UNED Escuela de Doctorado

PhD Thesis

2014

ENTITY-BASED FILTERING AND TOPIC DETECTION FOR ONLINE REPUTATION MONITORING IN TWITTER

DAMIANO SPINA VALENTI

MASTER IN LANGUAGES AND INFORMATION SYSTEMS, UNED

DOCTORAL PROGRAMME IN INTELLIGENT SYSTEMS

JULIO GONZALO ARROYO

ENRIQUE AMIGÓ CABRERA

UNED Escuela de Doctorado

PhD Thesis

2014

ENTITY-BASED FILTERING AND TOPIC DETECTION FOR ONLINE REPUTATION MONITORING IN TWITTER

DAMIANO SPINA VALENTI

MASTER IN LANGUAGES AND INFORMATION SYSTEMS, UNED

DOCTORAL PROGRAMME IN INTELLIGENT SYSTEMS

JULIO GONZALO ARROYO

ENRIQUE AMIGÓ CABRERA

Dona Maria,
Me dá água de beber,
A água do coco é doce,
Eu também quero beber.

Se iaiá quiser me ver,
Bote seu navio no mar.
Eu também sou marinheiro,
Também quero navegar, camará.
Água de beber. . .

(Capoeira's Popular Sayings)

Acknowledgements

It is hard to capture in a few words the gratitude I owe to my supervisors Dr. Julio Gonzalo and Dr. Enrique Amigó. All their patience, timely help, and continuous belief in me have been of immeasurable value to me. I always be indebted for all I have learned with them. If I were to describe only one thing, it would be that doing research is like playing music or playing capoeira: it must be done wholeheartedly.

I would like to express my gratitude to the external reviewers Dr. Manos Tsagkias —who also generously accepted to review the thesis proposal— and Dr. Alexandra Balahur. Their advice and comments have been a great help in the writing of this thesis.

I am very grateful and owe a very important debt to Prof. Maarten de Rijke, Dr. Edgar Meij and the Information and Language Processing Systems (ILPS) Group who hosted me during my 4-month stay at the University of Amsterdam. I want to thank them for all their help and support. The fruitful collaboration with them —during and after my stay— has been of significant benefit for this dissertation.

I am gratefully indebted to all my co-authors —especially to Dr. Arkaitz Zubiaga, Dr. Jorge Carrillo de Albornoz, Dr. Irina Chugur, Tamara Martín, Maria-Hendrike Peetz, Dr. Víctor Fresno, Dr. Raquel Martínez and Dr. Laura Plaza—for all the illuminating discussions, selfless dedication and interest for collaborating with me.

It would not have been possible to carry out this work without the help of Vanessa Álvarez, Ana Pitart, Adolfo Corujo and Miguel Lucas, who always expressed their full willingness to cooperate in this work as Online Reputation Management experts.

I want to thank all my former and current colleagues at the UNED NLP&IR Group and at the Lenguajes y Sistemas Informáticos (LSI) Department at UNED. It is hard to imagine a better and more enjoyable work environment. I indebted to Prof. Felisa Verdejo and Dr. Fco. Javier Garijo for giving me the opportunity to work here and for their guidance in my professional career.

Last, but not least, I would like to acknowledge all the support provided by my parents Giovanni M. Spina and Caterina N. Valenti, my brothers Christian, Leonardo and Michelangelo during all these years. I will always owe a very important debt to my beloved Stenia Stein for her patience and encouragement. Finally, I want also to thank Carlos A. Moreira da Silva (Mestre Pantera) and my friends from the ACDP (Associação de Capoeira Descendente do Pantera) for helping me on maintaining the body, mind and soul balance throughout this long way. I also want to thank all my friends and family who —near or far— always supported me.

Institutional Acknowledgements. The research presented in this thesis was partially supported by the European Community's FP7 Programme under grant agreement nr. 288024 (LiMoSINe), the Spanish Ministry of Education (FPU grant nr. AP2009-0507), the Spanish Ministry of Science and Innovation (Holopedia Project, TIN2010-21128-C02), the Regional Government of Madrid under MA2VICMR (S2009/TIC-1542) and the European Science Foundation (ESF) Research Network Programme ELIAS.

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Abstract

Programa de Doctorado en Sistemas Inteligentes
Escuela de Doctorado de la UNED

Doctor of Philosophy in Computer Science

Entity-Based Filtering and Topic Detection for Online Reputation Monitoring in Twitter

by Damiano Spina Valenti

With the rise of social media channels such as Twitter —the most popular microblogging service— the control of what is said about entities —companies, people or products— online has been shifted from them to users and consumers. This has generated the necessity of monitoring the reputation of those entities online. In this context, it is only natural to witness a significant growth of demand for text mining software for Online Reputation Monitoring: automatic tools that help processing, understanding and aggregating large streams of facts and opinions about a company or individual. Despite the variety of Online Reputation Monitoring tools on the market, there is no standard evaluation framework yet —a widely accepted set of task definitions, evaluation measures and reusable test collections to tackle this problem. In fact, there is even no consensus on what the tasks carried out during the Online Reputation Monitoring process are, on which a system should minimize the effort of the user.

In the context of a collective effort to identify and formalize the main challenges in the Online Reputation Monitoring process in Twitter, we have participated in the definition of tasks and subsequent creation of suitable test collections (WePS-3, RepLab 2012 and RepLab 2013 evaluation campaigns) and we have studied in depth two of the identified challenges: *filtering* (Is a tweet related to a given entity of interest?) —modeled as a binary classification task— and *topic detection* (What is being said about an entity in a given tweet stream?), that consists of clustering tweets according to topics. Compared to previous studies on Twitter, our problem lies in its *long tail*: except for a few exceptions, the volume of information related to a specific entity (organization or company) at a given time is orders of magnitude smaller than Twitter trending topics, making the problem much more challenging than identifying Twitter trends.

We rely on three building blocks to propose different approaches to tackle these two tasks : the use of *filter keywords*, *external resources* (such as Wikipedia, representative pages of the entity of interest, etc.) and the use of *entity-specific training data* when available.

We have found that the notion of *filter keywords* —expressions that, if present in a tweet, indicate a high probability that it is either related or unrelated to the entity of interest— can be effectively used to tackle the filtering task. Here, (i) specificity of a term to the tweet stream of the entity

is a useful feature to identify keywords, and (ii) the association between a term and the entity's Wikipedia page is useful to differentiate positive vs. negative filter keywords, especially when it is averaged by considering its most co-occurrent terms. In addition, exploring the nature of filter keywords also led us to the conclusion that there is a gap between the vocabulary that characterizes a company in Twitter and the vocabulary associated to the company in its homepage, in Wikipedia, and even in the Web at large.

We have also found that, when entity-specific training data is available —as in the known-entity scenario— it is more cost effective to use a simple Bag-of-Words classifier. When enough training data is available (around 700 tweets per entity), Bag-of-Words classifiers can be effectively used for the filtering task. Moreover, they can be used effectively in an active learning scenario, where the system updates its classification model with the stream of annotations and interactions with the system made by the reputation expert along the monitoring process. In this context, we found that by selecting the tweets to be labeled as those on which the classifier is less confident (margin sampling), the cost of creating a bulk training set can be reduced by 90% after inspecting 10% of test data. Unlike many other applications of active learning on Natural Language Processing tasks, margin sampling works better than random sampling.

As for the topic detection problem, we considered two main strategies: the first is inspired on the notion of filter keywords and works by clustering terms as an intermediate step towards document clustering. The second — and most successful — learns a pairwise tweet similarity function from previously annotated data, using all kinds of content-based and Twitter-based features; and then applies a clustering algorithm on the previously learned similarity function. Our experiments indicate that (i) Twitter signals can be used to improve the topic detection process with respect to using content signals only; (ii) learning a similarity function is a flexible and efficient way of introducing supervision in the topic detection clustering process. The performance of our best system is substantially better than state-of-the-art approaches and gets close to the inter-annotator agreement rate of topic detection annotations in the RepLab 2013 dataset —to our knowledge, the largest dataset available for Online Reputation Monitoring. A detailed qualitative inspection of the data further reveals two types of topics detected by reputation experts: reputation alerts / issues (which usually spike in time) and organizational topics (which are usually stable across time).

Along with our contribution to building a standard evaluation framework to study the Online Reputation Monitoring problem from a scientific perspective, we believe that the outcome of our research has practical implications and may help the development of semi-automatic tools to assist reputation experts in their daily work.

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Resumen

Programa de Doctorado en Sistemas Inteligentes
Escuela de Doctorado de la UNED

Doctor en Informática

Filtrado y Detección de Temas para la Monitorización de la Reputación en Línea de Entidades en Twitter

por Damiano Spina Valenti

Con el crecimiento de los medios sociales de comunicación en línea como Twitter (el servicio más popular de *microblogging*), los usuarios y consumidores han pasado a tener el control de lo que se dice acerca de una entidad (p.e., una compañía, un personaje público o una marca) en la Web. Este fenómeno ha creado la necesidad de monitorizar la reputación de dichas entidades en línea. En este ámbito, es esperable un aumento de la demanda de software de minería de textos para la monitorización de la reputación en línea (en inglés, Online Reputation Monitoring): herramientas automáticas que ayudan a procesar, analizar y agregar grandes flujos de menciones acerca de una compañía, organización o personaje público. A pesar de la gran variedad de herramientas disponibles en el mercado, no existe aún un marco de evaluación estándar (es decir, un conjunto de tareas bien definidas, métricas de evaluación y colecciones reutilizables ampliamente aceptados) que permita abordar este problema desde un punto de vista científico.

En un marco de esfuerzo colectivo para identificar y formalizar los principales desafíos en el proceso de gestión de reputación en Twitter, hemos participado en la definición de tareas de acceso a la información, así como en la creación de colecciones de test (utilizadas en las campañas de evaluación WePS-3, RepLab 2012 y RepLab 2013) y hemos estudiado en profundidad dos de los desafíos identificados: *filtrado* de contenido no relevante (¿está relacionado un tweet dado con la entidad de interés?), modelado como una tarea de clasificación binaria, y *detección de temas* (¿qué se dice de la entidad en un flujo de tweets dado?), donde los sistemas deben agrupar los tweets en función de los temas tratados. En comparación con otros estudios sobre Twitter, nuestro problema se encuentra en su *cola larga*: salvando algunas excepciones, el volumen de información relacionado con una entidad dada (organización o compañía) en un determinado intervalo de tiempo es varios órdenes de magnitud más pequeño que los *trending topics* de Twitter, aumentando así su complejidad respecto a la identificación de los temas más populares en Twitter.

En esta tesis nos basamos en tres conceptos para proponer distintas aproximaciones para abordar estas dos tareas: el uso de términos clave filtro (*filter keywords*), el uso de recursos externos (como Wikipedia, páginas web representativas de la entidad, etc.) y el uso de datos de entrenamiento específicos de la entidad (cuando éstos estén disponibles). Nuestros experimentos

revelan que la noción de términos clave filtro (palabras que indican una alta probabilidad de que el tweet en el que aparecen esté relacionado o no con la entidad de interés) puede eficazmente ser utilizada para resolver la tarea de filtrado. En concreto, (a) la especificidad de un término con respecto al flujo de tweets de la entidad es un rasgo útil para identificar términos clave; y (b) la asociación entre el término y la página de la entidad en Wikipedia es útil para distinguir entre términos filtro positivos y negativos, especialmente cuando se calcula su valor medio teniendo en cuenta los términos más co-ocurrentes. Además, estudiando la naturaleza de los términos filtro hemos llegado a la conclusión de que existe una brecha terminológica entre el vocabulario que caracteriza la entidad en Twitter y el vocabulario asociado a la entidad en su página principal, Wikipedia o en la Web en general. Por otro lado, hemos hallado que, cuando se dispone de material de entrenamiento para la entidad en cuestión , es más efectivo el uso de un simple clasificador basado en bolsa de palabras. Existiendo suficientes datos de entrenamiento (unos 700 tweets por entidad), estos clasificadores pueden ser utilizados eficazmente para resolver la tarea de filtrado. Además, pueden utilizarse con éxito en un escenario de aprendizaje activo (*active learning*), en el que el sistema va actualizando su modelo de clasificación en función del flujo de anotaciones realizadas por el experto de reputación durante el proceso de monitorización. En este contexto, seleccionado los tweets en los que el clasificador tiene menos confianza (muestreo basado en márgenes) como aquellos que deben ser etiquetados por el experto, el coste de crear el conjunto inicial de entrenamiento puede llegar a reducirse en un 90% sólo inspeccionando el 10% de los datos de test. A diferencia de otras tareas de Procesamiento del Lenguaje Natural, el muestreo basado en márgenes funciona mejor que un muestreo aleatorio.

Con respecto a la tarea de detección de temas, hemos considerado principalmente dos estrategias: la primera, inspirada en la noción de palabras término filtro, consiste en agrupar términos como un paso intermedio para la agrupación de tweets. La segunda, más exitosa, se basa en aprender una función de similitud entre pares de tweets a partir de datos previamente anotados, utilizando tanto rasgos basados en contenido como el resto de señales proporcionadas por Twitter; luego se aplica un algoritmo de agrupación sobre la función de similitud aprendida previamente. Nuestros experimentos revelan que (a) las señales Twitter pueden usarse para mejorar el proceso de detección de temas con respecto a utilizar sólo señales basadas en contenido; (b) aprender una función de similitud a partir de datos previamente anotados es una forma flexible y eficiente de introducir supervisión en el proceso de detección de temas. El rendimiento de nuestro mejor sistema es sustancialmente mejor que las aproximaciones del estado del arte, y se acerca al grado de acuerdo entre anotadores en las anotaciones de detección de temas incluidas en la colección RepLab 2013 (a nuestro conocimiento, la colección más grande para la monitorización de la reputación en línea). Una inspección cualitativa de los datos muestra que existen dos tipos de temas detectados por los expertos de reputación: alertas o incidentes de reputación (que normalmente sobresalen en el tiempo) y temas organizacionales (que, en cambio, suelen ser estables en el tiempo).

Junto con nuestra contribución para crear un marco estándar de evaluación para el estudio del problema de la monitorización de la reputación en línea desde una perspectiva científica, creemos que el resultado de nuestra investigación tiene implicaciones prácticas que pueden servir para beneficiar el desarrollo de herramientas semi-automáticas que asistan a los expertos en reputación en su trabajo diario de monitorización.

Contents

List of Figures

List of Tables

Abbreviations

AP	**A**verage **P**recision
AUC	**A**rea **U**nder the **C**urve
BoW	**B**ag **o**f **W**ords
CEO	**C**hief **E**xecutive **O**fficer
IR	**I**nformation **R**etrieval
HAC	**H**ierarchical **A**gglomerative **C**lustering
INEX	**IN**itiative for the **E**valuation of **X**ML Retrieval
KBP	**K**nowledge **B**ase **P**opulation
KLD	**K**ullback-**L**eibler **D**ivergence
lam%	**l**ogistic **a**verage **m**isclassification percentage
LDA	**L**atent **D**irichlet **A**llocation
LLR	**L**og-**L**ikelihood **R**atio
MAP	**M**ean **A**verage **P**recision
MAR	**M**ean **A**verage **R**eliability
ML	**M**achine **L**earning
MRR	**M**ean **R**eciprocal **R**ank
MS	**M**argin **S**ampling
NED	**N**amed **E**ntity **D**isambiguation
NER	**N**amed **E**ntity **R**ecognition
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
NLP	**N**atural **L**anguage **P**rocessing
ODP	**O**pen **D**irectory **P**roject
ORM	**O**nline **R**eputation **M**onitoring
PLM	**P**arsimonious **L**anguage **M**odels
PLSA	**P**robabilistic **L**atent **S**emantic **A**nalysis

ROUGE	**R**ecall-**O**riented **U**nderstudy for **G**isting **E**valuation
R&S	**R**eliability & **S**ensitivity
RS	**R**andom **S**ampling
SMO SVM	**S**equential **M**inimal **O**ptimization **S**upport **V**ector **M**achines
SMS	**S**hort **M**essage **S**ervice
SVM	**S**upport **V**ector **M**achines
UGC	**U**ser **G**enerated **C**ontent
URL	**U**niform **R**esource **L**ocator
TDT	**T**opic **D**etection & **T**racking
TF	**T**erm **F**requency
TF.IDF	**T**erm **F**requency **I**nverse **D**ocument **F**requency
TREC	**T**ext **RE**trieval **C**onference
WePS	**We**b **P**eople **S**earch
WSD	**W**ord **S**ense **D**isambiguation

1 Introduction

1.1 Motivation

Reputation Management

Reputation Monitoring

Reputation management may be defined as the process of (i) monitoring the reputation of an *entity*—individual, company, organization or brand—in order to know and understand which information, opinions and comments are being shared about it, and (ii) addressing negative contents that can potentially damage its corporate image [55]. Focusing on reputation monitoring, this has been traditionally based on manual analysis of media contents and consumers' complaints, as well as on conducting targeted and expensive market surveys. However, with the rise of online social media, new channels of communication between companies and their audience have appeared, and the nature of the reputation monitoring process has significantly changed. People talk and complain on blogs, forums, opinion sites, social networks such as Facebook and LinkedIn, on Twitter, and in YouTube videos —to name only a few of the channels available. Social media provide means for information to emerge and move constantly. Mentions about people, companies or products are generated 24 hours a day and are quickly spread over large communities. The control of what is said about public figures and organizations, at least partly, has moved from them to users and consumers [59, 80]. Thus, the vast use of social media to share facts and opinions about entities, such as companies, brands and public figures has generated the necessity of managing the reputation of those entities online.

ORM: Online Reputation Monitoring

In this new scenario, Online Reputation Monitoring (ORM) faces two main and novel challenges: the dramatic growth of the data potentially important for an entity's reputation, and the need to quickly detect potentially dangerous contents [44, 72, 144, 182]. The enormous amounts of User Generated Content (UGC) require a much bigger effort in terms of human resources. The cost of an error—e.g., overlooking a possible reputation menace—is higher than in traditional mass media, due to the real-time nature and the potential impact enabled by social media. Besides, the big amounts of mentions generated in social media give the opportunity to discover new insights about the entity's image. A number of recent studies have shown that aggregating millions of apparently insignificant comments can reveal interesting data about users [49, 103, 106, 140].

Online reputation sources offer new opportunities for information technologies to improve and facilitate the reputation management process. In this context, it is only natural to witness a significant growth of demand for text mining software for ORM, automatic tools that could help to process, understand and aggregate large streams of facts and opinions about a company or individual. Over the last years, a wide variety of tools have been developed that facilitate the Online Reputation Monitoring (ORM) task. At the time of writing this, some popular ORM tools are Trackur [169], BrandChats [30], Nielsen BuzzMetrics [136], Salesforce Marketing Cloud [152] or SocialMention [159], among others. ORM Tools

These tools typically follow a three-step process:

1. **Retrieval of potential mentions.** The user gives as input a set of keywords (e.g., the company name, products, CEO's name...), and the service uses these keywords to retrieve documents and content generated by users from different sources: broadcast news sources, social media, blogs, microblogging services, etc.
2. **Analysis of results.** Retrieved documents are automatically processed in order to get relevant information to the user: sentiment, authority and influence, background topics, etc.
3. **Results visualization.** Analyzed data is presented to the user in different ways: ranking documents, drawing graphics, generating tag clouds, etc.

Despite the variety of solutions available on the market, current tools do not seem to perfectly match user needs and are difficult to personalize. For instance, while most of the tools claim to get over 70% accuracy —on average— on detect sentiment in social media texts,[1] it is not trivial for reputational experts to predict how the tool will perform on the real data of their daily work.

More importantly, there is no standard evaluation framework —a defined set of evaluation measures and reusable test collections to tackle this problem— that allows tools benchmarking. In fact, there is no even consensus on what the tasks carried out during the ORM process on which a system should minimize the effort of the user are. For instance, some aspects worth to be tackled are: how many relevant mentions are covered by the system? To what extent irrelevant mentions are correctly filtered out? Is it possible to detect topics that may damage the reputation of the entity automatically? Evaluation

Therefore, there is a need to formalize the ORM problem from a scientific perspective, in order to state the main research challenges and provide a standard evaluation framework that will allow us —and, in general, the Information Retrieval (IR) and Natural Language Processing (NLP) research communities— to explore novel solutions to the detected challenges. Additionally, it will help the experts to understand how much of the problem can be solved automatically and obtain the maximum benefit from the tools that may be developed from our research findings.

[1] http://www.theguardian.com/news/datablog/2013/jun/10/social-media-analytics-sentiment-analysis

1.2 Scope of the Thesis

Twitter

In this thesis we focus on monitoring the reputation of entities in Twitter, the most popular microblogging service, since it is the most important social media source for ORM [80]. Among other things, it is considered the first source for the latest news [98]. Besides, there are some characteristics that make Twitter a challenging source from an IR & NLP point of view. First, tweets are short (only 140 characters), ubiquitous (most of them are publicly accessible from anywhere) and have a real-time nature (more than 5k tweets are produced per second[2]). Second, tweets have little context due to its short nature (140 characters, around 15 words), and their language is not always standard: freely-chosen tags (*folksonomies*, known as *hashtags* in Twitter), slang expressions, abbreviations and misspellings are common. Finally, given the multilingual nature of the ORM problem, we want to consider different languages in our problem setting.

Social Media Sources

Monitoring and linking together heterogeneous social media sources (Twitter, Facebook, weblogs, online forums, news, news comments, etc.) is itself a complex problem that needs different solutions for different media [170]. For instance, considering sources with longer and threaded text units like forums, blogs and news implies the task of identifying parts of the document that are relevant to the entity of interest (paragraphs, sentences, comments, etc.). In order to keep the scope of our work manageable, we will focus on Twitter data only.

Head vs. *Long* Tail in Twitter

Most research about Twitter tackles the problem of analyzing the microblogging phenomenon and detecting events that are relevant for a large audience (e.g., Twitter trending topics), which corresponds to a *dense* scenario. Different from this, in our ORM context we will focus on the Twitter's long tail, analyzing the presence of entities—companies, organizations or music bands

Sparsity

that are not necessarily trending—in Twitter. Therefore, we will have to deal with *sparsity* issues (e.g., term frequency is insufficient to detect tweets that are talking about the same topic) which may invalidate the assumptions under which some existent techniques have been explored so far.

While in this thesis we will focus on algorithmic effectiveness —i.e., accuracy and coverage—, it is worth caring about the suitability of the proposed techniques to analyze and mine social

Real-Time

streams in *real-time*, i.e., processing Twitter data as it is made available. Therefore, we will make sure that our proposed approaches can be used in real-time monitoring scenarios, which is one of the desired requirements for an ORM assistant tool.

1.3 Problem Statement

The main purpose of this thesis is to understand, formalize and explore the scientific challenges

ORM in Twitter: Tasks

inherent to the problem of Online Reputation Monitoring in Twitter. With the help of reputation experts, we will decompose the ORM process in different tasks: (1) continuously searchning the stream of tweets for potential mentions to the entity of interest, and filtering out those that do

[2] http://www.statisticbrain.com/twitter-statistics/

not refer to the entity, (2) detecting the topics the different tweets talk about, and (3) ranking such topics based on their potential impact on the reputation of the entity.

More in detail:

Filtering: Analysts are asked to determine which tweets are related to the entity and which are not. For instance, distinguishing between tweets that contain the word *Stanford* referring to the University of Stanford and filtering out tweets about Stanford as a place.

Topic detection: Related tweets referring to the same subject or event are grouped into a same *topic*. The aim is to identify the most *popular* issues about the entity in Twitter. Examples of possible topics are, for instance, `Mortgages` and `Money Laundering`, when analyzing a banking company's reputation.

Priority assignment: Previously identified topics are ranked depending on their relevance for the entity's reputation. In the top of the ranking are those topics that deserve immediate attention of reputation managers, while topics that can be neglected from a reputation management perspective are listed at the bottom. Some of the factors that play a role in the priority assessments are: the *centrality* of the topic, *influence* of users that discuss on the topic, *freshness* of the topic or reputation *polarity* (positive or negative) of the tweets that talk about the topic.

1.3.1 Scenarios

The problem of Online Reputation Monitoring in Twitter can be considered in two different scenarios, hereafter referred as *unknown-entity* and *known-entity*, that we introduce now.

Unknown-Entity Scenario

In order to illustrate the *unknown-entity* scenario, let us consider that the reputation manager uses services like Topsy[3] or SocialMention[4] for monitoring the reputation of an entity (Figure 1.1).

Here, the user typically issues a query (e.g., `ford`) and the system returns a list of potentially relevant mentions, accompanied by some related keywords and some statistics—strength, passion, sentiment—that might be helpful to analyze the public image of the entity. From the algorithmic perspective, the system has to identify relevant mentions and compute statistics for *any* entity submitted as query by the users. Given that our user is a reputation manager that is looking for a particular entity, in our setting we assume that the entity of interest is identified by a representative URL —for instance, the entity's homepage or its Wikipedia article— in addition to the text query. A crucial feature of this scenario if that, since the entity is unknown, supervised models have to be learned from data associated to other entities.

Known-Entity Scenario

The *known-entity* scenario corresponds to the iterative monitoring process carried out by reputation experts in a Public Relations consultancy. For each customer —typically, the entity of interest— reputation managers have to spend big efforts in retrieving, analyzing and reporting

[3] http://topsy.com
[4] http://socialmention.com

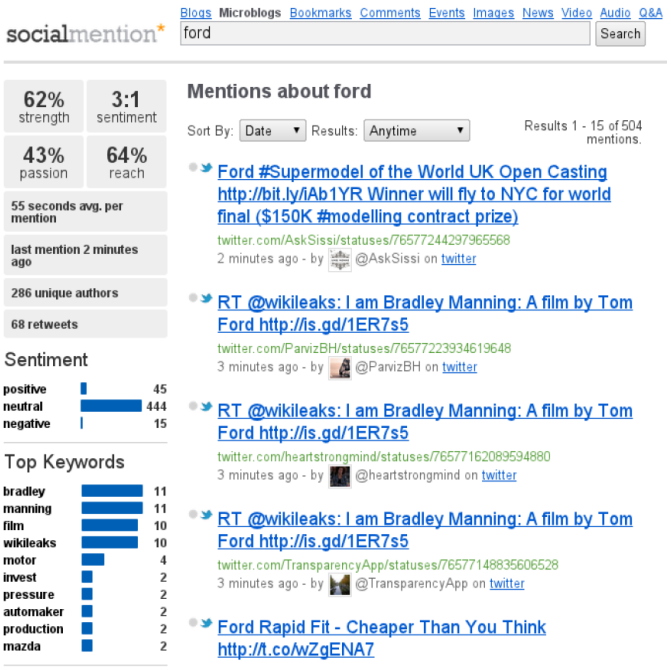


FIGURE 1.1: SocialMention's search results and statistics for the query `ford`.

what people say about a given company, and what the main topics they should keep tracking are. This process is done manually and repeated daily, weekly or monthly, depending on the entity of interest and the volume of relevant data generated in Twitter.[5] Unlike the *unknown-entity* scenario, in the *known-entity* scenario it is assumed that there is already some annotated data about the entity.

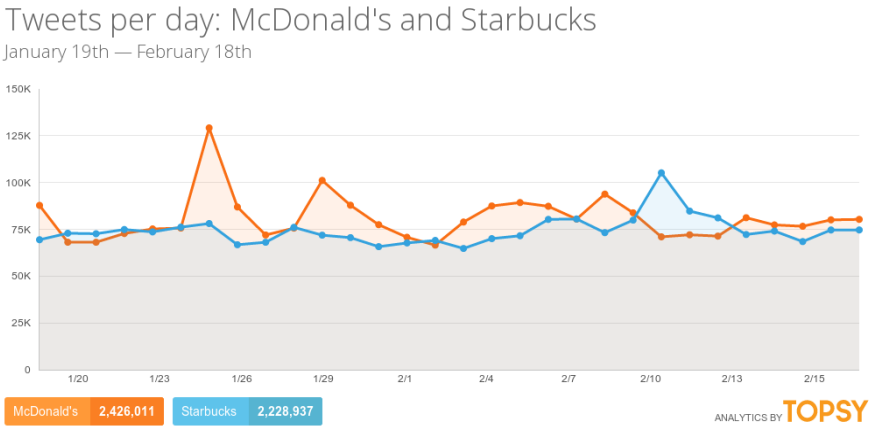


FIGURE 1.2: Frequency of tweets containing the queries `McDonald's` and `Starbucks` by Topsy Analytics.

Although we are looking at the long tail of Twitter, and there may not be enough redundancy for applying simple statistical methods, it is still too much data to tackle the problem manually. In both scenarios, alleviating the tedious task of manually analyzing thousands of tweets with the help of information access technologies will significantly reduce the effort of retrieving and

[5] An entity that is usually analyzed monthly might be analyzed daily when some specific event is going on, e.g., two companies are being merged together.

analyzing relevant mentions of a given entity in Twitter. Moreover, information access is an *enabling technology* in order to cover all mentions about popular entities. For instance, more than 75k tweets containing the query `McDonald's` or `Starbucks` are produced daily (Figure 1.2).

Note that while in the *unknown-entity* scenario the goal is to maximize the effectiveness of automatically analyzing an unseen entity, in the *known-entity* scenario it is also crucial to reduce the effort of manual annotating each of the entities. One way to minimize this effort is considering the task as an *active learning* problem. Here, the learning process is seen as a semi-automatic procedure. The system first asks to the user to manually annotate a few instances—the least certain to be predicted correctly by the current model. Once the user provides the annotations, the new annotated instances are fed back into the model. In this way, the model is always updated and therefore, is more likely to accurately classify new instances.

Active Learning

1.3.2 Research Goals

Once the scientific challenges are identified, we will focus on how much—qualitatively and quantitatively—of the ORM problem can be solved automatically by our proposed approaches. To this aim, we will address the filtering and the topic detection tasks automatically by applying text mining and machine learning techniques. Here, we will make use of Twitter data and external resources, such as Wikipedia or the Web at large. Both scenarios introduced above —unknown-entity and known-entity— will be explored.

There are three main concepts that will be thoroughly analyzed in this thesis: the discovery of keywords, the use of external data and the use of manually annotated training data.

Keywords. As in other information retrieval tasks, we assume that there exists a set of salient terms that will help the user to understand what is being said about an entity in a given Twitter stream. For the filtering task, we will start by validating our hypothesis of *filter keywords*: those whose presence in a tweet reliably confirm (positive keywords) or discard (negative keywords) that the tweet refers to the entity. This hypothesis —that we will consider as the *building block* to propose similar approaches for the other ORM tasks— is based on the observation that manual annotation can be simplified by picking up filter keywords that reliably signal positive or negative information.[6] Figure 1.3 illustrates this phenomenon. In the example, we have ten tweets returned by querying `apple` when we look for relevant information for the company Apple Inc. All the tweets that contain the keyword `store` refer to the Apple's retail chain. Thus, `store` can be considered as a positive keyword. Reversely, the keyword `eating` can be considered as a negative keyword, since it directly identifies tweets that are not related to Apple. Considering only two keywords, we have perfectly disambiguated six out of ten tweets. We will study to what extent filter keywords can be discovered automatically and their suitability in the two scenarios described above. In the context of the topic detection task we will study the benefits

Filter Keywords

[6] This observation derived from the set-up and the analysis of the results of the WePS-3 Online Reputation Management Task [7], where the author was involved in the definition process of the task and the creation of the collection.

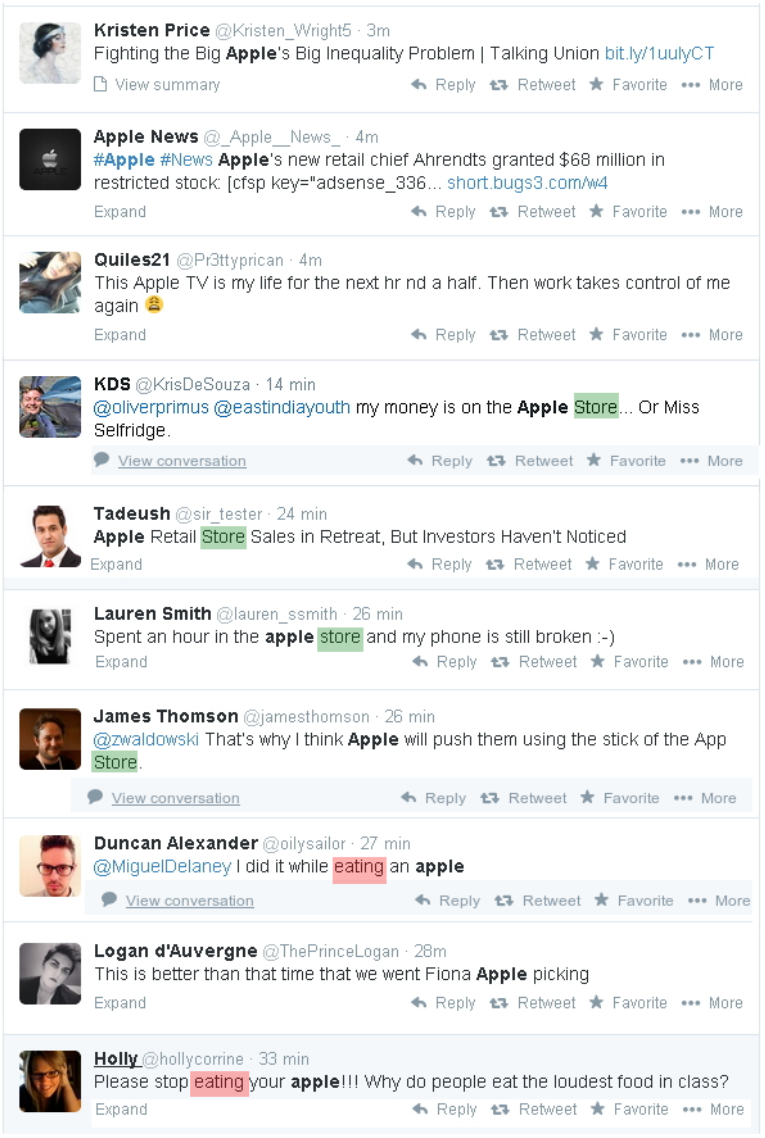


FIGURE 1.3: An example of the *filter keyword* strategy applied to tweets containing the query `apple`.

and limitations of considering cluster keywords, i.e., keywords that clearly identify the topics that are being discussed in a stream of tweets that are relevant to the entity.

Benefits of using external resources. Twitter as a microblogging service offers different signals —authors, hashtags, URLs, timestamps— that may help when discovering useful associations between similar tweets. Besides, web resources such as the entity's homepage or Wikipedia entry, among others, may be useful for discovering keywords or relevant concepts to the entity of interest, specially when there is no entity-specific training data (unknown-entity scenario). We will study the use of both Twitter signals and external resources when tackling the tasks of filtering and topic detection. Additionally, we will study whether topics discussed about an entity can be discovered by *wikification*, i.e., linking tweets to Wikipedia articles.

Use of training data. As we have seen before, the main difference between the two scenarios is the availability of either generic or entity-specific training data. We will study the impact—in terms of effectiveness—of our approaches on the unknown-entity and the known-entity scenarios. Moreover, we will explore to what extent it is possible to reduce the effort of annotating tweets manually for each of the entities, by using active learning.

On top of these main concepts, the following research questions will be investigated throughout this thesis:

> **RQ1:** *Which challenges —when monitoring the reputation of an entity in Twitter— can be formally modeled as information access tasks? Is it possible to make reusable test beds to investigate the tasks?*

Studying the concept of *keywords*:

> **RQ2:** *Can we use the notion of filter keywords effectively to solve the filtering task?*

> **RQ3:** *Can we generalize the idea of "filter keywords" to "cluster keywords", i.e., can we use it for topic detection?*

Related to the use of Twitter signals and *external resources*:

> **RQ4:** *Where should we look for filter keywords in order to find them automatically?*

> **RQ5:** *Wikipedia is a knowledge base that is continuously being updated, and can be a relevant source to discover filter keywords automatically. Are the topics discussed about an entity in Twitter represented somehow in Wikipedia?*

> **RQ6:** *Can Twitter signals be used to improve entity-specific topic detection?*

Related to the use of *training data*:

> **RQ7:** *When entity-specific training data is available, is it worth looking for filter keywords in external resources or is it better to learn them automatically from the training data?*

> **RQ8:** *In an active learning scenario, what is the impact in terms of effectiveness of an informative sampling over a random sampling? How much of the (initial) annotation effort can be reduced by using active learning?*

> **RQ9:** *Can previously annotated material be used to learn better topic detection models?*

1.4 Research Methodology

At the moment of starting this research, there was no standard evaluation setting, and little academic work had been published on the ORM problem. Therefore, formalizing the problem and providing an evaluation framework seemed the best way to start. To this aim, in the framework of two research projects (Holopedia [74] and LiMoSINe [108]) we actively participated in the organization of evaluation campaigns using a *living lab* approach [19], i.e., we actively collaborated with experts in online reputation management in order to model and solve the real problems they have from a scientific point of view. In this way, we involved both the scientific and user communities during the definition and evaluation of the challenges.

