a single (e.g., Texas) or a composed word (e.g., Texas shooting). An example of this annotation task is given below for the following Tweet: "What we know about the victims of the Texas shooting", for which a random user chooses the four coming Keywords: "victims"; "Texas", "Texas shooting" and "victims of the Texas shooting". An interface of the tagging system for the previous example is shown in Fig. 11.

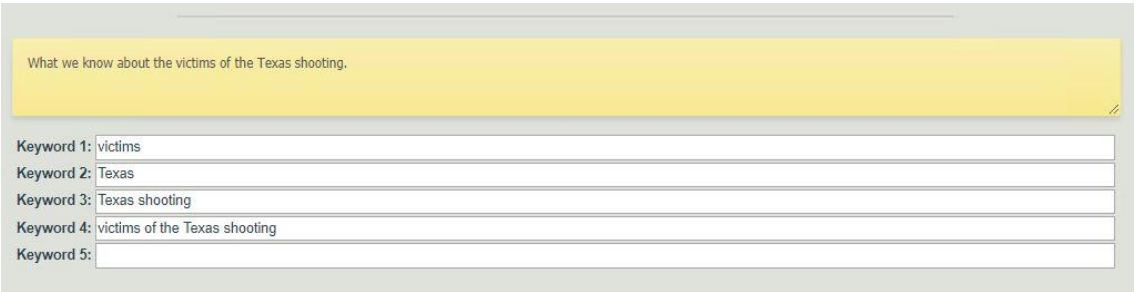

Fig. 11 - Annotation task interface

Note that there may be some cases where a tweet, due to its noisy nature, may not be assigned any relevant keyword by the annotator. Examples of this are the following tweets: "@A_MermaidsTale https://t.co/9HK21R2l3A"; or "@mcgc1998 ✿💮🌷✿□". These will be simply disregarded. Overall, of the initial 8722 tweets, 7736 were annotated. Of these, 4892 were annotated by two annotators and 2844 by three, thus guaranteeing that each tweet was labelled by at least two annotators. This resulted in an accumulated value of 18,722 tweet annotations and 39,959 keywords. Each one of these annotations were manually analysed to guarantee the quality of the annotators work and to remove noisy content. Table 6 shows the number of tweets tagged per group and per annotator.

Table 6 - Number of tweets labelled per annotator

Twitter User Groups	**# Annotated Tweets per Group**	**Annotator**	**# Annotated Tweets per Annotator**	**Total**
1	1706 out of 1730	Annotator 1	1696	**3880**
		Annotator 2	1649	
		Annotator 3	535	
2	1693 out of 1706	Annotator 4	1688	**3781**
		Annotator 5	1686	
		Annotator 6	407	
3	1552 out of 1582	Annotator 7	1512	**3969**
		Annotator 8	1488	
		Annotator 9	969	
4	1619 out of 1737	Annotator 10	1592	**2628**
		Annotator 11	986	
		Annotator 12	50	
5	1775 out of 1775	Annotator 13	1774	**4464**
		Annotator 14	1763	
		Annotator 15	927	

To calculate the inter-agreement between the annotators we resort to the Fleiss Kappa [100] statistic, however given that not all the tweets were labelled by the same number of annotators, we had to perform two different calculations, one considering the

tweets labelled by two annotators, and another one considered those annotated by three annotators. Note that, while Cohen's kappa [101] work for only two raters, Fleiss' kappa works for any number of raters (not necessarily the same). This suits our task, as we have five different groups of raters to annotate the entire set of observations. Table 7 shows an example of this task when considering three annotators. In this example, the first keyword was considered as relevant by three annotators. In contrast, the last one was deemed as relevant by just a single annotator.

Table 7 - Example of the data used to calculate the inter-agreement of 3 annotators

Case	Keyword	Not Keyword
Keyword 1	3	0
….	2	1
….	0	3
Keyword 2,816	1	2

The results obtained point to 9.44% of inter-agreement in the annotations made by two annotators, and to 37.31% when considering three annotators, which reflects the difficulty of this task in considering coincident keywords. These low results may also be justified by the fact that, unlike usual inter-agreement tasks, where annotators are offered a few possibilities from which to choose on, in this task, annotators had to specify keywords by themselves, from the scratch, which as referred by Sterckx et al. [102] will hardly lead to a consensual list of keywords.

4.3.3. GitHub

The result of this annotation is made publicly available at the LIAAD INESC TEC research center GitHub[22]. Note that due to the restrictions set by the tweets redistribution[23] developer policy (F. Be a Good Partner to Twitter), which in its number 2 states that "If you provide Content to third parties, including downloadable datasets of Content or an API that returns Content, you will only distribute or allow download of Tweet IDs, Direct Message IDs, and/or User IDs", we are not allowed to distribute the content of a tweet, though we may use its id to guarantee that researchers may access it in the future. Thus,

[22] https://github.com/LIAAD/KeywordExtractor-Datasets [accessed on 02/03/2018]
[23] https://developer.twitter.com/en/developer-terms/agreement-and-policy [accessed on 02/03/2018]

instead of providing the tweet and the corresponding relevant keywords, we make available a text file whose name is the id of the tweet and whose contents are the corresponding relevant keywords as determined by the annotators[24]. For instance, the file (910583891408424960.key) contains the gold keywords ('intersectionality'; 'black women'; 'Ericka Hart'; 'life'; 'impact of intersectionality'; 'illustrate') of the tweet identified with the id "910583891408424960". A python script is also made available in order to ease the process of obtaining the tweets content. Note however, that, the fact that the id of the tweet is made available, does not guarantee, per se, the access to its content, as tweets may become inaccessible overtime (either because they were deleted or made private).

4.4. KWTweet Dataset Analysis

In this section we conduct a brief analysis to get insights about the data collected. We begin by plotting the number of gold keywords per tweet. By looking at Fig. 12 we can observe that most of the tweets were annotated with 4 gold keywords. In contrast, only a few tweets were annotated by more than 8 different gold keywords.

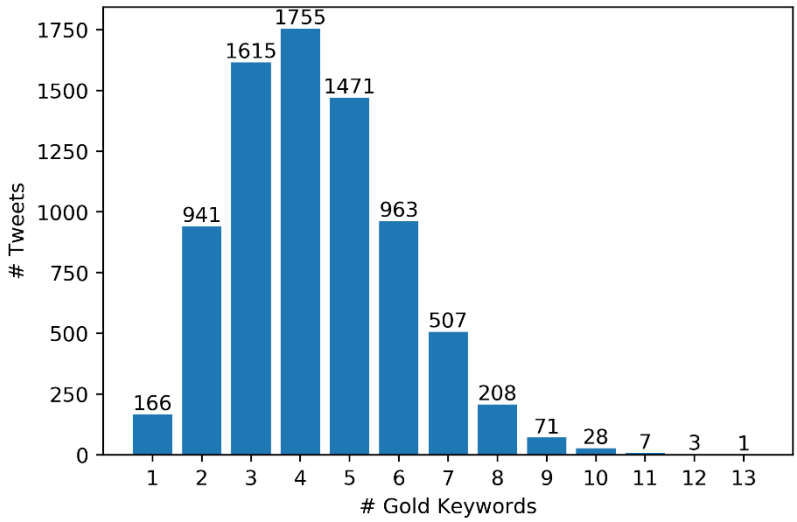

Fig. 12 - Number of gold keyword per tweet

Next, we aim to understand the distribution in the number of terms present in gold keywords. By looking at Fig. 13, we can observe that the number of terms that form a gold

[24] Note that in some particular cases, there may be some empty files, if none of the three annotators defined a keyword. This may happen in cases where the tweet is just a portion of noisy content. In contrast, a maximum of 15 keywords may be found, should the 3 annotators define 5 different keywords

keyword, seem to follow an exponential curve, that is, gold keywords with only one term (e.g., “discrimination”), two terms (e.g., “Puerto Rico”), or three terms (e.g., “Donald J. Trump”), tend to occur to a high extent, while keywords formed by a higher number of n-grams, tend to become scarce as the number of grams increases.

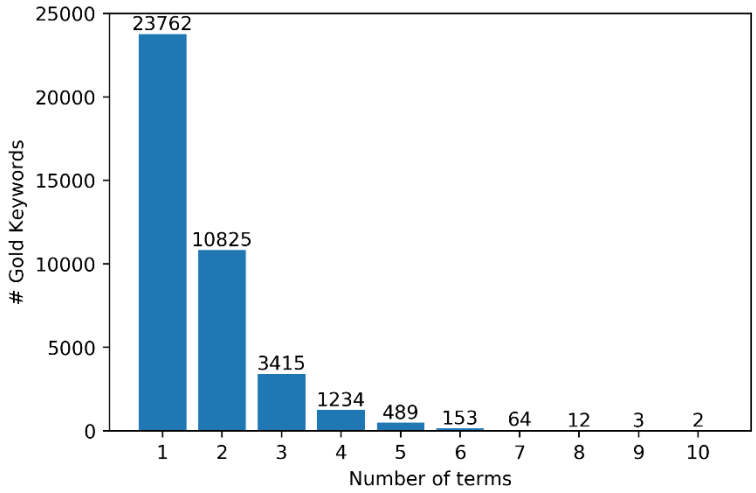

Fig. 13 - Number of terms per keyword

In order to understand the distribution of tweets per day we also plot (see Fig. 14) their frequency publication over the period of our study. Although one month is a too short period to take some valid conclusions, it appears this dataset follows a random behaviour as for daily patterns tweet publications is concerned, with the occurrence of several peak days (particularly in the middle – 13, 15 and 20 - and at the end of the month - 26) followed by subsequent falls.

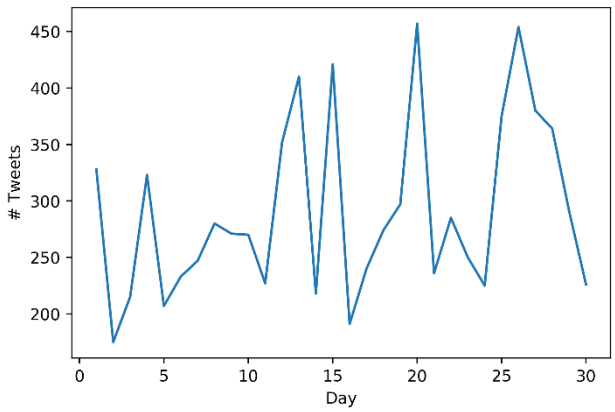

Fig. 14 - Publications of tweets per day

To understand this volatility, we decided to look in more detail at the top-3 twitter users, i.e., Google, FC Barcelona and ESPN. Fig. 15 shows that Google apparently follows the same trend as before, with some sparse peaks and falls.