Living Lab

The definition of an evaluation framework for the two scenarios described above allowed us to explore different methods for the two main tasks in Online Reputation Monitoring: filtering and topic detection.

The development of this thesis followed an iterative process, as shown in Figure 1.4.

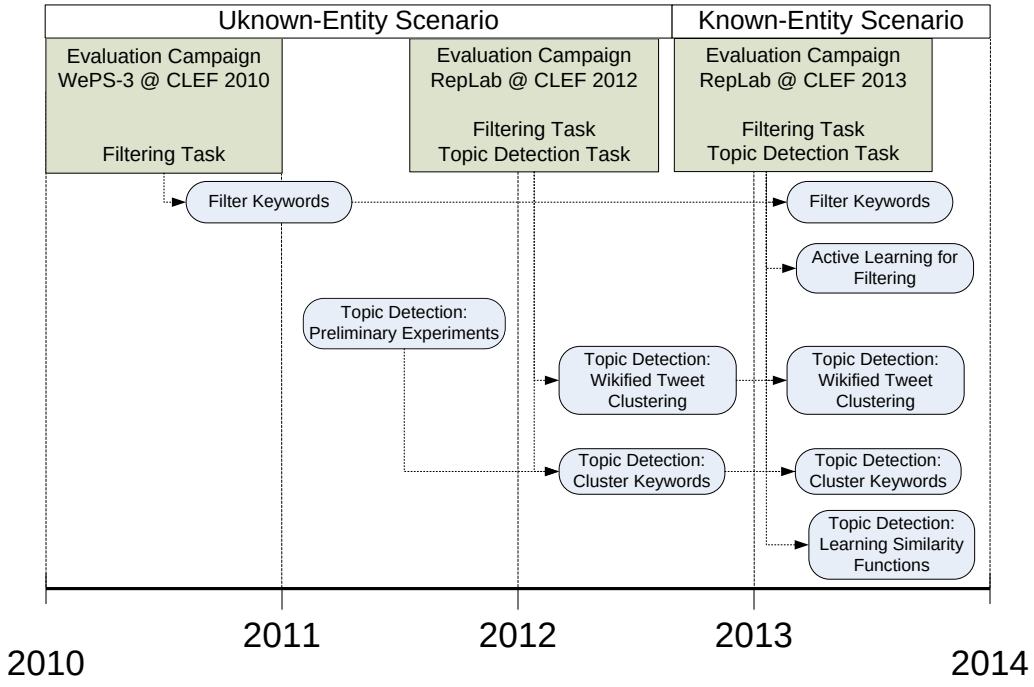


FIGURE 1.4: Iterative process followed in the development of this thesis.

We started by defining the filtering task in the unknown-entity scenario, tackled at the WePS-3 ORM Task at CLEF 2010. The definition of this evaluation framework allowed us to explore our filter keywords approach. Then, we tried to go through the definition of the post-filtering ORM task: topic detection. Note that the evaluation framework for this task was not ready until RepLab 2012. Therefore, we carried our some preliminary experiments in more controlled topic detection scenarios: sub-event detection for summarizing scheduled events in real-time and aspect identification.

In 2012, we had the opportunity to collaborate with our partners Llorente & Cuenca—an online communication firm—and the University of Amsterdam to define the full stack of tasks carried by reputation experts in the Online Reputation Monitoring process: filtering, topic detection, polarity for reputation and topic priority. We ran together the RepLab 2012 evaluation campaign, which comprised for the first time a test collection for tackling these tasks in the unknown-entity scenario. As participants in RepLab 2012, we tested two topic detection approaches: wikified tweet clustering and cluster keywords.

The RepLab 2013 edition included a new dataset for the same tasks. This time, entity-specific training data was provided, enabling the study of the known-entity scenario. As participants, we performed a second iteration of our filtering and topic detection approaches. In collaboration with UvA, we tested a preliminary version of our active learning approach for filtering. Finally, we proposed a topic detection approach which combines the signals used in our previous methods to learn a similarity function for detecting topics.

1.5 Structure of the Thesis

This thesis is organized in 7 chapters. Below we provide a brief overview summarizing the contents of each of these chapters.

Chapter 1 on page 1

Introduction

We provide the motivation of formalizing the ORM problem as information access tasks and we state the scope and the research goals of this thesis.

Chapter 2 on page 13

Background: State of the Art Before 2010

We present the State of the Art before the beginning of this thesis, giving some background about Twitter and covering some information access tasks and techniques that help us to contextualize our work.

Chapter 3 on page 21

State of the Art: Recent Progress

We survey the work related to ours that have been produced in the last years, from the beginning of this thesis. We decided to separate it from the background (Chapter 2) for two reasons. First, most research on Twitter has been published after 2010. Second, our contributions to formalizing the ORM problem and making reusable test collections have had some impact on the state-of-the-art, and therefore, the thesis concurred with the development of ORM in the NLP&IR community.

Chapter 4 on page 33

ORM Framework: Tasks and Datasets

We present the formalization of the ORM challenges as information access tasks and the datasets built for tackling these tasks, including the ones built in the context of the evaluation campaigns (WePS-3, RepLab 2012 and RepLab 2013). Note that most of the work

presented in this chapter was made in collaboration with the organizers of the evaluation campaigns and the partners of the Holopedia [74] and LiMoSINe [108] projects.

Chapter 5 on page 59

Filtering

We propose different approaches for tackling the filtering task in the *unknown-entity* and *known-entity* scenarios. In particular, we study the use of filter keywords and the suitability of the filtering task in an active learning scenario.

Chapter 6 on page 93

Topic Detection

Before tackling the ORM topic detection task, we start by studying the use of simple statistical methods in two more controlled scenarios: real-time summarization of scheduled events and entity aspect identification. Then, we propose three different topic detection systems to study our research goals: (i) wikified tweet clustering that represents tweets by linking them to Wikipedia concepts; (ii) cluster keywords, and extension of the filter keyword intuition for the topic detection task; and (iii) learning similarity functions to combine different Twitter signals for grouping tweets according to topics.

Chapter 7 on page 133

Conclusions and Future Research

We discuss and summarize the main conclusions and contributions of the work. We present the answers to the formulated research questions, the practical outcome of our work for building ORM systems and the outlook on future directions of the work.

Additionally, the thesis contains at the end an appendix with the conclusions in Spanish and three appendices with complementary information about the main generated datasets: WePS-3 ORM, RepLab 2012 and RepLab 2013 datasets.

2

Background: State of the Art Before 2010

In this chapter, we depict the context in which our work started, summarizing related work and the technological background before 2010. This will help us to define our ORM problem according to previous work, as well as identify research issues.

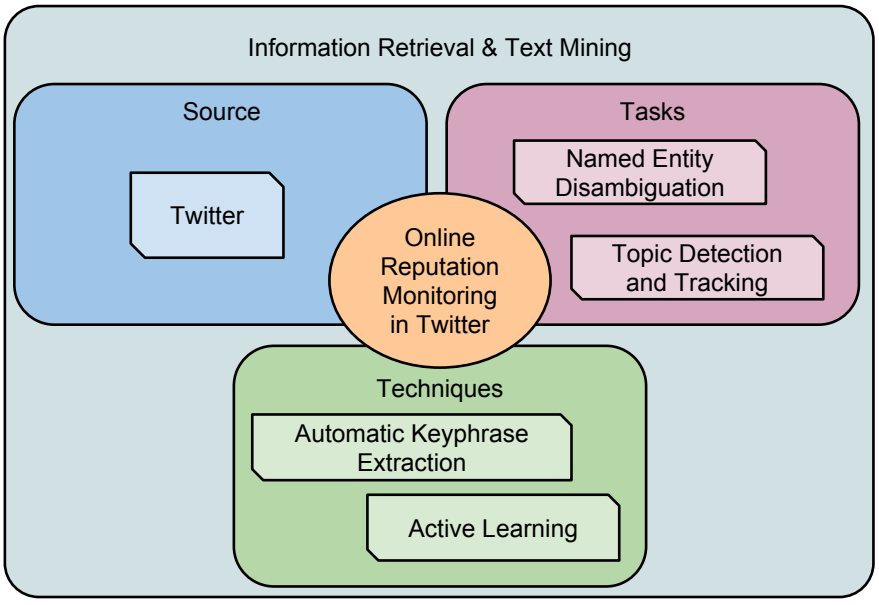


FIGURE 2.1: Graphical representation of the related source, tasks and techniques.

The chapter is organized according to the diagram in Figure 2.1. We start by giving some background about Twitter (§2.1). Then, we introduce previous work done on two tasks that contextualize our work: Named Entity Disambiguation (NED) in Section 2.2 and Topic Detection & Tracking (TDT) in Section 2.3. Then, we describe related work about the information retrieval and text mining techniques that we will apply to our problems, such as Automatic Keyphrase Extraction (§2.4) and Active Learning (§2.5).

2.1 The Data Source: Twitter

Twitter —the most popular microblogging service— is the most important social media source for ORM [80] and, among other things, it is considered the first source for the latest news [98]. Besides, tweets are particularly challenging for text mining [45], given that (i) tweets are short (i.e., 140 characters), and (ii) users post text using a non-standard language with similarities to SMS style [62, 93, 102].

Twitter has become a popular social media service where millions of users contribute on a daily basis and which stands out as the microblogging service par excellence [98]. Launched in July of 2006, Twitter is today one of the top 10 most visited Internet sites. As from February 2014, Twitter has more than 241 million of monthly active users and more than 500 million tweets are published per day.[1]

Two features have been fundamental in its success: the shortness of tweets, which cannot exceed 140 characters, facilitates creation and sharing of messages in a few seconds; and the easiness of spreading those messages to a large number of users in very little time [29, 101, 133].

Throughout time, the community of users on Twitter has established a syntax for interaction, which has been later officially adopted by its developers. Most major Twitter clients have implemented this standard syntax as well. The standards in the interaction syntax include: Twitter Syntax

- **User mentions:** when a user mentions another user in their tweet, an at-sign is placed before the corresponding username, e.g., `You should all follow @username, she is always abreast of breaking news and interesting stuff`.
- **Replies:** when a user wants to direct to another user, or reply to an earlier tweet, they place the `@username` mention at the beginning of the tweet, e.g., `@username I agree with you`.
- **Retweets:** a retweet is considered a re-share of a tweet posted by another user, i.e., a retweet means the user considers that the message in the tweet might be of interest to others. When a user retweets, the new tweet copies the original one in it. Furthermore, the retweet attaches an `RT` and the `@username` of the user who posted the original tweet at the beginning of the retweet. For instance: if the user `@username` posted the tweet `Text of the original tweet`, a retweet on that tweet would look this way: `RT @username: Text of the original tweet`. Moreover, retweets can further be retweeted by others, which creates a retweet of level 2, e.g., `RT @username2: RT @username: Text of the original tweet`. Similarly, retweets can go deeper into 3rd level, 4th, and so forth.
- **Hashtags:** similar to tags on social tagging systems or other social networking systems, hashtags included in a tweet tend to group tweets in conversations or represent the main terms of the tweet, usually referred to topics or common interests of a community. A hashtag is differentiated from the rest of the terms in the tweet in that it has a leading hash, e.g., `#hashtag`.

[1] https://about.twitter.com/company

Up to 2010 —when this Ph.D. work started—, most of the work on Twitter had focused on analyzing the microblogging phenomenon [29, 34, 82, 101] and modeling the propagation of information in the social network [98, 189]. Kwak et al. [101] use a representative sample of 467 million tweets from 20 million users covering a 7 month period from 1st June 2009 to 31th December 2009 to study the information diffusion and the topological characteristics of Twitter. Yang & Leskovec [189] used a dataset of 500 million tweets to model the influence of nodes and temporal dynamics of information diffusion. Cha et al. [34] built a dataset comprising 54,981,152 users, connected to each other by 1,963,263,821 social links and including a total of 1,755,925,520 tweets to analyze users' influence and how to measure it on the Twittersphere.

A number of publications focused on the analysis of the content of tweets [38, 80, 150, 166]. Jansen et al. [80] showed that Twitter can be considered as an electronic word-of-mouth channel, and therefore a crucial source for Online Reputation Monitoring. In their experiments, they automatically categorize opinions and comments about brands and products in microblog posts, in order to determine the overall sentiment about a brand in Twitter and how it evolves over time. A corpus of 900,000 tweets has been provided by the *Content Analysis in Web 2.0* Workshop (CAW 2.0),[2] to tackle the problem of text normalization on user generated contents. Here, the objective is to correct the texts present in the Web 2.0, including Twitter, in order to produce well-written and syntactically correct sentences. The Edinburgh Twitter Corpus [143] consists of a collection of 97 million Twitter posts collected from 11th November 2009 until 1st February 2010.

As we have seen, the first Twitter datasets were mainly used to study the peculiarities of Twitter that make it different from other social networks. However, there were no test collections to study entity-oriented problems such as Online Reputation Monitoring, where specific manual annotations are needed for tackling tasks like filtering or topic detection.

Chapter 3 will cover related work on Twitter during the period of this thesis (2010–2014), which is the time where most research has been done.

2.2 Named Entity Disambiguation

Entities play a crucial role in ORM. First, the monitoring process itself is driven by the entity of interest: the company, organization or brand to be analyzed. Second, the topics and events being discussed in tweets typically involve entities or concepts in a knowledge base (e.g., products, key people, competitors, etc.). From an information access perspective, the problem lies in the fact that named entities are often ambiguous, and identifying the correct entity/concept on which a term, n-gram or document refers to is not trivial.

Word Sense Disambiguation

Ambiguity in texts have been widely studied in Natural Language Processing, specifically the Word Sense Disambiguation (WSD) problem. WSD deals with the polysemy of common words, and consists of assigning the appropriate sense to an occurrence of a word in a given context [2].

[2] http://caw2.barcelonamedia.org/node/41

In WSD, the senses of a word are often looked up in a dictionary, thesaurus or lexical resource such as WordNet [48].

One of the main NLP tasks that involves the identification of entities in texts is the Named Entity Recognition (NER). NER consists of identifying and classifying phrases in a text that refer to people, places or organizations, as well as temporal expressions or numeric quantities [64, 135]. Named Entity Recognition

A step further to WSD and NER is Named Entity Disambiguation (NED). NED involves the association of mentions in one or more texts (also called references or surface forms) of an entity with the concrete object that they are actually referencing [20]. For instance, in the sentence: Named Entity Disambiguation

The Big Apple*'s new* Apple *retail store was officially opened today.*

the two occurrences of the word *Apple* refer to two different entities. The former refers to New York City (nicknamed as "The Big Apple"[3]), while the latter refers to the consumer electronics and software company Apple Inc[4].

In the literature, we found the NED problem tackled as (i) a *linking* problem, i.e., associating a source form (term, n-gram or document) to an entity in a knowledge base (entity linking, §2.2.1) and (ii) a *clustering* problem as in Web People Search (§2.2.2). Here, search results returned by querying a person name have to be grouped in as many sets as entities sharing that name.

2.2.1 Disambiguation as Entity Linking

Entity linking consists of associating a mention in a text with the corresponding entity in a Knowledge Base.[5] Linking texts —like news articles, blogposts or tweets— to entities in a knowledge base —like Wikipedia or Freebase— can be used to provide relevant context that may facilitate the understanding of raw texts to the reader [129], as well as a way to improve the performance of information retrieval and question answering systems [31, 94]. Entity Linking

We are interested in Entity Linking for two main reasons. First, the filtering task for ORM can be seen as a particular case of entity linking where, unlike document enrichment, it is not required to link every mention of entities in the collection, but just to decide whether each mention refers or not to the entity of interest. Second, in the context of the topic detection task it is worth exploring whether linking concepts mentioned in tweets can be used as a signal to improve the process. Here, we will take advantage of the coverage provided by Wikipedia to link not only to entities, but also to concepts that are represented by Wikipedia articles.

In the Knowledge Base Population scenario (KBP) [46, 83, 84, 123], Entity Linking is used as a first step to discover facts about entities and augment a knowledge base with these facts and with newly discovered entities. Given an entity name and a background document where the name occurs, systems typically perform three steps to link the entity to the knowledge base: (i) query Knowledge Base Population

[3] http://en.wikipedia.org/wiki/Big_Apple
[4] http://en.wikipedia.org/wiki/Apple_Inc
[5] An updated bibliography on entity linking can be found at http://ejmeij.github.io/entity-linking-and-retrieval-tutorial/

expansion (enrich the query by mining the Wikipedia structure or resolving co-reference in the background document); (ii) candidate generation (find all possible entries in the knowledge base that the query might link to) and (iii) candidate ranking (rank candidate entities by computing similarity between the represented query and the entities, and fixing a threshold to decide when the entity does not exist in the knowledge base).

There is a variety of systems that automatically annotate a document by linking names appearing in the document with Wikipedia articles [31, 99, 129, 132, 167].

In general, NED in these systems is carried out in three steps:

– **Mention or surface form representation.** In this step, the context of the mention to disambiguate is defined. The most common representations used are vector space model [31, 43, 129], the set of named entities that occur in the text [58], and the resolved Wikipedia links of unambiguous entities next to the mention [51, 99, 126, 132].

– **Candidate entities retrieval and representation.** The system retrieves all possible entities that could be referenced by the mention from the knowledge base (i.e., Wikipedia pages) and represents each entity as a bag-of-words from the page [31, 43, 129], extracting features from the page structure (such as the categories which the page belongs to) [31, 43, 58], or syntactic features [129] or also exploiting the hyperlink structure of Wikipedia, retrieving all pages that link to the candidate entity page [51, 99, 126, 132].

– **Disambiguation.** In the final step, the best candidate entity is selected by computing a distance or similarity function between the surface form and each of the candidate entities. The most common functions are cosine [31] and other vector similarity functions [43, 129], random walk graph models to compute semantic relatedness [58] and finally relatedness or coherence functions that involve all the entity links made in the text [51, 99, 132].

In this thesis we will use Wikipedia as knowledge base for two purposes: first, we will compute term overlap between the tweets and the Wikipedia's page of the entity of interest for automatically discovering filter keywords; second, in the context of the topic detection task, we will *wikify* tweets, by using the hyperlink structure of Wikipedia to retrieve Wikipedia entities which are semantically related to a given tweet [126].

2.2.2 Disambiguation as Clustering: Web People Search

Web People Search

Web People Search is defined as the task of clustering a set of web pages, which are the result of a Web search for a person name, in as many groups as entities sharing that name [12]. This was the original goal of the Web People Search campaign (WePS) [15], complemented in subsequent editions by the task of person attribute extraction [14, 16].

In the web people search scenario, Hierarchical Agglomerative Clustering (HAC) seems to be the most competitive (and most frequently used) clustering technique [14–16, 61]. Documents are typically represented as bag-of-words [15, 20, 89]. Other approaches use smaller portions of the document, such as sentences where the ambiguous name occurs or pre-defined windows of

words [16, 61, 115]. Obviously, this does not apply to microblog posts; on the contrary, we will see that the lack of context is typically a bottleneck in the ORM scenario.

Named entities are also a frequently used feature for people name disambiguation [15, 16, 89], and biographical features (e.g., title, organization, email address, phone number, etc.) are used to a lesser extent [3, 114, 178]. The most common similarity measure in this scenario is cosine [15, 16, 20], while some authors also use Kullback-Leibler Divergence (KLD) [61, 121].

Although Web People Search is a different task, it shares some similarities with the ORM problem: (i) in both scenarios we are looking for named entities that may be ambiguous —people or companies— (ii) in both scenarios Web data can be used to extract relevant features (e.g., extracting keywords from web pages) and (iii) clustering techniques that work well in WePS —such as HAC— may be helpful for the clustering task of topic detection.

2.3 Topic Detection and Tracking

One of the main tasks in the ORM process consists of identifying what are the topics discussed in a given Twitter stream. In this section we present the literature related to the task of Topic Detection and Tracking. We start describing the Topic Detection and Tracking (TDT) NIST's initiative, that was the first evaluation framework built to investigate the state of the art in finding and following new events in a stream of broadcast news stories. We then survey the techniques used in the detection of topics and clustering of weblogs.[6]

2.3.1 The Topic Detection and Tracking (TDT) Initiative

For several years, NIST has been organizing evaluation campaigns to formalize, evaluate and study the problem of Topic Detection and Tracking (TDT) in texts [5, 6, 53]. The scenario was an event-based organization of newswire stories task. Given a news stream, for each piece of news systems must decide whether is about a new topic (*detection*) or belongs to an already seen topic (*tracking*). Here, a *topic* is defined as a set of news stories that are strongly related to a real-world event. The TDT initiative introduced a streaming scenario for detecting popular events (generic topics that cover more than one piece of news).

The ORM scenario on Twitter includes a topic detection and tracking task, with a couple of substantial differences with respect to the NIST TDT track: (i) tweets are very different from news, and (ii) differences between topics, being all related to the same entity, tend to be much more fine-grained.

[6] Event detection in Twitter is a recent research area which is discussed in §3.2.

2.3.2 Topic Detection and Clustering in the Blogosphere

Over the last years, the advent of social media made possible (and necessary) the study of topic detection in the blogosphere [1]. In the problem of detecting salient topics in blogposts, some scientific work has been done looking at word co-occurrence bursts over time as a key signal for detecting topics [60, 65, 107]. On the other hand, it has been shown that, taking advantage of the large text collections extracted from the blogosphere, probabilistic graphical models like Probabilistic Latent Semantic Analysis (PLSA) or Latent Dirichlet Allocation (LDA) perform well [113, 124, 125].

In §3.2 we will see that both alternatives —co-occurrence-based bursts and probabilistic graphical models— have been recently explored for detecting topics or events in Twitter.

2.4 Automatic Keyphrase Extraction

One of our research goals consists of validating the usefulness of using *keywords* for monitoring the reputation of entities in Twitter. A first step that will help us to tackle both the filtering and the topic detection ORM tasks consists of identifying salient terms or keywords in the tweet stream. Therefore, we include here an overview of Automatic Keyphrase Extraction techniques, as this is a technique that plays a crucial role in our approaches.

Automatic Keyphrase Extraction

Automatic Keyphrase Extraction is the task of identifying a set of relevant terms or phrases that summarize and characterize one or more given documents [185]. It is typically used to characterize the content of one or more documents, using features intrinsically associated with that documents. In order to automatically discover filter keywords in the *unknown-entity* ORM scenario, we need to look into external resources in addition to the target Twitter stream. Thus, automatic keyword extraction methods are not directly applicable to our filter keyword approach.

Most of the literature about automatic keyphrase extraction is focused on (well-written) technical documents, such as scientific and medical articles, since the keywords given by the authors can be used as gold standard [56, 95, 130, 131, 173, 185]. Some authors address automatic keyword extraction as a way of automatically summarizating web sites [195–198]. In [195] different keyword extraction methods are compared, including Term Frequency Inverse Document Frequency (TF.IDF) weighting, supervised methods, and heuristics based on both statistical and linguistic features of candidate terms.

Similar to Zhang et al. [195], who extract keywords from website descriptions, in our work we will explore an automatic keyword extraction approach that extracts filter keywords over a tweet stream.

2.5 Active Learning

Twitter streams are highly dynamic and what is being said about an entity always changes over time. It is difficult to ensure that models trained in previous data may remain effective for all the lifetime of the ORM process. Therefore, one of our proposed approaches for solving the ORM filtering task will rely on active learning. Active learning [157] is a subfield of machine learning that has been gaining great interest in the last years. Unlike *passive* supervised learning, where the goal of a learner is to infer a classification model from labeled and static training data, active learning interacts with the user for updating the classifier. This learning framework has been widely used in information access tasks [155, 186] and, in particular, in text categorization [76, 111, 187]. As in passive text categorization, Support Vector Machines (SVM) have proved to be one of the most competitive learning models in active learning [111, 155, 187]. We will analyze the impact in terms of effectiveness of using an SVM-based active learning model with respect to the passive models in the state-of-the-art.

Active Learning

3 State of the Art: Recent Progress

In this chapter, we cover recent progress (i.e., since the beginning of our own research in 2010) on the problems related to Online Reputation Monitoring in Twitter and the filtering and topic detection tasks. We decided to separate it from the background (Chapter 2) for two reasons. First, most of the research on Twitter have been published as of 2010. Second, our contributions to formalize the ORM problem and making reusable test collections have had some impact on the state-of-the-art, and therefore, the thesis concurred with the development of ORM in the NLP&IR community.

We start by describing recent work on Named Entity Disambiguation in Twitter in Section 3.1. In Section 3.2, we discuss related work on Topic Detection and Event Summarization in Twitter. Next, we discuss work derived from the three ORM evaluation campaigns —WePS-3, RepLab 2012 and RepLab 2013— in the context of the filtering and topic detection tasks. Then, other evaluation campaigns that tackle IR and text mining tasks upon Twitter data are presented in Section 3.4.

3.1 Named Entity Disambiguation in Twitter

As we have seen in Section 2.2, NED is one of the information access tasks to consider as essential background for our ORM problem. Most NED techniques have been applied to disambiguate entities in reasonably long texts such as news articles or blog posts. However, little work has been done on NED over microblogging posts. In this scenario, disambiguation is harder due to the fact that texts are short —limited to 140 characters— and hence the context of a mention is minimal.

Most of the research of NED in Twitter tackles the entity linking problem, where the use of Wikipedia as knowledge base is a de facto standard [51, 112, 126, 127]. Here, tweets are linked to Wikipedia pages, which typically represent entities and concepts. Systems rely on the hyperlink structure of Wikipedia, exploiting the links between pages and the anchor texts of the links [51, 112, 126]. To be able to link entities in real-time, they first build an inverted index

of Wikipedia, i.e., with an '`anchor text` → `Wikipedia page`' structure. When the system receives a text, it detects the anchors on the text and retrieves the possible senses for each anchor.

Finally, ambiguity is resolved with different strategies that rely on the global consistence of the linked entities in a text [51, 112] or on machine learning [126]. For instance, Ferragina et al. [51] use a collective agreement function among all senses associated to the anchors detected in the text [99], and, taking advantage of the unambiguous anchors, boost the selection of these senses for the ambiguous anchors [132]. They do not evaluate the accuracy of their disambiguation component over tweets. However, they report that almost 95% of the 5,000 analyzed tweets have at least 3 phrases with an entry in Wikipedia (which is not necessarily an entity). This result shows that Wikipedia has a high coverage as a catalog of senses for tweet disambiguation.

A simple but competitive baseline for disambiguation consists of considering the most probable Wikipedia concept for each n-gram, i.e., the *commonness* probability [126]. We used the commonness probability in our experiments for *wikifying* tweets in the context of the ORM topic detection task.

Commonness Probability

Summing up, Wikipedia seems an effective resource for entity disambiguation and linking in Twitter for three reasons: (i) it has high coverage (ii) can be accurately used by exploiting its hyperlink structure, e.g., matching source forms to anchor texts and (iii) can be indexed for use in real-time scenarios.

3.2 Topic Detection and Event Summarization in Twitter

As we have seen before, Twitter has some peculiarities that make the Topic Detection and Tracking a different problem. Different from other media sources, like news articles or blogposts, topics in Twitter are more related to events, where they drift more often and faster. Moreover, different from previously TDT scenarios, Twitter continuously serves a large amount of streaming data, where most of the tweets are dismissable: a huge number of them are spam or about personal affairs (e.g., `On the go to the office...`).

In this section, we describe recent work on one of the most trending research topic in Twitter. We start by surveying the techniques used to identify trending topics, i.e., new topics that are being discussed by a large volume of users (*dense* scenarios), then we describe topic models in Twitter and we finish by exploring topic tracking and event summarization in *sparse* scenarios.

3.2.1 Trending Topics

One of the main features on the Twitter's homepage shows a list of top terms so-called *trending topics* at all times. These terms reflect the topics that are being discussed most at the very moment on the site's fast-flowing stream of tweets [17]. Twitter defines trending topics as

Trending Topics

"topics that are immediately popular, rather than topics that have been popular for a while or on a daily basis".[1]

In the context of detecting the most popular topics in the Twittersphere, simple statistical approaches that look at term frequency shifts over time have been proven to be effective [24, 25, 122, 180]. For instance, Mathioudakis & Koudas [122] group keywords that suddenly gain an unusually high tweet rate into related groups based on their co-occurrences and Benhardus and Kalita [25] outlines methodologies for using streaming data, e.g., analysis of tf.idf term weighting and normalized term frequency to identify trending topics. Weng et al. [180] detect events by grouping a set of signals (words) with similar patterns of bursts using a modularity-based graph partitioning method. Moreover, bursty patterns in the tweet histogram have been explored to detect specific events such as earthquakes [150] or local events [4].

Besides Twitter, event detection have been also studied in other social media channels. Becker et al. [23] showed the effectiveness of considering meta-data information (tags, time, location, etc.) in combination to textual information to cluster social media documents (such as Flickr images) according to (unknown) real-life events. In our ORM topic detection experiments, we will see that learning similarity functions taking into consideration Twitter information (authors, timestamp, links, hashtags, etc.) significantly outperforms clustering systems based on textual data only.

There are two factors that make our setting different from the dense trending topics scenario. First, topics are part of the *long tail* of Twitter topics, i.e., topics are far in the tweet distribution from the popular and most discussed topics in Twitter. This is due to the fact that our topic detection task is entity-oriented: topics are about a given entity in a short time frame. Moreover, some of the topics are small (tens of tweets) even in comparison with the size of the entity-specific Twitter stream. This causes data sparsity, which makes the topic detection for ORM a challenging problem. Second, coverage in ORM is crucial in order to minimize the damage of overlooking a potential menace. Therefore, all tweets have to be analyzed by the experts or systems and, in particular, assigned to a topic. Unlike our scenario, coverage in trend and event detection in Twitter is usually not as essential as precision.

Data Sparsity

Coverage

3.2.2 Topic Models

Topic Models

Topic models are a family of statistical models that help to summarize large collection of documents by extracting the latent topics within the texts [27]. Recently, topic models like LDA [28] and PLSA [73] have been adapted to the context of Twitter.

The general assumption is that each author has a certain distribution of topics, while each tweet is associated only to one topic [148, 200] —contrary to other larger documents where each document is about multiple topics. Hong & Davison [75] conducted experiments on the quality of topics derived from different aggregation strategies. They concluded that topic models learned from messages posted by the same user may lead to superior performance in classification problems.

[1] http://support.twitter.com/articles/101125-about-trending-topics

The common characteristic of all previous work is that there are contexts where there is enough information and redundancy to work with topic models (i.e., millions of tweets [75, 148, 200]). However, detecting topics of a given entity of interest in a given time span in Twitter has data sparsity, which is a bottle-neck for topic models [134].

3.2.3 Topic Tracking and Event Summarization in Sparse Scenarios

The common characteristic of all previous works is that are contextualized in a *dense* scenario. This means that in most cases, there is enough information and redundancy to extract the main topics just looking for salient terms. However, in the ORM scenario, the user will be interested in a particular entity, that is, only a small subset of the Twitter stream.

Automatic summarization of events from the Twitter's long tail is still in its infancy as a research field. For instance, Hannon et al. [69] present an approach for the automatic generation of video highlights for soccer games after they finish. They set a fixed number of sub-events that want to be included in the highlights, and select that many video fragments with the highest tweeting activity. While some have studied events after they happened, there is very little research dealing with the real-time study of events to provide near-immediate information. Zhao et al. [199] detect sub-events occurred during NFL games, using an approach based on the increase of the tweeting activity. We set this approach as the baseline in one of our preliminary experiments for topic detection: real-time summarization of scheduled events. Afterwards, they apply a specific lexicon provided as input to identify the type of sub-event. Different from this, we focus on event and entity-independent strategies, providing a summarized stream instead of categorizing sub-events or topics. Chakrabarti and Punera [35] were the first to present an approach –which is based on Hidden Markov Models– for constructing real-time summaries of events from tweets. However, their approach requires prior knowledge of similar events, and so it is not easily applicable to previously unseen types of events.

On the other hand, the problem of splitting a given sparse tweet stream in topics has recently studied as the task of microblog summarization [79, 158]. Remarkably, this task is not evaluated in terms of matching between the system and reference topics. Besides, they are evaluated in terms of summarization-oriented measures such as ROUGE.

Summing up, topic detection at sparse Twitter streams, as in the ORM scenario, currently represents a gap in the state-of-the-art.

3.3 Online Reputation Monitoring from an Information Access Perspective

We now describe the state-of-the-art in Online Reputation Monitoring systems, focusing on the two tasks tackled in this thesis: Filtering (§3.3.1) and Topic Detection (§3.3.2). We analyze the different techniques proposed by the participants in the three evaluation campaigns for ORM that

have been run so far: WePS-3 Online Reputation Management Task and the first two editions of RepLab (2012 and 2013).

3.3.1 Filtering

The filtering task consists of classify Twitter posts containing a given company name, depending on whether the post is actually related with the company or not.

We first describe the different approaches proposed to tackle the filtering task in the *unknown-entity* scenario (WePS-3 and RepLab 2012) and then we analyze the RepLab 2013 participation, which tackles the filtering problem in the *known-entity* scenario.

3.3.1.1 WePS-3 Online Reputation Management Task

The Online Reputation Management task of the WePS-3 evaluation [7] was the first of evaluation campaign organized to start establishing an evaluation framework for the ORM problem. The task tackled here was the filtering task, defined to deal with the previously described *unknown-entity* scenario. Here, an online system accepts any company name as input, and has to learn to disambiguate entries about that company without a set of previously disambiguated tweets for the entity of interest. Therefore, the set of organization names in the training corpora is different from the set of companies in the test set.

A total of five research groups participated in the campaign. The best two systems were LSIR [190] and ITC-UT [192]. The LSIR system builds a set of profiles for each company, made of keywords extracted from external resources such as WordNet or the company homepage, as well as a set of manually defined keywords for the company and the most frequent unrelated senses for the company name. These profiles are used to extract tweet-specific features that are added to other generic features that give information about the quality of the profiles to label the tweets as related or unrelated with an SVM classifier.

The ITC-UT system is based on a two-step classification. Firstly, it predicts the class of each query/company name according to the ratio of related tweets of each company name and secondly applies a different heuristic for each class, basically based on the PoS tagging and the named entity label of the company name.

The SINAI system [57] also uses a set of heuristic rules based on the occurrence of named entities both on the tweets and on external resources like Wikipedia, DBPedia and the company homepage. The UvA system [171] does not employ any resource related to the company, but uses features that involve the use of the language in the collection of tweets (URLs, hashtags, capital characters, punctuation, etc.). Finally, the KALMAR system [90] builds an initial model based on the terms extracted from the homepage to label a seed of tweets and then uses them in a bootstrapping process, computing the point-wise mutual information between the word and the target's label.

3.3.1.2 Other Work using WePS-3 ORM Dataset

Recently, Yerva et al. [191] have explored the impact of extending the company profiles presented in [190] by using the related ratio and considering new tweets retrieved from the Twitter stream. By estimating the degree of ambiguity per entity from a subset of 50 tweets per entity, they reach 0.73 accuracy.[2] Using this related ratio and considering co-occurrences with a given company profile, the original profile is extended with new terms extracted from tweets retrieved by querying the company name in Twitter. The expanded profile outperforms the original, achieving 0.81 accuracy.

Zhang et al. [194] presents a two-stage disambiguation system. They combine supervised and semi-supervised methods. Firstly, they train a generic classifier using the training set, similarly to [190]. Then, they use this classifier to annotate a seed of tweets in the test set, using the Label Propagation algorithm to annotate the remainder tweets in the test set. Using Naïve Bayes and Label Propagation they achieve a 0.75 accuracy, that matches the performance of the best automatic system in WePS-3.

3.3.1.3 Filtering at RepLab 2012

Similar to Yerva et al. [190], most of the approaches make term or concept profiles for the entity of interest either manually (Daedalus [174], OXYGEN [91]) or automatically by using external resources. The most used external resource is Wikipedia, used by CIRGDISCO [193], ILPS [142], BMedia [39] and UNED [33]. Freebase, DBPedia and Linked Open Data have been also used to extract related and unrelated concepts [39, 63]. Particularly, UIOWA group [188] propose the use of Google Adwords Keyword Tool to make a profile of keywords related to the entity of interest. Term or concepts profiles are typically used to compute features that are combined with entity features (similar to the features computed by ITC-UT [192]) and tweet generic features (like the ones computed by UvA [171]) to feed a machine learning algorithm such as SVM (BMedia) or Naive Bayes (OXYGEN).

Similar to Zhang et al.[194], a common strategy —that we will also explore in our filter keyword approach— relies on first label a high-precision seed of tweets and then propagate the labels to the remainder tweets (CIRGDISCO, Gavagai [92]).

Accuracy and Reliability&Sensitivity have been used as evaluation metrics. Different from WePS-3, there is a large bias to the related class, i.e., most of the tweets refer to the entity of interest. Therefore, almost all systems have relatively high accuracy scores (above 0.7).

3.3.1.4 RepLab 2013 Filtering Subtask

Different from WePS-3 and RepLab 2012, for the first time, RepLab 2013 tackles the filtering problem in the known-entity scenario. It provides a dataset with entity-specific training data,that

[2] Note that, unlike the original formulation of the WePS-3 task, this is a supervised system, as it uses part of the test set for training. Hence, their results cannot be directly compared with the results in our work.

can be used to simulate the daily work of reputation analysts. Here, most of the systems exploit the entity-specific training data to build a binary classifier. The system that obtained the best results at RepLab 2013, POPSTAR [151], is based on supervised learning, where tweets are represented with a variety of features to capture the relatedness of each tweet to the entity. Features are extracted both from the collection (Twitter metadata, textual features, keyword similarity, etc.) and from external resources such as the entity's homepage, Freebase and Wikipedia. The best run from the SZTE_NLP group [67] applies Twitter-specific text normalization methods as a preprocessing step and combines textual features with Latent Dirichlet Allocation to extract features representing topic distributions over tweets. These features are used to train a maximum entropy classifier for the filtering task. The best run submitted by LIA [42] is based on a k-nearest-neighbor classifier over term features. Similar to the official RepLab 2013 baseline, which labels each tweet in the test set with the same label as the closest tweet in the training set, LIA matches each tweet in the test set with the n most similar tweets. The best run submitted by UAMCLyR [154] uses a linear kernel SVM model with tweets represented as Bag-of-Words.

Most of the evidences revealed in previous work on the ORM filtering task will be validated in our proposed approaches. First, similar to [190], we will explore the use of keywords for filtering, analyzing its suitability in both unknown-entity and known-entity scenarios. Second, we will see that, as the results obtained by Zhang et al.[194], filter keywords can be used to label a high-precision seed of tweets that are propagated to the remainder tweets in a subsequent step. In our propagation step, we will study the impact of the ratio of related tweets of each entity name, which is one of the key signals used by Yoshida et al. [192]. Finally, we will see that a Bag-of-Words classifier similar to the one proposed by UAMCLyR [154] can be effectively used in an active learning setting for tackling the filtering task in the known-entity scenario —where we exploit the entity-specific training data to build a binary classifier.

3.3.2 Topic Detection

Besides filtering, the second task tackled in this thesis is topic detection. Here, systems have to group the tweets related to the entity by topics.

3.3.2.1 RepLab 2012 Topic Detection Subtask

RepLab 2012 was the first evaluation campaign that The first evaluation campaign that In RepLab 2012, three systems —besides ours— have been evaluated for the Topic Detection task. Balahur and Tanev [21] customize a clustering method that had initially employed in the case of news to deal with news reported in tweets. They use multilingual lists of keywords, extracted by Europe Media Monitor that are used as features for clustering. Each tweet is represented as a vector that contains the Europe Media Monitor keywords present in the tweet. Log likelihood ratio is used as weighting function, considering probability of appearance of the keyword in a large news corpus of 100,000,000 words. They finally use Hierarchical Agglomerative Clustering (HAC) to cluster the tweet's vectors. CIRGDISCO [147] uses POS to identify terms likely to be concepts (nouns or adjectives) and uses text similarity measures to group tweets into clusters. Martín et

al. [119, 120] tested Twitter-LDA —an extension of the LDA probabilistic graphical model that considers the author of the tweet to generate the topics— for detecting topics that are related to the entity of interest.

Results showed the difficulty of this clustering pilot task: a simple agglomerative clustering over word similarity performs similarly or better to more complex systems. In our experiments, we will see that this is a barrier that is difficult to overcome.

3.3.2.2 RepLab 2013 Topic Detection Subtask

RepLab 2013 intended to evaluate the Topic Detection task in the *known-entity* scenario, where some entity-specific training data is provided. Participation in RepLab 2013 included both supervised and unsupervised techniques. On one hand, different clustering techniques such as HAC, VOS clustering [26] —a community detection algorithm— and K-star [154] were used by the participants. The most common similarity functions are cosine [26] and Jaccard similarity [154] over terms. In our experiments, we will explore the use of supervised learned similarities over Twitter signals in two ways: defining similarities between concurrent terms— in order to detect keywords associated to clusters— and learning similarity functions between tweets.

On the other hand, LIA [42] and UAMCLYR [154] tackled the Topic Detection task as a multi-class classification problem. LIA [42] used Maximum A Posteriori probability of the most pure headwords of the topics in the training set to assign the tweets in the test set. UAMCLYR [154] used standard multi-class classification techniques, such as Naive Bayes and Sequential Minimal Optimization Support Vector Machines (SMO SVM).

Overall, the results of official RepLab systems were the first set of experiments on the RepLab 2013 dataset. Here, a HAC algorithm over term similarity outperforms all the RepLab systems: this result —which will be further analyzed in our topic detection experiments— is another evidence that corroborates the issue of data sparsity in our Online Reputation Monitoring problem.

Besides the RepLab Topic Detection Task, Chen et al. [36] have recently studied the problem of discovering hot topics about an organization in Twitter. The problem tackled here is slightly different to our scenario: instead of clustering all the tweets related to an entity of interest, they are only interested in detecting the *hot emerging* topics from an initial clustering generated by cosine similarity. Their ground truth doesn't include clustering relationships between tweets. Instead of this, they align topics with online news and they manually evaluate the aggregated output of different hot topic detection methods to create the ground truth deciding whether a topic is emerging or not.

3.4 Other Evaluation Campaigns on Twitter

It is worth mentioning that other evaluation campaigns have been organized to tackle complementary tasks to ORM in Twitter data: the most salient are the TREC Microblog Track (§3.4.1),

the SemEval-2013 Task on Sentiment Analysis in Twitter(§3.4.2) and the INEX Tweet Contextualization Track (§3.4.3).

3.4.1 The TREC Microblog Track

The last editions of the TExtual Evaluation Conference (TREC) held a track to evaluate real-time search systems in Twitter: the TREC Microblog Track.[3] The real-time ad-hoc search task consist of retrieving the most recent but relevant tweets according to a given query at a certain time.

The Tweets2011 corpus was the dataset used on the first two editions (2011 and 2012). The corpus is a representative sample of the twittersphere, that includes also spam tweets. It consists of approximately 16 million tweets over a period of 2 weeks (24th January 2011 until 8th February, inclusive), which covers both the time period of the Egyptian revolution and the US Superbowl, among others. The last edition of the TREC Microblog Track was designed as a *track-as-a-service*: participants interacts with the tweet collection stored remotely via a search API, allowing to scale up the size of the collection. The collection consists of approximately 240 million tweets, collected via the Twitter streaming API over a two-month period —from 1st February 2013 to 31th March 2013.

Note that this real-time ad-hoc search task is different from our ORM problem. While in the ad-hoc task the user wishes to see relevant and recent content about a topic, in the ORM problem the user is interested in retrieving *all* related mentions to a given entity of interest. Therefore, results from TREC are not applicable (at least directly) to our scenario.

3.4.2 SemEval-2013 Task 2: Sentiment Analysis in Twitter

The last Semantic Evaluation Exercise (SemEval-2013) held a task to tackle the problem of sentiment analysis in Twitter [183]. There were two classification subtasks: contextual polarity disambiguation and message polarity classification. In the contextual polarity disambiguation subtask, systems have to determine whether, given a marked instance of a word or a phrase in a tweet, it is positive, negative or neutral. In the message polarity subtask, systems have to determine the polarity of sentiment at tweet level. The dataset comprises both tweets and SMS messages. It includes around 17k annotated phrases for the first subtask, and 11k tweets and 2k SMS for the second subtask. The manual annotation of tweets was crowdsourced using Mechanical Turk, where tweets where pre-filtered using SentiWordNet: they keep only tweets with at least one word with positive or negative sentiment score greater than 0.3 in SentiWordNet for at least one sense of the words.

[3] https://sites.google.com/site/microblogtrack/

3.4.3 INEX Tweet Contextualization Track

Running since 2010, the INEX Tweet Contextualization Track tackles the problem of providing some context about the subject of a given tweet automatically. This challenging task involves the use of information retrieval, text mining, multi-document summarization and entity linking techniques. For each tweet, systems have to generate a readable summary of at most 500 words, made of passages from a provided Wikipedia dump.

Systems typically combine passage retrieval, sentence segmentation and scoring, NER, and POS tagging. Anaphora detection and diversity content measure as well as sentence reordering can also help. The resulting summaries are evaluated according to readability and informativeness, which is measured by comparing the proposed summary to a text reference. Results in both evaluation criterion show that the task is challenging, leaving a large room for improvement: for instance, the best informativeness scores obtained so far are between 10% and 14%. We believe that advances in this tweet contextualization track —which is complementary to our ORM problem— may be used to improve filtering and topic detection systems in a future.

3.5 Wrap Up

In this chapter, we have seen that the Online Reputation Monitoring problem has not been previously formalized from a scientific perspective. In the next chapter of this thesis we fill this gap in the state-of-the-art by formalizing the ORM problem and making reusable test collections which facilitated the development and evaluation of ORM systems. We have discussed the approaches developed by the NLP&IR community to tackle the two ORM tasks on which this thesis is focused: filtering and topic detection.

Since the filtering task can be seen as an Named Entity Disambiguation problem, we surveyed recent work on NED in Twitter. Moreover, in our experiments we will use an NED approach that link tweets to Wikipedia concepts for wikifying tweets in the context of the ORM topic detection task. The lessons learned from systems developed upon the built collections will be used to define our filtering approaches. We will explore the use of keywords for filtering, analyzing its suitability in both unknown-entity and known-entity scenarios. Here, we will use a propagation step to automatically label the tweets that are not covered by filter keywords, studying the impact of the ratio of related tweets of each entity name. Finally, we will see that a Bag-of-Words classifier can be effectively used in an active learning setting for tackling the filtering task in the known-entity scenario.

On the other hand, we have seen that little work has been done in sparse topic detection scenarios. Contrary to the typical topic detection scenario (e.g., Twitter trending topics) —where there is enough information and redundancy to extract the main topics by simply looking for salient terms—, in the ORM scenario, the user will be interested in a particular entity, that is, only a little subset of the Twitter stream. Results obtained by the systems proposed so far show the difficulty of the task. Remarkably, we will see that a simple agglomerative clustering over term

similarity outperforms all RepLab systems, corroborating the issue of data sparsity in our Online Reputation Monitoring problem.

Finally, we have summarized others evaluation contests focused on Twitter. We believe that, being complementary to ORM evaluation campaigns, the progress in these other tracks may help to better process Twitter data and hence, to build better ORM systems in the future.

4

ORM Problem Framework: Tasks and Datasets

The work presented in this chapter has been done in collaboration with Vanessa Álvarez, Javier Artiles, Jorge Carrillo de Albornoz, Irina Chugur, Adolfo Corujo, Bing Liu, Tamara Martín, Edgar Meij, Ana Pitart and Maarten de Rijke.

Online reputation sources offer new opportunities for information technologies to improve and facilitate the reputation monitoring process. Over the last years, a wide variety of tools have been developed that facilitate the Online Reputation Monitoring (ORM) task. Despite of the variety in the market, current tools do not seem to perfectly match user needs and the quality of their language processing machinery is not up to the tasks. More importantly, there is no standard evaluation framework —a defined set of evaluation measures and reusable test collections to tackle this problem— that allows tools benchmarking. In this chapter, we tackle the first research question of this thesis:

> **RQ1:** *Which challenges when monitoring the reputation of an entity in Twitter can be formally modeled as information access tasks? Is it possible to make reusable test beds to investigate the tasks?*

ORM Evaluation Campaigns

We formalize the ORM problem from a scientific perspective, in order to state the main research challenges and provide a standard evaluation framework that will allow us —and, in general, the NLP & IR research communities— to explore novel solutions to the detected challenges. To do so, we organized different international evaluation campaigns —WePS-3 [7], RepLab 2012 [9] and RepLab 2013 [8]— that cover the defined ORM tasks in the *unknown-entity* and *known-entity* scenarios. We covered the different tasks and the scenarios in three yearly iterations and in an incremental way, starting from the *unknown-entity* scenario and finishing with the *known-entity* scenario. The evaluation campaigns have been organized in the context of the projects Holopedia [74] and the EU project LiMoSINe [108]. In particular, the author of this thesis has been collaborating on (i) the definition and formalization of the tasks, participating in several interviews and discussions with reputational experts; (ii) the creation of the different datasets, which includes the definition of the schema, the development of annotation tools and

the management of the annotation process; and (iii) the assistance to participants by providing technical support during the participation process.

The chapter is organized as follows. In Section 4.1 we formally define the filtering and topic detection tasks. For each task, we provide a notation and describe the official evaluation measures, which will also be used throughout the thesis. Then, we describe other ORM tasks included in the organized evaluation campaigns. After defining the ORM tasks, we describe the reusable test collections built for the evaluation campaigns to tackle these tasks, as well as other datasets used in our experiments (§4.2).

4.1 Tasks

We start by describing the daily work of an analyst when monitoring the reputation of an entity in Twitter (§4.1.1). Then, we define the filtering task (§4.1.2) and the topic detection task (§4.1.3), which are the tasks that we tackle in this thesis. Finally, we describe other ORM tasks, which were most of them defined in the context of the organized evaluation campaigns (§4.1.4).

4.1.1 Analyst's Workflow

Annotation Process

In order to gather the requirements from the ORM analysts, we have defined a set of annotation guidelines created in collaboration with two reputation experts from a leading multinational Public Relations consultancy firm, Llorente&Cuenca[1]. The annotation process was designed to mimic the real workflow of reputation analysts (see Figure 4.1).

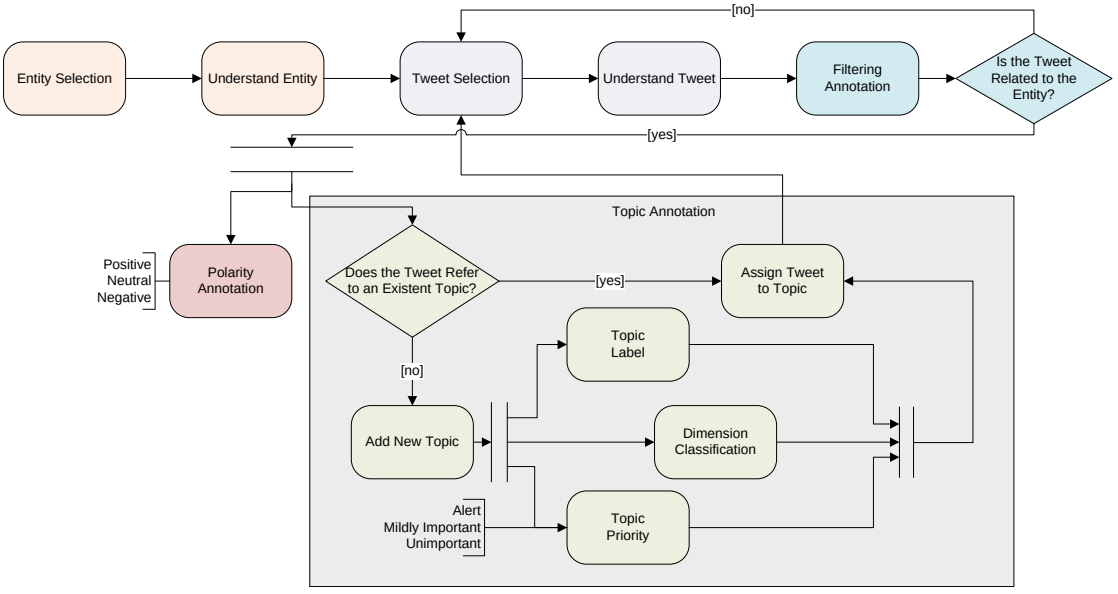


FIGURE 4.1: Workflow of the ORM annotation process.

[1] http://www.llorenteycuenca.com/index.php?&idioma=en

The monitoring process of a reputation analyst in Twitter consists of finding, analyzing and annotating tweets that mention the entity of interest (i.e., the customer). Tweets are usually ordered by creation time. The expert selects a tweet and tries to understand its meaning. The tweet understanding process involves: (i) reading the text, (ii) reading the author name, (iii) going to the URLs included in the tweet, if any, (iv) visiting the author's profile (v) identifying keywords or hashtags to use them as queries in the Twitter stream and also to facilitate the annotation of further tweets. When the expert has a clear understanding of what the tweet is about, the annotation process starts. The first decision the expert has to make is whether the tweet refers to the entity of interest or not (e.g., to distinguish between tweets that contain the word `Stanford` referring to the University of Stanford and filtering out tweets about Stanford as a place). If the tweet is related to the entity, it is annotated with polarity, topic, dimension and topic priority; otherwise, the tweet is discarded.

For polarity, the expert has to decide if the tweet affects the entity's reputation in a positive or negative way. Note that reputational polarity differs form traditional sentiment analysis [110, 139]: first, when analyzing reputational polarity both facts and opinions must be considered (i.e., the goal is to find what implications a piece of information might have on the reputation of a given entity); second, negative sentiments do not always imply negative polarity for reputation and vice versa [9].

For instance, `I LIKE IT.....NEXT...MITT ROMNEY...Man sentenced for hiding millions in Swiss bank account`, has a positive sentiment (joy about a sentence) but has a negative implication for the reputation of the entity Mitt Romney.

The topic annotation process is meant to identify the issues or conversations around the entity, grouping tweets with the same subject together. Topic descriptions are created freely by the analyst. New tweets similar to ones that have been already analyzed are assigned to an existing topic. For a new tweet dealing with a completely new subject, the expert defines a new topic. Topics discussed about an entity in Twitter are typically about a specific dimension of the entity business (e.g., topics are about products, employee satisfaction, economic affairs, etc.). Dimensions can represent different departments of the entity. Therefore, analysts often classify topics according to a predefined taxonomy of dimensions in order to better organize the information in their monitoring reports —for instance, to give different advices to each department. Finally, identified topics are assigned a priority value, according to the implications that each topic might have to the entity's reputation (e.g.,the topic is an alert, is mildly important or unimportant from a reputational perspective). The annotation of the current tweet finishes at this point.

We now formally define the tasks covered in this monitoring process. We pay special attention to the two tasks tackled in this thesis: filtering (§4.1.2) and topic detection (§4.1.3) and then we describe other ORM tasks (§4.1.4), such as topic priority, polarity for reputation, dimension classification, author profiling for ORM and aspect and opinion target identification.

4.1.2 Filtering

As we have seen before, the ORM process is recall-oriented, i.e., the expert has to look at every possible mention of the entity of interest. Taking this into account and giving that entity names are often ambiguous, discarding non relevant tweets can be a tedious and time consuming task. From an information access perspective, this filtering task can be seen as a binary classification problem (Is a given tweet related/unrelated to the entity of interest?).

Given an entity of interest e and the set of tweets D_q in a time span that contain the entity name/query q, the *filtering* task consists in defining a function

$$f(d) : D_q \rightarrow \{\text{related}, \text{unrelated}\} \tag{4.1}$$

Table 4.1 summarizes the notation that we use for defining the task.

TABLE 4.1: Notation used for defining the tasks.

Item	Description
q	query or (ambiguous) name that identifies an entity (e.g., `jaguar`)
e	entity of interest (e.g., Jaguar Cars)
D_q	set of tweets in the collection for a given entity query q.
D_e	set of tweets related to the entity of interest e.
d, d_i	tweets in D_q.

Filtering systems are those that implement a function $f_S(d)$ that approximates the optimal solution represented by a gold standard $f_G(d)$:

$$f_G(d) : \begin{cases} \text{related} & \text{if } d \in D_e \\ \text{unrelated} & \text{in other case} \end{cases} \tag{4.2}$$

where

$$D_e = \{d_i \in D_q | d_i \text{ is related to the entity of interest } e\} \tag{4.3}$$

We can build a confusion matrix for evaluating the filtering task:

TABLE 4.2: Confusion matrix for evaluating filtering systems.

		$\mathbf{f_G}$	
		related	**unrelated**
$\mathbf{f_S}$	**related**	TP	FP
	unrelated	FN	TN

where

$$TP = |\{d \in D_q | f_S(d) = f_G(d) \wedge f_G(d) = \text{related}\}|$$

$$FP = |\{d \in D_q | f_S(d) \neq f_G(d) \wedge f_G(d) = \text{related}\}|$$

$$TN = |\{d \in D_q | f_S(d) = f_G(d) \wedge f_G(d) = \text{unrelated}\}|$$

$$FN = |\{d \in D_q | f_S(d) \neq f_G(d) \wedge f_G(d) = \text{unrelated}\}|$$

Unless otherwise stated, we will use Accuracy and the metric pair Reliability&Sensitivity to evaluate the filtering task.

Accuracy

Accuracy is the most common metric for evaluating classification problems [184]. Intuitively, it measures the ratio of correctly classified instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

In scenarios where classes are highly unbalanced (such as our filtering task), it is not trivial to understand the effectiveness of a system by measuring accuracy: a system that simply assigns all the instances to the majority class may have a 0.9 accuracy if 90% of the instances do belong to that class.

Reliability & Sensitivity

Metrics such as Reliability & Sensitivity (R&S), are more appropriate than accuracy for measuring how informative a filtering system is. In our evaluations, we will use R&S to complement accuracy. Reliability & Sensitivity [11] have been recently proposed as two complementary measures to evaluate document organization tasks involving classification, clustering and ranking. The harmonic mean of R&S tends to zero when the system has a "majority class" behavior, and a high score according to $F_1(R, S)$ ensures a high score for most of the popular evaluation metrics in filtering tasks.

When evaluating a binary classification task, Reliability corresponds to the product of the precision of the classes, and Sensitivity to the product of the recall of both classes:

$$R = \frac{TP}{TP + FP} \cdot \frac{TN}{TN + FN} \qquad S = \frac{TP}{TP + FN} \cdot \frac{TN}{TN + FP}$$

and

$$F_1(R, S) = 2 \cdot \frac{R \cdot S}{R + S}$$

Macro-averaged Scores

Note that, these metrics are typically used for evaluate each entity (test case) separately, i.e., one confusion matrix per entity, to then average the results to get the final score of a given system. That is, we use macro-averaged accuracy and macro-averaged $F_1(R, S)$ scores. Averaged scores are useful for ranking systems, but may hide significant differences on how systems behave across test cases, especially when the classes are skewed and not following a normal distribution. Complementary to these metrics, we propose a visualization technique for representing filtering system results in skewed scenarios.

4.1.2.1 A *Fingerprint* Visualization of Filtering Systems

We will see that the ratio of related tweets does not follow a normal distribution in our filtering datasets, which are a representative sample of the problem in Twitter. In general, for a given entity name and a (small) period of time, either most tweets refer to the company, or most tweets are unrelated. We will see that, even in the WePS-3 dataset, where the organizers made an effort to include company names covering all the spectra of positive ratio values, the ratio is very low or very high for many companies.

In this context, a *fingerprint* representation technique to visualize system results is a useful tool to understand systems' behavior —complementary to average performance measures (accuracy or F_1). Figure 4.2 illustrates our method, which consists of displaying the accuracy of the system (vertical axis) for each company vs. the ratio of related (positive) tweets for the company (horizontal axis). Each dot in the graph represents one of the test cases (i.e., the accuracy of the system for the set of tweets containing the name of one of the companies in the dataset).

The advantage of this representation is that the three basic baselines (all related, all unrelated and random) are displayed as three fixed lines, independently of the dataset. The performance of the "all related" baseline classification corresponds exactly with the proportion of true cases, and therefore is the $y = x$ diagonal in the graph. The "all unrelated", correspondingly, is represented by the $y = 1 - x$ diagonal. And, finally, the random baseline is the horizontal line $y = 0.5$. Fixed Baselines

Note that, for averaged measures —such as accuracy or F_1—, the results of the baselines depend on the dataset: for instance, the "all related" baseline depends on the average number of true cases in the corpus used for evaluation. As our fingerprint representation is constant for that baselines, it is easier to identify when a system is having a baseline-like approach, and for which subset of the data.

4.1.3 Topic Detection

After filtering out unrelated tweets, tweets about the entity of interest have to be grouped in topics. That is, given an entity (e.g., *Yamaha*) and a set of tweets relevant to the entity in a certain time span, the task consists of identifying tweet clusters, where each cluster represents a topic/event/issue/conversation being discussed in the tweets, as it would be identified by reputation management experts. Therefore, we define this topic detection task as a *clustering* problem with no overlap (i.e., each tweet is assigned to one and only one topic/cluster).

Note that this is not a standard Topic Detection setting, because in our scenario each of the tweets must be assigned to a topic. From the perspective of reputation management, reputation alerts —issues that may affect the reputation of the client— must be detected early, preferably before they explode, and therefore the number of tweets involved may be small at the time of detection. That makes the task harder than standard topic detection, mainly due to sparsity issues: topics about a given entity in a short time frame are part of the "long tail" of Twitter topics, and some of them are small even in comparison with the size of the entity-specific Twitter stream.

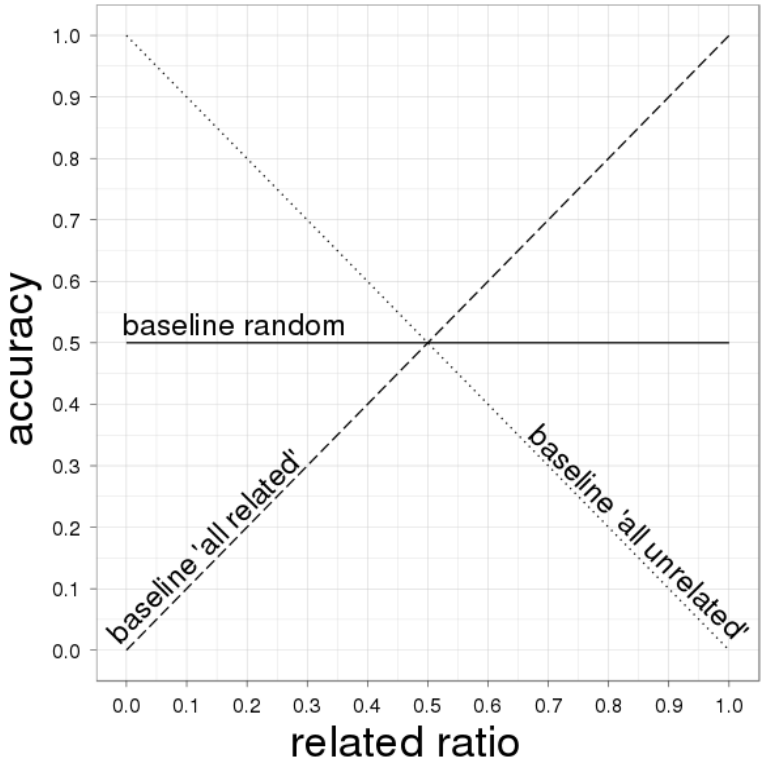


FIGURE 4.2: Fingerprint technique for visualizing results of filtering systems.

Table 4.3 illustrates some examples of tweets belonging to the same topics, extracted from the RepLab 2013 dataset (described in detail in Section 4.2.3) and corresponding to entities *Maroon 5*, *Yamaha*, *Ferrari*, *Bank of America* and *Coldplay*.

TABLE 4.3: Examples of annotated tweets in the RepLab 2013 training dataset.

Entity	Id	Tweet	Topic
Maroon 5	d_1	maroon 5 quedará excitado con las mexicanas (? jajaja.	Promotion of Concerts
	d_2	Oigan ! Creo vendrá a México Maroon 5 quien sabe bien? Quiero ir *n*	
Yamaha	d_3	Just saw Valentino Rossi going into Yamaha's hospitality!! Don't get too excited though, he just attending Agostini's 70th birthday do	MotoGP - User Comments
	d_4	Big piece of 2013 puzzle solved then with Jorge Lorenzo signing a new 2-year deal with Yamaha	
Ferrari	d_5	Alonso pierde la carrera por la mala estrategia de Ferrari, adicional al gran trabajo de Hamilton	(F1) Strategies in the Race
	d_6	Siempre igual Alonso hace el maximo, lo da todo, pero es que las estrategias de Ferrari... son para morirse...	
	d_7	@alo_oficial:"A ver si podemos confirmar la mejoría del coche, es una buena prueba para Ferrari" #A3F1Canada	(F1) GP of Montreal
	d_8	@alo_oficial Qué crack. La que organizas. En Canadá vince la Ferrari. xD	
	d_9	#F1 Fernando Alonso says Montreal will be 'crucial indicator' for Ferrari's title bid.	
	d_{10}	Vídeo - La Scuderia Ferrari (@InsideFerrari) y Martin Brundle (@MBrundleF1) nos traen el previo del GP de Canadá: URL #F1	
Bank of America	d_{11}	Cons Prod Strategy Manager at Bank of America (Jacksonville, FL) SAME_URL	Vacancy
	d_{12}	Part Time 20 Hours Bartram Lake Village at Bank of America (Jacksonville, FL) SAME_URL	
	d_{13}	Irony: Bank of America is directly across the street from the Dept of the Treasury. Must make it easy to get those bailouts!	Criticism of BofA Bad Behavior
	d_{14}	In 2010 Bank of America seized three properties that were not under their ownership, 'apparently' due to incorrect addresses.	
Coldplay	d_{15}	and so to mourn the loss of may, a trip to see coldplay is in order. i hope they play that uplifting number the scientist.	Fans go to Concert
	d_{16}	Can't get over how fast this day has come !! @coldplay @USER1 @USER2 @USER3	

Following Wagner&Wagner notation [177], let D be a finite set of tweets with cardinality $|D| = n$. A clustering $\mathcal{C}$ is a set $\{C_1, \ldots, C_k\}$ of non-empty disjoint subsets of D such that their union equals D. The set of all clusterings of D is denoted by $\mathcal{P}(D)$. For a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ we assume $|C_i| > 0$ for all $i = 1, \ldots, k$. We will use $d \sim_{\mathcal{C}} d'$ to denote that documents d and d' belong to the same cluster in a clustering $\mathcal{C}$.

Being D_e the tweets related to the entity of interest e, let $\mathcal{G} = \{C_1^{\mathcal{G}}, \ldots, C_l^{\mathcal{G}}\} \in \mathcal{P}(D_e)$ denote the topic detection gold standard. Here, $d \sim_{\mathcal{G}} d'$ denotes a clustering relationship between d and d' in the clustering gold standard $\mathcal{G}$. Note that the topic detection gold standard is defined for $d \in D_e$, and not for the complete tweet stream D_q. Therefore, in a full monitoring scenario, the topic detection depends on the filtering task, as we are interested on detecting the topics that are discussed in the related tweets only.

For evaluation, we will use Reliability&Sensitivity, which in the non overlapped clustering scenario are equivalent to Bcubed Precision and Recall [11]. They correspond to the average of the precision and recall of clustering relationships $d \sim_{\mathcal{C}} d'$ for each document $d \in \mathcal{D}_e$, respectively:

$$R_{\mathcal{C},\mathcal{G}} \equiv Avg_d \left(\frac{|d \sim_{\mathcal{G}} d' \wedge d \sim_{\mathcal{C}} d'|}{|d \sim_{\mathcal{C}} d'|} \right)$$

$$S_{\mathcal{C},\mathcal{G}} \equiv Avg_d \left(\frac{|d \sim_{\mathcal{G}} d' \wedge d \sim_{\mathcal{C}} d'|}{|d \sim_{\mathcal{G}} d'|} \right)$$

where

$$d \sim_{\mathcal{C}} d' \Leftrightarrow \exists C_i \in \mathcal{C}. d \in C_i \wedge d' \in C_i$$

4.1.4 Other ORM Tasks

Here we describe other tasks involved in the Online Reputation Monitoring process, such as Topic Priority, RepTrak Dimension Classification, Polarity for Reputation and Author Profiling.

4.1.4.1 Topic Priority

Not all the topics discussed in the given tweet stream have the same consequences in the entity's reputation: topics that are more likely to damage its reputation (alerts) must be handled first. The Topic Priority task is defined as a *ranking* problem according to its potential influence in the entity's reputation. In the annotation process, it is encoded as grades of priority: alerts, medium and low priority.

Following Krehler's formalism [97], let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a clustering output representing the detected topics in D_e:

$$\text{priority} : \mathcal{C} \rightarrow \{1, \ldots, N\}$$

where $N \leq k$, i.e., two clusters may have the same priority.

Systems have to optimize a priority function that approximates a $\text{priority}_{\mathcal{G}}$ represented in the gold standard:

$$\text{priority}_{\mathcal{G}} : \mathcal{G} :\rightarrow \{\text{alert}, \text{medium}, \text{low}\}$$

Note that priority values in the gold standard are annotated in a graded scale. Intuitively, the output given by an optimal priority system should correlates with the gold standard priority values, which are ordered by alert, medium and low. Since $N \leq k$, standard IR evaluation metrics used to evaluate ranking systems with graded relevance —such as Normalized Discounted Cumulative Gain (NDCG) [81]— are not applicable in our scenario.

The topic priority task can be evaluated by looking at the precision and recall of the priority relationships captured by the systems, using Reliability&Sensitivity metrics [8].

4.1.4.2 Polarity for Reputation

Experts have to keep tracking of the polarity pulse in the tweets that refer to the entity of interest. Here, the expert has to decide if the tweet affects the entity's reputation in a positive or negative way. Note that reputational polarity differs form traditional sentiment analysis [110, 139]: first, when analyzing reputational polarity both facts and opinions must be considered (i.e., the goal is to find what implications a piece of information might have on the reputation of a given entity); second, negative sentiments do not always imply negative polarity for reputation and vice versa. For instance, the tweet "`I LIKE IT.....NEXT...MITT ROMNEY...Man sentenced for hiding millions in Swiss bank account`", has a positive sentiment (joy about a sentence) but has a negative implication for the reputation of the politician Mitt Romney.

The polarity for reputation task can be seen as three-class classification task. Given a tweet, systems have to decide whether the tweet is positive, negative or neutral for the reputation of the entity of interest. Similar to the filtering task, a 3×3 confusion matrix can be built, where accuracy is the most common metric for evaluation.

4.1.4.3 RepTrak Dimension Classification

Topics discussed about an entity in Twitter are typically about a specific dimension (e.g., topics are about products, employee satisfaction, economic affairs, etc.). Most existing models of reputation measure perceptions of an organization in terms of a taxonomy of dimensions or attributes of the corporate business. The Reputation Institute, for instance, defines the RepTrak Framework[2], that comprises a set of seven dimensions [145] that reflect the feelings and intentions

[2] http://www.reputationinstitute.com/about-reputation-institute/the-reptrak-framework

of the different stakeholder groups: *products and services*, *innovation*, *workplace*, *governance*, *citizenship*, *leadership*, and *performance*. Table 4.4 shows the definition of these dimensions, supported by an example of labeled tweet. This classification is based on that of Fombrum [54], and describes reputation in terms of the different stakeholders' expectations (customers, employees, investors and general public).

TABLE 4.4: RepTrak dimensions. Definition and examples of tweets

Dimension	Definition and Example
Products & Services	It is to do the products/services offered by the company and the consumers' satisfaction. `BMW To Launch M3 and M5 In Matte Colors: Red, Blue, White but no black...`
Innovation	Its innovativeness, nurturing good and novel ideas and translating them into products. `Listening to Ian Sayers speak on Barclays Pingit #mobilemoney app, built in 90 days in-house by 12 people, now with over 1M users`
Workplace	It is to do with employees satisfaction and the company's ability to attract, develop and keep talented people. `HSBC cuts 30,000 jobs, sees more to come: HSBC has reduced its number of staff by almost 30,000 in the last two...`
Governance	The quality of the government management. `Watch Stanford's & Columbia's Ed Deans talk about 'No Child Left Behind' & schemes like Teach First my GSEForum`
Citizenship	Its acknowledgement of community and environmental responsibility. `Tyler serves pizza slices in New York: New York, Nov 21 (IANS) Aerosmith frontman Steven Tyler left a smile on ...`
Leadership	Integrity, transparency and accountability. `Tips on how to apply for funding for your franchise http://ow.ly/fo0KL #HSBC`
Performance	Its long term business success and financial soundness. `#Yahoo #Trend Cancer Center: Torrington's Charlotte Hungerford May Join With Yale http://dlvr.it/2M58v1 #IFollowAll Qo`

Classifying mentions of a company on social web streams into these dimensions is a very important and time-consuming task in the daily work of ORM experts in communication consulting firms.