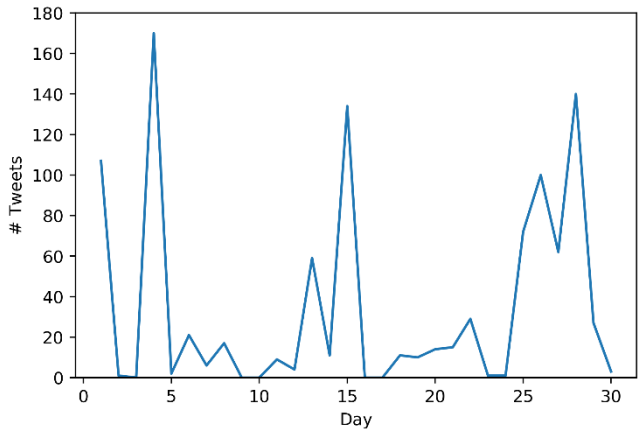

Fig. 15 - Total Google tweets per day

In contrast, the results of FC Barcelona (see Fig. 16) show, despite a fewer number of tweets, a more balanced approach with peaks being followed by falls in a rhythmic cadence, which may be due to the nature of the twitter user id itself.

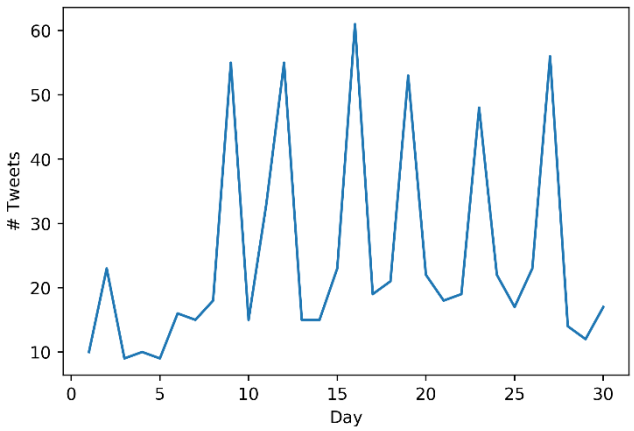

Fig. 16 - Total FC Barcelona tweets per day

Finally, Fig. 17 presents the number of daily tweets for ESPN, a sport TV channel. Similarly to the two previous plots, some peaks may be observed, though to a smaller extent. Overall, we may conclude that users follow a behaviour which favors peaks of publications followed by considerable falls that may lead to larger or smaller valleys. A more detailed analysis on this, however, may be conducted in order to take valid conclusions.

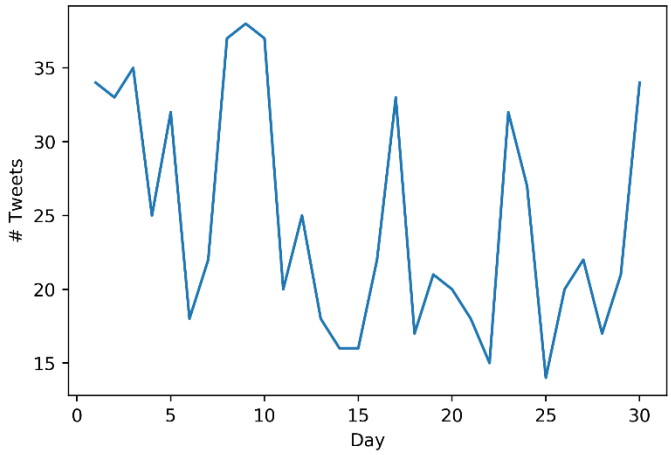

Fig. 17- Total ESPN tweets per day

Next, we aim to understand if the number of tweets is influenced by the number of followers as different users, publish a highly diverse number of tweets. Our (naïve) assumption is that the higher the number of followers, the higher the number of tweets, as people with more followers, thus audience, is likely to publish more often than people with less followers.

To conduct this analysis and understand this possible relationship we resort to a scatter plot. Fig. 18 shows this relation by plotting the dependent variable (#Tweets) on the y axis, and the independent variable (#Followers) in the x axis. By looking at the plot, we can observe that, unlike expected, there is a somewhat negative relationship between the two variables, meaning that as the number of followers increases the number of tweets decreases. However, we can observe that, through the red line therein plotted, there is a somewhat weak relationship, as several contra-examples can be observed in the plot. This can be confirmed by the Pearson Correlation Coefficient, which points to a negative moderate relationship around -0.415. A more detailed analysis however, should be conducted in order to take valid conclusions. One possibility is to analyze the behavior of all these followers during a one-year period. This however is out of the scope of this work.

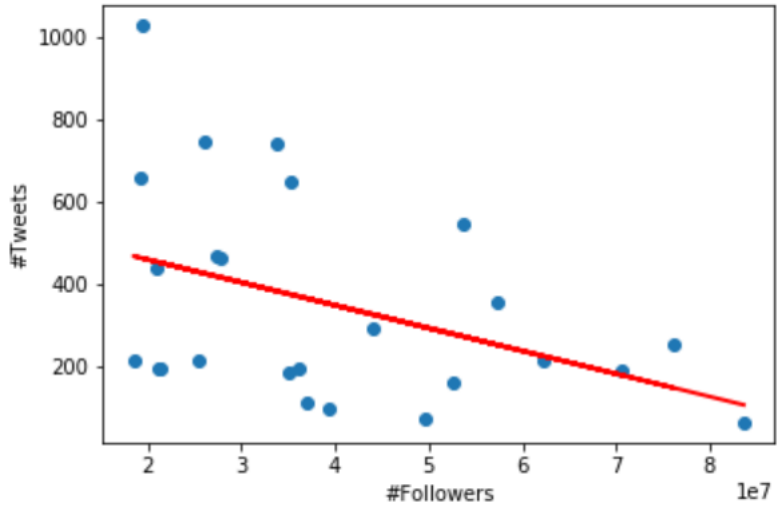

Fig. 18 - Relationship between the #Followers and the #Tweets. 1e7 means 1*10^7 number of tweets.

In the following, we plot the keywords cloud as determined among all the annotations of the tweets of each of the 25 Twitter users to understand if there are any keywords that may characterize the user's posting behaviour during the period of one month. By looking at Fig. 19 we can observe that, although there are one or two users for which finding repeated keywords seems to be a difficult process (as is the case of "Kloe Kardashian" and "National Geographic"), it turns out easy to observe that almost all the twitter users may be characterized by a number of keywords which stand out particularly during a one-month period. This means that several different tweets, eventually posted in different days, may refer to the very same issue.

Fig. 19 - Word cloud of the 25 Twitter user's keywords

For instance, Kevin Hart (upper left-hand side of the figure) held a concert in "Asheville" on September 16 for which "tickets" were on sale. In the very same month he also set up a food bank in "Houston". Nicki Minaj, a well-known rapper has a partnership with "MAC Cosmetics". Instagram in turn, held a "Weekend Hashtag Project" during this period, encouraging users to create photos or videos about the hobbies they most like. During this period, a few hurricanes, like "Irma" and "Maria", were also reported by NASA, together with their mission named "Saturn". Likewise, the official user of the President of the United States (@POTUS), also tweeted about the "Irma hurricane". When talking about the Kardashian's, two USA socialites, most of the references either pertain to themselves or to their collection of "beauty" articles such as "powders" and "creams". In turn, keywords related to Niall Horan, a famous singer, refer to his new "album" and to his "World Tour" projected to happen on 2018 ESPN as expected, mostly refers to sports keywords, such as "football", "game" and "season", besides "state" to refer to the different teams of every state of USA. National Geographic in turn, refers to several different words, with "hurricane" being referred by the annotators more often than all the others. Another sports tweet account is that of Barcelona, with most of the terms being related with "Suárez" one of its players and with "BarçaEibar" a match played during September 2017.

By looking at Shakira's word cloud, we can find references to "Fiel Perro" her new song, that was also presented in a "video", and to "Barcelona" as she is married with Pique, the well-known Barcelona football player. In Donald Trump personal profile, one can also find references to "Puerto Rico", due to the passage of the tornado "Maria", and not surprisingly to both "Koreas". Instead, the most common words for Real Madrid are "rmucl", a hashtag that aims to gather information on Real Madrid in the UEFA Champions League, and "Cristiano" (Ronaldo) Moving to Rihanna we may also found, words such as "beauty" and "fenty" like with did with the Kardashian's. Similar to other users, Alejandro Sanz, a famous singer, also used the words "CDMX", "Mexico" and "Puerto Rico" to refer to Hurricanes Irma and Maria. In turn, The Ellen Show, a popular TV show in USA, mostly refers to the word "birthday" due to the anniversary show due to be celebrated in September 2017. BBC Breaking TV channel is characterized be references to "Hurricane" and "Mexico" related to the Hurricane Irma. References to "Korea" involving "Donald Trump" and the tension between North and South Korea may also be found in this account. In contrast, the Google account uses twitter as a way to help their users when having problems accessing their accounts. Instead, YouTube is mostly characterized by words such as "dance", "watch", "creators" and "Ellen Show". Another sports account is that of the Champions League, which despite referring to several teams playing on this competition (e.g., "Celtic", "Bayern", "Manchester United", "Chelsea", "Roma", and so on and so forth) is also characterized by the word "UCL", an acronym for the UEFA Champions League. Similarly to BBC Breaking news channel, CNN also make use of words related to recent events. The Sports Center user, instead, is mostly characterized by the word "ESPN" (another user presented in this analysis) and "NFL" a professional sports league of USA football. In contrast, Demi Lovato, a famous singer popularized with her new song "Tell me you love me", has several words related to this new title as is the case of "new", "album" and "love". Annotators recruited to this task also found that "Fabletics", a brand advertised by Demi Lovato, was worth of interest. Finally, Twitter is characterized by the keywords "Happy", "Birthday", "Welcome" and "Friends" much due to the fact that it often congratulates people who celebrate a birthday and welcomes users who follow the twitter of Twitter. In addition to this, words such as "live" have also been characterized as descriptive much due to their live streams relative to Hurricane IRMA.

In order to confirm if there is any relationship between the annotated keywords and the Twitter Users hashtags we plot the hashtags word cloud for each one of the 25 Twitter users. To conduct this analysis, we looked at each tweet and registered the absolute frequency of each hashtag found. By looking at Fig. 20 we can observe that, except to some users (as is the case of Donald Trump which shares the word Puerto Rico), there is no visible relationship between the hashtags and the keywords. One such conclusion, confirms our assumption that simply relying on hashtags is a too limited approach to find relevant keywords, and definitely requires other type of features. More to the point, it may also put in question the usefulness of formal evaluation datasets automatically built on top of hashtags.