The Dimension Classification Task can be modeled as a *multi-class classification* problem: given a tweet in D_e, classify it into one of the seven dimensions defined by RepTrak (Table 4.4). This

is one of the novel tasks held in RepLab 2014.[3]

4.1.4.4 Author Profiling

A complementary task in Online Reputation Monitoring consists of taking a snapshot of the most influencers Twitter profiles in a domain (e.g., banking, automotive). Author Profiling includes two subtasks: Author Categorization and Author Ranking.

4.1.4.4.1 Author Categorization. Given a Twitter profile, systems have to classify it in one of the following categories: *journalist*, *professional*, *authority*, *activist*, *investor*, *company* or *celebrity*.

4.1.4.4.2 Author Ranking. Given a domain (such as banking or automotive), systems are expected to find out which authors have more reputational influence (who the influencers or opinion makers are) and which profiles are less influential or have no influence at all. Therefore, this is a *ranking* task, on which profiles are ranked according to their probability of being an influencer.

Some aspects that determine the influence of an author in Twitter—from a reputation analysis perspective—can be the number of followers, the number of comments on a domain or the type of author. As an example, below is the profile description of an influential financial journalist:

> **Description:** New York Times Columnist & CNBC Squawk Box (@SquawkCNBC) Co-Anchor. Author, Too Big To Fail. Founder, @DealBook. Proud father. RTs $\neq$ endorsements
>
> **Location:** New York, New York · nytimes.com/dealbook
>
> **Tweets:** 1,423
>
> **Tweet examples:** Whitney Tilson: Evaluating the Dearth of Female Hedge Fund Managers http://nyti.ms/1gpClRq @dealbook
> Dina Powell, Goldman's Charitable Foundation Chief to Lead the Firm's Urban Investment Group http://nyti.ms/1fpdTxn @dealbook

These two author profiling subtasks are also part of the RepLab 2014 evaluation campaign.

4.1.4.5 Entity Aspect and Opinion Target Identification

Complementary to the topic detection task, knowing what is being said about an entity in Twitter can be also seen as identifying the specific *aspects* that people discuss. Aspects refer to "hot" topics that are talked about in the context of an entity and are of particular interest for companies. Aspects can cover a wide range of issues and include (but are not limited to) company

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

products, services, key people, advertisements, potential competitors and events. They are typically nouns, but can also be verbs, and (rarely) adjectives. They can change over time as public attention shifts from some aspects to others. For instance, for a company releasing its quarterly earnings report, its earnings can become a topic of discussion for a certain period of time and, hence, an aspect.

The entity aspect identification task can be modeled as a *ranking* problem. Given a stream of microblog posts related to an entity, we are interested in a ranked list of aspects that are being discussed with respect to the entity. We formulate our scenario as an information retrieval (IR) task, where the goal is to provide a ranking of terms, extracted from tweets that are relevant to the company.

Brand Identity vs. Brand Image

Although Twitter is a communication channel for conversations between consumers, it is often harnessed by companies and organizations to engage with customers and to promote their products and services. For instance, companies typically run social marketing campaigns to improve the *brand identity*, i.e., how the entity wants to be perceived by consumers and competitors. Here, identifying aspects are a powerful tool to capture the brand identity in Twitter and may be used to validate the effectiveness of social marketing campaigns. Sometimes, however, reputation experts are interested on taking a snapshot of the *brand image*, which is what consumers actually think. To this aim, identifying the target of the opinions expressed by users—which is a fine-grained sentiment analysis task [85]—can be more convenient. Here, the opinion target identification task consists of detecting opinionated tweets and, if so, which part of the tweet is (i) subjective and (ii) what the target of the sentiment is, if any.

4.2 Datasets

In this section we describe the datasets built for most of the ORM tasks defined above. We start by describing the datasets used in three ORM evaluation campaigns organized so far: WePS-3 ORM Task (§4.2.1), RepLab 2012 (§4.2.2) and RepLab 2013 (§4.2.3). Finally, we present a corpus we built for the tasks of aspect and opinion target identification (§4.2.4).

4.2.1 WePS-3 ORM Dataset

The Online Reputation Management task of the WePS-3 evaluation campaign [7] tackles the ORM filtering task in the *unknown-entity* scenario for organizations such as companies, music bands or soccer clubs.

As we have seen before, the filtering task consists of classifying Twitter posts containing a given entity name, depending on whether the post is actually related with the company or not. In the *unknown-entity* scenario, an online system accepts any entity name as input, and has to learn to disambiguate entries about that entity without entity-specific training data (i.e., without a set of previously disambiguated tweets for the entity of interest). Therefore, the set of entities in the training corpora is different from the set of entities in the test set.

For each entity in the dataset, systems are provided with the entity name c (e.g., `apple`) used as query to retrieve the stream of tweets to annotate T_c, and a representative URL (e.g., `http://www.apple.com`) that univocally identifies the target entity. The input information per tweet consists of a tuple containing: the tweet identifier, the organization name, the query used to retrieve the tweet, the author identifier, the date and the tweet content. Systems must label each tweet as *related* (i.e., the tweet refers to the entity) or *unrelated* (the tweet does not refer to the given entity).

The WePS-3 ORM task dataset comprises 52 training and 47 test cases, each of them including an entity name, its URL, and an average of 436 tweets manually annotated as related/unrelated to the entity. Figure 4.5 shows the total number of entities and tweets in the training and test dataset.

The dataset has been annotated using the Mechanical Turk crowdsourcing marketplace[4], with the following manual annotation options: *related*, *unrelated* and *undecidable*. Each hit has been redundantly annotated by five Mechanical Turk workers. A total of 902 annotators have participated in the annotations of 43,740 tweets. Finally, an agreement analysis was done in order to decide the final annotation for each tweet.

An interesting property of the dataset is that there is a great variability of the degree of ambiguity across the training and test cases. That is, there are entities with low occurrence in tweets (e.g., `Delta` Holdings, `Zoo` Entertainment), entities with medium ambiguity (e.g., `Luxor` Hotel and Casino, `Edmunds`.com) and entities with high presence in tweets (e.g., `Yamaha`, `Lufthansa`).

TABLE 4.5: WePS-3 ORM Dataset.

	Training	**Test**
Entities	51	47
# Tweets	23,740	19,402
# Related Tweets	9,687	8,092
# Unrelated Tweets	14,053	11,310

The WePS-3 ORM dataset is publicly available at `http://nlp.uned.es/weps`.

4.2.2 RepLab 2012 Dataset

The RepLab 2012 dataset [9] included for the first time manual annotations for post-filtering ORM tasks: filtering, topic detection, topic priority and polarity for reputation. Officially, filtering and polarity for reputation are considered as a subtask of a *profiling* task, while topic detection and priority are included in a *monitoring* task. In this context, *entity profiling* refers to the analysis of the reputation of the company in online media at a certain point in time (e.g., what is the overall polarity in the last months?). In this thesis, we focus on Online Reputation

Entity Profiling

[4] `https://www.mturk.com`

Monitoring, which consists of a dynamic process of searching and analyzing every mention of the entity of interest.[5]

The RepLab 2012 Dataset comprises Twitter data in English and Spanish. The balance between both languages depends on the availability of data for each of the entities included in the dataset.

Similar to WePS-3, the RepLab 2012 dataset tackles the ORM tasks in the *unknown-entity* scenario, i.e., entities in the trial dataset are different from entities in the test set. Different from WePS-3 annotations (that were annotated via crowdsourcing), RepLab assessments are provided by ORM experts from the Public Relations consultancy *Llorente & Cuenca*. Table 4.6 shows the total number of entities and tweets manually annotated for the trial and test dataset.

TABLE 4.6: RepLab 2012 Dataset.

	Trial	**Test**
Entities	6	31
# Labeled Tweets	1,800	6,782
# Labeled Tweets EN	1,385	3,114
# Labeled Tweets ES	415	3,668
# Background Tweets	276,941	1,366,243

The information associated to the entities includes: the complete name of the entity, the query used to retrieve the tweets, as well as the homepage and the Spanish and English Wikipedia pages associated to it.

Trial Dataset

Trial data consists of at least 30,000 tweets crawled per each entity name, for six entities (*Apple, Lufthansa, Alcatel, Armani, Marriott, Barclays*) using the entity name as query, in English and Spanish. The time span and the proportion between English and Spanish tweets depends on the entity.

For each entity's timeline, 300 tweets (approximately in the middle of the timeline) have been manually annotated by reputation management experts. This is the *labeled* dataset. The rest (around 15,000 unannotated tweets before and after the annotated set, for each entity), is the *background* dataset. Tweets in the background set have not been annotated.

Test Dataset

Test data are identical to trial data , for a different set of 31 entities (Telefónica, BBVA, Repsol, Indra, Endesa, BME, Bankia, Iberdrola, "Banco Santander", Mediaset, IAG, Inditex, Mapfre, Caixabank, "Gas Natural", Yahoo, Bing, Google, ING, "Bank of America", Blackberry, BMW, BP, Chevrolet, Ferrari, Fiat, VW, Wilkinson, Gillette, Nivea, Microsoft). The tweets have been crawled using the entity identifier as query. There are between 19,400 and 50,000 tweets per entity name, in English and Spanish. Similarly to the trial dataset, the time span, and the proportion between English and Spanish tweets here depend on the entity.

Analogously to the trial data, a set of tweets from the middle of each entity's timeline, has been extracted to be annotated by reputation management experts.

[5]For the sake of clarity, we consider hereafter the four RepLab 2012 subtasks as ORM tasks, without distinguishing between profiling and monitoring tasks.

The tweets corresponding to each entity are annotated with the following ORM tasks described in §4.1: filtering, polarity for reputation, topic detection and topic priority. Table 4.7 shows the distribution of annotations in the RepLab 2012 trial and test datasets.

TABLE 4.7: RepLab 2012 dataset distribution in classes.

Subtasks	Labels	Trial	Test	All
Filtering	Related	1,640	4,746	6,386
	Unrelated	160	2,036	2,196
Polarity for Reputation	Positive	899	1,739	2,638
	Neutral	656	1,718	2,374
	Negative	85	1,289	1,374
Topic Detection	# Topics	198	627	825
	Average # Tweets per Topic	8	10	9
Priority Assignment	# Alert Topics	28	93	121
	# Mildly Important Topics	77	302	379
	# Unimportant Topics	93	255	348

The RepLab 2012 dataset is available for research purposes via the RepLab 2012 organizers.[6]

4.2.3 RepLab 2013 Dataset

RepLab 2013 intended to evaluate the Topic Detection task in the *known-entity* scenario, where some entity-specific training data is provided. In order to ensure representativeness, we selected 61 entities from four domains that illustrate different ways of building reputation: (i) based on the products (*automotive*); (ii) largely dependent on transparency and ethical side of the entity's activity (*banking*); (iii) based on a very broad and intangible set of products (*universities*) and (iv) depending almost equally on the products and personal qualities of the members (*music*). The RepLab 2013 corpus comprises English and Spanish tweets. The balance between both languages depends on the availability of data for each entity.

TABLE 4.8: RepLab 2013 Dataset.

	All	Automotive	Banking	Universities	Music
Entities	61	20	11	10	20
# Training Tweets	45,679	15,123	7,774	6,960	15,822
# Test Tweets	96,848	31,785	16,621	14,944	33,498
# Total Labeled Tweets	142,527	46,908	24,395	21,904	49,320
# Background Tweets	1,038,003	250,961	68,127	120,117	598,798
# Tweets EN	113,544	38,614	16,305	20,342	38,283
# Tweets ES	28,983	8,294	8,090	1,562	11,037

Table 4.8 shows the number of tweets by domain, giving an idea of the size of the training and test sets, providing the distribution per language. Crawling was performed between the 1st June

[6] http://www.limosine-project.eu/events/replab2012#Organizers

of 2012 and the 31th December of 2012 using one canonical query per entity. For each entity, at least $2,200$ tweets were collected: the first 700 were reserved for the training set and the last $1,500$ for the test collection. The distribution was set in this way to obtain a temporal separation (ideally of several months) between the training and test data. Apart from the training and test datasets, the corpus also contains additional background tweets for each entity (up to $50,000$) posted between the training and the test set time frames. As in RepLab 2012, the background corpus has not been annotated.

Figure 4.3 shows the actual distribution of tweets in the corpus over time.[7]

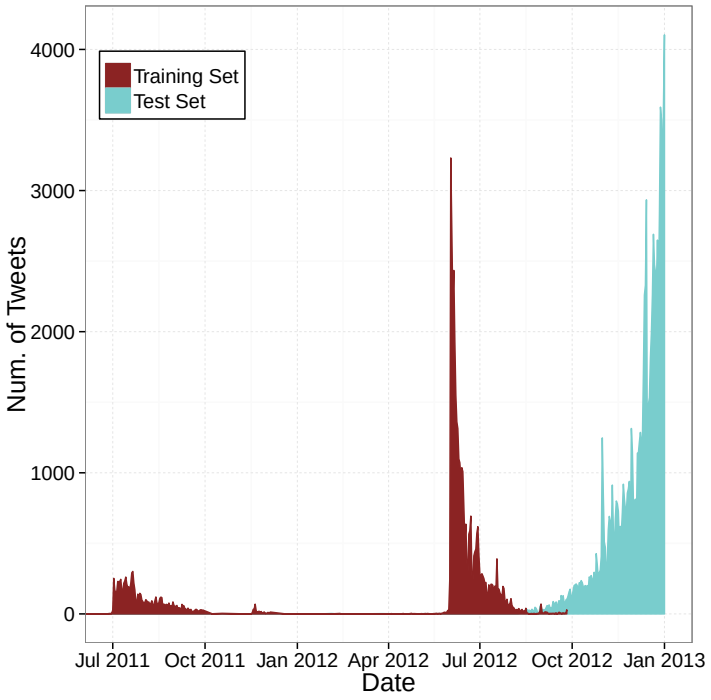


FIGURE 4.3: Distribution of tweets over time in the RepLab 2013 training and test sets.

In the collection each tweet is annotated as follows:

- `RELATED/UNRELATED`: the tweet is/is not about the entity
- `POSITIVE/NEUTRAL/NEGATIVE`: the information contained in the tweet has positive, neutral or negative implications for the entity's reputation.
- Label of the topic cluster the tweet has been assigned to.
- `ALERT/MILDLY_IMPORTANT/UNIMPORTANT`: the priority of the topic cluster the tweet belongs to.

Annotation Process

Once the data was collected, we hired 13 annotators to label the tweets with reputational information. The 13 annotators hold, at least, graduate studies in the areas of marketing, journalism

[7]While the majority of the tweets belong to the period considered in the crawling, some of them were originally published before June 2012 and were later retweeted. Due to the impossibility of retrieving the date of the retweet publication, we had to use the original tweet posting date

and computer science, and all had a high level of English language understanding. The annotation process was supervised by two methods: first, by a distribution list where all the annotators present their doubts to the reputational experts that were daily answered; and second, by weekly meetings between the annotators and the experts, where a collection of the most commons doubts was explained by the reputational experts in order to homogenize the annotation process. All the tweets concerning the same entity were assigned to the same annotator. Annotators were assisted with an annotation tool built specifically to this aim, which was recently extended to use in a semi-automatic environment [32].

Table 4.9 shows the statistics for the annotated subtasks. For the filtering subtask, the collection contains $110,352$ *related* tweets, out of which $34,882$ belong to the training and $75,470$to the test set. The $32,175$ *unrelated* tweets of the dataset are distributed as follows: $10,797$ tweets are in the training and $21,378$ in the test set.

TABLE 4.9: RepLab 2013 dataset distribution in classes.

Subtasks	**Labels**	**Training**	**Test**	**All**
Filtering	Related	34,882	75,470	110,352
	Unrelated	10,797	21,378	32,175
Polarity for Reputation	Positive	19,718	43,724	63,442
	Neutral	9,753	20,740	30,493
	Negative	5,409	11,006	16,415
Topic Detection	# Topics	3,813	5,757	9,570
	Average # Tweets per Topic	14.40	21.14	17.77
Priority Assignment	# Alert Topics	134	170	304
	# Mildly Important Topics	2,212	3,448	5,660
	# Unimportant Topics	1,412	2,082	3,494

With regards to the *polarity* subtask, the RepLab 2013 collection contains $63,442$ tweets classified as *positive*, $30,493$ labeled as *neutral* and $16,415$ marked as *negative*. For the topic detection subtask, Table 4.9 displays the number of topics per set as well as the average number of tweets per topic, which is 17.77 for the whole collection, varying from 14.40 in the training set to 21.14 in the test set. Finally, the distribution of topics in *priority* classes are summarized. The less populated class is *alert*, with 304 topics classified as a possible reputation alert in the whole corpus. *Mildly_Important* has $5,660$ topics, while *Unimportant* received $3,494$ topics.

Inter-Annotator Agreement

To compute inter-annotator agreement , 14 entities (4 automotive, 3 banking, 3 universities, 4 music) were labeled by two annotators. This subset contains $31,381$ tweets which represent 22% of the RepLab 2013 dataset. Table 4.10 shows inter-annotator agreement in terms of percentage of agreement and Kappa coefficients —both Cohen's and Fleiss'— for filtering, polarity and priority detection subtasks, and F_1 measure of Reliability and Sensitivity [11] for the topic detection subtask. When applied to the evaluation of clustering tasks, Reliability and Sensitivity correspond to the standard BCubed Precision and BCubed Recall, respectively. The filtering subtask obtains the highest level of agreement, both in terms of percentage of agreement and in terms of kappa, that corresponds to "substantial agreement" according to [104].

The percentage of agreement for polarity for reputation is lower than in similar studies for sentiment analysis[8], which is probably an indication of the complexity of polarity classification in ORM. The agreement for the topic detection subtask is, on the other hand, higher than expected, given the complexity and subjectivity of this subtask. Finally, the relatively low agreement in priority assignment highlights the difficulty of this knowledge-intensive subtask. Note that the kappa relative agreement is lower than for the others subtasks, due to the skewness of the priority levels, e.g., 304 alerts vs. $5,660$ mildly important topics.

TABLE 4.10: RepLab 2013 agreement: analysis of 14 entities labeled by two annotators.

Subtask	**Set**	**% Agreement**	**Cohen's Kappa**	**Fleiss' Kappa**	$\mathbf{F_1(R,S)}$
Filtering	Training	94.80	0.70	0.69	-
	Test	96.46	0.68	0.68	-
	Total	95.94	0.67	0.66	-
Polarity for Reputation	Training	68.27	0.41	0.39	-
	Test	68.81	0.42	0.40	-
	Total	68.59	0.42	0.40	-
Topic Detection	Training	-	-	-	0.50
	Test	-	-	-	0.48
	Total	-	-	-	-
Priority Assignment	Training	58.41	0.24	0.16	-
	Test	60.88	0.29	0.21	-
	Total	60.07	0.28	0.20	-

The RepLab 2013 dataset is publicly available at http://nlp.uned.es/replab2013. To the best of our knowledge there is no similar dataset in the state-of-the-art in terms of volume of data and high quality manual annotations for Online Reputation Monitoring.

4.2.4 A Corpus for Entity Aspect and Opinion Target Identification in Twitter

The work presented in this section has been done in collaboration with Edgar Meij, Andrei Oghina, Minh Thuong Bui, Mathias Breuss and Maarten de Rijke.

In this section we present a manually annotated corpus suitable to evaluate the task of identifying either aspects or opinion targets in the context of ORM on microblog streams. Both annotations are based on the dataset used in WePS-3 ORM task and available online.[9] The first aspect identification annotations were created using a pooling methodology. Here, we have implemented various methods for automatically extracting aspects from tweets that are relevant for an entity. We subsequently generate a ranked list of aspects using each method, take the highest ranked aspects, and pool them. Then, human assessors consider each aspect and determine whether it is relevant in the context of the entity or not.

[8] http://www.informationweek.com/software/business-intelligence/expert-analysis-is-sentiment-analysis-an/224200667

[9] http://nlp.uned.es/~damiano/datasets/entityProfiling_ORM_Twitter.html

The opinion target annotations methodology is similar, but more fine-grained. Here, annotators consider individual tweets related to an entity and manually identify whether the tweet is opinionated and, if so, which part of the tweet is (i) subjective and (ii) what the target of the sentiment is, if any.

4.2.4.1 Annotating Aspects

Let us consider the aspect identification task described above: given a stream of tweets that are related to a company, we are interested in a ranked list of aspects (products, services, key people, events, etc.) representing the hot topics that are being discussed with respect to the company. We formulated this task as an information retrieval task, where the goal of a system implementing a solution to this task is to provide a ranking of terms, extracted from tweets that are relevant with respect to the company.[10] We have implemented various methods addressing this task. For each company, each method returns a ranked list of terms associated with each company. The underlying principle for all methods is a comparison of the contents of the relevant tweets —henceforward, the *foreground* corpus— with a common *background* corpus, e.g., the whole WePS-3 collection. Using this comparison we identify and score terms based on their relative occurrence. Our methods include TF.IDF [153], the log-likelihood ratio [47] and parsimonious language models [71]. Since aspects can be opinion targets, we also applied an opinion-oriented method [86] that extracts potential targets of opinions to generate a topic-specific sentiment lexicon. We use the targets selected during the second step of this method.[11]

This dataset is then created using a pooling methodology [70]: the 10 highest ranking terms from each method are merged and randomized. Then, human assessors consider each term and determine whether it is relevant in the context of the company or not.

Annotations

The annotators were presented with an annotation interface, where they could select one of the companies from a list. Once a company is selected, the interface shows a randomized list of aspects. The interface also facilitated looking up a term; when clicked, the system would present all tweets that are relevant to the company and contain that particular term. The annotators could indicate one of the following labels for each aspect:

- **Relevant:** A relevant aspect can include, e.g., product names, key people, events, etc. Relevant aspects are in general nouns, but can also be verbs, and (rarely) adjectives. Relevant aspects can include terms from compound words, mentions or hashtags. Aspects should provide some insight into the hot topics discussed regarding a company, topics that would also differentiate it from other more general discussions, or its competitors.
- **Not relevant:** Common words and words not representing aspects or sub-topics are not relevant.
- **Competitor:** A term is (part of) a competitor name, including an opponent team name, a competing company or a product from a competing company.

[10] In our current setup, we only consider unigrams as aspects. When a unigram is an obvious constituent of a larger, relevant aspect, it is considered relevant.

[11] Further details about the methods and their effectiveness on the task of aspect identification are presented in §6.1.2.1.

- **Unknown:** If, even after inspecting the tweets where the term occurs, the judge still cannot use the other labels.

In this work we treat the label *Competitor* as being *Relevant*, although the data set contains this explicit label for possible follow-up work. Table 4.11 shows some examples of the aspects annotated in the corpus.

TABLE 4.11: Examples of aspects annotated for some of the entities in the corpus.

Entity	Aspects
A.C. Milan	milanello, ac, football, milan, galliani, berlusconi, brocchi, leonardo
Apple Inc.	ipad, iphone, prototype, apple, store, gizmodo, employee, gb
Sony	advertising, set, headphones, digital, pro, music, sony, xperia, dsc, x10, bravia, camera, vegas, battery, ericsson, playstation
Starbucks	coffee, latte, tea, frappuccino, starbucks, shift, pilot, barista, drink, mocha

Analysis In order to determine the level of agreement between the three annotators J_i, we calculate *Cohen's kappa* and *Fleiss' kappa* [105] and compare the annotators both pairwise and overall. The results are given in table 4.12. All of the obtained kappa values are above 0.6, which indicates a substantial agreement.

TABLE 4.12: Inter-annotator agreement for the aspects dataset.

Method	J_1–J_2	J_1–J_3	J_2–J_3	All
Cohen's κ	0.691	0.62	0.676	-
Fleiss' κ	0.69	0.62	0.676	0.662

In the WePS-3 ORM dataset, the number of tweets relevant to each company is highly variable [7]. Thus, one could expect correlations between the ratio of relevant tweets and the ratio of relevant aspects annotated for each company.

TABLE 4.13: Distribution of relevant aspects, binned by the number of relevant tweets per company.

Tweets	$\|C\|$	AvgTw	AvgTerms	AvgRel	Rel%
$0-10$	19	4.05	12.47	2.79	22.36%
$11-50$	15	22.20	22.00	8.53	38.79%
$51-150$	12	97.67	26.75	13.58	50.78%
$151-300$	25	219.40	28.80	16.40	56.94%
$301+$	28	381.43	30.64	19.46	63.52%

Table 4.13 shows the number of tweets, the number of extracted terms (*AvgTerms*), and the number of identified relevant aspects (*AvgRel*) based on the annotations. For this, we consider

all terms included in the pooling, and divide the entities in five groups, based on the number of tweets available for each company ($0-10$, $11-50$, $51-150$, $151-300$, $301+$). For each group C, we count how many companies are part of the group ($|C|$) and the average number of tweets for these entities (*AvgTw*). We also compute the percentage of the aspects that are relevant (*Rel%*).

We observe that the percentage of relevant aspects increases with the amount of data available. For companies that have at most 10 tweets, only 22.36% of extracted aspects are annotated as being relevant. On the other hand, for entities with more than 300 tweets, 63.52% of all extracted aspects were annotated as being relevant. This suggests that the amount of data available plays an important role in the performance of the methods used for the pooling.

4.2.4.2 Annotating Opinion Targets

The second dataset we present consists of the tweets of 59 entities from the WePS-3 dataset, manually annotated at the phrase-level. Here, we aim to identify opinion targets in tweets, related to an aspect of a company. We define an opinion target as a phrase p that satisfies the following properties: (i) p is an aspect of the entity, (ii) p is included in a sentence that contains a direct subjective phrase (i.e., an expression that explicitly manifests subjectivity or an opinion) and (iii) p is the target of the expressed opinion.

Annotation Guidelines

With the help of an annotation tool specially developed to the purpose, the annotators were asked to indicate the following.

- **Subjectivity:** Tweet-level annotation that indicates whether the tweet contains an explicit opinionated expression.
- **Subjective phrase:** If the tweet is opinionated, identify the phrase that express subjectivity. In our annotation schema, we only considered direct private states [181].
- **Opinion target:** If the tweet contains opinionated phrases, identify the target of the opinion expressed in that phrases.

Table 4.14 show some examples of opinionated tweets.

Phrase-level annotation require much more effort than tweet-level annotations or aspect assessments. In order to maximize the number of annotated entities, 59 entities were randomly distributed over seven different annotators, making a disjoint assignment of annotators to data. Moreover, besides simple string matching, there is no established statistical method for measuring inter-annotator agreement on this type of data.

In total, $9,396$ tweets were annotated. Only $1,427$ (15.16%) tweets contain subjective phrases and $1,308$ (13.82%) contain opinion targets. There are 119 tweets where the annotators identified subjective phrases but not opinion targets. Most of them are tweets containing either emoticons or phrases expressing subjectivity at tweet-level (e.g., `LOL`, `Yay!`, `#fail`).

Analogous to the first dataset, we divided the annotated entities in groups based on the number of annotated tweets and computed the average of tweets with subjective phrases (*AvgSubj*) and

TABLE 4.14: Examples of phrase-level annotated tweets, having subjective phrases (italic) and opinion targets (boldface).

Entity	**Tweet**
Linux	Lxer: A Slimline Debian Install: It's Easier Than You Might Think: There are some *superb* **desktop Linux distributions**... http://bit.ly/8ZSaF
MTV	@MTV has the *best* **shows** ever. i watch it all day every day (:
Oracle	IMHO, the *best part of* Oracle now **owning Java** is that whenever **Java** is *criticized* for something, Oracle's name is attached.
Sony	@user Welll Im not getting one then. **Sony** is *expensive*
Starbucks	**The Dark Cherry Mocha** from @Starbucks is just *the best Mocha ever!*

opinion targets (*AvgOT*). Table 4.15 reports these averages as well as the averaged percentage of subjective tweets (*Subj%*).

TABLE 4.15: Distribution of subjective phrases and opinion targets, binned by the number of relevant tweets per company.

Tweets	**|C|**	**AvgTw**	**AvgSubj**	**AvgOT**	**Subj%**
$0-10$	7	3.57	0.85	0.85	35.11%
$11-50$	11	23.36	3.64	3.09	14.24%
$51-150$	9	96.22	11.77	10.33	11.88%
$151-300$	19	218.68	25.21	23.10	14.22%
$301+$	13	392.54	61.23	56.61	15.8%

4.2.4.3 Aspects vs. Opinion Targets

In this section we analyze the vocabulary overlap between the terms identified in the two annotation efforts, i.e., between aspects and opinion target terms. For the first dataset we consider a majority vote, labeling terms as relevant when they are annotated as such by two or more judges. We further restrict ourselves to the same 59 entities annotated with opinion targets in the second dataset. We tokenize the phrases identified as opinion targets, keeping the constituent terms that occur in them after removing stopwords and symbols.

As an example, Table 4.16 shows the opinionated aspects for some of the entities in the datasets. The percentage of how many times an aspect has been annotated as a target of an opinion (w.r.t. its occurrence in the aspects corpus) is shown in parentheses. In general, the percentage of an aspect identified as an opinion targets is low, suggesting that an aspect appears more often in factual statements than in opinionated contexts.

TABLE 4.16: Examples of aspects that are included in opinion target phrases, with the percentage of the aspect identified as a target in parentheses.

Entity	Aspects in opinion targets
Jaguar Cars Ltd.	jaguar (12.00%), xj (15.91%), cars (11.76%), auto (9.09%), xf (6.67%), car (3.57%), rover (1.67%)
Linux	linux(1.59%), multitouch (9.09%)
Sony	sony (1.42%), music (44.44%), vegas (27.27%), headphones (33.33%), battery (9.52%), pro (12.50%), xperia (5.00%), playstation (4.76%), x10 (4.35%), camera (2.50%), ericsson (1.33%)
Starbucks	starbucks (9.34%), coffee (26.19%), tea (37.50%), frappuccino (50.00%), drink (25.00%), latte (18.18%)

From a total of 783 aspects, 209 (26.69%) occur in opinion target phrases. Vice versa, the total number of terms extracted from the opinion target phrases is 1650; only 12.66% of those are also identified as relevant aspects. The overlap between aspects and opinion targets is lower than expected. The low overlap is probably given by the different methodologies used to annotate the both corpus. While aspects were annotated using a pooling methodology that considers the 10 highest ranking terms retrieved from each method, opinion targets were manually annotated inspecting the tweets related to each company. We observe that, instead of an aspect, the actual name of the entity has a tendency to occur as a target. However, the remaining aspects occur only a few times, suggesting a power-law distribution. In fact, terms in opinion targets are very sparse. The average occurrence of a term in an opinion target equals 1.78 and more than 75% of all terms occur only once. This suggests that the WePS-based sample of around 150 tweets per entity might not be enough for opinion-based entity profiling. We leave verifying this hypothesis (and possibly creating a larger dataset) for future work.

The low overlap between relevant aspects and terms occurring in opinion target phrases shows the different nature of the two corpora built. We believe that these resources will allow to evaluate different entity profiling systems in microblog posts and to make progress in the use of human language technologies for online reputation management.

4.3 Wrap Up

In this chapter, we have identified the main research challenges in the Online Reputation Monitoring (ORM) process and formalized them as information access tasks. We have provided a standard evaluation framework —which includes reusable test collections— defined in the context of international evaluation campaigns —WePS-3, RepLab 2012 and RepLab 2013— that cover the defined ORM tasks in the *unknown-entity* and *known-entity* scenarios.

One of the crucial tasks on which text mining and IR techniques may significantly reduce the effort of the ORM process is the filtering task (*Is the tweet about the entity of interest?*). Next chapter tackles the filtering task in both *unknown-entity* and *known-entity* scenarios, where we will study the effectiveness of using filter keywords and the filtering task in an active learning setting.

5 Filtering

A major problem concerning the retrieval of potential mentions in ORM is that brand names are often ambiguous. For instance, the query "Ford" retrieves information about the motor company, but also might retrieve results about Ford Models (the modeling agency), Tom Ford (the film director), etc.

One might think that the query is too general, and the user should provide a more specific query, such as "ford motor" or "ford cars". In fact, some tools explicitly suggest the user to refine possibly ambiguous queries[1].

This approach has two main disadvantages: (i) users have to make an additional effort when defining unambiguous queries and (ii) query refinement harms recall over the mentions of the brand in the Web, which can be particularly misleading in an ORM scenario. Ideally, a filtering process in ORM should remove spurious mentions as much as possible, without harming the coverage of relevant mentions.

Moreover, note that filtering out the mentions that do not refer to the monitored entity is also crucial when estimating its visibility. Quantifying the number of mentions about an entity on the Web —and, particularly, on Twitter—, and how this number changes over time, is essential to track marketing or Public Relationship campaigns. When the entity name is ambiguous, indicators given by tools such as Google Trends[2] or Topsy[3] can be misleading.

We think that a component capable of filtering out mentions that do not refer to the entity being monitored (specified by the user as a keyword plus a representative URL) would be a substantial enhancement of current ORM tools, and would also facilitate the analysis of the online presence/visibility of a brand.

In this chapter, we tackle the filtering task with two approaches. Firstly, we will explore the notion of using filter keywords to solve the task (§5.1). We will focus on the *unknown-entity*

[1] Alterian SM2 service: http://www.sdlsm2.com/social-listening-for-business/industry/.

[2] http://www.google.com/trends

[3] http://www.topsy.com

scenario, on which we will study if filter keywords exist and how can be extracted automatically from Web resources. Secondly, we will analyze the task in an active learning scenario (§5.2), studying the capability of this machine learning technique to keep the filtering model updated over time with minimum effort.

5.1 Filter Keywords

In this section we will validate an intuitive observation derived by the set-up and the analysis of the results of the WePS-3 Online Reputation Management Task [7], which tackles for the first time the filtering task in the *unknown-entity* scenario. As we have seen, in the unknown-entity scenario, systems do not use any previously annotated data about the target entity.

Filter Keywords Hypothesis

The observation is that manual annotation can be simplified by picking up special keywords —henceforth called *filter keywords*— that reliably signal positive or negative information. For instance, "ipod" is a positive filter keyword for the Apple company, because its presence is a highly reliable indicator that the tweet is about the company. Reversely, "crumble" is a negative filter keyword for Apple, because it correlates with unrelated tweets. The intuition is that automatic detection of such filter keywords can be a valuable signal to design an automatic solution to the problem.

Our goal is to provide quantitative evidence supporting (or rejecting) our intuition, and to answer some related questions:

- *Is the notion of filter keywords useful? (i.e., Do filter keywords exist in a given tweet stream? If so, are filter keywords present in representative Web resources of the entity of interest?)*
- *Can filter keywords be automatically extracted from Web resources (e.g., Wikipedia, ODP, entity's homepage) ?*
- *What is the effectiveness of filter keywords for solving the filtering task?*
- *What is the effectiveness of using entity-specific training data when available (*known-entity-scenario*)?, i.e., Is it worth looking for filter keywords in external resources or it is better to learn them automatically from the training data?*

We will focus in the unknown-entity scenario —which is the most challenging, using the WePS-3 Task 2 test collection described in §4.2.1. The WePS-3 ORM dataset contains 52 entities as training cases and 47 entities as test cases, each of one comprising an average of 486 tweets. Otherwise stated, results are macro-averaged across the 47 entities in the test set.

We start by tackling the first research question by investigating the upper bound of the filter keyword strategy (§5.1.1. Then, we analyze how to discover filter keywords automatically with the use of external resources (§5.1.2) and how to use filter keyword for solving the filtering task (§5.1.3). Finally, we analyze the usefulness of filter keywords in the known-entity scenario—where entity-specific training data is available—-and we conclude in §5.1.5.

5.1.1 Is the Notion of Filter Keywords Useful for the ORM Filtering Task?

A positive/negative filter keyword is an expression that, if present in a tweet, indicates a high probability that the tweet is related/unrelated to the entity.