Fig. 20 - Word cloud of the hashtags from tweets text

Finally, Fig. 21 plots the frequency of the user's tweets hashtags per day to understand whether there is any prevalent topic among all the users.

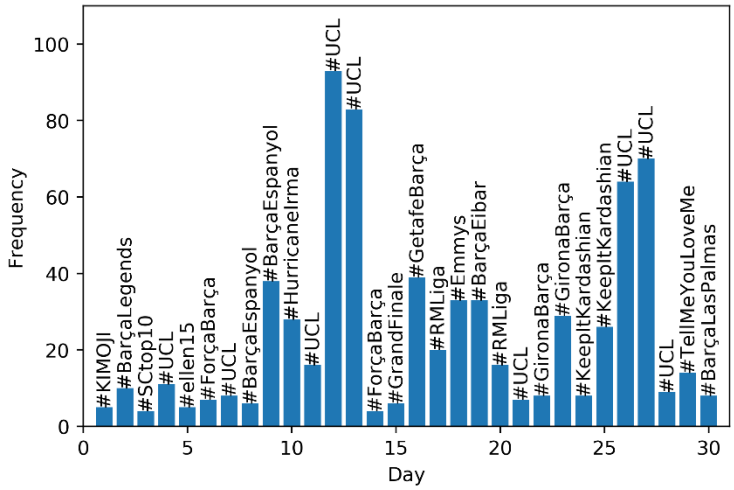

Fig. 21 - Number of hashtags per day

By looking at the figure, we can observe that, most hashtags are related to sport (e.g. "#BarçaLegends", "#UCL", "#SCtop10", "#ForçaBarça", "#BarçaEspanyol", "#GrandFinale", "#GetafeBarça", "#RMLiga", "#GironaBarca" and "#BarçaLasPalmas"). Other hashtags however worth to be cited. The hashtag "#Ellen15" represents the 15th season of Ellen's show, and appears as one of the most frequent hashtags on September 5, the first day of the season. Although having lasted a couple of days, "#HurricaneIrma" was only referred on day 10 when it hit Florida. Another important hashtag refers to the Emmy Awards, which is frequently cited on September 18th, one day after the event took place. "Keeping Up With The Kardashians" was then the most prevalent hashtag on two consecutive days (September 24 and 25) coinciding with the 10th anniversary. Finally, "#TellMeYouLoveMe" was the most frequent hashtag on September 29 as a result of a release of the new Demi Lovato album.

With this, we conclude the analysis of the KWTweet Dataset, which, to the best of our knowledge, is the first dataset made publicly available for twitter keyword research purposes. This will enable us to test and evaluate the suitability and the effectiveness of YAKE [65] on top of a fully formal twitter dataset. We plan to explain its architecture in the coming chapter.

Chapter 5

Detecting Keywords on Twitter

Over the last few years the emergence, the use and the establishment of social networks as the preferred form of communication between people and organizations has led to an unprecedent amount of data being generated. Among these social networks, Twitter is, perhaps with Facebook, Instagram, YouTube and LinkedIn, one of the most important and leading platforms for common people, enterprises, politicians and news outlets to communicate, changing the way people disseminate, share and consume information. Twitter, particularly, has gained attention as a disruptive platform for news consumption and distribution. For instance, personalities such as Kim Kardashian, politicians such as Donald Trump or companies such as Microsoft use Twitter to promote interaction with their users. News outlets, instead, use it to cover important events, some of them being broadcasted in real-time, and to promote engagement with their audience.

A recent work of Orellana-Rodriguez & Keane [103] shows how journalists and news outlets use Twitter as a platform to disseminate news, highlighting the factors that impact reader's attention and engagement with that news on Twitter.

This micro-blogging platform poses however some challenges when handling the information it contains. One such problem, relates to its short length text nature which makes it difficult to extract important information. In this research, we aim to understand whether Twitter may be used as a knowledge data source. In particular, we aim to understand how the newly developed YAKE! keyword extractor system applies to texts of short length nature. The rest of this chapter is organized as follows. Section 5.1 introduces our problem definition. Section 5.2 describes the architecture of YAKE!.

5.1.Problem Definition

Every day an amount of information on the internet grows in a striking way both in the form of simple web documents, news articles, files (e.g. pdf or text) but also through social networks like Facebook, LinkedIn and Twitter. This information is at the disposal of everyone and when digested may turn into valuable knowledge. Twitter is a large social network where the data posted by its users is constantly growing. One such information

may be of the utmost importance for anyone looking for a summary content. For instance, a journalist looking for the most cited topics of a Twitter user, such as Donald Trump, may find it useful to be given its most relevant keywords. In this work, we aim to evaluate whether keywords extractors may be applied to tweet posts. Our problem can be defined as follows:

Given a tweet *t*, extract top-*n k* keywords from it.

Based on this, we aim to evaluate how YAKE! [1] behaves when compared to state-of-the-art approaches. To better understand this problem, we describe in the next section the architecture of YAKE!.

5.2. YAKE! Architecture

Common keyword extraction algorithms take their foundations on the architecture previously presented in Section 3.2. YAKE! keyword extractor system [1], which will guide us through the rest of this work, has a very similar architecture to the one therein presented. In particular, it consists of five main steps: (1) Text pre-processing and Candidate Term Identification; (2) Feature Extraction; (3) Single Term Weight; (4) *n*-gram Generation and Keyword Weight Assignment; (5) Data Deduplication and Ranking. Fig. 22 portraits the overall architecture of YAKE!.

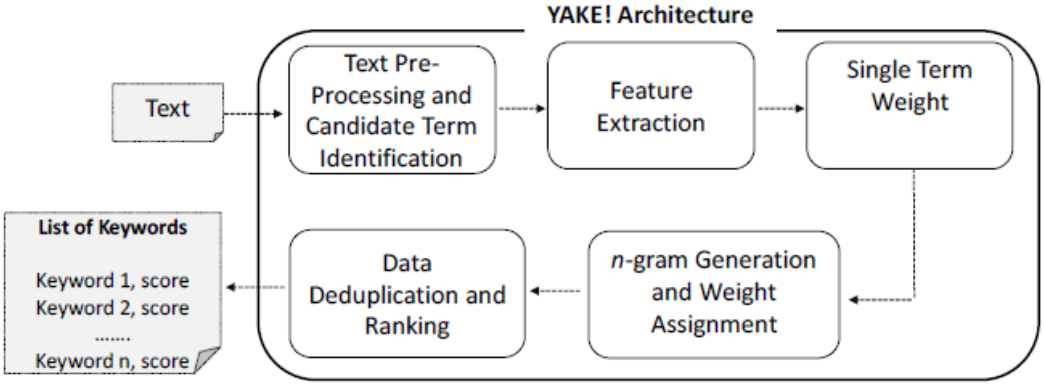

Fig. 22 - YAKE! Architecture. Obtained from Campos et al. [1]

5.2.1. Text pre-processing and Candidate Term Identification

The first step, known as the pre-processing stage, receives a raw document and applies natural language processing (NLP) techniques to generate a list of candidate keywords. At

this step, YAKE! segments the text into sentences (through segtok[25] rule-based sentence segmenter) and then the sentences into individual tokens (through the web_tokenizer module of the segtok). Each token is then annotated with some tag delimiters: (1) <d> for digits or numbers; (2) <u> for unparsable content; (3) <a> for acronyms; (4) <c> for uppercase; and (5) <p> for parsable content. Finally, a static list of stopwords is used to mark meaningless words. Note that, words with less than three characters are also considered a stopword in YAKE!'s approach.

After the pre-processing stage, each candidate term is represented through a series of five features capable of conveying its term importance. The features used are: (1) Casing; (2) Term Positional; (3) Term Frequency Normalization; (4) Term relatedness to Context; and (5) Term Different Sentence.

Casing (T_{Case}) is related to terms that begin with capital letter as well as acronyms on the assumption that terms with these characteristics are usually more relevant. Term Positional ($T_{Positional}$) relies on the belief that relevant terms have a tendency to concentrate on the beginning of a document, as opposed to least significant ones which may be found to a higher extent in the middle or at the end of a text. Term Frequency Normalization (TF_{Norm}) aims to evaluate the frequency of a candidate term on the assumption of Luhn [104] who states that the frequency of a word in a text provides a useful measure for its significance. Term relatedness to Context (T_{Rel}) aims to determine the dispersion of a candidate in relation to its context on the assumption of Machado et al. [105] who states that, the higher the number of different terms that co-occur with the candidate term on both sides, the less important it will be. Finally, Term Different Sentence ($T_{Sentence}$), is related to how often a term appears within different sentences, on the belief that terms that appear in many different sentences, tend to be more significant. A more detailed analysis of these features can be found in the work of Campos et al. [1]. In the following, we discuss how these features are combined into a single weight.

5.2.2. Single Term Weight

After the feature extraction step, the single term weight stage takes place. This is a core task of the architecture of YAKE! [1] as it gathers all these features into a single score capable of conveying the importance of a 1-gram term.

[25] https://pypi.python.org/pypi/segtok [accessed on 15-09-2018]

Equation 1 gathers also these features together such that the smaller the value S(t), the more significant the candidate term (t) would be, where S(t) means the weight of a term (t). This weight will feed the process of generating keywords to be explained in the next section.

$$S(t) = \frac{T_{Rel} * T_{Positional}}{T_{Case} + \frac{TF_{Norm}}{T_{Rel}} + \frac{T_{Sentence}}{T_{Rel}}} \quad (1)$$

5.2.3. *n*-gram Generation and Keyword Weight Assignment

While evaluating the importance of a single term (*1*-gram) may be an important first step, YAKE! [1] still needs to find a way of gathering the importance of gold keywords composed of more than one term. To form the final candidate keywords, YAKE! considers a sliding window of *n*-grams, ranging between *1*-gram to *n*-grams. During this process YAKE! will not consider selecting candidate keywords, whose individual terms are tagged as unparsable content (<*u*>), numbers, or beginning or ending with a stopword. Once the candidate keywords are formed, YAKE! determines its final score through the following equation:

$$S(kw) = \frac{\prod_{t \in kw} S(t)}{KF(kw) * (1 + \sum_{t \in kw} S(t))} \quad (2)$$

In this equation, kw represents a candidate keyword of one (e.g., "content") or more terms (e.g., "content atomism") and $S(kw)$ represents the final score, the lower the better. The score of a candidate keyword is determined by multiplying (in the numerator) the $S(t)$ score of the first term, by the subsequent $S(t)$ scores of the remaining terms (if they exist). YAKE! then computes the final score by dividing the numerator by the sum of the $S(t)$ scores, weighted by the candidate keyword frequency $KF(kw)$.