Given the tweets D_q for an entity name q, let D_q^w be the tweets in D_q on which a given term w occurs. Analogously D_e^w are the related tweets D_e containing the term w and f_G the filtering gold standard (as defined in the previous chapter in Eq. 4.2):

$$\text{filterKeyword}(w) = \frac{|D_e^w|}{|D_q^w|} \vee \frac{|D_{\neg e}^w|}{|D_q^w|} \approxeq 1 \tag{5.1}$$

where

$$D_e^w = \{d \in D_q^w | f_G(d) = \text{related}\} \qquad D_{\neg e}^w = \{d \in D_q^w | f_G(d) = \text{unrelated}\} \tag{5.2}$$

In this section, we investigate whether filter keywords exist in the tweet stream and, if so, if we can find them from web pages representative to the entity of interest.

Firstly, we consider manual annotations in the WePS-3 collection to derive *oracle* (optimal) keywords. Secondly, we manually extract keywords from representative Web pages about the entity (manual keywords). Then, we will see if we can found oracle keywords in those representative web pages, more specifically, the entity's Wikipedia article and its homepage. Finally, we study how to use these filter keywords to solve the WePS-3 ORM task.

Oracle Keywords

Oracle Keywords. The most useful filter keywords are those with a high coverage, i.e., those which appear in as many tweets as possible.

As all tweets in the WePS-3 collection are manually annotated as related/unrelated to their respective company name, we can find exactly how many filter keywords there are (by definition, filter keywords are those terms that only appear in either the positive or the negative tweets), and how much recall they provide. Figure 5.1 shows the coverage of the first n filter keywords (for $n = 1 \ldots 20$) in the test collection.

Coverage at step n is the proportion of tweets covered by adding the keyword that filters more tweets among those which were not still covered by the first $n-1$ keywords. We will hereafter refer to this optimal keyword selection as *oracle keywords*.

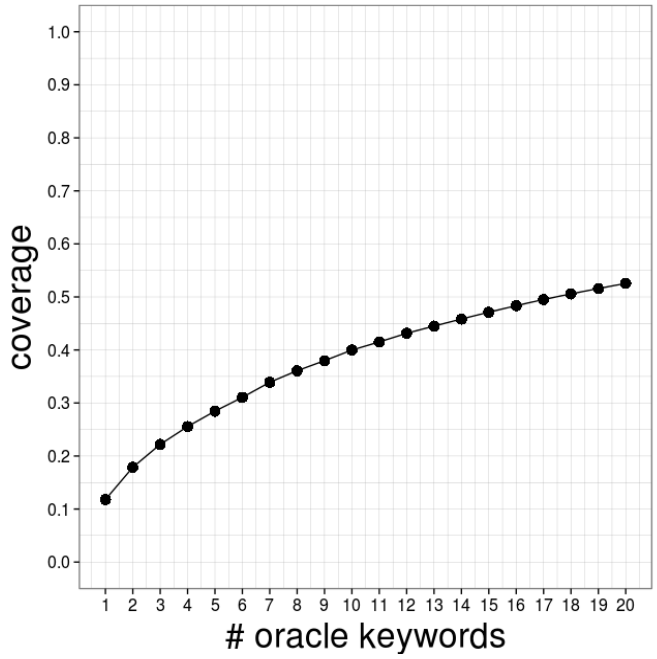


FIGURE 5.1: Upper bound of the filter keyword strategy.

The graph shows that, on average, the best five oracle keywords cover around 30% of the tweets, and the best ten cover around 40% of the tweets. Note that the best five discriminative terms already cover, on average, 130 out of 435 tweets in each stream, and those could in turn be used to build a supervised classifier for the rest of the tweet stream. This indicates that filter keywords are, potentially, a relevant source of information to address the problem.

Manual Keywords. In principle, the natural place to find filter keywords is the Web: the company's web domain and the reference to this domain in Wikipedia, ODP, etc. and the Web at large. Using the company's URL and web search results for the company name, we performed a manual selection of positive and negative keywords for all the companies in the WePS-3 corpus. Note that the annotator inspected pages in the web search results, but did not have access to the tweets in the corpus. Remarkably, manual keywords extracted from the Web (around 10 per company) only reach a 15% coverage of the tweets (compare with the 40% coverage using 10 oracle keywords extracted from the tweet stream), with an accuracy of 0.86 (which is lower than expected for manually selected filter keywords). This seems an indication that the vocabulary and topics of microblogging are different from those found in the Web. Our experiments in Section 5.1.2 corroborate this finding.

Manual Keywords

Oracle vs. Manual Keywords. Tables 5.1 and 5.2 show some examples for positive and negative keywords, respectively. Note that in the set of Oracle keywords there are expressions that a human would hardly choose to describe a company (at least, without previously analyzing the tweet stream). For instance, the best positive oracle keywords for the *Fox Entertainment Group* do not include intuitive keywords such as `tv` or `broadcast`; instead, they include keywords closer to breaking news (`leader`, `denouncing`, etc.).

TABLE 5.1: Differences between oracle and manual positive keywords for some of the company names in the test collection.

Entity Name	**Oracle Positive Keywords**	**Manual Positive Keywords**
amazon	sale, books, deal, deals, gift	electronics, apparel, books, computers, buy
apple	gizmodo, ipod, microsoft, itunes, auction	store, ipad, mac, iphone, computer
fox	money, weather, leader, denouncing, viewers	tv, broadcast, shows, episodes, fringe, bones
kiss	fair, rock, concert, alleseg-retti, stanley	tour, discography, lyrics, band, rockers, make up design

Looking at negative keywords (Table 5.2), we can find occasional oracle keywords that are closely related with the vocabulary used in microblogging services, such as `followdaibosyu`, `nowplaying` or `wanna`, while intuitive manual keywords like `wildlife` for jaguar are unlikely to occur in the Twitter collection.

TABLE 5.2: Differences between oracle and manual negative keywords for some of the company names in the test collection.

Entity Name	**Oracle Negative Keywords**	**Manual Negative Keywords**
amazon	followdaibosyu, pest, plug, brothers, pirotta	river, rainforest, deforestation, bolivian, brazilian
apple	juice, pie, fruit, tea, fiona	fruit, diet, crunch, pie, recipe
fox	megan, matthew, lazy, valley, michael	animal, terrier, hunting, Volkswagen, racing
kiss	hug, nowplaying, lips, day, wanna	french, Meg Kevin, bang bang, Ryan Kline

Oracle Keywords: Web vs. Twitter. We have seen that manual keywords in the Web are not completely equivalent to oracle keywords in Twitter. Remarkably, manually selecting around 10 salient terms from Web search results retrieved using the entity name and its representative URL only covers 15% of the tweets. This indicates that the vocabulary that characterizes a company on Twitter substantially differs from the the vocabulary associated to the company on the Web.

In order to corroborate this finding, we have explored the association between the best 10 oracle keywords for each tweet stream and its occurrences in both the company's homepage and its Wikipedia article[4]. The terms from each page have been extracted using the

[4]We manually extended the input data of each organization on the WePS-3 dataset with its Wikipedia page (or its homepage in the cases which the Wikipedia page is provided as the representative page)

`lynx -dump` *url* Linux command. Table 5.3 shows the average percentage of the best 10 oracle keywords that occur on the company's homepage, on the Wikipedia page, and both.

TABLE 5.3: Percentage of the 10 best *oracle* keywords extracted from the tweet stream covered by the company's homepage, its Wikipedia article and both.

Filter Keywords	Homepage	Wikipedia	Both
positive *oracle* keywords	36%	68%	33%
negative *oracle* keywords	9%	19%	6%

Overall, the only substantial overlap is for positive keywords in Wikipedia, indicating that representative Web pages are not the ideal place to look for effective filter keywords in Twitter.

Note that the overlap of related oracle keywords with the company's Wikipedia page roughly doubles the overlap with its homepage. The same thing happens with unrelated keywords: almost 20% on Wikipedia and almost 10% on the homepage. The percentage of oracle keywords that occur both in the homepage and in the Wikipedia article is similar to the homepage alone, indicating that Wikipedia basically extends the keywords already present in the homepage.

Therefore, none of the entity's homepage and its corresponding Wikipedia article cover the best oracle keywords. However, it seems more likely to find them in the Wikipedia article than in the entity's homepage.

Solving the Filtering Task. So far, we have seen that filter keywords do not cover all tweets in the collection, but a part of them. In order to annotate the remainder tweets to complete the task, we need a propagation step: given a seed of tweets, annotate the tweets that remain uncovered by filter keywords on each test case. To this aim, we experiment with a standard *bootstrapping* method. Tweets are represented as Bag-of-Words (BoW) —produced after tokenization, lowercase and stop word removal—and term occurrence is used as weighting function; then we have employed a C4.5 Decision Tree classification model[5] [146] —with its default parameters— using the implementation provided by the Rapidminer toolkit [128]. For each stream, we use the tweets retrieved by the keywords as seed (training set) in order to classify automatically the rest of tweets.

Propagation Step: Bootstrapping

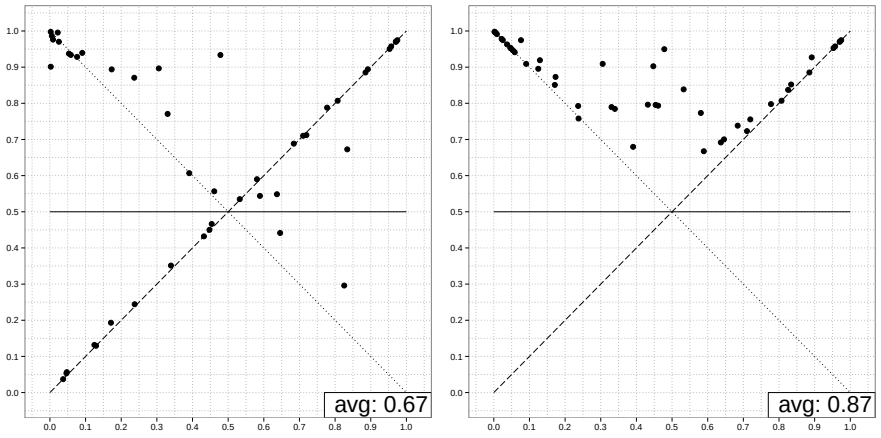
Bag-of-Words (BoW) Classifier

Table 5.4 displays results for different amounts of filter keywords: the bootstrapping strategy ranges from 0.81 (with 5 keywords) up to 0.87 with 20 keywords. On the other hand, using the tweets covered by manual keywords as training set, the bootstrapping achieves only a 0.67 accuracy.

[5]We also tried with other machine learning methods, such as linear SVM and Naive Bayes, obtaining similar results; therefore we only report results on C4.5 for the sake of clarity.

TABLE 5.4: Quality of seed sets and the bootstrapping classification strategy when applying oracle/manual filter keywords.

Keyword Selection Strategy	**Seed Set**		**Bootstrapping**
	Coverage	**Accuracy**	**Overall Accuracy**
5 oracle keywords	28%	1.00	0.81
10 oracle keywords	40%	1.00	0.85
15 oracle keywords	47%	1.00	0.86
20 oracle keywords	53%	1.00	0.87
manual keywords	15%	0.86	0.67



(A) manual keywords (B) 20 oracle keywords

FIGURE 5.2: Fingerprints for the bootstrapping classification strategy when applying manual keywords (a) or 20 oracle keywords (b).

In order to better understand the results, Figure 5.2 shows the *fingerprint* representation—previously presented at Section 4.1.2.1—for manual keywords and 20 oracle keywords. This visualization technique consists of displaying the accuracy of the system (vertical axis) for each company/test case (dots) vs. the ratio of related (positive) tweets for the company (horizontal axis). The three basic baselines (all true, all negative and random) are represented as three fixed lines: $y = x$, $y = 1 - x$ and $y = 0.5$, respectively. The fingerprint visualization method helps in understanding and comparing systems' behavior, specially when class skews are variable over different test case.

Using 20 oracle keywords (see Figure 5.2b), the obtained average accuracy is 0.87. The fingerprint shows that the improvement resides in cases with a related ratio around 0.5, i.e., the cases where it is more likely to have enough training samples for both classes. Still, on average, there is almost 15% of tweets which remain missclassified by this upper bound.

Manual keywords, on the other hand, lead to annotations that tend to stick to the "all related" or "all unrelated" baselines, which indicates that they tend to describe only one class, and then the

learning process is biased. This is may be due to the fact that negative manual keywords defined from inspecting Web search results are unlikely to occur in the tweet stream.

In summary, our results validate the idea that filter keywords can be a powerful tool in our filtering task, but also suggest that they will not be easy to find: there is a gap between the most useful or accurate keywords in the Twitter stream and the vocabulary found in descriptive web sources that can be attributed to the entity of interest.

In the next section, we study the benefits and limitations of automatically discovering filter keywords from different Web sources.

5.1.2 Automatic Discovery of Filter Keywords

Our next step is to study to what extent filter keywords can be automatically discovered from the information available in the *unknown-entity* scenario. Note that in the unknown-entity scenario, there is no entity-specific training data available. Therefore, the information available for representing terms are: the term frequency and co-occurrence distribution in the tweet stream and the term frequency distribution in Web sources that are potentially related to the entity of interest.

The goal is to automatically identify the terms which are most strongly associated to the entity (*positive* filter keywords) and to the alternative meanings of the entity name (*negative* filter keywords), and to discard those which are not discriminative (*skip* terms).

In this section, we start discussing the features that we propose to represent terms taking into account the information available. Then, we perform a statistical analysis of the features and finally, we study the effectiveness of different feature combinations for predicting if a term is a positive/negative filter keyword or is a skip term, reporting the results of experiments with the WePS-3 ORM dataset.

5.1.2.1 Features

We define three families of features: collection-based features, web-based features, and web-features expanded by term co-occurrence in the collection. Note that the only information available for representing a term in the *unknown-entity* scenario are: (i) its frequency and co-occurrence distribution in the collection and (ii) its frequency distribution in external Web sources.

Table 5.5 summarizes the notation that we will use in this section to describe the features.

We have considered a total of 18 features grouped in the three classes mentioned above:

TABLE 5.5: Notation used to describe the features used to represent terms.

Item	Description
w, w_i	term
q	query or (ambiguous) name that identifies an entity (e.g., `jaguar`)
$\mathcal{C}$	set of tweets in the WePS-3 collection[a]
D_q	set of tweets in the collection for a given company name q.
D_q^w	Documents containing the term w in the collection D_q.
D_Ψ^x	Documents returned by the Yahoo! Search BOSS API (`http://developer.yahoo.com/search/boss/`) for the query x.
M	an approximation of the size of the search engine index (30×10^9).
domain_e	domain of the website given as reference for the entity of interest e.
$\text{wikipedia}(q)$	set of Wikipedia pages returned by the MediaWiki API (`http://www.mediawiki.org/wiki/API`) for the query q.
$\text{dmoz}(q)$	set of items (composed by an URL, a title, a description and a category) returned by searching q on the Open Directory Project (`http://www.dmoz.org/search`)

[a] For each entity name, only the dataset to which the company belongs is used (either training or test).

Collection-based features (`col_*`) Terms that co-occur frequently with the (ambiguous) company name q, or terms written as hashtags should have more probability to be (positive/negative) keywords than others. These features combine information about the occurrence of the term in the collection: document frequency in the whole corpus, document frequency in the set of tweets for the company, how many times the term occurs as a twitter hashtag, and the average distance between the term and the entity name in the tweets.

- **`col_1`**: Normalized document frequency in the collection of the tweets for the company D_q:

$$\frac{|D_q^w|}{|D_q|} \tag{5.3}$$

- **`col_2`**: Ratio of document frequency in the tweets for the company D_q over the document frequency in the whole corpus $\mathcal{C}$:

$$\frac{|D_q^w|}{|\mathcal{C}|} \tag{5.4}$$

- **`col_3`**: Number of occurrences of the term as a hashtag (e.g., `#jobs`, `#football`) in D_q.
- **`col_4`**, **`col_5`**, **`col_6`**: Terms that occur close to the entity name might have more probability of being a keyword (e.g., part of a complete name or product like "apple

store"). Respectively, these features correspond to the mean, standard deviation and median of the distance (number of terms) between the term w and the entity name q in the tweets.

Web-based features (`web_*`) These features are computed from information about the term in the all the Web (approximated by search counts), the website of the entity, Wikipedia and the Open Directory Project (ODP).[6]

- **`web_1`**: Intuitively, a term which is close to the company name has more chances to be a keyword (either positive or negative) than more generic terms. This feature represents the association, according to the search counts, between the term w and an entity name q.

$$\frac{|D_\Psi^w \cap D_\Psi^q|}{|D_\Psi^q|} \cdot \frac{M}{|D_\Psi^w|} \tag{5.5}$$

- **`web_2`**: An alternative way of capturing the association between the term w and the entity name q is given by the Normalized Google Distance [40] (applied to the Yahoo! search engine), which is a measure of semantic distance between two terms from the search counts. Then, for a term w and an entity name q, the Google Normalized Distance is given by (5.6):

$$\frac{\max(\log(|D_\Psi^q|), \log(|D_\Psi^w|)) - \log(|D_\Psi^w \cap D_\Psi^q|)}{M - \min(\log(|D_\Psi^w|), \log(|D_\Psi^q|))} \tag{5.6}$$

- **`web_3`**: Frequent terms in the entity's website should be meaningful to characterize positive keywords. `web_3` is the normalized document frequency of the term in the website of the entity e.

$$\frac{|D_\Psi^w \cap D_\Psi^{\text{site:}domain_e}|}{|D_\Psi^{\text{site:}domain_e}|} \tag{5.7}$$

- **`web_4`**: Degree of association of the term with the website in comparison with the use of the term in the web. This feature is analogous to `web_1`, using the website domain instead of the company name q.

$$\frac{|D_\Psi^w \cap D_\Psi^{\text{site:}domain_e}|}{|D_\Psi^{\text{site:}domain_e}|} \cdot \frac{M}{|D_\Psi^w|} \tag{5.8}$$

- **`web_5`**: The DMOZ Open Directory Project is a collaborative web directory that includes manual summaries of Web pages. Terms occurring in ODP items related to the entity's domain are likely to be positive keywords for the entity e. This feature corresponds to the number of occurrences of the term in all the items in $dmoz(domain_e)$. Each item is composed by an URL, a title, a description and the ODP category to which it belongs.

[6] Some entities in the Weps-3 collection have a Wikipedia page as reference page instead of the entity website. In these cases, the feature `web_3` (that is also the numerator in the feature `web_4`) is computed as the presence of the term w in the Wikipedia page. Also, the query used to get the values of the features `web_5` and `web_6` is the title of the Wikipedia page.

- **`web_6`**: Likewise, Wikipedia articles relevant to the entity's domain are—a priori—a useful place to find positive filter keywords. Here, we compute the number of occurrences of the term in the first 100 results in $wikipedia(domain_e)$. In order to filter pages returned by the API that could be unrelated to the company, only pages that contain the string $domain_e$ are considered.

Features expanded with co-occurrence (`cooc_*`) In order to avoid false zeros in web-based features, we expand some of the previous term features with the value obtained by the five most co-occurrent terms.

Given a feature f, a new feature is computed as the Euclidean norm (Eq. 5.9) of the vector with components $f_{w_i} * s(w, w_i)$ for the five most co-occurrent terms with w in the set of tweets D_q (Eq. 5.10), where f_{w_i} is the web-based feature value f for the term w_i and $s(w, w_i)$ is the grade of co-occurrence of each term (Eq. 5.11):

$$cooc\_agg(w, f) = \sqrt{\sum_{i \in cooc_w} (f(w_i) * s(w, w_i))^2} \tag{5.9}$$

$$cooc_w = \text{ set of the five terms which most co-occur with } w \tag{5.10}$$

$$s(w, w_i) = \frac{|D_q^w \cap D_q^{w_i}|}{|D_q|} \tag{5.11}$$

$$f(w_i) = \text{value of the feature } f \text{ for the term } w_i$$

This formula is applied to each of the web-based features described above, resulting to the analogous `cooc_1`,...,`cooc_6` features.

5.1.2.2 Feature Analysis

The first step for the feature analysis is to develop a gold standard set of positive and negative keywords.

In order to get sufficient training data and to deal with possible miss-annotations in the corpus, we set a precision of 0.85 of a term w in a related/unrelated set of tweets as a feasible threshold to annotate a term as a keyword. Those terms with precision lower than 0.85 in both classes are labeled as *skip* terms (Eq. 5.12):

$$label(w) = \begin{cases} \text{positive} & \text{if } \frac{|D_e^w|}{|D_q|} > 0.85 \\ \text{negative} & \text{if } \frac{|D_{\neg e}^w|}{|D_q|} > 0.85 \\ \text{skip} & \text{otherwise} \end{cases} \tag{5.12}$$

where f_G represents the filtering goldstandard.

Labeling all suitable terms of the WePS-3 training dataset we end up with a total of 6,410 terms, where 34% were labeled as positive keywords, 44% as negative keywords and the remaining 22% as skip. The test dataset, on the other hand, produces a total of 4,653 candidate terms, where 33% were labeled as positive keywords, 40% as negative keywords and 27% as skip. We

only work with terms which are not stop words and appear at least in five different tweets on the set D_q, given an entity name q.

In order to study feature behavior, we calculate the distribution of each feature in the three classes: positive, negative and skip. We rely on box/whisker plots to show these distributions and differences or similarities between classes (see Figure 5.3). Each box/whisker plot shows the distribution of values of a feature for the three classes. The bottom and top of the box are the 25^{th} and 75^{th} percentile (the Q_1 and Q_3 quartiles, respectively), and the band near the middle is the 50^{th} percentile (the median, Q_2). The whiskers extend to the most extreme data point (1.5 times the length of the box away from the box: $-1.5 \cdot \text{IQR}$ and $1.5 \cdot \text{IQR}$, where $\text{IQR} = |Q_3 - Q_1|$).

These plots help visualizing the range of values for each feature, as well as where most of the values lie, allowing for a qualitative analysis of the features.

Some features are little or not informative at all. We can see that features `col_3` (Fig. 5.3c), `web_5` (Fig. 5.3o) and `cooc_5` (Fig. 5.3q) are not discriminative, because almost all of their values are zero. There are less than 1% of the terms in the test set that occur at least one time as hashtag (Fig. 5.3c). Also, less than 1% are terms that appear in descriptions and titles of ODP search results (Figs. 5.3o and 5.3q).

Features describing term-company distance seem to capture differences between keyword and skip terms: both negative and positive keywords, generally occur closer to the company name than skip terms. While positive and negative keywords share similar median and standard deviation (Figs. 5.3f and 5.3e) of proximity to the company name, average distance for positive keywords is slightly smaller than for negative keywords (Fig. 5.3d). Moreover, features `col_1`, `col_2`, `web_1`, `web_2` and their expanded (by co-occurrence) versions `cooc_1` and `cooc_2` are also able to discriminate filter keywords from skip terms. Here, specificity to the tweet stream (`col_2`) seems to be the most discriminative feature (Fig. 5.3b).

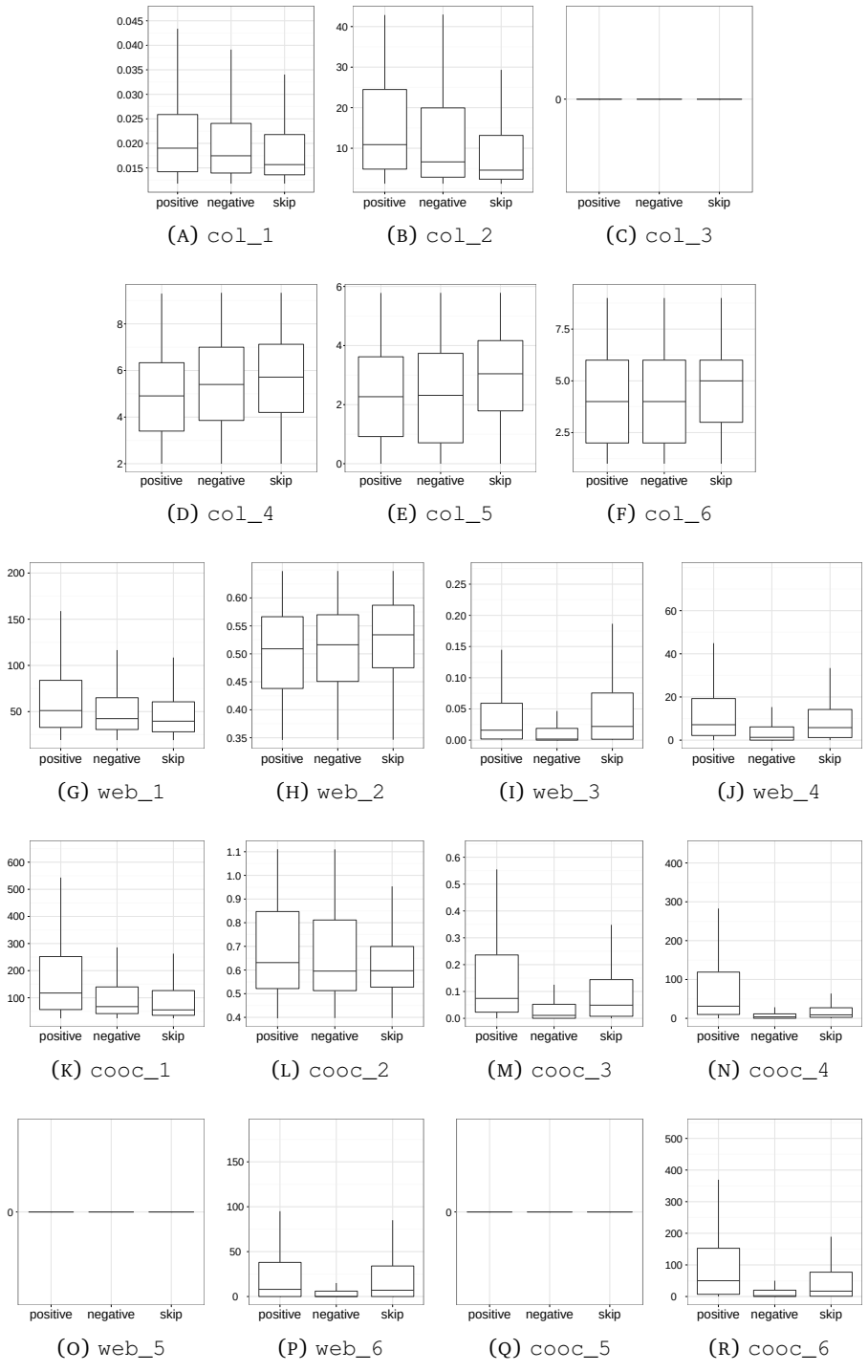
Specificity

On the other hand, features `web_6`, `web_3`, `web_4`, `cooc_3`, `cooc_4` and `cooc_6` were designed to distinguish between positive and negative filter keywords. At a first glance, positive and negative keywords have different distributions in all the features. Besides, skip terms tend to have distributions similar to those of positive keywords. The features `cooc_4` (Fig. 5.3n) and `cooc_6` (Fig. 5.3r) seem to be the best to discriminate positive keywords from negative and skip terms.

Remarkably, features expanded by co-occurrence seem to be more informative than the original features, which tend to concentrate on low values (the median is near zero). When expanding the original values by co-occurrence, positive terms receive higher values more consistently. Therefore, co-occurrence expansion seems to work well to alleviate the effect of false negatives.

In order to quantitatively evaluate the quality of features, we compute the Mann-Whitney U test [117], which is a non-parametric test used in statistical feature selection when a normal distribution of the features cannot be assumed. The p-value could be used to rank the features, since the smaller the value of the p-value, the more informative the feature is [66].

Non-parametric Test



(A) `col_1` (B) `col_2` (C) `col_3`

(D) `col_4` (E) `col_5` (F) `col_6`

(G) `web_1` (H) `web_2` (I) `web_3` (J) `web_4`

(K) `cooc_1` (L) `cooc_2` (M) `cooc_3` (N) `cooc_4`

(O) `web_5` (P) `web_6` (Q) `cooc_5` (R) `cooc_6`

FIGURE 5.3: Box-plots representing the distribution of each of the features in the positive, negative and skip classes. The bottom and top of the box are the Q_1 and Q_3 quartiles, respectively, and the band near the middle is the Q_2 quartile—i.e., the median. The whiskers extend to the most extreme data point (1.5 times the length of the box away from the box: $-1.5 \cdot \text{IQR}$ and $1.5 \cdot \text{IQR}$, where $\text{IQR} = |Q_3 - Q_1|$).

TABLE 5.6: U test p-value and ranking position of the features, comparing filter keywords (both positive and negative) with skip terms and comparing positive with negative filter keywords.

	Filter Keywords vs. Skip Terms		Positive vs. Negative Filter Keywords	
	p-value	Rank	p-value	Rank
col_1	2.11×10^{-19}	7	4.55×10^{-31}	8
col_2	$\mathbf{8.18 \times 10^{-50}}$	**1**	4.40×10^{-22}	9
col_3	1.49×10^{-02}	15	5.40×10^{-04}	12
col_5	1.87×10^{-33}	2	4.20×10^{-02}	15
col_4	2.03×10^{-20}	6	4.18×10^{-02}	14
col_6	5.79×10^{-14}	8	1.62×10^{-01}	16
web_1	4.76×10^{-06}	12	8.39×10^{-19}	10
web_3	6.67×10^{-30}	4	5.33×10^{-92}	6
web_4	7.14×10^{-14}	9	7.01×10^{-138}	4
web_2	5.12×10^{-12}	10	3.38×10^{-08}	11
web_5	1.96×10^{-01}	16	3.63×10^{-01}	18
web_6	1.68×10^{-20}	5	4.86×10^{-115}	5
cooc_1	2.91×10^{-30}	3	2.04×10^{-54}	7
cooc_3	1.53×10^{-05}	13	1.42×10^{-189}	3
cooc_4	8.35×10^{-01}	18	$\mathbf{7.27 \times 10^{-233}}$	**1**
cooc_2	3.54×10^{-07}	11	2.41×10^{-01}	17
cooc_5	3.15×10^{-01}	17	3.92×10^{-03}	13
cooc_6	2.36×10^{-03}	14	2.13×10^{-211}	2

Table 5.6 shows the p-value and the rank of each feature for the U test. The most remarkable aspect of this table is that—in agreement with the boxplot analysis—the *col_2* feature discriminates between filter keywords and skip terms better than other features. In addition, the feature *cooc_4*, which measures the association between the term and the entity website, is the best feature to discriminate between positive and negative filter keywords. These results confirm our assumptions that salient terms in the set of tweets of the company tend to be discriminative and salient terms associated with the company in tweets are also associated with the company website.

Although the features analyzed above are signals that help differentiating between positive, negative and skip terms, it seems that the vocabulary that characterizes a company in microblog streams is different from the vocabulary associated to the entity in its website, in ODP entries or in Wikipedia.

5.1.2.3 Keyword Discovery

After analyzing the features independently, we now study how to combine them to automatically discover filter keywords.

The features described above have been combined in three different ways. The first one, (*machine learning-all features*), consists of training a positive-negative-skip classifier over the training corpus in WePS-3 by using all the features. We combine two classifiers: positive versus others and negative versus others, using the confidence thresholds learned by the classifiers. Terms which are simultaneously under/over both thresholds are tagged as skip terms.

machine learning - all features

heuristic

The second approach, (*heuristic*), is inspired by the analysis of the signal provided by each of the features (in the previous section). It is a simple heuristic which looks at only the best two features according to the Mann-Whitney U test: first, we define a threshold to remove skip terms according to the specificity w.r.t. the collection of the tweets for the entity (`col_2` feature). Then we state, for the feature that measures association with the website (`cooc_4` feature) a lower bound to capture positive filter keywords and an upper bound to capture negative filter keywords. These three thresholds have been manually optimized using the training data set.

machine learning - 2 features

Finally, we also explored a third option: we apply machine learning using only the best two features instead of the whole feature set. We will refer hereafter to this method as *machine learning - 2 features*.

We have experimented with several machine learning methods using Rapidminer tool[128]: Neural Nets, C4.5 and CART Decision Trees, Linear Support Vector Machines (SVM) and Naive Bayes. All methods have been used with "out-of-the-box" parameters. All the terms labeled over the WePS-3 training dataset were used to train the models. In the same way, terms extracted from the test dataset were used as test set. Table 5.7 shows the values of the Area Under the ROC Curve (AUC) of each of the binary classifiers evaluated. AUC is an appropriate metric to measure the quality of binary classification models independently of the confidence threshold [50].

Area under the ROC Curve

We analyzed three different subsets of features to represent the terms: (i) using all but the six features expanded by co-occurrence, (ii) using only the best two features (those used by the *heuristic* and *machine learning - 2 features* classifiers), and (iii) using all the features.

TABLE 5.7: Area Under the ROC Curve (AUC) values of the five classification models and the three feature sets used to classify positives and negatives keywords.

Machine Learning Algorithm	Not Expanded by Co-occurrence Features		2 Best Features		All Features	
	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
Neural Net	**0.68**	**0.67**	**0.73**	**0.72**	**0.75**	**0.73**
CART Dec. Trees	0.58	0.61	0.72	0.71	0.63	0.64
Linear SVM	0.50	0.50	**0.73**	0.71	0.50	0.50
Naïve Bayes	0.64	0.64	0.71	0.71	0.72	0.72
C4.5 Dec.Trees	0.50	0.61	0.50	0.50	0.59	0.66

The results obtained are similar for all models, except for C4.5 and SVM that in several cases do not provide any useful information for classification (AUC $= 0.5$). Keeping out the "expanded by co-occurrence" features, the performance is, in general, lower for all the algorithms. This corroborates the results of our previous feature analysis.

In the following experiments, we focus on the Neural Net algorithm to train both (positive versus others and negative versus other) classifiers, because it is consistently the best performing algorithm according to the AUC measure.

For each of the feature combinations described at the beginning of this section (*machine learning - all features*, *heuristic* and *machine learning - 2 features*), below we analyze the obtained results. The methods were trained using terms from the WePS-3 training dataset and evaluated with the WePS-3 test set.

Table 5.8 shows the confusion matrix obtained for the *machine learning - all features* method. The precision for the positive and negative classes is 62% and 56%, respectively, while recall is 52% and 72%. In order to obtain a few representative keywords, this recall levels are good enough; but the precision may compromise the final accuracy of the filtering process.

TABLE 5.8: Confusion matrix for the *machine learning-all features* classifier.

		Actual Class			
		Positive	**Negative**	**Skip**	**Class Precision**
Predicted Class	**Positive**	790	190	304	**62%**
	Negative	483	1,334	583	**56%**
	Skip	242	330	375	40%
	Class Recall	**52%**	**72%**	30%	

Table 5.9 shows the confusion matrix for the *heuristic* method, that only uses the `col_2` and `cooc_4` features. This method is more precision-oriented than *machine learning - all features*: precision values of positive and negative class are higher (68% and 75%), but recall is significantly lower (26% and 19%).

TABLE 5.9: Confusion matrix for the *heuristic* classifier.

		Actual Class			
		Positive	**Negative**	**Skip**	**Class Precision**
Predicted Class	**Positive**	391	60	122	**68%**
	Negative	23	352	94	**75%**
	Skip	1,102	1,453	1,056	29%
	Class Recall	**26%**	**19%**	83%	

Finally, Table 5.10 shows the contingency matrix for the *machine learning - 2 features* method, that represents terms with the features `col_2` and `cooc_4` and uses the Neural Net machine learning algorithm to build the model.

TABLE 5.10: Confusion matrix for the *machine learning-2features* classifier.

		Actual Class			
		Positive	**Negative**	**Skip**	**Class Precision**
Predicted Class	**Positive**	438	102	139	**65%**
	Negative	85	399	101	**68%**
	Skip	993	1,364	1,032	30%
	Class Recall	**29%**	**21%**	81%	

As expected, its performance lies between *machine learning - all features* and *heuristic* methods, with a precision higher than the former (65% and 68%) and a recall higher than the latter (29% and 21%).

5.1.2.4 How Much of the Problem is Solved by Filter Keywords?

In order to shed light on the trade-off between quality and quantity of filter keywords, here we analyze the relation between accuracy and coverage of the tweets classified by considering different sets of filter keywords. Tweets containing only positive keywords are classified as related. Analogously, tweets that contain negative keywords are classified as unrelated. Otherwise, tweets are considered as not covered by filter keywords.

Accuracy-Coverage Curves

Figure 5.4 shows the accuracy-coverage curves for oracle, manual and automatic filter keywords.

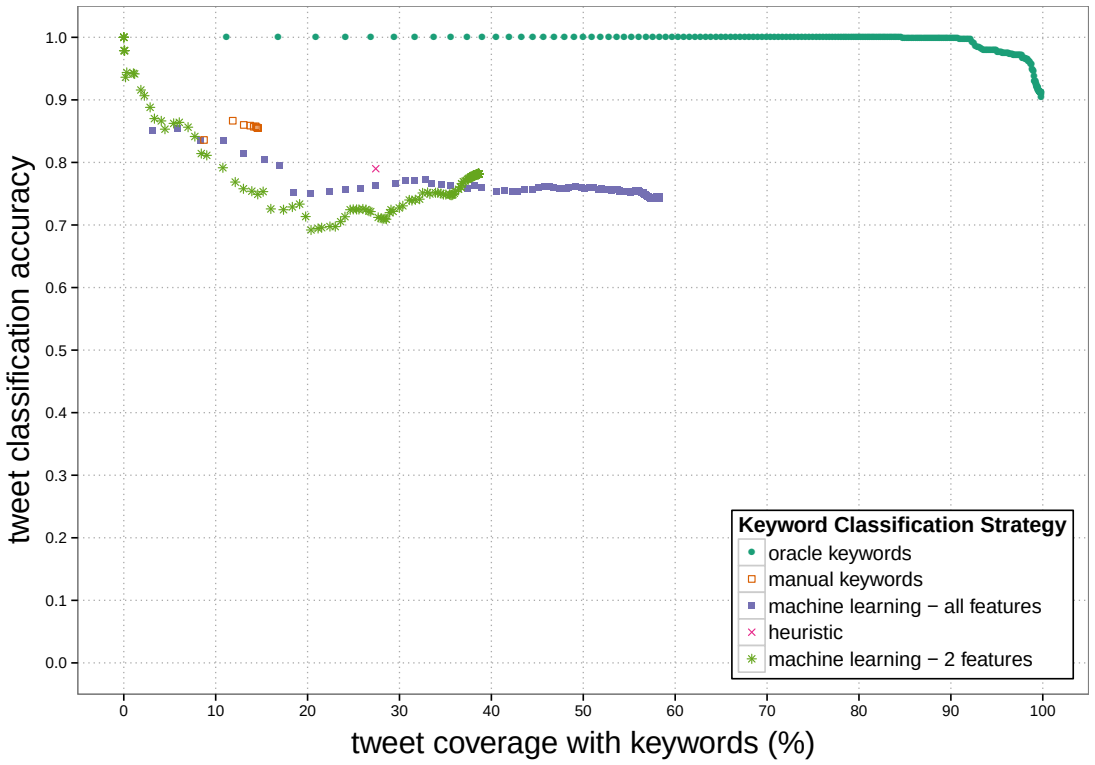


FIGURE 5.4: Coverage/accuracy curves for oracle, manual and automatic filter keywords.

Curves were generated as follows:

Oracle keywords. It represents a statistical upper bound of filter keywords in the dataset, since are extracted from the gold standard. At step n, we consider the n^{th} positive/negative oracle keywords that maximizes accuracy and—in case of ties—coverage of tweets,i.e., in the case of two keywords having same accuracy, the one that covers more tweets is considered first.

Manual keywords. At each step n we consider the n^{th} positive/negative manual keywords that maximize coverage of tweets. Manual keywords represents an upper bound from a user that identifies filter keywords from inspecting external resources such as relevant web search results for the entity of interest.

Machine learning - all features. This curve is generated automatically, i.e., without manual supervision. The set of terms considered in this analysis are those that were classified as positive or negative by using the confidence thresholds learned by the two classifiers in the method `machine learning - all features`. Skip terms are (i) those classified as skip by both binary classifiers; (ii) those classified simultaneously as positive and negative keywords. Then, we use the maximum of the confidence scores returned by the two classifiers[7] to sort the keywords. The keyword with highest confidence score is added at

[7]The maximum of the confidence scores returned by the two classifiers is computed as $\max(\text{conf}(\text{positive}), \text{conf}(\text{negative}))$

each step. The point in the curve with highest coverage corresponds to the classifier used in the experiments explained in Section 5.1.2.

Machine learning - 2 features. This curve is generated by using the two classifiers learned in "machine learning - 2 features", similarly to the curve generated for "machine learning - all features".

Heuristic. Since this classifier consists of manually defining thresholds using the training set, it doesn't provide any confidence score for the test cases. Hence, in the graphic it is represented as a single point ($\times$).

The curve for Oracle keywords provides a statistical upper bound of how many tweets can be directly covered using filter keywords. Considering the best 100 oracle keywords for each test case/company name, it is possible to directly tag 85% of the tweets with 0.99 accuracy. Note that—despite of firstly considering the keyword that covers more tweets in the case of two keywords having same accuracy—there may exists a bias to false keywords when the frequency is low (i.e., term frequency near to 5).

On the other hand, a more realistic upper bound is given by manual keywords. Here, we can observe how the accuracy remains stable around 0.85, while the coverage grows from 10% to 15% approx. In the best possible case, with more keywords the curve would continue as the line $y = 0.85$. Note that manual keywords have been annotated by inspecting representative Web pages (from Google search results) rather than inspecting tweets. Therefore, an automatic keyword classifier cannot be expected to achieve an accuracy above 0.85. Our automatic approaches, on the other hand, establish a strong lower bound of 0.7 accuracy. In conclusion, it seems that a filter keyword classifier should reach an accuracy between 0.7 and 0.85 to be competitive.

Finally, the strategy used to sort keywords in *machine learning - all features* and *machine learning - 2 features* seems effective but not optimal. As expected, both curves tend to decrease accuracy as long as more tweets are covered, being *machine learning - all features* more stable. However, considering a longer list of automatic keywords in *machine learning - 2 features* has a noise-reduction effect. When the list of filter keywords is long, the probability of a tweet belonging to the seed set is higher. However, note that tweets sharing negative and positive keywords are not taken into account. The latter is more likely to happen when one of the automatic filter keyword is misclassified.

In summary, exploring the nature of filter keywords in the *unknown-entity* scenario leads us to the conclusion that the vocabulary characterizing a company in Twitter is substantially different from the vocabulary associated to the company in its homepage, in Wikipedia, and apparently in the Web at large. These findings are in line with the "vocabulary gap" that has been shown between Twitter and other Web sources such as Wikipedia or news comments [172]. One way of alleviating this problem is using co-occurrence expansion of web-based features, which allows to better recognize automatically filter keywords. While the company's Wikipedia article seems to have more coverage of (perfect) filter keywords than the company's homepage, further investigation is needed on how to automatically infer the company's Wikipedia page from its homepage URL in order to extract additional keyword features from it.

Results obtained for the different feature combinations indicate that automatic detection of keywords is plausible and challenging at the same time. Which of the three approaches is better for our problem depends on their performance on the filtering task. In the next section we explore how to use these filter keywords to complete the task by classifying tweets not covered by filter keywords.

5.1.3 Completing the Filtering Task using Filter Keywords as Seeds

As we have seen, not every tweet in the stream contains filter keywords. But filter keywords can be used to produce a seed of tweets that are in turn used to feed a propagation step, and thereby cover the remaining tweets. After detecting the filter keywords automatically, we directly classify the subset of tweets that contain only negative or only positive keywords to produce the seed.

Propagation Step: Bootstrapping

Then, the *bootstrapping* strategy—based on Bag-of-Words—described in Section 5.1.1 is used to complete the task. Here, we assume that there is some degree of term co-occurrence between the tweets in the seed and those uncovered by filter keywords.

winner-takes-all

It is also interesting to validate two other assumptions. Similar to Yoshida et al.[192], we can assume that the ratio of related tweets is extremely biased to 0 or 1. Therefore, we consider a naïve *winner-takes-all* baseline, which directly classifies all the tweets as related or unrelated depending on which is the dominant class in the seed of tweets. On the other hand, this assumption can be relaxed by considering that the related ratio in both the seed and the uncovered tweet set is the same. This is the assumption held by the *winner-takes-remainder* strategy, which consists of applying the winner-takes-all strategy only to those tweets that were not covered by some of the filter keywords.

winner-takes-remainder

TABLE 5.11: Results for automatic keyword detection strategies (wta=winner-takes-all, wtr=winner-takes-remainder). Statistical significance w.r.t. the `ml-all features` selection strategy was computed using two-tailed Student's t-test. Significant differences are indicated using ▲ (or ▼) for $\alpha = 0.01$ and $\triangle$ (or $\triangledown$) for $\alpha = 0.05$.

Keyword Selection Strategy	Seed Set Coverage	Seed Set Acc.	Overall Accuracy wta	Overall Accuracy wtr	Overall Accuracy Bootstrapping
20 oracle keywords	53%$^{\blacktriangle}$	1.00$^{\blacktriangle}$	0.80$^{\triangle}$	0.85$^{\blacktriangle}$	0.87$^{\blacktriangle}$
manual keywords	15%$^{\blacktriangledown}$	0.86	0.61	0.63	0.67
m. learning - all feat.	58%	0.75	0.69	0.71	0.73
heuristic	27%$^{\blacktriangledown}$	0.79	0.64	0.65	0.71
m. learning - 2 feat.	39%$^{\blacktriangledown}$	0.78	0.70	0.72	0.72

Table 5.11 shows the results and Figure 5.5 shows the fingerprint of each of the combinations tested.

The best automatic method, which combines `machine learning-all features` to discover keywords and bootstrapping with the tweets annotated using that keywords gives an accuracy of 0.73, which is higher than using manual keywords from the Web (0.67) and is close to the best automatic result reported in the WePS-3 competition (0.75). In addition, the bootstrapping process almost doubles the coverage (from 58% to 100%) with only 2.7% of accuracy loss.

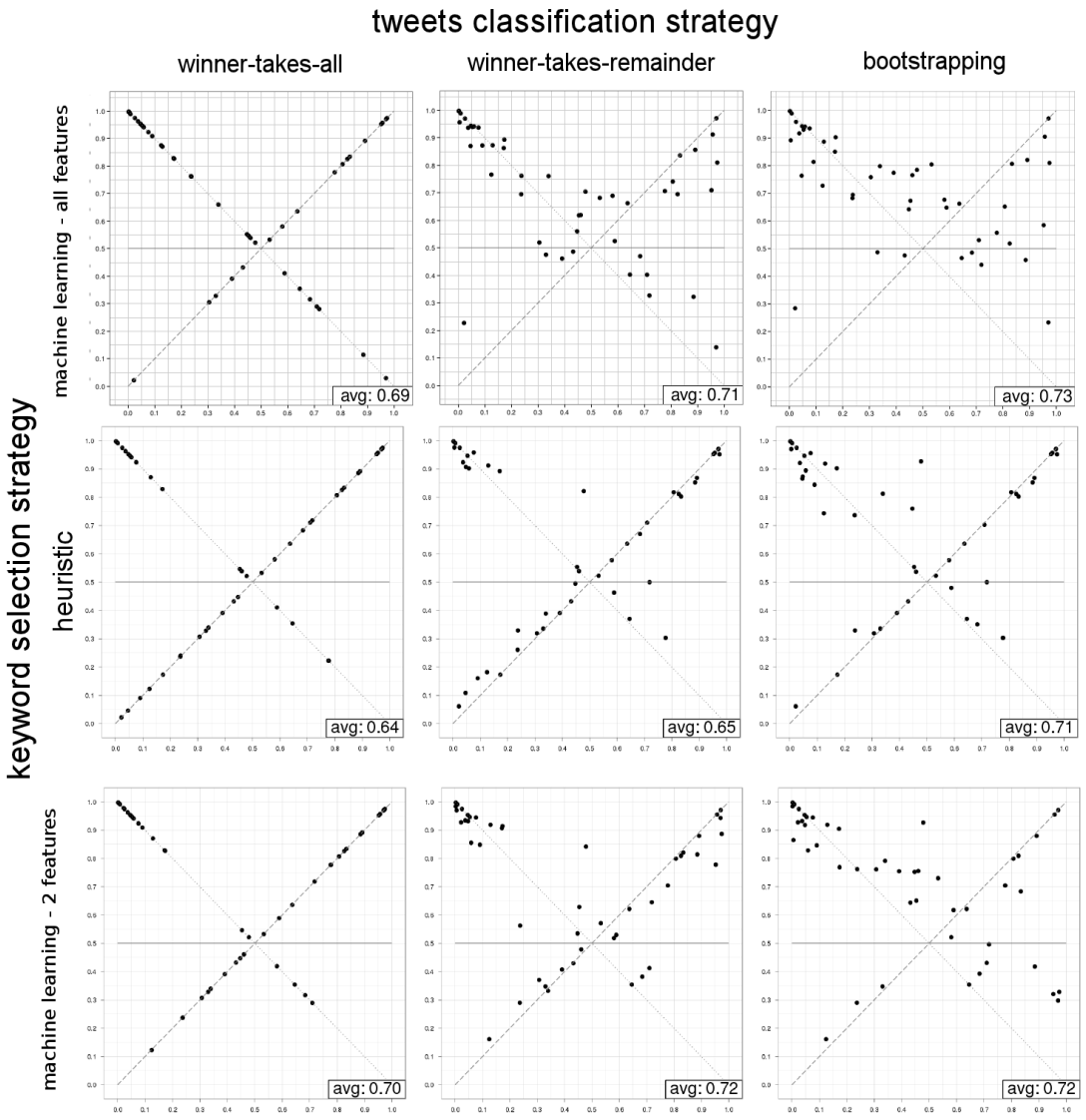


FIGURE 5.5: Fingerprints for each of the keyword selection strategies combined with each of the different tweet classification strategies.

In general, the more tweets covered by filter keywords (seed coverage), the lower the loss in accuracy: the `heuristic` keyword selection covers 27% of tweets with 0.79 accuracy and achieves a 0.71 accuracy with the bootstrapping process, while `machine learning-2 best features` covers 39% of the tweets with 0.78 accuracy and finishes with 0.72 accuracy. Note that, in this case, having more seed coverage at the expense of accuracy leads to highest accuracy at the completion of the task. Remarkably, the bootstrapping process outperforms the winner-takes-all and winner-takes-remainder baselines in all the cases. Therefore, it exists a lexical overlap between the seed and the set of tweets uncovered by filter keywords that can be harnessed to complete the task automatically.

Discovery of filter keywords has proved to be challenging using signals from the Web: the accuracy of the resulting seed set ranges between 0.75 and 0.79, with a potentially useful coverage (58%) in the case of the *machine learning - all features*. Overall, this result reinforces the conclusion that the characterization of companies in Twitter, in terms of vocabulary, is probably

different from the characterization that can be derived from the Web.

5.1.3.1 Comparing Systems with Different Metrics

We have seen that related/unrelated tweets are not balanced in most of the test cases in WePS-3, and the proportion does not follow a normal distribution—extreme values seem to be as plausible as values around the mean. Because of this, accuracy may be not sufficient to understand the quality of systems, and that is why we have complemented it with the fingerprint representation (§4.1.2.1). Here, we evaluate (and compare) results with the most popular alternative evaluation metrics found in the literature.

Considering the confusion matrix given by each system, where TP=true related tweets, FP=false related tweets, TN=true unrelated tweets, and FN=false unrelated tweets, we compute the following metrics, in addition to accuracy:

Normalized Utility. Utility has been used to evaluate document filtering tasks in TREC [77, 78] and is commonly used assigning a relative α weight between true positives and false positives:

$$u(S,T) = \alpha \cdot TP - FP$$

As in the TREC-8 filtering task [78], here Utility is normalized by means of the following scaling function:

$$u_s^*(S,T) = \frac{max\left(u(S,T), U(s)\right) - U(s)}{MaxU(T) - U(s)}$$

where $u(S,T)$ is the original utility of system output S for topic T, $U(s)$ is the utility of retrieving s non-relevant documents, and $MaxU(T) = \alpha \cdot (TP + FN)$ is the maximum possible utility score for topic T. In this study, we set $\alpha = 2$ and $U(s) = -25$.

lam%. $lam\%$ (logistic average misclassification percentage) has been used in TREC to evaluate the problem of spam detection [41]. It was defined as the geometric mean of the odds of $hm\%$ (ham misclassification percentage) and $sm\%$ (spam misclassification percentage). More precisely, $lam\%$ is defined as

$$lam\% = logit^{-1}\left(\frac{logit(hm\%) + logit(sm\%)}{2}\right)$$

where

$$hm\% = \frac{FN}{FN+TP} \quad sm\% = \frac{FP}{FP+TN}$$

$$logit(x) = log\left(\frac{x}{1-x}\right) \quad logit^{-1}(x) = \frac{e^x}{1+e^x}$$

Note that $lam\%$ is an error-based metric—i.e., maximum scores represent minimum quality.

One remarkable property of this metric is that, when a system has a non-informative behavior—that is, its output does not depend on the input—lam% score is around 0.5.

Reliability & Sensitivity. As we have seen in Section 4.1.2, Reliability & Sensitivity (R&S), are appropriate for measuring how informative a filtering system is [11]. R&S was the official pair of metrics used for evaluating filtering systems in RepLab 2012 [9] and RepLab 2013 [8].

As $lam\%$, Reliability & Sensitivity also penalizes systems that do not provide any useful information, giving zero — the minimal score — to systems that assign the same class to all documents.

$\mathbf{F_1}$ measure. The most standard combination of Precision and Recall is F_1, or balanced F measure. Here we focus on the "related" class, where

$$\textit{Precision} = \frac{TP}{TP + FP} \qquad \textit{Recall} = \frac{TP}{TP + FN}$$

and

$$F_1 = 2 \cdot \frac{\textit{Precision} \cdot \textit{Recall}}{\textit{Precision} + \textit{Recall}}$$

Table 5.12 reports the results for the baselines, the WePS-3 systems and our proposed systems for the metrics described above. All metrics were macro-averaged by topics, and undefined scores were considered as zero values.

TABLE 5.12: Results for proposed systems, WePS-3 systems and baselines compared with different evaluation metrics. Best automatic runs are in boldface. (ml=machine learning, wta=winner-takes-all, wtr=winner-takes-remainder). Statistical significance w.r.t. the `ml-all feat. + bootstrapping` run was computed using two-tailed Student's t-test. Significant differences are indicated using ▲ (or ▼) for $\alpha = 0.01$ and $\triangle$ (or $\triangledown$) for $\alpha = 0.05$.

System	Accuracy	Utility	lam%	$\mathbf{F_1}(\mathbf{R},\mathbf{S})$	$\mathbf{F_1}$
Gold standard	1.00▲	1.00▲	0.00▲	1.00▲	1.00▲
WePS-3: LSIR (manual)	0.83▲	0.66▲	0.28▲	0.27	0.62▲
ml-all feat. + bootstr.	0.73	0.47	0.37	**0.27**	0.49
ml-2 feat. + wtr	0.72	0.49	0.43	0.16$\triangledown$	0.49
ml-2 feat. + bootstr.	0.72	0.47	0.43	0.17$\triangledown$	0.50
heuristic + bootstr.	0.71	0.45	0.42	0.11▼	0.46
ml-all feat. + wtr	0.71	0.44	0.39▼	0.21▼	0.43▼
ml-2 feat. + wta	0.70	0.48	0.50▼	0.00▼	0.39$\triangledown$
ml-all feat. + wta	0.69$\triangledown$	0.40$\triangledown$	0.50▼	0.00▼	0.27▼
heuristic + wtr	0.65▼	0.46	0.42	0.10▼	0.46
heuristic + wta	0.64▼	0.44	0.50▼	0.00▼	0.39$\triangledown$
WePS-3: ITC-UT	**0.75**	**0.52**	0.37	0.20	0.49
WePS-3: SINAI	0.64$\triangledown$	0.38$\triangledown$	**0.35**	0.12▼	0.30▼
WePS-3: UvA	0.58▼	0.22▼	0.46▼	0.17▼	0.36▼
WePS-3: KALMAR	0.47▼	0.35▼	0.43	0.16$\triangledown$	0.48
baseline: all unrelated	0.57▼	0.20▼	0.50▼	0.00▼	0.00▼
baseline: random	0.49▼	0.20▼	0.49▼	0.16▼	0.37▼
baseline: all related	0.43▼	0.40	0.50▼	0.00▼	**0.52**

Results show that, according to Reliability & Sensitivity, our best automatic system `ml-all features + bootstrapping` achieves the same score as the WePS-3 LSIR semi-automatic system (0.27)—which is the best result at the competition and involves manual processing—and outperforms the best automatic system in WePS-3 (ITC-UT= 0.20), with a 35% of relative improvement. Remarkably, our filter keyword approach has similarities with these two systems. On one hand, WePS-3 LSIR makes use of both positive and negative keywords. On the other hand, WePS-3 ITC-UT assumes that predicting the bias of the related ratio is useful.

In terms of $lam\%$, the SINAI system achieves the best automatic score of 0.35, followed by ITC-UT & `ml-all features + bootstrapping` that reach 0.37 $lam\%$. Note that $lam\%$ and R&S penalize non-informative/baseline-like behaviors. Because of this, the `winner-takes-all` systems and the "all (un)related" baselines get the worst scores in these metrics.

According to utility, ITC-UT is still the best automatic system (0.52). Our best runs are between 0.47 and 0.49, being `ml-2 features + bootstrapping` the best of them. Finally, F_1 rewards systems that tend to return all tweets as related. Indeed, the best score given by an automatic system is achieved by the "all_related" baseline, that has perfect recall and enough precision to get the highest score.

There are a number of empirical observations that can be made on this comparison of metrics:

- In general, high $F_1(R,S)$ implies high accuracy, but not vice-versa: $F_1(R,S)$ is a stricter metric, at least in this dataset.
- Metrics such as $lam\%$ and $F_1(R,S)$ are suitable to identify baseline-like behaviors, while F_1 is not.
- ITC-UT and `ml-features + bootstrapping` perform consistently well across metrics.
- Different metrics illustrate different aspects of the behavior of systems: If we need to penalize non-informative behavior, we should look at results with $lam\%$ or $F_1(R,S)$. Accuracy and utility directly show misclassification errors, but are sensitive to collections where class skews are variable over different test cases, such as our dataset.

In summary, we have seen that filter keywords can be used to solve the filtering task in the *unknown-entity* scenario. An extended comparison with the most used evaluation metrics in filtering scenarios show that our approach is competitive with respect to the best WePS-3 systems. We now move to the *known-entity* scenario, in order to study if filter keywords are also effective when entity-specific training data is available.

5.1.4 Known-Entity Scenario: Filter Keywords

So far, we have studied the suitability of filter keywords in the unknown-entity scenario. An interesting question is how our approach—which does not require entity-specific training data—behaves in the known-entity scenario, in which training data for each of the companies in the test set is available. Here, we first compare our runs in the WePS-3 dataset with a supervised upper bound and then we present our participation at RepLab 2013, which consists of applying the filter keyword approach to the known-entity scenario.

Known-Entity Scenario

The supervised upper bound in WePS-3 uses the same machine learning algorithm and the same Bag-of-Words feature representation of the bootstrapping method presented in §5.1.1, but built upon perfect training material—taken from the test set—for each of the companies. To this aim, we carried a 10-fold validation of the model on the test set. This supervised upper bound achieves 0.85 accuracy, that is 14% higher than its unsupervised counterpart (0.73). If we use $F_1(R,S)$ as evaluation metric, differences are significantly higher: the supervised upper bound achieves $0.62^{\blacktriangle}$, which is 27% higher than the performance of filter keywords in the unknown-entity scenario (0.49).

Supervised Upper Bound in WePS-3

In Figure 5.6 we can see the fingerprint representation of the supervised upper bound. The fingerprint shows that almost all the test cases are in the triangle of the top, i.e, the BoW classifier is able to predict the majority class. In the cases on which the related ratio is close to 0.5, the classifier is able to correctly classify around 70–80% of the instances.

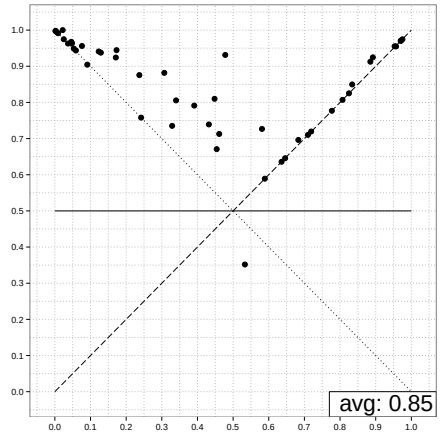


FIGURE 5.6: Fingerprint for the supervised upper bound (10-fold cross-validation).

In order to better quantify the effectiveness of our filter keywords approach in the known-entity scenario , we now present the official results of our participation at RepLab 2013. As we have seen in Section 4.2.3, the RepLab 2013 comprises entity-specific training data for the entities in the test set, making feasible the evaluation of filtering systems in the known-entity scenario.

Here, tweets are tokenized using a Twitter-specific tokenizer [138], frequent words are removed using both English and Spanish stop word lists, and terms occurring less than 5 times in the collection are discarded. The machine learning algorithm used in all the experiments was Naïve Bayes, using the implementation provided by the Rapidminer toolkit [128].

There are three main filtering runs. The *BoW Classifier* run corresponds to the use of the bootstrapping method presented above—Naive Bayes with BoW feature representation—built upon the entity-specific training data. Then, there are two variations of the filter keyword approach. On one hand, we simulate the unknown-entity scenario by using a "leave-one-entity-out" approach: for each entity in the test set, positive and negative filter keyword classifiers are learned using training data from all but the same entity. On the other hand, filter keywords are learned using only training data corresponding to the entity on each test case, which corresponds to the filter keyword approach in the *known-entity* scenario. These three runs correspond to the official runs named `replab2013_UNED_filtering_2`, `replab2013_UNED_filtering_5` and `replab2013_UNED_filtering_4`, respectively.

Table 5.13 reports the scores obtained for the evaluation metrics used in the filtering subtask: Accuracy, Reliability (R), Sensitivity (S) and $F_1(R, S)$. For each of the runs, the position on the official $F_1(R, S)$ RepLab rank is also shown.

RepLab 2013 Baseline

We also report the results for the official baseline and the best filtering system at RepLab 2013. The official baseline, which consists of an instance-based learning method which tag each tweet

TABLE 5.13: Results of the runs submitted for the filtering subtask. Best results in boldface; significant changes are w.r.t. the filter keyword strategy in the *known-entity* scenario.

Run	Accuracy	R	S	$\mathbf{F_1(R,S)}$	Rank
Best RepLab 2013 System [151]	$\mathbf{0.91}^{\blacktriangle}$	$\mathbf{0.73}^{\blacktriangle}$	$\mathbf{0.45}^{\blacktriangle}$	$\mathbf{0.49}^{\blacktriangle}$	1
BoW Classifier (i.e., Bootstrapping Step)	$0.86^{\blacktriangle}$	$0.42^{\blacktriangledown}$	$0.38^{\triangle}$	$0.34^{\blacktriangle}$	19
RepLab 2013 Baseline	$0.87^{\blacktriangle}$	$0.49^{\blacktriangledown}$	0.32	$0.32^{\blacktriangle}$	21
Filter Keywords (*known-entity*)	0.84	0.67	0.26	0.25	42
Filter Keywords (*unknown-entity*)	$0.50^{\blacktriangledown}$	$0.17^{\blacktriangledown}$	0.29	$0.14^{\blacktriangledown}$	61

on the test set with the same label of the closest tweet in the training set, according to Jaccard's word similarity. The system that obtained the best results at RepLab 2013, POPSTAR [151], is based on supervised learning, where tweets are represented with a variety of features to capture the relatedness of each tweet to the entity. Features are extracted both from the collection (Twitter metadata, textual features, keyword similarity, etc.) and from external resources such as the entity's homepage, Freebase and Wikipedia.

Best RepLab System

Results obtained by the Best RepLab 2013 system shows that, with enough entity-specific training data, it is possible to reach a reasonably accurate filtering system ($0.91^{\blacktriangle}$ accuracy). However, the margin of improvement is still large according to Reliability&Sensitivity. As expected, runs that use previously annotated data from the entity are significantly better than applying approaches oriented to the unknown-entity scenario. Comparing filter keywords in the two scenarios, there is a 40% of significant improvement in terms of accuracy and 44% in terms of $F_1(R,S)$ from the unknown-entity to the known-entity scenario. Compared to filter keywords, the BoW classifier obtains little —but statistically significant— relative improvement in terms of accuracy (3%), but large improvement in terms of $F_1(R,S)$ (28%).[8]

In conclusion, when entity-specific training data is available, it seems more appropriated to use directly the bootstrapping method from the training data, instead of classifying filter keywords.

5.1.5 Conclusion

In this section we tackled the ORM filtering task —defined as a binary classification task— by studying the use of filter keywords: expressions that, if present in a tweet, indicate a high probability that it is related/unrelated to the entity of interest.

In our experiments in the unknown-entity-scenario, automatically discovered filter keywords are able to classify a subset of 30%-60% tweets with an accuracy range of $.75 - .79$.

We defined features that characterize terms in the Twitter dataset, the company's website, ODP, Wikipedia and the searchable Web. We found that (i) term specificity in the tweet stream of each company name is a feature that discriminates between filter keywords and skip terms and (ii) the association between the term and the company website is useful to differentiate positive

[8] Later on, in the experiments shown in the next section, we will see that the BoW classifier can be optimized to reach a 0.9 accuracy and a 0.42 $F_1(R,S)$.

vs. negative filter keywords, specially when it is averaged by considering its most co-occurrent terms. Tweets classified by these filter keywords can be used to feed a supervised machine learning process to obtain a complete classification of all tweets for an overall accuracy of 0.73.

We also found that, on average, the best five optimal keywords can directly classify around 30% of the tweets. Nevertheless, keywords defined by a human by inspecting web search results relevant to the company name only cover 15% of the tweets and accuracy drops to 0.86.

Exploring the nature of filter keywords also led us to the conclusion that there is a gap between the vocabulary characterizing a company in Twitter and the vocabulary associated to the company in its homepage, in Wikipedia, and apparently in the Web at large.

These findings and results at RepLab show that, when entity-specific training data is available—as is the case of the known-entity scenario—it is more appropriated to use a BoW classifier than filter keywords.

5.2 Known-Entity Scenario: Active Learning for Filtering

The work presented in this section has been done in collaboration with Maria-Hendrike Peetz and Maarten de Rijke.

Ideally, a filtering system must perform effectively during all the lifecycle of the monitoring process. This assumption does not always hold, since what is being said about an entity in Twitter—and what is being said about the other interpretations of the ambiguous entity name—has a high probability of changing over time.

A possible way of minimizing this effect is keeping the model updated with fresh manual annotations. However, manual annotations are costly and, therefore, optimizing the trade-off between quality and annotation effort is crucial. Active learning is a machine learning paradigm that directly tackles this problem. In active learning, the learner samples instances, tweets, that should be annotated manually. These annotations then feed back into the model. This sampling can be informed or random. Active learning is especially attractive in this setting of filtering for ORM because it promises to (a) use the analysts' background knowledge and understanding to improve a filtering system, and (b) capture new topics and problems without an exhaustive annotation effort.

In this section, we investigate the suitability of active learning for the ORM filtering task in the *known-entity* scenario. We test the effectiveness of an active learning approach by comparing it to current *passive* supervised learning approaches to the problem at RepLab 2013. We use state-of-the-art active learning approaches for text analysis and analyze the annotation effort needed to outperform over-engineered filtering systems at RepLab. We show that for our filtering task, unlike for other text analysis tasks [18], margin sampling outperforms random sampling.

We aim to answer the following research questions:

Algorithm 1: Active learning approach for the ORM filtering task

```
1  Initialization;
   Data: Training data set
2  begin
3      Initialize model;
4      return initialized model
5  end
6  Training phase;
   input : Current model
   Data: Training data set
7  begin
8      Extract features;
9      Retrain model;
10     return (re)trained model
11 end
12 Test phase;
   Data: Test data set
13 Extract features;
14 repeat
15     Run current model on test data;
16     Select candidate samples for feedback;
17     Collect feedback;
18     Update training and test data sets;
19     Run training phase with updated training data set;
20 until a suitable termination condition;
```

- *How does our active learning approach to the ORM filtering task compare against the state-of-the-art?*
- *Does margin sampling improve effectiveness over random sampling?*
- *Using active learning, how much can the cost of training the initial model be reduced?*

We start by presenting our active learning approach for the filtering task (§5.2.1). Then, we describe our experimental setup in §5.2.2. We analyze our results in §5.2.3 and we conclude in §5.2.4.

5.2.1 Approach

Our approach to entity filtering is based on active learning, a semi-automatic machine learning process interacting with the user for updating the classification model. It selects instances that may maximize the classification performance with minimal effort.

Algorithm 1 sketches the main steps of our active learning approach to entity filtering. First, the instances are represented as feature vectors. Second, the instances from the training dataset are used for building the initial classification model. Third, the test instances are automatically

classified using the initial model. Fourth, we sample candidates to be offered to the user for additional labeling; this step is performed by margin sampling: the instance closest to the class separation is selected. Fifth, the user manually inspects the instance and labels it. The labeled instance is then considered when updating the model. The active learning process is repeated until a termination condition is satisfied.

5.2.1.1 Feature Extraction

The content we work on, tweets, is represented as a Bag-of-Words (BoW), using the vocabulary of the training set and the name of the author of the tweet.[9] The advantage of this approach is that it is not over-engineered and does not make extensive use of additional data or external resources, unlike, e.g., the best performing systems at RepLab 2013 [8]. We use an entity-dependent approach, i.e., we train and test on specific training and test sets for entities.

5.2.1.2 Learning Model

We use Support Vector Machines[10] (SVM) as a classifier. Our active learning approach can be split into the *selection of candidates* for active annotations, *annotation of the candidates* and *updating the model*. Therefore, one iteration of our learning model follows the following three steps: select the best candidate x from the test set T (line 16 in Algorithm 1); annotate the candidate x (line 17 in Algorithm 1); and update the model (line 19 in Algorithm 1). If the resources are available, the training data used to initialize the model can be a large manually annotated (bulk) set of tweets published before the test set. If this training set is available, we call this a *warm* start. Without a warm start, we have a *cold start*, where the initial model selects and classifies tweets randomly; the bulk set of training data facilitates a strong initial model.

Candidate Selection. We consider two sampling methods for selecting candidate tweets for annotation: random and margin sampling.

Random Sampling

For *random sampling*, the candidate instance is sampled without replacement from the training set. There is no informed prior on the instances. Random sampling proved to be effective for other tasks, e.g: building dependency treebanks [18], or clinical text classification [52].

Marging Sampling

The most commonly used sampling method in binary classification problems is uncertainty sampling [157]. We consider a specific uncertainty sampling method especially suitable for support vector machines [168]: *margin sampling*.

Candidates are sampled based on the classification difficulty, thereby selecting candidates where the classifier is less confident. Following this, the candidate x to be annotated from the test set

[9] We also considered alternative representations, but these did not outperform this simple BoW representation. E.g., this BoW representation outperformed a BoW representation that also used linked entities, using 10 fold cross-validation on the training set.

[10] http://scikit-learn.org/stable/modules/svm.html

T is selected as follows:

$$x = \arg\min_{i \in T} |P(C_1 \mid F_x) - P(C_2 \mid F_x)| \,. \tag{5.13}$$

where $P(C_1|F_x)$ and $P(C_2|F_x)$ are the probabilities that the candidate x, as represented by the feature vector F_x, generates the classes C_1 and C_2, respectively. This candidate x is then annotated and used to update the model. In a linear case this means: instances (tweets here) that are closest to the class separation are selected.

Candidate Annotation. In this step of the algorithm, annotations for the selected candidates are collected.

Model Updating. The training of the model is fast. We therefore decided to *retrain* the model with every freshly annotated instance. The instance and its annotation are added to the training set and the model is retrained. The weight for training and new instances is uniform.

As explained earlier, the initial model can be created based on a warm start or a cold start. A warm start enables the algorithm perform an informed, i.e., margin based, sampling from the first iteration onwards. Without a warm start, the algorithm can only select candidates randomly. Furthermore, we do not actually include users in the active learning procedure, but simulate their influence. The following section, on our experimental setup, details our choices for the various options in the algorithm.

5.2.2 Experimental setup

In the following we introduce the settings and parameters needed to evaluate the effectiveness of active learning for the entity filtering task.

We train on the dedicated training set and sample from the entire test set. We compare the effectiveness using different percentages N_{test} of sampled tweets with the effectiveness of two passive supervised learning approaches: the initial model and the best approach at RepLab2013. We compare both sampling methods: random and margin sampling.

Tweets are represented as Bag-of-Words (BoW) with binary occurrence (1 if the word is present in the tweet, 0 if not). The BoW representation was generated by removing punctuation, lowercasing, tokenizing by whitespaces, reducing multiple repetitions of characters (from n to 2) and removing stopwords. Document representation

For our experiments we use Support Vector Machines, using a linear kernel.[11] The penalty parameter C is automatically adjusted by weights inversely proportional to class frequencies. We use the default values for the rest of parameters.