5.2.4. Data Deduplication and Ranking

As the final step, YAKE! aims to verify if the removal of similar candidate keywords improves the final results. To study the effect of this deduplication phase and its suitability in discarding potential similar candidate keywords, YAKE!'s authors make use of three different distance similarity measures: Levenshtein [63], Jaro-Winkler [64] and the sequence matcher. The results of their experiments however, show that, unlike expected, the data deduplication stage does clearly impact the improvement of the ranking results,

while increasing the time it takes to get the final results. Based on this consideration, we decided not to apply any deduplication stage.

Chapter 6

Evaluation

In this section, we aim to evaluate the effectiveness of YAKE! [1] on top of a set of tweets when compared to baseline similar approaches. Based on this, we conduct a set of experiments. Section 6.1 describes the metrics used to evaluate the effectiveness of YAKE! and of baseline methods. Section 6.2 presents information about YAKE! parameters, and feature importance, before presenting the results of the comparison between YAKE! and the baselines.

6.1.Evaluation Metrics

For the evaluation of the results, the automatic extracted keywords are compared with the manually annotated gold keywords using the exact match criteria. Traditionally, keyword extraction is by nature a ranking problem. Based on this, we opt to calculate Precision at k (P@k), Recall at k (R@k), F1-Measure at k (F1@k) and Mean Average Precision at k (MAP@k), where k = 10, a value that is commonly used by similar research approaches. Each metric will be detailed to a higher extent in the four coming sub-sections.

In order to avoid over-fitting and understand the generalization of the results, we followed a 5-fold cross validation approach, which operates by randomly partitioning the set of documents into five folds. A t-student test was used to assess the validity of the proposed solutions with statistical significance (p-value < 0.01 or p-value < 0.05) using matched paired one-sided t-test.

6.1.1. Precision at k

Precision at k (P@k) refers to the proportion of *k* returned keywords that are relevant. The higher the precision the better the result. Equation 3 formalizes P@k:

$$P@k = \frac{TP}{TP+FP} \tag{3}$$

where TP (True Positive) is the number of keywords correctly identified as relevant and FP (False Positive) is the number of keywords wrongly identified as relevant.

6.1.2. Recall at k

Recall at k (R@k) is the proportion of relevant keywords that are retrieved at the top-k results. The higher the precision the better the result. R@k is formalized in Equation 4:

$$R@k = \frac{TP}{TP+FN} \tag{4}$$

where TP (True Positive) is the number of keywords correctly identified as relevant and FN (False Negative) is the number of relevant keywords that are not retrieved by the system. For the computation of the metrics we applied a micro-average approach where TP and FN are first summed up before being computed.

6.1.3. F1-Measure at k

F1-Measure at k (F1-M@k) is used to define a balance between precision and recall and has the advantage of summarizing effectiveness in a single number. Its formalization is given in Equation 5:

$$F1 - M@k = 2\ x\ \frac{P@k * R@k}{P@k + R@k} \tag{5}$$

6.1.4. Mean Average Precision at k

Mean Average Precision at k (MAP@k) in turn, enables to distinguish between differences in the rankings at position 1 to k. Thus, the result is based on a set of queries rather than just one. Equation 6 formalized this metric:

$$\text{MAP@k} = \frac{\sum_{tw=1}^{|TW|} AP(tw)}{|TW|} \tag{6}$$

where AP is the average precision of each tweet (tw) and |TW| is the number of tweets.

6.2. Results and Discussion

In this section, we aim to analyse the effectiveness of YAKE! [1] on top of a collection of tweets. In Section 6.2.1, we begin by evaluating the results of YAKE! under different *n*-gram sizes. In particular, we aim to understand whether n = 2 is the best combination as this parameter has shown to achieve the best results on Campos et al. [1] when evaluated on top of datasets with characteristics different than those of Twitter. Then in Section 6.2.2, we evaluate the importance of each individual feature. Finally, in Section 6.2.3, we compare the results of YAKE! against baseline approaches.

6.2.1. n-Gram Parameter

A common way to generate a list of candidate keywords is through PoS patterns. An alternative to this approach, is to rely on an *n*-gram sliding window. YAKE! relies on this methodology to generate a list of candidates. The experiments here conducted enable us to understand the behaviour of YAKE! under different *n* parameter settings. In particular, we aim to understand whether there is any substantial difference for $1 \leq n \leq 5$. Fig. 23 shows the results obtained by YAKE! on top of the KWTweet dataset.

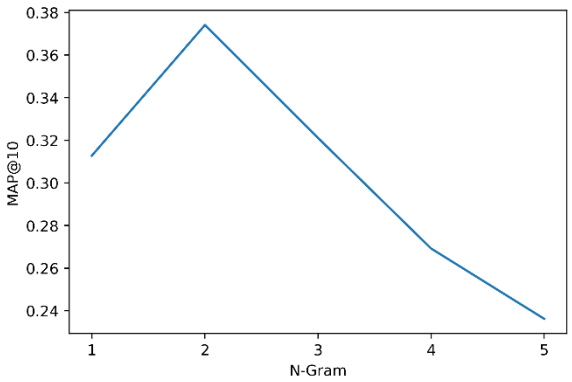

Fig. 23 - YAKE! MAP@10 Effectiveness on top of the KWTweet dataset when $1 \leq n \leq 5$

Fig. 23 confirms the results obtained by Campos et al. [1] who pointed out that the best results are obtained when n = 2. By looking at the plot we can confirm that a MAP@10 score of 0.3741 is obtained when n = 2, which is slight superior when compared to both n = 1 (particularly this one) and n = 3. However, the results worsen considerable for n = 4 and n = 5. This may be explained by the fact that the complexity of the system in detecting the most relevant keywords, increases with the number of grams, as more candidate terms are gathered into the pool, thus making it harder to make better decisions.

6.2.2. Feature Importance

YAKE! [1] consists of 5 different features (already introduced in section 5.2.1: T_{Case}, TF_{Norm}, $T_{Positional}$, T_{Rel} and $T_{Sentence}$ plus KF which is used in the Equation 2 upon the keyword weight assignment. In this section, we aim to evaluate feature importance. To this regard, we follow a backward-like elimination approach, which studies the impact of each individual feature, one feature at a time, by simply removing it from the single term weight $S(t)$ (recall Equation 1). That is, we consider a zero value for the corresponding feature in the equation $S(t)$ when talking about sums of features, and a 1 value if the feature is to be

multiplied by another one. The results of Fig. 24 clearly show that removing TF_{Norm} and $T_{Sentence}$, one at a time, might improve the results with statistically significance, thus suggesting that both features, though important to other kind of texts, may be eventually disregarded in short length texts. One possible reason for this might be due with the fact that tweets, due to its short size, do not have enough evidence in terms of frequency and of different sentences. Another thing that stands out here is that $T_{Positional}$ doesn't seem to either affect or improve the results (though without statistically significance), which once again, may be related to the fact that one short text may not embody enough evidence as for position is concerned. In contrast, removing both T_{Case} and T_{Rel} seems to negatively impact the results with statistically significance. This is particularly evident for T_{Case}, for which results may worse to a high extent. This clearly suggests that uppercase words may entail considerable evidence, in terms of what is or not relevant to a user when given a short text. Finally removing KF (which is solely used in Equation 2) seems to negatively impact the results, though not to a high extent.

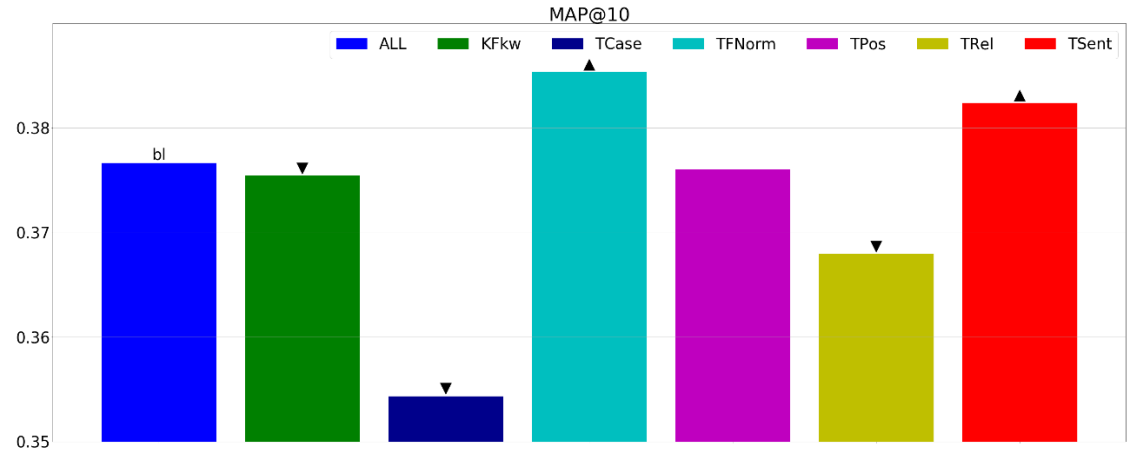

Fig. 24 - YAKE! Feature Importance - MAP@10 effectiveness of YAKE – (KFkw, TCase, TFNorm, TPos, TRel and TSent) features. bl means baseline.

In an attempt to better understand the behaviour of YAKE! [1] on top of tweets, we decided to deepen our analysis by evaluating the feature importance when removing more than one feature together. To conduct this task, we begin by studying the removal of several combinations involving TF_{Norm} and $T_{Sentence}$ (as these have proven to improve the results) with the remaining features, i.e., $T_{Positional}$ and T_{Rel}. A combination with T_{Case} will be left for further analysis. By looking at Table 8 we can observe that, perhaps not surprisingly, removing both TF_{Norm} and $T_{Sentence}$, at the same time, clearly impacts the

results with statistically significance. This was already expected as both features have shown that removing each one at a time would improve the results. We believe this is an important contribution as it gives us insights into the behaviour of YAKE! under a different kind of collection. Based on these results, we could for example simplify the S(t) equation of YAKE! as depicted in Equation 3, whenever dealing with tweets.

$$S(t) = \frac{T_{Rel} * T_{Positional}}{T_{Case}} \quad (3)$$

Table 8 - YAKE! Feature Importance - MAP@10 effectiveness of YAKE – (KFkw, TCase, TFNorm, TPos, TRel, TSent, TFNormTSent, TFNormTSentTRel, TFNormTPos, TSenTPos, TSenTRel, TFNormTRel, TFNormTSenTPos) features. bl means baseline.

Feature	MAP@10
YAKE	0.373900 bl
$YAKE - TF_{Norm}$	0.381000 ▲
$YAKE - T_{Sentence}$	0.378600 ▲
$YAKE - T_{Positional}$	0.373400 ∇
$YAKE - KF$	0.372900 ▼
$YAKE - T_{Rel}$	0.366800 ▼
$YAKE - T_{Case}$	0.355700 ▼
$YAKE - TF_{Norm} T_{Sentence}$	**0.385000 ▲**
$YAKE - TF_{Norm} T_{Sentence} T_{Rel}$	0.378400 ▲
$YAKE - TF_{Norm} T_{Positional}$	0.373600 ∇
$YAKE - T_{Sentence} T_{Positional}$	0.366700 ▼
$YAKE - T_{Sentence} T_{Rel}$	0.366700 ▼
$YAKE - TF_{Norm} T_{Rel}$	0.367300 ▼
$YAKE - TF_{Norm} T_{Sentence} T_{Positional}$	0.351200 ▼

In addition to this, we decided to conduct a further analysis, to shed light in the behaviour of the T_{Case} feature. Table 9 confirms the results already plotted in Fig. 24 which point towards a negative impact should the T_{Case} is removed.

Table 9 - YAKE! Feature Importance - MAP@10 effectiveness of YAKE – (TCase, TCaseTFNorm, TCaseTSent, TCaseTrel, TCaseTPos, TCaseTFNormTPos, TCaseTFNormTRel, TCaseTSentTpos, TCaseTsentTPos) features. bl means baseline.

Feature	MAP@10
YAKE	0.373900 bl
YAKE $-\mathrm{T}_{\mathrm{Case}}$	0.355700 ▼
YAKE $-\mathrm{T}_{Case}\mathrm{TF}_{\mathrm{Norm}}$	0.355100 ▼
YAKE $-\mathrm{T}_{Case}\mathrm{T}_{Sent}$	0.355100 ▼
YAKE $-\mathrm{T}_{Case}\mathrm{T}_{\mathrm{Rel}}$	0.347400 ▼
YAKE $-\mathrm{T}_{Case}\mathrm{T}_{Pos}$	0.339100 ▼
YAKE $-\mathrm{T}_{Case}\mathrm{TF}_{\mathrm{Norm}}\mathrm{T}_{Pos}$	0.339000 ▼
YAKE $-\mathrm{T}_{Case}\mathrm{TF}_{\mathrm{Norm}}\mathrm{T}_{Rel}$	0.347100 ▼
YAKE $-\mathrm{T}_{Case}\mathrm{T}_{Sent}\mathrm{T}_{Pos}$	0.342900 ▼
YAKE $-\mathrm{T}_{Case}\mathrm{T}_{Sent}\mathrm{T}_{Rel}$	0.346900 ▼

Finally, we decided to test the hypothesis of defining the S(t) equation on top of the results obtained individually by each feature. That is, instead of having the equation defined in Equation 1 or even in Equation 3, we would define S(t) on top of the values obtained by the $\mathrm{T}_{\mathrm{Case}}$ (and similarly on top of $\mathrm{TF}_{\mathrm{Norm}}$, $\mathrm{T}_{\mathrm{Positional}}$, $\mathrm{T}_{\mathrm{Rel}}$ and $\mathrm{T}_{\mathrm{Sentence}}$). As expected, the results worsen considerably, meaning that none of these features, by itself, are able to outperform the results of YAKE! when using the usual S(t) equation. Interestingly one can observe from Table 10 that, solely using the $\mathrm{T}_{\mathrm{Case}}$ feature would result in the worst effectiveness among all features, meaning that, although being an important feature, it cannot get the keywords essence per se. Also recall that, the S(t) equation is just a part of YAKE! procedure which will feed Equation 2. What these results show, is that Equation 2 great benefits should we use YAKE! or at most YAKE $-\mathrm{TF}_{\mathrm{Norm}}\mathrm{T}_{\mathrm{Sentence}}$ as shown in Table 8.

Table 10 - YAKE! Feature Importance - MAP@10 effectiveness of YAKE considering that S(t) = TCase, S(t) = TRel, S(t) = TSent, S(t) = TPos, S(t) = TFNorm. bl means baseline.

Feature	MAP@10
YAKE	0.373900 bl
$\mathrm{S(t)} = \mathrm{T}_{\mathrm{Case}}$	0.315100 ▼
$\mathrm{S(t)} = \mathrm{T}_{\mathrm{Rel}}$	0.340300 ▼
$\mathrm{S(t)} = \mathrm{T}_{\mathrm{Sentence}}$	0.347200 ▼
$\mathrm{S(t)} = \mathrm{T}_{\mathrm{Positional}}$	0.346800 ▼
$\mathrm{S(t)} = \mathrm{TF}_{\mathrm{Norm}}$	0.346700 ▼

6.2.3. YAKE! vs Baselines

In order to evaluate the effectiveness of YAKE! we carried out a final experiment, which compares YAKE!'s effectiveness against unsupervised state-of-the-art baseline methods with an available solution. On these grounds, we make use of PKE[26], a very useful open source python-based keyword extraction toolkit, made available by Boudin [106], for the experiments with KP-Miner [3], MultipartiteRank [7], PositionRank [5], TopicalPageRank [6], TopicRank [8], SingleRank [4] and TF.IDF [60]. For TextRank [2] we used the Kazi Hasan code[27] and for RAKE [9], RAKE-tutorial code[28].

In addition to this, we also compare YAKE! against three additional baselines that we name as *HashTags*, *Users* and *UserHashTags*. The first one, *HashTags*, considers as a relevant keyword, all the terms marked with an hashtag (#). Instead, *Users* considers as relevant keywords, all the twitter users (@) that appear in a tweet. Finally, *UserHashTags* ground-truth consists of all the hashtags and users that are found within each tweet. The extraction of both, was done with resort to the definition of simple regular expressions which extract all the words that are found with the respective symbol (# or @). Setting these baselines will enable us to understand whether a single hashtag or user strategy is not appropriate to determine relevant keywords within tweets.

Table 11 presents the results for P@10, R@10, F1-M@10 and MAP@10 for $n = 2$ as this parameter has achieved the best results. Again, we followed a 5-fold cross validation approach, which operates by randomly partitioning the set of documents into five folds. A t-student test was used to assess the validity of the proposed solutions with statistical significance (p-value < 0.01 (▲, ▼) or p-value < 0.05 (∇, Δ)) using matched paired one-sided t-test.

[26] https://github.com/boudinfl/pke [accessed on 15/09/2018]
[27] http://www.hlt.utdallas.edu/~saidul/code.html [accessed on 07/11/2018]
[28] https://github.com/zelandiya/RAKE-tutorial [accessed on 07/11/2018]

Table 11 - YAKE! effectiveness vs Baselines methods. P@10, R@10, F1@10 and MAP@10. Results are shown in descending order of the MAP@10 score.

Baseline	**P@10**	**R@10**	**F1@10**	**MAP@10**
YAKE!	0.245	0.626	0.340	0.373
YAKE $-\mathrm{TF}_{\mathrm{Norm}}\mathrm{T}_{\mathrm{Sentence}}$	0.245 ∇	0.627	0.340	0.385 ▲
TextRank	0.228 ▼	0.588 ▼	0.317 ▼	0.313 ▼
KP-Miner	0.211 ▼	0.549 ▼	0.294 ▼	0.273 ▼
SingleRank	0.189 ▼	0.502 ▼	0.265 ▼	0.235 ▼
MultipartiteRank	0.131 ▼	0.324 ▼	0.180 ▼	0.222 ▼
TopicPageRank	0.143 ▼	0.354 ▼	0.197 ▼	0.219 ▼
TopicRank	0.130 ▼	0.320 ▼	0.179 ▼	0.215 ▼
PositionRank	0.143 ▼	0.354 ▼	0.197 ▼	0.214 ▼
TF.IDF	0.144 ▼	0.390 ▼	0.202 ▼	0.163 ▼
UserHashTags	0.039 ▼	0.098 ▼	0.053 ▼	0.089 ▼
HashTags	0.026 ▼	0.070 ▼	0.037 ▼	0.069 ▼
Rake	0.015 ▼	0.039 ▼	0.020 ▼	0.030 ▼
Users	0.013 ▼	0.028 ▼	0.017 ▼	0.027 ▼

One thing that stands out here is that the results of the Twitter dataset are considerable better than those obtained by Campos et al. [1] who obtained much lower results when applying YAKE! on top of other datasets than this of Twitter. Unsurprisingly, this suggests that extracting keywords from short length texts turns out to be easier than in larger ones, which may be easily explained by the fact that less candidates are available.

Considering the results presented in, YAKE! (and similarly, YAKE $-\mathrm{TF}_{\mathrm{Norm}}\mathrm{T}_{\mathrm{Sentence}}$) is the one that presents the best results with a significant difference to the best baseline approach (TextRank). This is particularly evident when comparing the MAP scores, for which YAKE was able to obtain a 0.373 score (and 0.385 in the case of YAKE $-\mathrm{TF}_{\mathrm{Norm}}\mathrm{T}_{\mathrm{Sentence}}$) and TextRank a much lower score of 0.31. This difference is considerable superior as of that obtained in the F1 score, thus meaning that the ranking of the results in the top-10 list, plays an important and significant role for YAKE! by pushing to the top the best relevant keywords.

Another thing that stands out here, is that, unlike expected, TextRank is the best baseline approach, which somehow contradicts the results obtained by Campos et al. [1] who showed that TextRank effectiveness was among the ones with worst behavior when

evaluated on top of other kind of collections. To better understand this difference, we opt to analyze a few individual results, and came to the conclusion that the TextRank effectiveness is highly influenced by a huge portion of high recall scores (much of the times around 100%). It turns out evident that TextRank particularly suits this kind of short texts, for which extracting all the possible candidate keywords (especially when using PoS, as TextRank does) becomes easier, though not enough to beat YAKE!. This however, is an important obtained result in our research, and a valuable contribution to the research community, shedding light on the fact that some kind of approaches are more tuned to a specific type of collection.

The good results obtained by KP-Miner (best second baseline), which, like YAKE!, is built on top of statistical features, confirms the results of Campos et al. [1] who has shown that this approach has a good effectiveness across different types of documents. However, they also show that simple relying of term frequency, as TF.IDF does, may not be sufficient to obtain good results. Of particular importance, the fact that none of the additional approaches considered in this work, that is (Hashtags and / or Users), were able to obtain good results, which proves that methods based on this information will hardly obtain important results. In the next chapter, we present a demo of using YAKE! in real-time tweets.