[11] We tested different algorithms (Naïve Bayes, Decision Trees) and this is the one that obtained the best results in terms of the initial (passive learning) model.

Evaluating active learning is difficult and costly, since users should provide feedback on each iteration. In a real-life setting, the selected candidates would be annotated by users. Those labeled instances are then incorporated into the system and not predicted anymore. Without direct users, the usual approach to model the active learning setting is to take the annotations from the test set. This simulates the user feedback; this is what we do.

To ensure comparability with previous approaches (e.g., the approaches evaluated at RepLab 2013) we first consider a batch scenario. After that, we evaluate our active learning approach to entity filtering in a streaming scenario.

Table 5.14 provides an overview over the acronyms used for the runs. The *passive* run is the underlying baseline for active learning; it is based on the training set in the streaming scenario with the warm start and in the batch scenario. In a streaming scenario with cold start it is based

TABLE 5.14: Runs used in our experiments.

Acronym	Ref.	Active	Description
passive	§5.2.1.2	no	Passive learning
best	[151]	no	Best RepLab2013
RS	§5.2.1.2	yes	Random sampling
MS	§5.2.1.2	yes	Margin sampling

on a random sample of $N_{\text{test}}\%$ of the instances in the initial bin. The *best* run is the score for the best performing system at RepLab2013. This score is only available for the batch scenario. *RS* and *MS* are the active learning runs, using random and margin sampling, respectively.

Evaluation Metrics

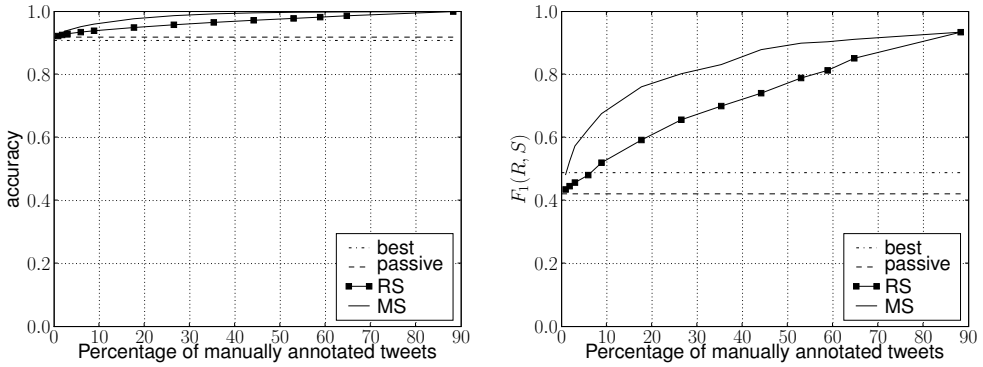
Unless stated otherwise, we use the official evaluation metrics from the RepLab2013 Filtering Subtask: accuracy and the harmonic mean of reliability and sensitivity ($F_1(R, S)$) [11]. Due to the randomness underlying the sampling methods, we report results averaged over 100 runs.

We use the Student's t-test to evaluate the significance of observed differences, using Bonferroni normalisation where appropriate. We denote significant improvements with $^{\blacktriangle}$ and $^{\triangle}$ ($p < 0.01$ and $p < 0.05$, respectively). Likewise, $^{\blacktriangledown}$ and $^{\triangledown}$ denote declines.

5.2.3 Results

We compare our active learning approach to entity filtering versus the state-of-the-art in the filtering task and versus our passive learning baseline. We then compare the effectiveness of two candidate selection methods, random and margin sampling. Finally, we examine to which degree active learning can reduce the cost of the initial training phase.

Fig. 5.7 compares the *best* and *passive* runs with the development of effectiveness of *RS* and *MS* in terms of accuracy (Fig. 5.7a) and $F_1(R, S)$ (Fig. 5.7b). A number of observations are worth making. First, in terms of accuracy, the effectiveness of all the runs is above 0.9, leaving little room for improvement. The *passive* run outperforms the *best* run, but the difference is not statistically significant. In contrast, $F_1(R, S)$ reveals more differences between the runs. *MS* outperforms *best* after inspecting only 2% of the test data (which, on average, corresponds to



(A) accuracy (B) $F_1(R,S)$

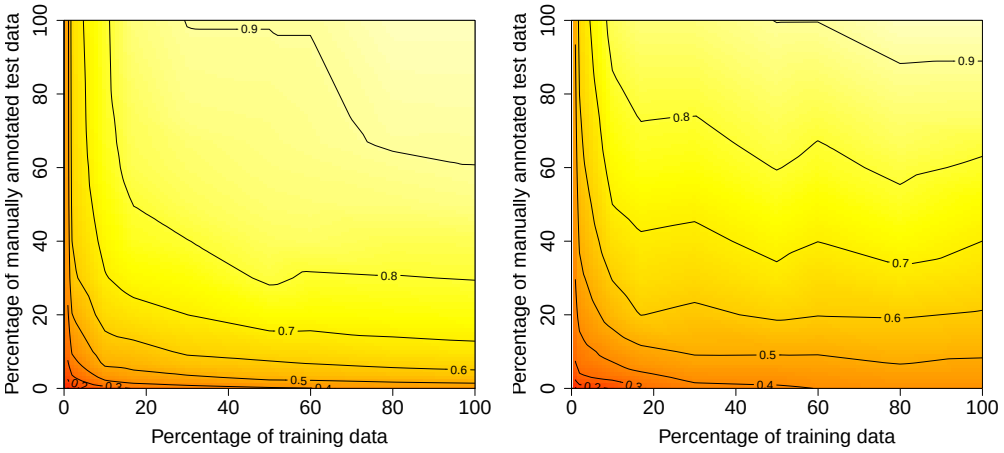
FIGURE 5.7: Accuracy and $F_1(R,S)$ vs. N_{test}.

30 tweets), obtaining a $F_1(R,S)$-score of 0.52 (vs. 0.49). Using $N_{\text{test}} = 5\%$, *MS* significantly outperforms *best*, obtaining an $F_1(R,S)$-score of $0.63^{\blacktriangle}$. On the other hand, *RS* needs more feedback to be able to reach *best*. Using $N_{\text{test}} = 5\%$ it achieves an $F_1(R,S)$-score of 0.48, while using 10% of sampled tweets achieves a score of 0.52. Moreover, differences between *best* and *RS* are statistically significant after inspecting 20% or 30% of the test data, achieving $F_1(R,S)$-scores of $0.59^{\triangle}$and $0.66^{\blacktriangle}$, respectively. The graphs also show *MS* outperforming *RS* consistently. Here, differences begin to be statistically significant from $3\% - 5\%$, with $F_1(R,S)$-scores of $0.57^{\triangle}$, $0.63^{\blacktriangle}$. Interestingly, while *RS* shows a linear behavior, *MS* starts with an exponential gain of effectiveness in terms of $F_1(R,S)$. In terms of $F_1(R,S)$, the effectiveness reached by *RS* after inspecting 10% of the test data can be achieved by *MS* considering only 2%. This amounts to an 80% reduction in cost.

We test an additional run —a training error oracle— to measure the learning capability of our active learning approaches. The initial model is trained on both training and test datasets. Then, we sample from the same test set using *MS* and *RS*. We find both *MS* and *RS* reach optimal accuracy and $F_1(R,S)$ after few iterations ($N_{\text{test}} = 5\%$). Thus, our classifier can actually learn a model of the data.

For compatibility with the RepLab2013 runs, the results above take into account the full test set for evaluation. We also evaluate *MS* and *RS* in a different setting that does not consider tweets that are selected to update the model in the evaluation. Here, *MS* reaches the performance of the best RepLab system (in terms of $F_1(R,S)$) after inspecting little more than 5% of the data (100 tweets); *RS* needs 20% of the test data. Therefore, using *MS* instead of *RS* reduces the cost of the user feedback by around 75%. Remarkably, even though sampled tweets are not considered in the evaluation, *MS* and *RS* continue learning as more feedback is given; again, *MS* is more efficient than *RS*.

We now address the question of cost reduction of the initial model. Initializing any supervised—either passive or active—approach for entity filtering has a cost derived from annotating the initial training data. We look at different percentages of training data used to initialize the model. Figure 5.8 shows heatmaps representing the evolution of $F_1(R,S)$ when considering different percentages of training data (x-axis) and different percentage of sampled test data (y-axis) for



(A) Margin sampling (B) Random sampling

FIGURE 5.8: $F_1(R, S)$-scores with different percentages of training data for the initial model (x-axis) and different percentages of test data for manually inspection during the active learning process (y-axis), using margin (5.8a) or random (5.8b) sampling. Red/-dark and yellow/light correspond to lower and higher $F_1(R, S)$ values, respectively.

MS (5.8a) and *RS* (5.8b). Red/dark and yellow/light correspond to lower and higher $F_1(R, S)$ values, respectively. *MS* needs less training data to obtain competitive $F_1(R, S)$-scores than *RS*. E.g., initializing the model with 10% and inspecting 10% of test data using *MS* achieves an $F_1(R, S)$-score of 0.55, while *RS* achieves only 0.44. Considering only 10% (i.e., 75 tweets) as the initial training set, the effectiveness of emphbest can be reached after 100 tweets ($N_{\text{test}} < 7\%$) using *MS*. In terms of annotation cost, this corresponds to a 75% reduction.

To summarize, our active learning approach requires small amounts of feedback to overperform state-of-the-art filtering systems. Additionally, margin sampling significantly outperforms random sampling. Finally, the cost of training the initial model can be reduced remarkably by using active learning, especially with margin sampling.

5.2.4 Conclusion

We analysed the effect of active learning on entity filtering of tweets. We found that, after only annotating 2% of the sampled test data, we reach 0.52 $F_1(R, S)$-score. We also found that, using active learning with margin sampling, the costs of creating a bulk training set can be reduced by 90% after inspecting 10% of test data. Unlike many other applications of active learning on NLP tasks, margin sampling works better than sampling instances randomly.

In summary, active learning seems to be a useful technique to keep the filtering system updated during the lifecycle of the ORM process.

6 Topic Detection

One of the tasks that motivates monitoring social media consists of knowing what is being said about an entity in Twitter. In this context, information access tools may assist the experts (i) to discover new topics that emerge in Twitter, sometimes as a response to an event that occurs in the real world (e.g., Toyota brake problem); (ii) to keep tracking of the brand identity (i.e., how the entity is characterized in Twitter), and (iii) to facilitate crisis management [1].

Topic Detection vs. Polarity for Reputation

Many ORM tools that are currently on the market opt for offering *sentiment analysis* or *polarity for reputation* as the key features [149, 156, 159, 169, 175]. We believe that, although polarity classification is a fundamental task for ORM, discovering the topics discussed by users might be more crucial than knowing the polarity of the tweets. After some iterations of the monitoring process, experts may know what is affecting positively or negatively the reputation of the entity, and a system that helps discovering unknown topics is more useful. Note that topics may be intrinsically associated with positive/negative polarity (e.g., "Problem with Iphone's antenna"). Therefore, knowing the topics beforehand may help in classifying the polarity of tweets.

While deciding the polarity of a tweet is an atomic task —each tweet can be classified independently— topic detection is a more complex and tedious task: for each tweet, the expert has to decide whether the tweet is talking about one of the already known topics. In that case, the tweet is assigned to the corresponding topic; if not, the expert creates a new topic and assigns the tweet to it. Therefore —and assuming that systems will not be perfect in terms of precision and recall— an automatic system that groups mentions according to topics is more useful than an automatic polarity classifier. In summary, grouping tweets automatically according to topics may help the user to carry out further analysis of the entity mentions.

In the ORM scenario, there is less volume of information —and hence, less redundancy— available than in other topic detection scenarios such as Twitter trending topic detection. Therefore, probabilistic generative approaches —which are a popular strategy to handle topic detection tasks—, might be less appropriate to solve this problem because of data sparsity [134]. Instead, we propose different clustering approaches that make use of Twitter signals and external resources such as Wikipedia to deal with sparsity issues.

[1] https://econsultancy.com/blog/63901-the-top-16-social-media-fails-of-2013

In this chapter we tackle the following research questions:

> **RQ5:** *Wikipedia is a knowledge base that is continuously being updated, and —as we have seen in the previous chapter— can be a relevant source to discover filter keywords automatically. Are the topics discussed about an entity in Twitter represented somehow in Wikipedia?*
>
> **RQ3:** *Can we generalize the idea of "filter keywords" to "cluster keywords", i.e., can we use it for topic detection?*
>
> **RQ6:** *Can Twitter signals be used to improve entity-specific topic detection?*
>
> **RQ9:** *Can previously annotated material be used to learn better topic detection models?*

Before the evaluation framework for the topic detection task was established, we did some preliminary experiments on two tasks that are strongly related to topic detection: real-time summarization of scheduled events (§6.1.1) and entity aspect identification (§6.1.2). Similar to topic detection for ORM, summarizing scheduled events like soccer matches from Twitter streams and aspect identification are focused in a target (e.g., the soccer match or the entity of interest) and the system have to detect relevant sub-events or sub-topics (e.g., goals, red cards or aspects). We will incorporate to our topic detection approaches features to capture those signals that were useful in these preliminary experiments.

After that, we present two approaches submitted to RepLab 2012 and 2013 for tackling the first two research questions for the topic detection task: wikified tweet clustering (§6.2) and cluster keywords (§6.3). Finally, inspired by these two approaches, we tackle the third and fourth research questions by exploring the suitability of combining different Twitter signals to learn a similarity function that groups tweets according to topics (§6.4).

6.1 Preliminary Experiments

In this section we present two preliminary experiments we did before tackling the ORM topic detection task. First, in §6.1.1 we study the capability of making real-time summaries from Twitter data in a more controlled scenario: scheduled events (e.g., soccer matches). Note that identifying salient sub-events that should be included in a real-time summary of an event is a scenario which is between the dense scenario of Twitter trending topics and our *sparse* ORM scenario. Our aim is to validate whether simple statistical methods such as TF.IDF or Kullback-Leibler Divergence (KLD) still holds in an intermediate scenario. On the other hand, a streaming scenario which is purely sequential, (i.e., each sub-event occurs only once and typically one at a time), facilitates the study of temporal signals, which will be included in our topic detection approaches. Second, in §6.1.2.1 we tackle the ORM aspect identification task defined in §4.1. Here, we will try to answer the question "what is being said about a given entity in a certain

tweet stream?" by identifying the specific *aspects* (products, related events, key people, etc.) that people discuss in a given tweet stream about the entity of interest.

6.1.1 Real-Time Summarization of Scheduled Events

The work presented in this section has been done in collaboration with Arkaitz Zubiaga.

Twitter has become a powerful tool to stay tuned to current affairs. It is known that, in particular, Twitter users exhaustively share messages about (all kinds of) events they are following live, occasionally giving rise to related trending topics [203]. The community of users live *tweeting* about a given event generates rich contents describing sub-events that occur during an event (e.g., goals, red cards or penalties in a soccer game). All those users share valuable information providing live coverage of events [22]. However, this overwhelming amount of information makes difficult for the user: (i) to follow the full stream while finding out about new sub-events, and (ii) to retrieve from Twitter the main, summarized information about which are the key things happening at the event. In the context of exploring the potential of Twitter as a means to follow an event, we address the —yet largely unexplored— task of summarizing Twitter contents by providing the user with a summed up stream that describes the key sub-events. We propose a two-step process for the real-time summarization of events —sub-event detection and tweet selection—, and analyze and evaluate different approaches for each of these two steps. We find that Twitter provides an outstanding means for detailed tracking of events, and present an approach that accurately summarizes streams to help the user find out what is happening throughout an event. We perform experiments on scheduled events, where the start time is known. By comparing different summarization approaches, we find that learning from the information seen before throughout the event is really helpful both to determine if a sub-event occurred, and to select a tweet that represents it.

To the best of our knowledge, our work is the first to provide an approach to generate real-time summaries of events from Twitter streams without making use of external knowledge. Thus, our approach might be straightforwardly applied to other kinds of scheduled events without requiring additional knowledge.

6.1.1.1 Dataset

We study the case of tweets sent during the games of a soccer competition. Sports events are a good choice to explore for summarization purposes, because they are usually reported live by journalists, providing a reference to compare with. We set out to explore the *Copa America 2011* championship, which took place from July 1^{st} to 24^{th}, 2011, in Argentina, where 26 soccer games were played. Choosing an international competition with a wide reach enables to gather and summarize tweets in different languages. The official start times for the games were announced in advance by the organization.

During the period of the *Copa America*, we gathered all the tweets that contained any of `#ca2011`, `#copaamerica`, and `#copaamerica2011`, which were set to be the official Twitter hashtags for the competition. For the 24 days of collection, we retrieved 1,425,858 unique tweets sent by 290,716 different users. These tweets are written in 30 different languages, with a majority of 76.2% in Spanish, 7.8% in Portuguese, and 6.2% in English. The tweeting activity of the games considerably varies, from 11k tweets for the least-active game, to 74k for the most-active one, with an average of 32k tweets per game.

In order to define a reference for evaluation, we collected the live reports for all the games given by Yahoo! Sports[2]. These reports include the annotations of the most relevant sub-events throughout a game. 7 types of annotations are included: goals (54 were found for the 26 games), penalties (2), red cards (12), disallowed goals (10), game starts (26), ends (26), and stops and resumptions (63). On average, each game comprises 7.42 annotations. Each of these annotations includes the minute when it happened. We manually annotated the beginning of each game in the Twitter streams, so that we could infer the timestamp of each annotation from those minutes. The annotations do not provide specific times with seconds, and the actual timestamp may vary slightly. We have considered these differences for the evaluation process.

6.1.1.2 Real-Time Event Summarization

Real-Time Event Summarization

We define *real-time event summarization* as the task that provides new information about an event every time a relevant sub-event occurs. To tackle the summarization task, we define a two-step process that enables to report information about new sub-events in different languages. The first step is to identify at all times whether or not a specific sub-event occurred in the last few seconds. The output will be a boolean value determining if something relevant occurred; if so, the second step is to choose a representative tweet that describes the sub-event in the language preferred by the user. The aggregation of these two processes will in turn provide a set of tweets as a summary of the game (see Figure 6.1).

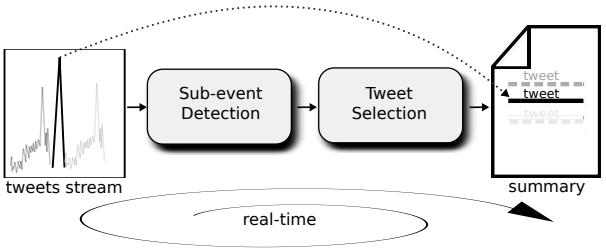


FIGURE 6.1: Two-step process for real-time event summarization.

6.1.1.2.1 First Step: Sub-Event Detection. The first part of the event summarization system corresponds to the sub-event detection. Note that, being a real-time sub-event detection, the system has to determine at all times whether or not a relevant sub-event has occurred,

[2] `http://uk.eurosport.yahoo.com/football/ copa-america/fixtures-results/`

clueless of how the stream will continue to evolve. Before the beginning of an event, the system is provided with the time that it starts, as scheduled in advance, so the system knows when to start looking for new sub-events. With the goal of developing a real-time sub-event detection method, we rely on the fact that relevant sub-events trigger a massive tweeting activity of the community. We assume that the more important a sub-event is, the more users will tweet about it almost immediately. This is reflected as peaks in the histogram of tweeting rates (see Figure 6.2 for an example of a game in our dataset). In the process of detecting sub-events, we aim to compare 2 different ideas: (i) considering only sudden increase with respect to the recent tweeting activity, and (ii) considering also all the previous activity seen during a game, so that the system learns from the evolution of the audience. We compare the following two methods that rely on these 2 ideas:

1. **Increase:** this approach was introduced by Zhao et al. [199]. It considers that an important sub-event will be reflected as a sudden increase in the tweeting rate. For time periods defined at 10, 20, 30 and 60 seconds, this method checks if the tweeting rate increases by at least 1.7 from the previous time frame for any of those periods. If the increase actually occurred, it is considered that a sub-event occurred. A potential drawback of this method is that not only outstanding tweeting rates would be reported as sub-events, but also low rates that are preceded by even lower rates.

2. **Outliers:** we introduce an outlier-based approach that relies on whether the tweeting rate for a given time frame stands out from the regular tweeting rate seen so far during the event (not only from the previous time frame). We set the time period at 60 seconds for this approach. 15 minutes before the game starts, the system begins to learn from the tweeting rates, to find out what is the approximate audience of the event. When the start time approaches, the system begins with the sub-event detection process. The system considers that a sub-event occurred when the tweeting rate represents an outlier as compared to the activity seen before. Specifically, if the tweeting rate is above 90% of all the previously seen tweeting rates, the current time frame will be reported as a sub-event. This threshold has been set a priori and without optimization. The outlier-based method incrementally learns while the game advances, comparing the current tweeting rate to all the rates seen previously. Different from the increase-based approach, our method presents the advantages that it considers the specific audience of an event, and that consecutive sub-events can also be detected if the tweeting rate remains constant without increase. Accordingly, this method will not consider that a sub-event occurred for low tweeting rates preceded by even lower rates, as opposed to the increase-based approach.

For evaluation purposes, we compare system outputs to the manually annotated gold standard created from the collected live reports. Since the annotations on the reference are limited to minutes, we round down the outputs of the systems to match the reference. Also, the timestamps annotated for the reference are not entirely precise, and therefore we accept as a correct guess an automatic sub-event detection that differs by at most one minute from the reference.

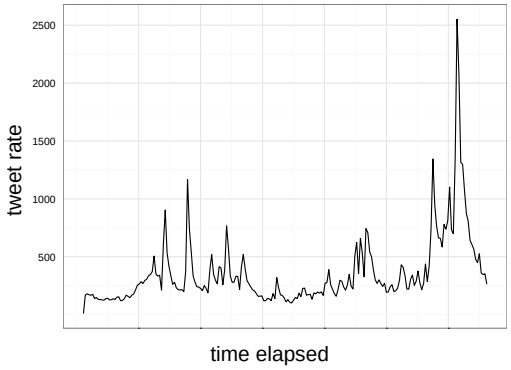


FIGURE 6.2: Sample histogram of tweeting rates for a soccer game (Argentina vs Uruguay), where several peaks can be seen.

This evaluation method enables us to compare the two systems to infer which of them performs best. Table 6.1 shows the precision (P), recall (R) and F-measure (F_1) of the automatically detected sub-events with respect to the reference, as well as the average number of sub-events detected per game (#). Our outliers approach clearly outperforms the baseline, improving both precision (75.8% improvement) and recall (3.7%) for an overall 40% gain in F_1. At the same time, the compression rate for the outliers approach almost doubles that of the baseline (56.4%). From the average of $32k$ tweets sent per game, the summarization to 25.6 tweets represents a drastic reduction to only 0.079% of the total. Keeping the number of sub-events small while effectiveness improves is important for a summarization system in order to provide a concise and accurate summary. The outperformance of the outlier-based approach shows the importance of taking into account the audience of a specific game, as well as the helpfulness of learning from previous activity throughout a game.

TABLE 6.1: Evaluation of sub-event detection approaches.

	P	**R**	**F_1**	**#**
Increase	0.29	0.81	0.41	45.4
Outliers	**0.51**	**0.84**	**0.63**	**25.6**

6.1.1.2.2 Second Step: Tweet Selection. The second and final part of the summarization system is the tweet selection. This second step is only activated when the first step reports that a new sub-event occurred. Once the system has determined that a sub-event occurred, the tweet selector is provided with the tweets corresponding to the minute of the sub-event. From those tweets, the system has to choose one as a representative tweet that describes what occurred. This tweet must provide the main information about the sub-event, so the user understands what occurred and can follow the event. Here we compare two tweet selection methods, one relying only on information contained within the minute of the sub-event, and another considering the knowledge acquired during the game. We test them on the output of the outlier-based sub-event detection approach described above, as the approach with best performance for the first step.

To select a representative tweet, we get a ranking of all the tweets. To do so, we score each tweet with the sum of the values of the terms that it contains. The more representative are the terms contained in a tweet, the more representative will be the tweet itself. To define the values of the terms, we compare two methods: (i) considering only the tweets within the sub-event (to give highest values to terms that are used frequently within the sub-event), and (ii) taking into account also the tweets sent before throughout the game, so that the system can make a difference from what has been the common vocabulary during the event (to give highest values to terms that are especially used within the minute and not so frequently earlier during the event). We use the following well-known approaches to implement these two ideas:

1. **TF:** each term is given the value of its frequency as the number of occurrences within the minute, regardless of its prior use.
2. **KLD:** we use the Kullback-Leibler Divergence [100] (see Equation 6.1) to measure how frequent is a term w within the sub-event (H), but also considering how frequent it has been during the game until the previous minute (G). Thus, KLD will give a higher weight to terms frequent within the minute that were less frequent during the game. This may allow to get rid of the common vocabulary all along the game, and rather provide higher rates to specific terms within the sub-event.

$$D_{\mathrm{KL}}(H\|G) = H(w)\log\frac{H(w)}{G(w)} \tag{6.1}$$

Ranking of Representative Tweets

With these two approaches, the sum of values for terms contained in each tweet results in a weight for each tweet. With weights given to all tweets, we create a ranking of tweets sent during the sub-event, where the tweet with highest weight ranks first. We create these rankings for each of the languages we are working on. The tweet that maximizes this score for a given language is returned as the candidate tweet to show in the summary in that language. The two term weighting methods were applied to create summaries in three different languages: Spanish, English, and Portuguese. We test them on the output of the outlier-based sub-event detection approach described above, as the approach with best performance for the first step. Thus, we got six summaries for each game, i.e., TF and KLD-based summaries for the three languages. These six summaries were manually evaluated by comparing them to the reference. Table 6.2 shows some tweets included in the KLD-based summary in English.

Manual Evaluation

In the manual evaluation process, each tweet in a system summary is classified as correct if it can be associated to a sub-event in the reference and is descriptive enough (note that there might be more than one correct tweet associated to the same sub-event). Alternatively, tweets are classified as novel (they contain relevant information for the summary which is not in the reference) or noisy. From these annotations, we computed the following values for analysis and evaluation: (i) recall, given by the ratio of sub-events in the reference which are covered by a correct tweet in the summary; and (ii) precision, given by the ratio of correct + novel tweets from a whole summary (note that redundancy is not penalized by any of these measures).

TABLE 6.2: Example of some tweets selected by the (outliers+KLD) summarization system, compared with the respective comments narrated on Yahoo! Sports.

Sub-event	Selected Tweet	Narrator's Comment
Game start	RT @user: Uruguay-Argentina. The Río de la Plata classic. The 4th vs the 5th in the last WC. History doesn't matter. Argentina must win. #ca2011	The referee gets the game under way
Goal	Gol! Gol! Gol! de Perez Uruguay 1 vs Argentina 0 Such a quick strike and Uruguay is already on top. #copaamerica	GOAL!! Forlan's free kick is hit deep into the box and is flicked on by Caceres. Romero gets a hand on it but can only push it into the path of Perez who calmly strokes the ball into the net.
Goal	Goooooooooooooooal Argentina ! Amazing pass from Messi, Great positioning & finish from Higuain !! Arg 1 - 1 Uru #CopaAmerica	GOAL!! Fantastic response from Argentina. Messi picks the ball up on the right wing and cuts in past Caceres. The Barca man clips a ball over the top of the defence towards Higuain who heads into the bottom corner.
Red card	Red card for Diego Pérez, his second yellow card, Uruguay is down to 10, I don't know if I would have given it. #CopaAmérica2011	You could see it coming. How stupid. Another needless free kick conceded by Perez and this time he is given his marching order. He purposely blocks off Gago. Uruguay have really got it all to do now.
Red card	#ca2011 Yellow for Mascherano! Double yellow! Adios! 10 vs 10! Mascherano surrenders his captain armband!	It's ten against ten. Macherano comes across and fouls Suarez. He's given his second yellow and his subsequent red.
Game stop (full time)	Batista didn't look too happy at the game going to penalties as the TV cut to hit at FT, didn't appear confident #CA2011	The second half is brought to an end. We will have extra time.
Game end	Uruguay beats Argentina! 1-1 (5-4 penalty shoot out)! Uruguay now takes on Peru in Semis. #copaamerica	ARGENTINA 4-5 - URUGUAY WIN. Caceres buries the final penalty into the top right-hand corner.

Table 6.3 shows recall values as the coverage of the two approaches over each type of sub-event, as well as the macro-averaged overall values. These results corroborate that simple state-of-the-art approaches like TF and KLD score outstanding recall values. Nevertheless, KLD shows to be slightly superior than TF for recall. Regarding the averages of all kinds of sub-events, recall values are near or above 80% for all the languages. It can also be seen that some sub-events are much easier to detect than others. It is important that summaries do not miss the fundamental sub-events. For instance, all the summaries successfully reported all the goals and

TABLE 6.3: Recall of reported sub-events for summaries in Spanish (es), English (en), and Portuguese (pt).

		es	**en**	**pt**
Goals (54)	TF	0.98	0.98	0.98
	KLD	**1.00**	**1.00**	**1.00**
Penalties (2)	TF	**1.00**	**0.50**	**1.00**
	KLD	**1.00**	**0.50**	**1.00**
Red Cards (12)	TF	0.75	0.75	40.83
	KLD	**0.92**	**0.92**	**1.00**
Disallowed Goals (10)	TF	**0.40**	**0.50**	**0.40**
	KLD	**0.40**	**0.50**	0.30
Game Starts (26)	TF	0.73	0.74	0.79
	KLD	**0.84**	**0.79**	**0.83**
Game Ends (26)	TF	**1.00**	**1.00**	**1.00**
	KLD	**1.00**	**1.00**	**1.00**
Game Stops & Resumptions (63)	TF	0.62	**0.60**	0.57
	KLD	**0.68**	**0.60**	**0.59**
Overall	TF	0.79	0.74	0.78
	KLD	**0.84**	**0.77**	**0.82**

all the game ends, which are probably the most emotional moments, when users extremely coincide sharing. However, other sub-events like game stops and resumptions, or disallowed goals, were sometimes missed by the summaries, with recall values near 50%. This shows that some of these sub-events may not be that shocking sometimes, depending on the game, so fewer users share about them, and therefore are harder to find by the summarization system. For instance, one could expect that users would not express high emotion when a boring game with no goals stops for half time. Likewise, this shows that those sub-events are less relevant for the community. In fact, from these summaries, users would perfectly know when a goal is scored, when it finished, and what is the final result.

TABLE 6.4: Precision of summaries in Spanish (es), English (en), and Portuguese (pt).

	es	**en**	**pt**
TF	0.79	0.74	0.79
KLD	**0.84**	**0.79**	**0.83**

Table 6.4 shows precision values as the ratio of useful tweets for the three summaries generated in Spanish, English and Portuguese. The results show that a simple TF approach is relatively good for the selection of a representative tweet, with precision values above 70% for all three languages. As for recall values, KLD does better than TF, with precision values near or above 80%. This shows that taking advantage of the differences between the current sub-event and tweets shared before considerably helps in the tweet selection. Note also that English summaries reach

0.79 precision even if the tweet stream is, in that case, an order of magnitude smaller than their Spanish counterpart, suggesting that the method works well at very different tweeting rates.

6.1.1.3 Lessons Learned

In this preliminary experiments, we have studied a two-step summarization approach that, without making use of external knowledge, identifies relevant sub-events in soccer games and selects a representative tweet for each of them. Using simple text analysis methods such as KLD, our system generates real-time summaries with precision and recall values above 80% when compared to manually built reports. As in our filtering experiments, term specificity is a key feature for detecting salient terms which help to select representative tweets for a sub-event. One of our ORM topic detection approaches —cluster keywords— will consider this information as one of the features to detect the keywords that characterize a topic.

Term Specificity

The fact that users tweet at the same time, with overlapping vocabulary, helps not only detecting that a sub-event occurs, but also selecting a representative tweet to describe it. Our study also shows that considering all previous information seen during the event is really helpful to this end, yielding superior results than taking into account just the most recent activity. The time when the tweets were published plays an important role for detecting sub-events in a tweet stream. Thus, it is worth exploring time signals in our ORM topic detection scenario.

Time Signals

The activity for the soccer games studied in this work varies from 11k to 74k tweets sent, showing that, regardless of the audience tweeting about an event —as long as there is enough information—, our method effectively reports the key sub-events occurred during a game. Finally, all of the most relevant types of sub-events, such as goals and game ends, are reported almost perfectly. Note that our method does not rely on any external knowledge about soccer events (except for the schedule time to begin), so it can be straightforwardly applied to other kinds of events.

Although term specificity and time signals can be used in our ORM topic detection scenario, considering a tweet stream that include only thousands of tweets, we will have to deal with a more sparse scenario than the summarization of soccer matches in Twitter.

6.1.2 Identifying Entity Aspects

> *The work presented in this section has been done in collaboration with Edgar Meij, Maarten de Rijke, Andrei Oghina, Minh Thuong Bui and Mathias Breuss.*

Before tackling the ORM topic detection problem —which was defined as a clustering—, we will try to answer the question “what is being said about a given entity in a certain tweet stream?” by identifying the specific *aspects* that people discuss. These aspects can be shown to the user as a tag-cloud, which summarizes the main aspects discussed in the tweet stream. Aspects refer to “hot” keywords that occurs in a stream of tweets about an entity and are of particular interest

for companies. Aspects can cover a wide range of issues and include (but are not limited to) company products, key people, other entities, services, and events.

Term Ranking Task

Given a stream of microblog posts related to an entity, we are interested in a ranked list of aspects that are being discussed with respect to the entity of interest. We formulate our scenario as an Information Retrieval (IR) task, where the goal is to provide a ranking of terms, extracted from tweets that are relevant to the company.[3] We compare different methods that address this task with the goal of analyzing how state-of-the art IR approaches perform, and to see how methods tailored specifically to identifying opinion targets perform. In our experiments, we use the dataset described in §4.2.4. Based on the WePS-3 ORM dataset, we run our aspect identification methods over the related tweets and the final gold standard is then created using a pooling methodology [70]: the 10 highest ranking terms from each method are merged and randomized. Then, human assessors consider each term and determine whether it is relevant in the context of the company or not.

6.1.2.1 Identifying Entity Aspects

We evaluate four models for identifying aspects, given an entity e and a stream of microblog posts related to that entity. All models work according to the same principle: comparing a pseudo-document D_e built from entity-specific tweets with a background corpus C. This comparison allows us to score a term w using a function $s(w, D_e, C)$.

We compare four methods for identifying entity aspects: TF.IDF, the Log-Likelihood ratio (LLR) [47], Parsimonious Language Models (PLM) [71] and an opinion-oriented method (OO) [86] that extracts targets of opinions to generate a topic-specific sentiment lexicon; we use the targets selected during the second step of this method.

We now describe how the scoring function is computed by each method. As usual, $tf(w, D_e)$ denotes the term frequency of term w in pseudo-document D; $cf(w)$ denotes the term frequency in the collection C and $df(w)$ denotes the total number of pseudo-documents $D_i \in C$ in which the term w occurs at least once.