Chapter 7

Demo

In this section, we present a demo of our work, to showcase how YAKE! behaves when fed by a tweet posts. TwitterYake!, which is available online at [http://bit.ly/TwitterYAKE], was developed in VueJS and NodeJS as a backend. The main libraries used were "node-cmd"[29] (for the connection with YAKE! [1]) and "twit"[30] (for the connection with Twitter API). Through this web application, one can choose between querying Twitter either with a Twitter user name or an hashtag The results obtained through the application of YAKE! [1] can then be interactively explored through a graphical representation made of word clouds. This may be understood as an important contribution to the research community as it will enable any user, to transparently test our proposed solution, and complements the work of Campos et al. [65] who has already made available a python package of YAKE! and a demo. The rest of this chapter is organized as follows. Section 7.1 presents the individual exploration of the tweet posts upon querying twitter with a twitter username or hashtag. Section 7.2 explores the aggregated temporal view, again through the specification of a username or an hashtag.

7.1. Individual Exploration of the Results

In this option, the user is given the chance to explore the 25 last tweets (and corresponding keywords) of a given specified username or hashtag. Once we get the results[31], YAKE! [1] is applied. Each tweet is then made available to the user, together with the corresponding date and the generated keyword cloud. The user is then given the chance to navigate through each of the 25 tweets using for that the arrow keys. Fig. 25 shows the final result for the "RealDonaldTrump" username. By looking at the results we can observe that YAKE! [1] is able to retrieve, from an apparent meaningless tweet, several interesting filtered information such as a reference to "great patriots", "America safe" and to a location, in the case, to "Florida" .

[29] https://www.npmjs.com/package/node-cmd [accessed on 15-09-2018]
[30] https://www.npmjs.com/package/twit [accessed on 15-09-2018]
[31] which consumes most of the processing time of our demo, as for a text of short nature get keywords is done in milliseconds

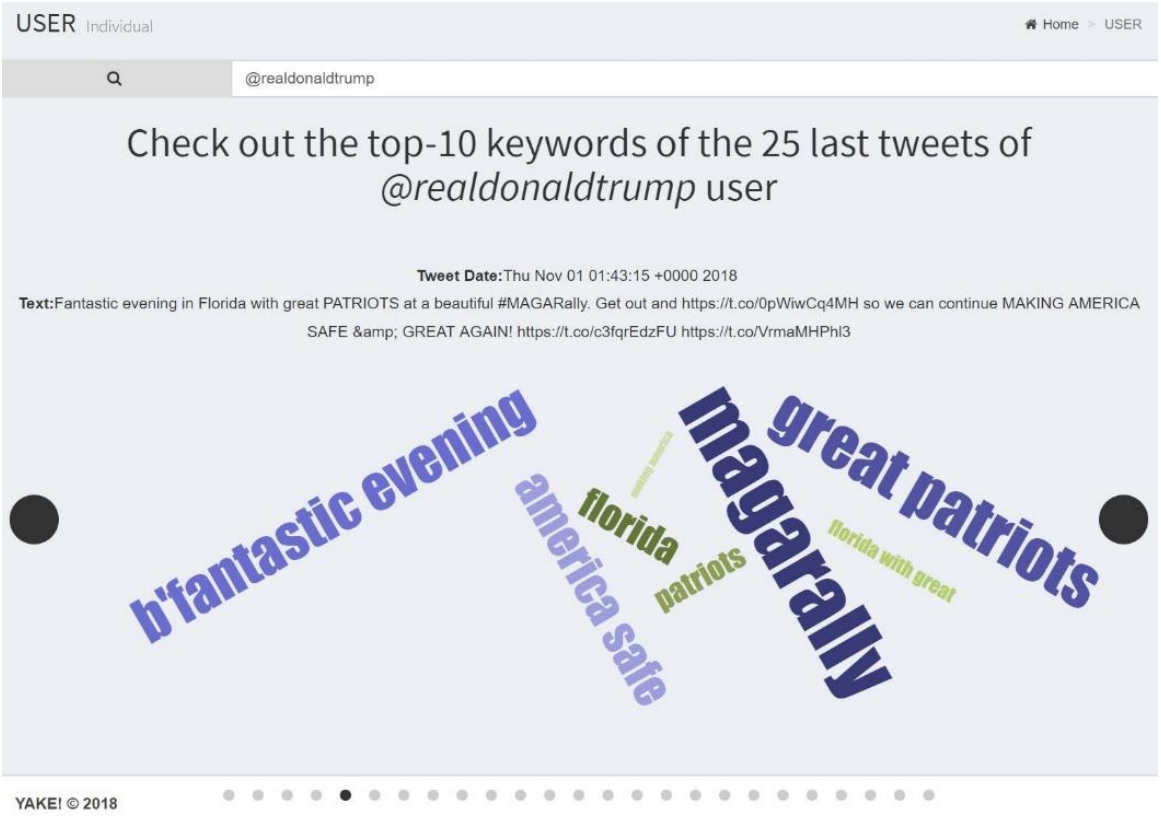

Fig. 25 - Individual keyword cloud for the “RealDonaldTrump” username

Fig. 26, instead, shows the very same results but for the “websummit18” hashtag. By looking at the figure, we can observe information such “security”, “trust”, “gdpr” which is the acronym of General Data Protection Regulation and a reference to a location, “lisbon”.

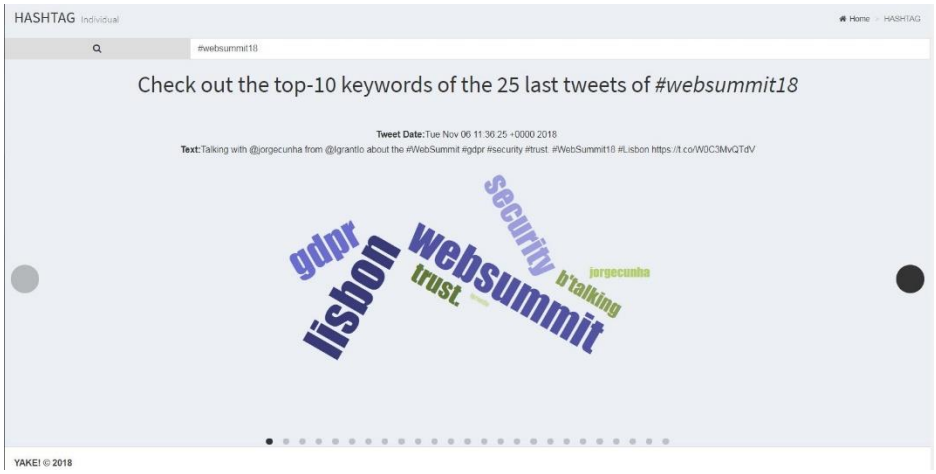

Fig. 26 - Individual keyword cloud for the “websummit18” hashtag

Next, we describe the aggregate exploration of the results.

7.2. Aggregated Exploration of the Results

While, extracting keywords from individual tweets may play an important role, we believe that a more aggregated view of the results gives the user the chance to get an overall analysis of the tweets in a quick fashion manner. To this regard, we gather all the 25 tweets collected, into a single text and apply YAKE! [1] to gather the most relevant keywords. Somehow, this simulates the behaviour of YAKE! [1] on top of texts made of medium size. Thus, interpretation of this as a contribution, may be cautious, as, though dealing with texts of tweet posts, we are no longer dealing with the extraction of keywords from short texts, but from medium size tweets posts (which still may differ from other texts by the fact that these may embed noisy information). A more principled solution should thus be developed in the future. Fig. 27 shows the result of the last 25 tweets of "RealDonaldTrump" username, in an aggregated fashion. In this functionality, browsing between tweets it is not possible, as the content here displayed refers to the aggregation of tweets. By looking at the picture we can observe as relevant keywords "great country" and "great state" which are commonly referred by Donald Trump, together with "crime", "military" and "illegal immigration", all of which have been in the root of recent problems in the United States of America.

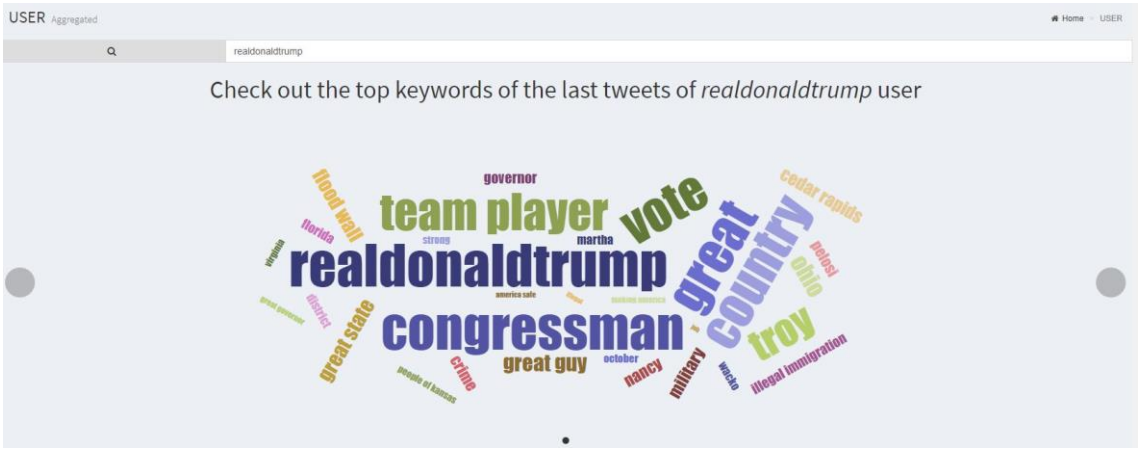

Fig. 27 - Aggregated keyword cloud for the "RealDonaldTrump" user

Instead, Fig. 28 shows the result of the last 25 aggregated tweets of the hashtag "websummit18". Fig. 28 presents some keywords that are also in Fig. 26 because this is the aggregated result of several individual tweets. We can observe as relevant keywords

“investor”, “europe”, “mobility”, “sharing” which defines the objective of this event but also exists a reference to a location, in this case the location of this event, Lisbon.

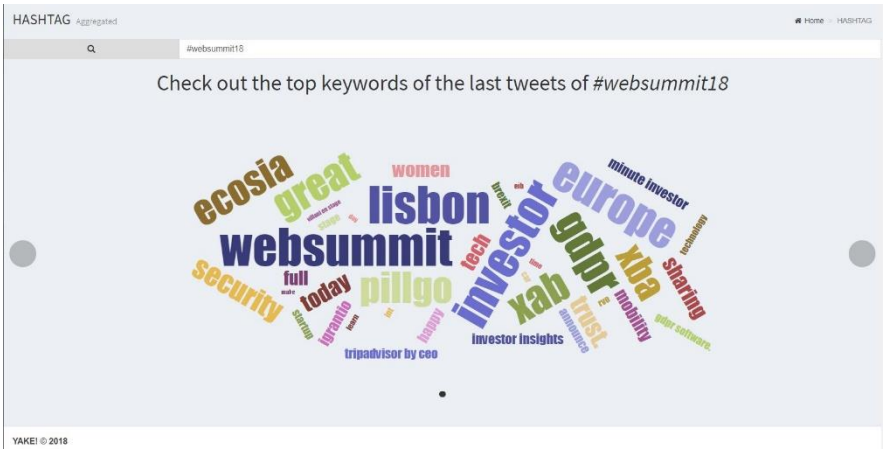

Fig. 28 - Aggregated keyword cloud for the “websummit18” hashtag

Chapter 8

Conclusion and Future Work

Despite the fact that tweets may contain valuable information, few studies have fully considered extracting relevant keywords from this kind of source. Much of this, is due to the fact that extracting keywords from texts of short nature is a difficult process, which makes this a challenging task. In this work, we aim to test YAKE! - an unsupervised keyword extractor algorithm - on top of a collection of tweets. Our purpose is to, given a tweet, identify its most relevant keywords, in a manner that one can get easily familiarized with the tweet subject, without the need to spend much time looking for information. In order to achieve this, we had to develop our own dataset with manually-tagged tweets. KWTweet dataset was made publicly available (according to the Twitter sharing rules), and may be faced as an important contribution for the research community by giving researchers the possibility to test future coming approaches in a formal manner. The results obtained show that YAKE! is able, not only to perform well on texts of medium and big size nature, but also on texts of short size, such as those evaluated here. However, the results also show that some specific features are more important than others in this kind of scenario, thus meaning that an adapted version of YAKE! may be better suited to this kind of text. Interestingly, we could also note that, while other approaches may not perform well among different types and size of texts, they may behave better in this short case scenario, which confirms that conclusions, must also be taken according to the domain, size and type of the collection being studied. These results give insights into the fact that obtaining relevant keywords through an unsupervised statistical approach, without any kind of training, is possible and can bring countless advantages in particular its capability to run on the fly, independently of its language, size or domain. In addition to this, we were able to make available a demo of our solution, thus enabling users to test our approach. This may be understood as an important contribution to the research community.

In this work, we were able to show that unsupervised algorithms, such as YAKE!, can be effectively applied to perform tasks not only in long texts but also in short texts as is the case of tweets. The study of this work was based on the most followed users, but it would be interesting to carry out a work based only on a specific area such as politics,

music, sports, news or technology to name but a few. With information focused on a single topic, it would be easier to create an history and perhaps get more concrete results. In addition, formally evaluating the effectiveness of YAKE! on extracting keywords from an aggregated temporal search, instead of solely based on a single tweet, may be an interesting future research direction.

References

[1] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes and A. Jatowt, "A Text Feature Based Automatic Keyword Extraction Method for Single Documents," in *40th European Conference on Information Retrieval*, Grenoble, France, 2018.

[2] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," in *Proceedings of EMNLP 2004*, Barcelona, Spain, 2004.

[3] A. Rafea and S. R. El-Beltagy, "KP-Miner: A keyphrase extraction system for English and Arabic documents," *Information Systems*, vol. 34, pp. 132-144, 2009.

[4] X. Wan and J. Xiao, "Single Document Keyphrase Extraction Using Neighborhood Knowledge," in *AAAI'08 Proceedings of the 23rd national conference on Artificial intelligence*, Chicago, Illinois, 2008.

[5] C. Florescu and C. Caragea, "PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017.

[6] Z. Liu, W. Huang, Y. Zheng and M. Sun, "Automatic Keyphrase Extraction via Topic Decomposition," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Massachusetts, USA, 2010.

[7] F. Boudin, "Unsupervised Keyphrase Extraction with Multipartite Graphs," in *Proceedings of NAACL-HLT*, New Orleans, Louisiana, 2018.

[8] A. Bougouin, F. Boudin and B. Daille, "TopicRank: Graph-Based Topic Ranking for Keyphrase," in *International Joint Conference on Natural Language Processing (IJCNLP)*, Nagoya, Japan, 2013.

[9] S. Rose, D. Engel, N. Cramer and W. Cowley, "Automatic keyword extraction from individual documents," in *Text Mining: Applications and Theory*, United States, John Wiley & Sons, Ltd, 2010, pp. 3-20.

[10] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, pp. 11-21, 1972.

[11] M. Jamali and H. Abolhassani, "Different Aspects of Social Network Analysis," in *WI '06 Proceedings of the 2006 IEEE/WIC/ACM International 66-72 Conference on Web Intelligence*, Hong Kong, China, 2006.

[12] N. Ellison, "Social network sites: Definition, history, and scholarship," *Journal of Computer-Mediated Communication*, vol. 13, pp. 210-230, 2007.

[13] D. Miller, E. Costa, N. Haynes, T. McDonald, R. Nicolescu, J. Sinanan, J. Spyer, S. Venkatraman and X. Wang, How The World Changed Social Media, London: UCL Press, 2016.

[14] D. J. Hughesa, M. Rowe, M. Batey and A. Lee, "A tale of two sites: Twitter vs. Facebook and the personality predictors of social media usage," *Computers in Human Behavior*, vol. 28, pp. 561-569, 2012.

[15] H. Jones and J. H. Soltren, "Facebook: Threats to Privacy," 2005.

[16] d. boyd and Eszter Hargittai, "Facebook privacy settings: Who cares?," *First Monday,* vol. 15, 2010.

[17] Y. Liu, K. P. Gummadi, B. Krishnamurthy and A. Mislove, "Analyzing Facebook Privacy Settings: User Expectations vs. Reality," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, Berlin, Germany, 2011.

[18] D. McCallig, "Facebook after death: an evolving policy in a social network," *International Journal of Law and Information Technology,* vol. 22, p. 107–140, 2014.

[19] L. Fu, M. A. Jacobs, J. Brookover, T. W. Valente, N. K. Cobb and A. L. Graham, "An exploration of the Facebook social networks of smokers and non-smokers," *PLOS ONE,* vol. 12, 2017.

[20] J. Fardouly, P. C. Diedrichs, L. R. Vartaniana and E. Halliwell, "Social comparisons on social media: The impact of Facebook on young women's body image concerns and mood," *Body Image,* vol. 13, pp. 38-45, 2015.

[21] H. Lin, W. Tov and L. Qiu, "Emotional disclosure on social networking sites: The role of network structure and psychological needs," *Computers in Human Behavior,* vol. 41, pp. 342-350, 2014.

[22] A. D. I. Kramer, J. E. Guillory and J. T. Hancock, "Experimental evidence of massive-scale emotional contagion through social networks," in *Proceedings of the National Academy of Sciences*, 2014.

[23] K. E. Anderson and J. M. Still, "An introduction to Google Plus," *Library Hi Tech News,* vol. 28, pp. 7-10, 2011.

[24] M. Landeweerd, T. Spil and R. Klein, "The Success of Google Search, the Failure of Google Health and the Future of Google Plus," in *International Working Conference on Transfer and Diffusion of IT*, Bangalore, India, 2013.

[25] M. Osborne and M. Dredze, "Facebook, Twitter and Google Plus for Breaking News: Is There a Winner?," in *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, Palo Alto, California, 2014.

[26] L. Shi, A. Agrawal, N. Agarwal and R. Garg, "Predicting US Primary Elections with Twitter," 2012. [Online]. Available: http://snap.stanford.edu/social2012/papers/shi.pdf.

[27] S. Vieweg, A. L. Hughes, K. Starbird and L. Palen, "Microblogging During Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness," in *CHI 2010: Crisis Informatics*, Atlanta, GA, USA, 2010.

[28] Z. Ashktorab, C. D. Brown, M. Nandi and A. Culotta, "Tweedr: Mining Twitter to Inform Disaster Response," in *Proceedings of the 11th International ISCRAM Conference*, University Park, Pennsylvania, USA, 2014.

[29] E. d. Quincey and P. Kostkova, "Early Warning and Outbreak Detection Using Social Networking Websites: The Potential of Twitter," in *Electronic Healthcare - Second International ICST Conference*, Istanbul, Turkey, 2009.

[30] M. J. Paul and M. Dredze, "You Are What You Tweet: Analyzing Twitter for Public Health," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, Barcelona, Spain, 2011.

[31] E. Aramaki, S. Maskawa and M. Morita, "Twitter Catches The Flu: Detecting Influenza Epidemics using Twitter," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK, 2011.

[32] E. Kouloumpis, T. Wilson and J. Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, Barcelona, Catalonia, Spain, 2011.

[33] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment Analysis of Twitter Data," in *Proceedings of the Workshop on Languages in Social Media*, Portland, Oregon, 2011.

[34] M. S. Gerber, "Predicting Crime Using Twitter and Kernel Density," *Decision Support Systems (Elsevier)*, vol. 61, pp. 115-125, 2014.

[35] X. Wang, M. S. Gerber and D. E. Brown, "Automatic Crime Prediction using Events Extracted from Twitter Posts," in *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, College Park, MD, 2012.

[36] C. Castillo, M. El-Haddad, J. Pfeffer and M. Stempeck, "Characterizing the life cycle of online news stories using social media reactions," in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, Baltimore, Maryland, USA, 2014 .

[37] B. Han, P. Cook and T. Baldwin, "Text-Based Twitter User Geolocation Prediction," *Journal of Artificial Intelligence Research*, vol. 49, pp. 451-500, 2014.

[38] S. Chandra, L. Khan and F. B. Muhaya, "Estimating Twitter User Location Using Social Interactions – A Content Based Approach," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom)*, Boston, MA, USA, 2011.

[39] T. Sakaki, M. Okazaki and Y. Matsuo, "Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors," in *Proceedings of the 19th international conference on World wide web*, Raleigh, North Carolina, USA, 2010.

[40] H. Abdelhaq, C. Sengstock and M. Gertz, "EvenTweet: Online Localized Event Detection from Twitter," *Proceedings of the VLDB Endowment*, vol. 6, pp. 1326-1329, 2013.

[41] K. Lee, A. Agrawal and A. Choudhary, "Real-Time Disease Surveillance Using Twitter Data: Demonstration on Flu and Cancer," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, Chicago, Illinois, USA, 2013.

[42] T. Cheng and T. Wicks, "Event Detection using Twitter: A Spatio-Temporal Approach," *PLOS ONE*, vol. 9, p. e97807, 2014.

[43] J. Krumm and E. Horvitz, "Eyewitness: Identifying Local Events via Space-Time Signals in Twitter Feeds," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle, Washington, USA, 2015.

[44] C. Zhang, G. Zhou, Q. Yuan, H. Zhuang, Y. Zheng, L. Kaplan, S. Wang and J. Han, "GeoBurst: Real-Time Local Event Detection in Geo-Tagged Tweet Streams," in *the 39th International ACM SIGIR conference*, Pisa, Italy, 2016.

[45] L. Zhao, J. Ye, F. Chen, C.-T. Lu and N. Ramakrishnan, "Hierarchical Incomplete Multi-source Feature Learning for Spatiotemporal Event Forecasting," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA, 2016.

[46] C. Zhang, L. Liu, D. Lei, Q. Yuan, H. Zhuang, T. Hanratty and J. Han, "TrioVecEvent: Embedding-Based Online Local Event Detection in Geo-Tagged Tweet Streams," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax, Canada, 2017.

[47] S. Beliga, "Keyword extraction: a review of methods and approaches," 2014.

[48] S. Jones and M. S. Staveley, "Phrasier: a system for interactive document retrieval using keyphrases," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, Berkeley, California, USA, 1999.

[49] C. Chua and C. Woodward, "Keyword Extraction from Meeting Documents for Search and Retrieval," in *Proceedings of the 8th National Natural Language Processing Research Symposium*, De La Salle University, Manila, 2011.

[50] Y. Zhang, N. Zincir-Heywood and E. Milios, "World wide web site summarization," *Web Intelligence and Agent Systems: An International Journal,* pp. 39-53, 2004.

[51] A. Hulth and B. B. Megyesi, "A study on automatically extracted keywords in text categorization," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Sydney, Australia , 2006.

[52] G. Berend, "Opinion expression mining by exploiting keyphrase extraction," in *Proceedings of the 5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand, 2011.

[53] M. Habibi and A. Popescu-Belis, "Keyword Extraction and Clustering for Document Recommendation in Conversations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 23, pp. 746 - 759, 2015.

[54] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan and Q. Zhang, "TIARA: A Visual Exploratory Text Analytic System," in *KDD '10 Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington DC, DC, USA, 2010.

[55] N. Diakopoulos, M. Naaman and F. Kivran-Swaine, "Diamonds in the Rough: Social Media Visual Analytics for Journalistic Inquiry," in *VAST 10 - IEEE Conference on Visual Analytics Science and Technology 2010, Proceedings*, Salt Lake City, UT, USA, 2010.

[56] P. Isenberg, T. Isenberg, M. Sedlmair, J. Chen and T. Möller, "Visualization as Seen Through its Research Paper Keywords," *IEEE Transactions on Visualization and Computer Graphics,* vol. 23, pp. 771-780, 2016.

[57] D. D. Palmer, "Tokenisation and sentence segmentation," *Handbook of Natural Language Processing*, pp. 11-36, 2000.

[58] J. Read, R. Dridan, S. Oepen and L. J. Solberg, "Sentence Boundary Detection: A Long Solved Problem?," in *Proceedings of COLING 2012: Posters*, Mumbai, India, 2012.

[59] A. Joorabchi and A. E. Mahdi, "Automatic Keyphrase Annotation of Scientific Documents Using Wikipedia and Genetic Algorithms," *Journal of Information Science,* vol. 39, pp. 410-426, 2013.

[60] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM Journal of Research and Development,* vol. 1, pp. 309-317, 1957.

[61] I. Witten, G. Paynter, E. Frank, C. Gutwin and C. Nevill-Manning, "KEA: Practical Automatic Keyphrase Extraction," in *DL '99 Proceedings of the fourth ACM conference on Digital libraries*, Berkeley, California, USA, 1999.

[62] O. Medelyan, E. Frank and I. H. Witten, "Human-competitive tagging using automatic keyphrase extraction," in *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009.

[63] B. Kessler, "Computational dialectology in Irish Gaelic," in *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics*, Dublin, Ireland, 1995.

[64] W. W. Cohen, P. Ravikumar and S. E. Fienberg, "A Comparison of String Metrics for Matching Names and Records," in *Proc. International Workshop on Data Cleaning and Object Consolidation at the 9th International Conference on Knowledge Discovery and Data Mining (KDD)*, Washington, DC, USA, 2003.

[65] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes and A. Jatowt, "YAKE! Collection-independent Automatic Keyword Extractor," in *40th European Conference on Information Retrieval*, Grenoble, France, 2018.

[66] L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," in *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, 1998.

[67] M. Timonen, T. Toivanen, Y. Teng, C. Cheng and L. He, "Informativeness-based Keyword Extraction from Short Documents," in *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, Barcelona, Spain, 2012.

[68] W. X. Zhao, J. Jiang, J. He, Y. Song, P. Achananuparp, E.-P. Lim and X. Li, "Topical Keyphrase Extraction from Twitter," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, 2011.

[69] L. Marujo, W. Ling, I. Trancoso, C. Dyer, A. W. Black, A. Gershman, D. M. d. Matos, J. P. Neto and J. Carbonell, "Automatic Keyword Extraction on Twitter," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, vol. 2, pp. 637-643, 2015.

[70] O. Medelyan, V. Perrone and I. H. Witten, "Subject metadata support powered by Maui," *Proceedings of the 2010 joint international conference on Digital libraries*, pp. 407-408, 2010.

[71] W. Wu, B. Zhang and M. Ostendorf, "Automatic Generation of Personalized Annotation Tags for Twitter Users," in *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles,

California, 2010.

[72] S. Wang, Z. Chen, B. Liu and S. Emery, "Identifying Search Keywords for Finding Relevant Social Media Posts," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, Phoenix, Arizona, 2016.

[73] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning,* vol. 20, pp. 273-297, 1995.

[74] I. Steinwart and A. Christmann, Support Vector Machines, Springer, 2008.

[75] J. R. Quinlan, Programs for Machine Learning, San Mateo, CA, USA: Morgan Kaufmann Publishers, 1993.

[76] D. J. Hand and K. Yu, "Idiot's Bayes: Not So Stupid after All?," *International Statistical Review / Revue Internationale de Statistique,* vol. 69, pp. 385-398, 2011.

[77] P. Rämö, R. Sacher, B. Snijder, B. Begemann and L. Pelkmans, "CellClassifier: supervised learning of cellular phenotypes," *Bioinformatics,* vol. 25, pp. 3028-3030, 2009.

[78] O. Kouropteva, O. Okun and M. Pietikainen, "Supervised Locally Linear Embedding Algorithm for Pattern Recognition," in *Pattern Recognition and Image Analysis*, Puerto de Andratx, Mallorca, Spain, 2003.

[79] J. Li, S. Li and C. Cardie, "TopicSpam: a Topic-Model-Based Approach for Spam Detection," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2013.

[80] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in *10th European Conference on Machine Learning*, Chemnitz, Germany, 1998.

[81] P. D. Turney, "Learning Algorithms for Keyphrase Extraction," *Information Retrieval,* vol. 2, p. 303–336, 2000.

[82] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky and Y. Chi, "Deep Keyphrase Generation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017.

[83] Q. Zhang, Y. Wang, Y. Gong and X. Huang, "Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, 2016.

[84] Y. Hu, S. Farnham and K. Talamadupula, "Predicting User Engagement on Twitter with Real-World Events," in *Proceedings of the Ninth International AAAI Conference on Web and Social Media*, 2015.

[85] B. Lott, "Survey of Keyword Extraction Techniques," UNM Education, 2012.

[86] K. S. Hasan and V. Ng, "Automatic Keyphrase Extraction: A Survey of the State of the Art," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, Maryland, USA, 2014.

[87] A. Hulth, "Improved Automatic Keyword Extraction Given More Linguistic Knowledge," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'03)*, Sapporo, Japan, 2003.

[88] S. N. Kim, O. Medelyan, M.-Y. Kan and T. Baldwin, "SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala, Sweden, 2010.

[89] M. Krapivin, A. Autaeu and M. Marc, "Large Dataset for Keyphrases Extraction," DISI-09-055, University of Trento, 2009.

[90] T. D. Nguyen and M.-Y. Kan, "Keyphrase Extraction in Scientific Publications," in *ICADL 2007: Proceedings of the 10th International Conference on Asian Digital Libraries*, Hanoi, Vietnam. December 10-13, 2007.

[91] J. Littman, L. Wrubel and D. Kerchner, "2016 United States Presidential Election Tweet Ids," Harvard Dataverse, 2016.

[92] L. Cram, C. Llewellyn, R. Hill and W. Magdy, "UK General Election 2017: a Twitter Analysis," 2017.

[93] K. Darwish, W. Magdy and T. Zanouda, "Trump vs. Hillary: What went Viral during the 2016 US Presidential Election," in *International Conference on Social Informatics*, 2017.

[94] L. Derczynski, K. Bontcheva and I. Roberts, "Broad Twitter Corpus: A Diverse Named Entity Recognition Resource," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan, 2016.

[95] A. Suarez, D. Albakour, D. Corney, M. Martinez and J. Esquivel, "A Data Collection for Evaluating the Retrieval of Related Tweets to News Articles," in *European Conference on Information Retrieval*, 2018.

[96] D. Corney, M.-D. Albakour, M. Martinez-Alvarez and S. H. Moussa, "What do a Million News Articles Look like?," in *Proceedings of NewsIR'16 Workshop at ECIR*, 2016.

[97] I. Brigadir, D. Greene and P. Cunningham, "Detecting attention dominating moments across media types," in *Proceedings of NewsIR'16 Workshop at ECIR*, Padua, Italy, 2016.

[98] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan and X. Li, "Comparing Twitter and Traditional Media Using Topic Models," in *Advances in Information Retrieval*, 2011.

[99] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan and N. A. Smith, "Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, Stroudsburg, PA, USA, 2011.

[100] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological Bulletin,* pp. 378-382, 1971.

[101] J. Cohen, "A coefficient of agreement for nominal scales," *Education and Psychological Measurement,* pp. 249-254, 1960.

[102] L. Sterckx, C. Caragea, T. Demeester and C. Develder, "Supervised Keyphrase Extraction as Positive Unlabeled Learning," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, Texas, USA, 2016.

[103] C. Orellana-Rodriguez and M. T. Keane, "Attention to news and its dissemination on Twitter: A survey," *Computer Science Review*, p. 94, 2018.

[104] H. P. Luhn, "The Automatic Creation of Literature Abstracts," *IBM Journal of Research and Development,* vol. 2, pp. 159 - 165, 1958.

[105] D. Machado, T. Barbosa, . S. Pais, B. Martins and G. Dias, "Universal Mobile Information Retrieval," in *UAHCI '09 Proceedings of the 5th International on ConferenceUniversal Access in Human-Computer Interaction. Part II: Intelligent and Ubiquitous Interaction Environments*, San Diego, CA, 2009.

[106] F. Boudin, "pke: an open source python-based keyphrase extraction toolkit," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, Osaka, Japan, 2016.