- **TF.IDF:**

$$s(w, D_e, C) = tf(w, D_e) \cdot \log \frac{N}{df(w)} \tag{6.2}$$

where

$$N = \text{number of pseudo-documents } D_i \text{ in } C$$

and

$$df(w) = |\{D_i \in C | tf(w, D_i) > 0\}|$$

- **Log-Likelihood Ratio (LLR):**

$$s(w, D_e, C) = 2 \cdot ((a \cdot \log(\frac{a}{E_1})) + (b \cdot \log(\frac{b}{E_2}))) \tag{6.3}$$

[3] We only consider unigrams. When a unigram is an obvious constituent of a larger, relevant aspect it is considered relevant.

where

$$E_1 = \frac{c \cdot (a+b)}{c+d} \quad E_2 = \frac{d \cdot (a+b)}{c+d}$$
$$a = tf(w, D_e) \quad b = cf(w)$$
$$c = \sum_i tf(w_i, D_e) \quad d = \sum_i cf(w_i)$$

- **Parsimonious Language Models (PLM):**

$$s(w, D_e, C) = P(w|D_e) \text{ (when model converges)} \tag{6.4}$$

PLM is an Expectation Maximization algorithm defined by the following steps:

$$\text{E-step: } e_w = tf(w, D_e) \cdot \frac{\lambda \cdot P(w|D_e)}{(1-\lambda) \cdot P(w|C) + \lambda \cdot P(w|D_e)}, \lambda = 0.1$$

$$\text{M-step: } P(w|D_e) = \frac{e_w}{\sum_w e_w}$$

The model is initialized with the following $P(w|D_e)$ and $P(w|C)$ values:

$$\text{initial } P(w|D_e) = \frac{tf(w, D_e)}{\sum_i tf(w_i, D_e)} \quad \text{initial } P(w|C) = \frac{cf(w)}{\sum_i cf(w_i)}$$

- **Opinion-Oriented (OO):**

$$s(w, D_e, C) = \chi^2(target(w, D), target(w, C)) \tag{6.5}$$

where

$$\chi^2(o, e) = \frac{(o-e)^2}{e}$$
$$target(w, D_e) = \text{ freq. of potential target } w \text{ in tweets } D_e$$

and

$$target(w, C) = \text{ freq. of potential target } w \text{ in background } C$$

6.1.2.2 Experiments

Here, we focus on the task of identifying aspects and base our annotations on the data used for the WePS-3 ORM Task [7]. Here, the task that participating systems need to solve is to decide which tweets containing a company name are actually related to the company. In total, 99 companies are used for testing, with around 450 tweets (manually annotated for relevance) on average for each company. In our experiments we only consider the tweets that are related to a company, adding up a total of 94 companies and 17,775 tweets with an average of 177 tweets per company. We lowercase, remove punctuation, and tokenize the tweets. We do not perform stopword removal or stemming, but only keep terms occurring at least 5 times in the corpus to remove noisy terms.

We evaluate the methods for ranking aspects using a pooling methodology [176]; the 10 highest ranked terms from each method are merged and randomized. Then, three human assessors consider each term and determine relevance in the context of the company; relevant aspects can include terms from compound words, mentions, or hashtags and should provide insight into the hot topics discussed regarding a company. We compute the inter-annotator agreement using both *Cohen's* and *Fleiss' kappa* and compare the annotators' pairwise and overall. All obtained kappa values are above 0.6, indicating a substantial agreement.

Table 6.5 (upper part) shows the results of all methods for identifying aspects. For evaluation, we consider the following standard IR metrics [118]: Mean Average Precision (MAP), Precision at k (P@5 and P@10, with $k = 5$ and $k = 10$, respectively) and Mean Reciprocal Rank (MRR). Since TF.IDF is the simplest approach, it is considered as the baseline. We use Student's t-test to test for statistical significance and indicate a significant difference with $\alpha = 0.01$ using $^{\blacktriangle}$(or $^{\blacktriangledown}$) and $^{\triangle}$(or $^{\triangledown}$) for $\alpha = 0.05$.

First, we observe that TF.IDF is a strong baseline. In terms of precision, it significantly outperforms PLM and OO, while differences between TF.IDF and LLR are not significant. The results for OO are much lower than for the other methods. Since terms that are (part of) the name of the entity were also annotated as aspects, and these terms are very frequent in the tweets related to the entity, they are often in the top of the ranking returned by the methods. This explains the high MRR values in the results.

When manually inspecting the results, we observe that the results for the frequency-based methods (TF.IDF, LLR and PLM) are very similar, while OO tends to return more subjective terms as aspects (e.g., *haha*, *pls*, *xd*, *safety*, *win*), probably because of errors in the syntactic parsing of tweets. Moreover, this approach has more difficulty to filter out generic terms (e.g., *new*, *use*, *today*, *come*).

Most of the true aspects are nouns (89.72%). Hence, in addition to the preprocessing steps detailed above, we experiment with applying a part-of-speech filter and only consider terms tagged as nouns (Penn Treebank's N* tags) [96]. Table 6.5 (lower part) shows the results when non-noun terms have been filtered out from the vocabulary. For all methods, MAP and precision values are slightly higher than in the all words condition: considering only nouns helps to identify aspects. Interestingly, the relative order of the approaches (as determined by the scores they achieve) changes with respect to the upper part. PLM now outperforms TF.IDF for two of the four metrics (significantly so for P10).

6.1.2.3 Lessons Learned

In this experiments, we addressed the task of identifying *aspects* that people discuss in a stream of microblog posts related to an entity, a problem that is strongly related to the ORM topic detection task. We modeled this task as a ranking problem and compared IR techniques and opinion target identification methods for automatically identifying aspects. We used a pooling methodology to evaluate the methods. Simple statistical methods such as TF.IDF are a strong baseline for the task. As in the real-time summarization scenario, term specificity is an effective

Term Specificity

TABLE 6.5: Aspect identification results. Best results per experimental condition in boldface; significant changes are w.r.t. the TF.IDF All words baseline.

	Method	MAP	P@5	P@10	MRR
All words	TF.IDF	**0.3953**	**0.6957**	**0.6426**	0.7908
	LLR	0.3879	**0.6957**	0.6309	**0.7979**
	PLM	0.3685$^{\blacktriangledown}$	0.6723$^{\triangledown}$	0.6096$^{\blacktriangledown}$	**0.7979**
	OO	0.1537$^{\blacktriangledown}$	0.4596$^{\blacktriangledown}$	0.2915$^{\blacktriangledown}$	0.7021
Noun filter	TF.IDF	0.4015	**0.7213**	0.6436	**0.7979**
	LLR	0.4055	0.7128	0.6511	**0.7979**
	PLM	**0.4097**	0.7106	**0.6617**$^{\triangle}$	**0.7979**
	OO	0.1635$^{\blacktriangledown}$	0.4809$^{\blacktriangledown}$	0.3000$^{\blacktriangledown}$	0.7021

approach for identifying aspects, and may effectively capture relevant signals in our ORM topic detection task. We will use pseudo-document TF.IDF as a feature to represent cluster keywords, as well as a weighting function to compute a similarity function between tweets.

Moreover, it is difficult to identify aspects by extracting opinion targets —deep linguistic processing such as dependency analysis are difficult to apply to tweets— mainly because the language used in tweets is often non-standard, hampering the performance of such techniques. We believe that there is a need of Twitter specific NLP tools —which are yet immature [45]— for an effective use of opinion target techniques in these ORM scenarios.

We now turn to the Topic Detection task in the ORM scenario. We start by analyzing a clustering approach that groups wikified tweets (§6.2); then we study the use of cluster keywords (§6.3) and finally we explore the suitability of combining different Twitter signals to learn a similarity function that groups tweets according to topics (§6.4).

6.2 Wikified Tweet Clustering

As we have seen in §3.1, a common approach to enrich the context of tweets consists of applying NED techniques that link n-grams to entities and concepts to a knowledge base such as Wikipedia [51, 112, 126, 127]. Besides, in Chapter 5 we have seen that Wikipedia is one of the most useful sources to represent the entity of interest on the Web.

A natural question that may come to mind is whether entity linking could be useful for detecting key concepts and entities involved in the topics discussed about the entity of interest, and therefore, applied effectively for topic detection. In this section we present a clustering approach that groups wikified tweets via entity linking (§6.2.1), which was submitted as a topic detection system for the last two RepLab editions (§6.2.2).

TABLE 6.6: Examples of tweets represented with Wikipedia concepts linked by using *commonness* probability.

Original Tweet	**Wikified Tweet Representation**
`Les presento el nuevo producto de la marca Apple...El iMeestaba-jando. pic.twitter.com/JPdR5Oct`	Brand, Product (business), Apple Inc.
`Apple ya ha comenzado con iOS 6 en el iPad 3 !!!! http://goo.gl/fb/aY0cO #rumor #ios6 #ipad #ios #ipad3g #apple`	IOS, Rumor, Apple Inc., IPad
`Server logs show Apple testing iPads with iOS 6, possible Retina Displays http://bit.ly/ysaFUA (via @appleinsider)`	Software testing, Retina, IOS, Display device, Apple Inc., IPad

6.2.1 Approach

The wikified tweet clustering approach relies on the hypothesis that tweets sharing concepts or entities defined in a knowledge base —such as Wikipedia— are more likely to talk about the same topic than tweets with none or less concepts in common.

The *wikified* tweet representation consists of the set of Wikipedia articles obtained by linking n-grams in the tweet content to Wikipedia. For wikifing tweets, we adopt an entity linking approach to gather Wikipedia entries that are semantically related to a tweet: the *commonness* probability [126]—based on the intra-Wikipedia hyperlinks— which computes the probability of a concept/entity c being the target of a link with anchor text q in Wikipedia by:

Tweet Wikification

$$commonness(c, q) = \frac{|L_{q,c}|}{\sum_{c'} |L_{q,c'}|} \tag{6.6}$$

where $L_{q,c}$ denotes the set of all links with anchor text q and target c.

As the dataset contains tweets in two languages, we use both (Spanish and English) Wikipedia dumps. Spanish Wikipedia articles are then translated to the corresponding English Wikipedia article by following the inter-lingual links, using the Wikimedia API[4].

Translation

Tweet Clustering After tweets were wikified, the Jaccard similarity between the sets of entities linked to the tweets is used to group them together: given two tweets d_1 and d_2 represented by the set of Wikipedia entities C_1 and C_2 respectively, if Jaccard$(C_1, C_2) > \alpha$, then d_1 and d_2 are grouped together to the same cluster. Note that this topic detection approach is applicable to both *unknown-entity* and *known-entity* scenario, since the α threshold is the only parameter that can be optimized by supervision —i.e., using training data, either entity-specific or not.

Table 6.6 shows some tweets that have been *wikified* using the method described above.

[4] http://www.mediawiki.org/wiki/API:Properties

Retweets and tweets generated automatically (e.g., by clicking "share" buttons in news or blog posts, tweets generated by third services like Foursqurare, etc.) are frequent in Twitter data that was crawled by querying about an entity, like in our ORM topic detection scenario. Moreover, tweets sharing a high percentage of words —near-duplicate tweets— are very likely to belong to the same cluster. Therefore, near-duplicates tweets that generate trivial clustering relationships will be treated independently in our experiments.

Trivial Clustering Relationships

6.2.2 Experiments

We now describe the experiments of the wikified tweet approach to tackle the topic detection task in RepLab 2012 and RepLab 2013. In both evaluations, we report the scores obtained with the official metrics used to evaluate the topic detection task: Reliability and Sensitivity [11], which, in the context of clustering, are equivalent to BCubed Precision and BCubed Recall, respectively [10].

6.2.2.1 RepLab 2012

In the wikified tweet clustering system submitted to RepLab 2012, tweet clustering is done by computing entity overlap, using a threshold of 40%, i.e., tweets sharing more than 40% of Wikipedia entities are grouped together). This threshold was empirically defined upon the RepLab 2012 trial data. In the RepLab 2012 evaluation campaign, this system corresponds to the `replab2012_monitoring_UNED_1` run.

Trivial clustering relationships between near-duplicate tweets are resolved a-priori as follows. Tweets with a term overlap higher than 70% are grouped together and removed from the input, except of one representative tweet for each of the trivial clusters. After running the system, we merge the output with the a-priori trivial clustering, i.e., each trivial cluster is joined to the cluster in the system output that contains its representative tweet).

RepLab 2012 Baseline: Term Jaccard + HAC

The RepLab 2012 official baseline consists of a Hierarchical Agglomerative Clustering (HAC) algorithm that uses single linkage over Jaccard word distances. Contrary to systems —which have to determine the used thresholds beforehand— different stopping thresholds were used for the baseline. Therefore, the baseline results can be represented as a Reliability/Sensitivity curve.

Figure 6.3 shows results as macro-averaged R&S in the RepLab 2012 test dataset. Reliability (y-axis), Sensitivity (x-axis) and $F_1(R,S)$ (dot size and number) are plotted. Note that $F_1(R,S)$ is not the harmonic mean of the average R&S, but the average of the harmonic mean for each test case (the 31 entities in the test collection). Each dot in the `Term Jaccard + HAC` curve represents the output of the HAC algorithm at different similarity thresholds (in percentiles). A lower similarity threshold gives larger clusters, increasing Sensitivity (BCubed Recall) at the expense of Reliability (BCubed Precision).

The figure shows the wikified tweet clustering result in comparison to the `Term Jaccard + HAC` baseline —which is a strong baseline, as it got the highest scores at RepLab 2012. Instead of

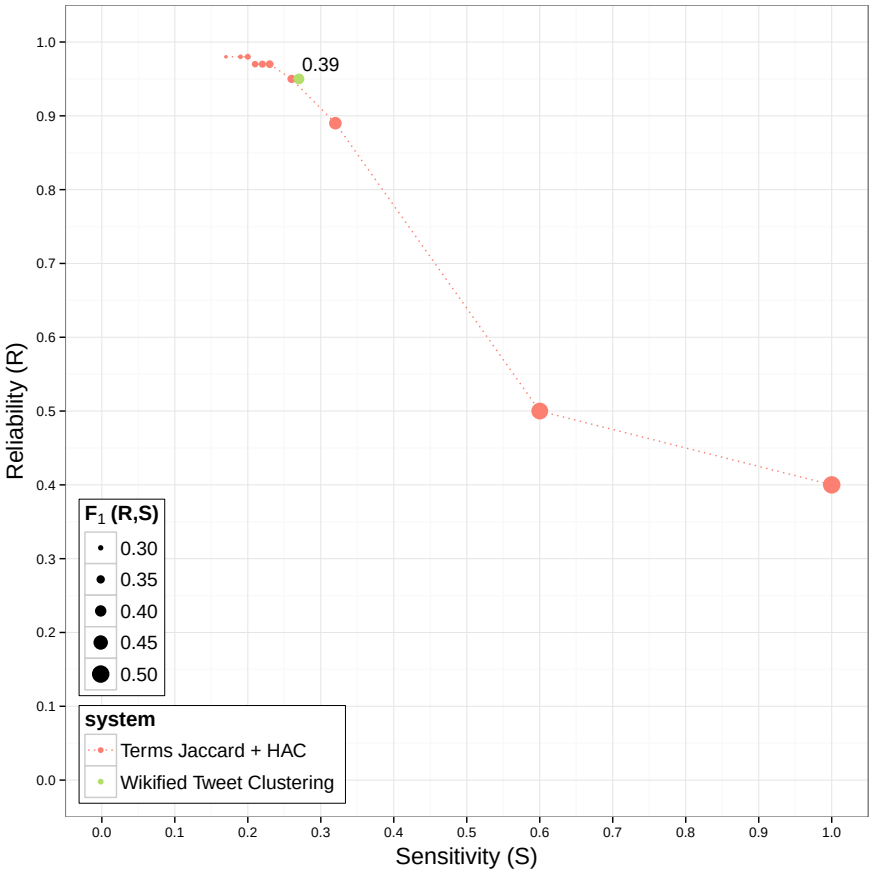


FIGURE 6.3: Wikified Tweet Clustering compared to `Term Jaccard + HAC` baseline in the RepLab 2012 Topic Detection Task.

the official run submitted to the evaluation campaign, we report a subsequent run that fixes some bugs, e.g., the translation of Spanish to Wikipedia concepts English. The wikified tweet clustering system achieves 0.39 $F_1(R,S)$, having 0.95 Reliability and 0.27 Sensitivity. The figure shows that, in the RepLab 2012, the approach performs similarly to the baseline. The baseline with 0% stopping threshold obtains 0.5 $F_1(R,S)$. It assigns all the tweets to a single cluster, corresponding to the so-called *all-in-one* system. This system reaches perfect recall, and is relatively high in precision for entities wich have associated only few topics. More precisely, it achieves a Reliability score above 0.95 in five of the 24 test cases (slightly more than 20%).

Note that the RepLab 2012 dataset contains retweets and near-duplicate tweets, which generate clustering relationships that are easy to detect with approaches similar to the baseline based on word overlap. Results achieved by the baseline with stopping threshold of 50% indicate this effect: on average, almost all the tweets that have a term Jaccard similarity above 0.5 belong to the same cluster (0.97 Reliability), and that corresponds to 20% of the clustering relationships between tweets (0.22 Sensitivity).

6.2.2.2 RepLab 2013

The same approach was submitted as a run for the topic detection task at RepLab 2013. Here, the tweet clustering step is carried out by computing Jaccard similarity between the set of Wikipedia concepts representing the tweets and using a threshold for the Jaccard similarity $\alpha = 0.2$. This threshold has been empirically optimized using the training dataset[5]. We tested both using entity-specific or general training data, obtaining no substantial differences. In the official RepLab 2013 evaluation, this system corresponds to the `UNED_ORM_topic_det_2` run.

RepLab 2013 Baseline

The official RepLab 2013 baseline consists of a memory-based supervised learning approach that assigns each tweet in the test dataset to the same topic of the closest tweet in the training set, according to term Jaccard similarity. This baseline obtains a 0.22 $F_1(R,S)$. On the other hand, the `Term Jaccard + HAC` baseline obtained significantly better results. Thus, we consider this stronger baseline to compare with. Analogously to RepLab 2012, Figure 6.4 compares the results of the wikified tweet clustering approach to the `Term Jaccard + HAC` curve in terms of Reliability and Sensitivity. Note that R, S and $F_1(R,S)$ for the RepLab systems reported are different from the official scores [8], because we are excluding unrelated tweets from our evaluation, and we are also excluding near-duplicates. Nevertheless, all systems benefit similarly from the normalization and it does not produce any change in the official ranking.

The wikified tweet clustering approach obtained 0.47 Reliability, 0.38 Sensitivity and $F_1(R,S)$ equal to 0.37. Although the wikified tweet clustering was the approach that performed best in the competition —getting the highest scores in all the metrics— the figure shows that our approach is under the curve obtained by the `Term Jaccard + HAC` baseline. Although both Reliability and Sensitivity are far from tolerable by an end user, it is worth noting that the inter-annotator agreement for the topic detection task in RepLab 2013 is 0.48 $F_1(R,S)$—which is lower than other information access tasks.[6]

Different from RepLab 2012, the results of RepLab 2013 show that it is more difficult to get high Reliability and Sensitivity scores in the RepLab 2013 dataset. For instance, the one-in-one baseline —which is equivalent to the dot with highest Reliability in the `Term Jaccard + HAC` curve— only obtains 0.05 of Sensitivity. One main difference between the two collections is the number of tweets per each entity or test case. While RepLab 2012 has only 220 tweets per test case on average, the RepLab 2013 comprises 1,500 non near-duplicate tweets per entity. Therefore, the results of RepLab 2013 are probably more representative than the results in the pilot task of RepLab 2012.

Unknown-Entity vs. Known-Entity Scenario

Contrary to the filtering task, where the use of entity-specific training data is crucial for obtaining an effective filtering system, it is not clear that its use in topic detection systems has substantial benefits in terms of effectiveness. The little effect of optimizing the stopping threshold per entity in our wikified tweet clustering approach —instead of optimizing the threshold globally— and the low performance of the official RepLab 2013 baseline corroborate this intuition.

[5] Note that this is the only supervised information used on this system.
[6] Refer to §4.2.3 for further details.

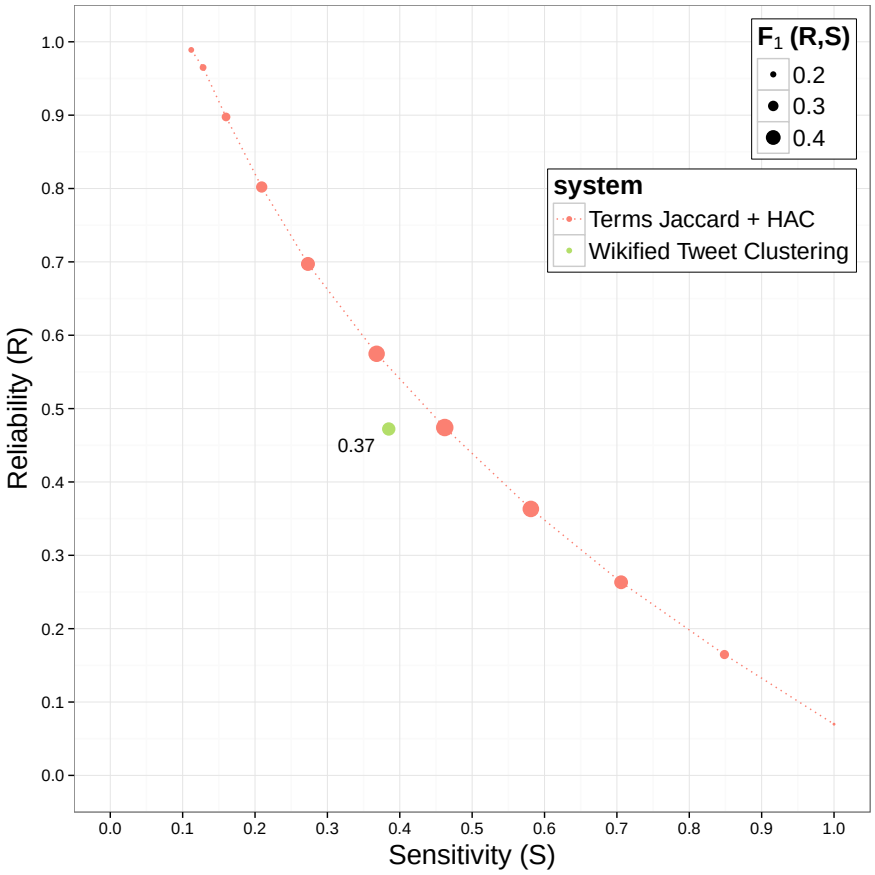


FIGURE 6.4: Wikified Tweet Clustering compared to `Term Jaccard + HAC` baseline in the RepLab 2013 Topic Detection Task.

6.3 Cluster Keywords

In Section 5.1, we have seen that discovering keywords is a valid approach for tackling the filtering task. In filtering, positive keywords often refer to topics related to the entity of interest (e.g., "ipod" or "itunes" for the entity Apple). In this section we extend the notion of filter keywords to *cluster keywords*, i.e., keywords that are strongly representative of the topics being discussed in a given Twitter stream about the entity of interest. We describe the approach in §6.3.1 and present the experiments in RepLab 2012 and 2013 in §6.3.2. Analogously to filter oracle keywords, we analyze the upper bound of oracle keywords for topic detection in §6.3.3.

6.3.1 Approach

Let us assume that each topic related to an entity can be represented with a set of keywords, that allow to the expert to understand what the topic is about. For instance, keywords such as "battery" or "lithium-ion" for the entity Nissan refer to the topic "Batteries for LEAF Car".

Considering this, we define a two-step algorithm that consists of (i) identifying the terminology associated to each topic —by clustering the terms— and (ii) assigning tweets to the identified

clusters. Note that most of the computational cost resides in the first step of our approach. Intuitively, the size of the vocabulary space grows much slower than the size of the tweet stream. Therefore, this has a potential advantage in terms of scalability, which can be crucial in real-time scenarios.

Term Clustering

To carry out the first step (term clustering) we simply use Hierarchical Agglomerative Clustering (HAC), which is the top performer in similar tasks [16]. As similarity function, we use the confidence score returned by a classifier that, given a pair of co-occurrent terms, guesses whether both terms belong to the terminology of the same topic or not.

Pairwise Term Representation

Note that, different from filter keywords —where we consider each term as an instance— in our cluster keyword approach we model instances as pair of co-occurrent terms. Besides efficiency, data sparsity is another main issue that motivated us to define this approach. Pairwise term representation enables us to handle more information (i.e., all the tweets on which the terms co-occur). We used different families of features to represent each of the co-occurring pairs:

- **Term features:** Features that describe each term of the co-occurrence pair. These are: term occurrence, normalized frequency, pseudo-document TF.IDF as presented in §5.1.2.1 and, analogously, pseudo-document KL-Divergence. These features were computed in two ways: (i) considering only tweets in the labeled corpus, and (ii) considering tweets in both the labeled and the background corpus provided by the RepLab datasets.
- **Pairwise content features:** Features that consider both terms of the co-occurrence pair, such as Levenshtein's distance between terms, normalized frequency of co-occurrences, term Jaccard similarity of the tweets in which the terms co-occur.
- **Pairwise meta-data features:** Jaccard similarity and Shannon's entropy of named users, URLs, hashtags and authors of the tweets where both terms co-occur.
- **Pairwise time features:** Features based on the date of the creation of the tweets where the terms co-occur. We consider the following features: median, minimum, maximum, mean, standard deviation, Shannon's entropy and Jaccard similarity. These features were computed considering four different time scales: milliseconds, minutes, hours and days.

In our classification model each instance corresponds to a pair of co-occurrent terms $\langle w, w' \rangle$ in the related tweets of a given entity. In order to learn the model, we extract training instances from the trial and training dataset —for RepLab 2012 and RepLab 2013, respectively— considering the following labeling function:

$$\mathit{label}(\langle w, w' \rangle) = \begin{cases} \texttt{clean} & \text{if} \quad \max_j \mathit{Purity}(C_{w \cap w'}, G_j) > 0.9 \\ \texttt{noisy} & \text{in other case} \end{cases} \tag{6.7}$$

where $C_{w \cap w'}$ is the set of tweets where terms w and w' co-occur and $\mathcal{G}$ is the set of topics in the goldstandard, and

$$\mathit{Purity}(C_i, G_j) = \frac{|C_i \cap G_j|}{|C_i|} \tag{6.8}$$

After this process, term pairs with 90% of the tweets belonging to the same cluster in the gold-standard (i.e., purity=0.9) are considered *clean* pairs. Otherwise, co-occurrent terms with less purity are labeled as *noisy* pairs.

Similarity Function

We rely on Machine Learning (ML) for building a similarity function. Using all the co-occurrence pair instances in the training set, we build a single binary classifier that will be applied to all the entities in the test set. Then, the confidence of a co-occurrence pair belonging to the *clean* class is used to build a similarity matrix between terms. A Hierarchical Agglomerative Clustering is then applied to cluster the terms, using the previously built similarity matrix. Finally, a cut-off threshold based on the number of possible merges is used to return the final term clustering solution.

Hierarchical Agglomerative Clustering

Tweet to Topic Assignment

The second step of this algorithm consists of assigning tweets to the identified term clusters. Each tweet is assigned to the cluster with the highest Jaccard similarity.

6.3.2 Experiments

We now describe the results of the cluster keyword clustering approach in RepLab 2012 and RepLab 2013, to which it has been submitted as a system in the official evaluation.

6.3.2.1 RepLab 2012

In the official RepLab 2012 participation, our cluster keywords approach corresponds to the `replab2012_monitoring_UNED_2` run. Here, tweets were lowercased and tokenized using a Twitter-specific tokenizer [138], and punctuation was removed. Only terms after stopword removal and with occurrence greater than five are considered.

As above, trivial clustering relationships between near-duplicate tweets were resolved a-priori and then merged with the output of the system.

We used a Naïve Bayes classifier to learn the clean/noisy pair-terms classifier. We experimented with several ML methods using Rapidminer[128]: Multilayer Perceptron with Backpropagation (Neural Net), C4.5 and CART Decision Trees, Linear Support Vector Machines (SVM), and Naïve Bayes. We used a *leave-one-entity-out* cross-validation strategy to evaluate the performance of the models on the RepLab 2012 trial data. On each fold, all but one entities are used to extract the pair terms for feeding the classifier, while the pair terms related to an remainder entity are used as test data. This process is repeated 6 times (as many as entities in the trial corpus) and AUC is computed to evaluate the classifiers. In the trial data, Naïve Bayes significantly outperformed the other tested models, obtaining AUC values above 0.8 in all trial entities except one, `Alcatel-Lucent` (entity id `RL2012E02`), which got a lower score.

Leave-One-Entity-Out Cross-Validation

The Hierarchical Agglomerative Clustering (HAC) was performed applying average linkage (i.e., considering the mean similarity between elements of each cluster) and using the S-Space package implementation [88]. The cut-off threshold of the HAC was empirically established to 0.9999 after analyzing the runs with the trial data.

Figure 6.5 shows the results for the cluster keyword system in comparison to the RepLab 2012 official baseline —explained in the previous section. Again, the results show that the official baseline —a simple agglomerative clustering based on term Jaccard similarity— is difficult to beat in this collection.

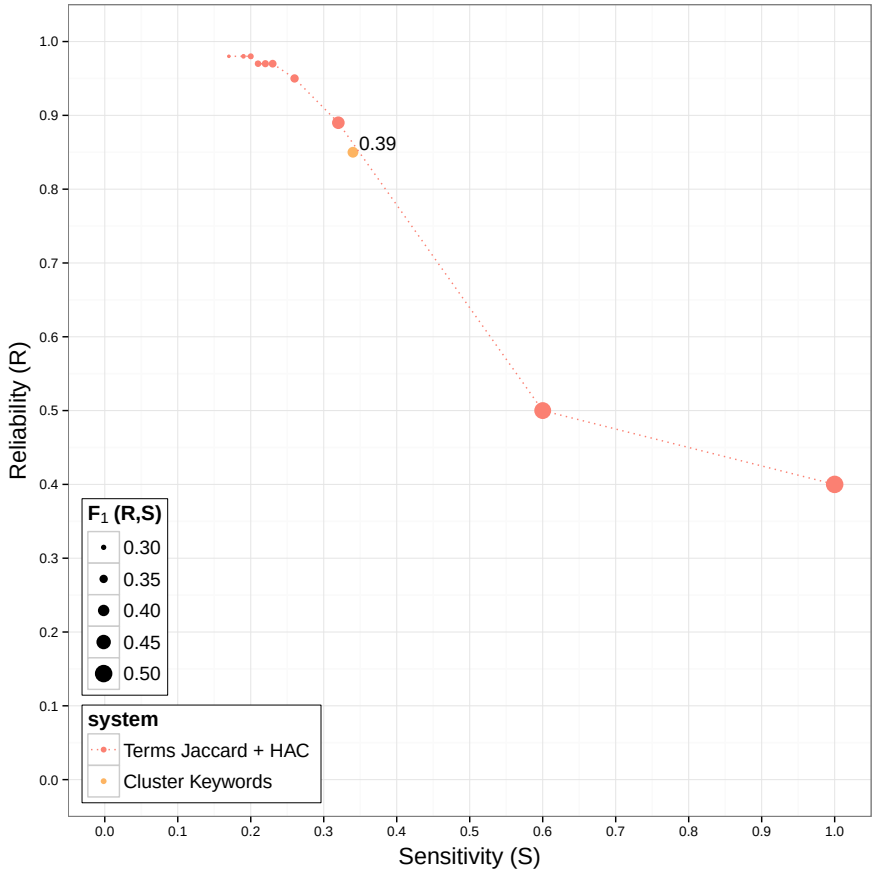


FIGURE 6.5: `Term Jaccard+HAC` baseline and Cluster Keywords in the RepLab 2012 Topic Detection Task.

In terms of the averaged $F_1(R, S)$, the cluster keywords approach performs as the wikified tweet clustering explained in the section above (0.39). Using cluster keywords though, the resulting clusters are less pure: it has 0.85 Reliability, which is 0.10 less than the wikified tweet clustering. Besides, the cluster keywords are able to capture more clustering relationships, achieving a Sensitivity score of 0.34.

Note that in these topic detection experiments —submitted as official runs to RepLab 2012—, filtering was not previously performed, i.e., the system received all the tweet stream D_q instead of D_e. This may had some negative impact when grouping tweets according to topics. Remarkably, in some test cases where most of the tweets are not related to the entity of interest such as `Indra` (`RL2012E12`), `ING` (`RL2012E15`) or `BP` (`RL2012E27`), both proposed systems obtain F_1 scores below 0.25. This suggests that an explicit treatment of ambiguity is needed, at least when the entity's name may refer to multiple entities or concepts (e.g., acronyms).

6.3.2.2 RepLab 2013

Likewise, we tested our cluster keywords approach in the RepLab 2013 Topic Detection Task. Cluster keywords have been submitted to RepLab 2013 with different combinations of parameters that are summarized in Table 6.7. We tested two different machine learning algorithms to combine the features (Naïve Bayes and Logistic Regression) and different thresholds were applied to the HAC. For instance, a threshold of 0.30 indicates that the final clustering considered is the one produced when 30% of all possible merges are applied. As in RepLab 2012, the merges in the HAC were carried out by the average linkage criterion.

TABLE 6.7: Overview of the cluster keyword runs submitted to the RepLab 2013 topic detection task.

Approach	**Parameters**	**Run**
Cluster Keywords	combination=N.Bayes, HAC threshold= 0.3	`UNED_ORM_topic_det_3`
Cluster Keywords	combination=N.Bayes, HAC threshold= 0.7	`UNED_ORM_topic_det_4`
Cluster Keywords	combination=Log.regression, HAC threshold= 0.7	`UNED_ORM_topic_det_5`

Surprisingly, neither of the tested parameters —ML algorithm used to combine features and the HAC threshold— had a significant impact in the effectiveness of the cluster keywords. Using Naive Bayes or Logistic Regression as ML algorithms obtained identical results in the runs, while considering a threshold of 0.3 has only an absolute improvement of 0.0012 in terms of $F_1(R, S)$. Figure 6.6 compares the latter run to the `Term Jaccard+HAC` baseline.

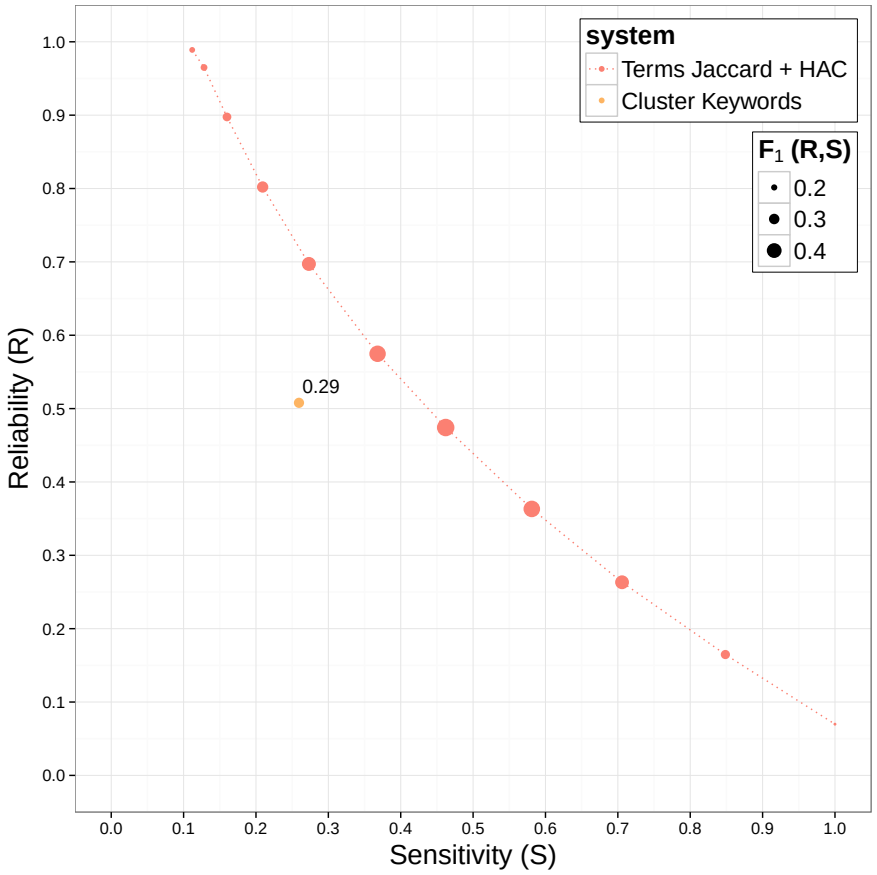


FIGURE 6.6: `Term Jaccard+HAC` baseline and Cluster Keywords in the RepLab 2013 Topic Detection Task.

On average, the cluster keyword strategy achieves a Reliability of 0.51 and a Sensitivity of 0.26, obtaining a $F_1(R, S)$ of 0.29. We can see in the figure that, again, our approach is below the Reliability/Sensitivity curve obtained by the baseline. Although our results are competitive with other RepLab 2013 systems, the `Term Jaccard+HAC` is hard to beat, showing the difficulty of the task.

6.3.3 Oracle Cluster Keywords

As for filter keywords, one may wonder to what extent cluster keywords are useful for detecting the topics discussed about an entity of interest in Twitter. To answer this question, we define the *oracle cluster keywords* as the purest keywords that cover most of the tweets in the stream. Intuitively, we look at the most discriminative terms a posteriori, i.e., knowing the clustering relationships annotated in the gold standard.

Algorithm 2 sketches the process of generating a clustering output by considering a given n number of oracle keywords. It starts by initializing the clustering output $\mathcal{C}_\mathcal{O}$ to the one-in-one clustering, i.e., each tweet belongs to a different cluster (line 4).

For each iteration i, the best candidate oracle keyword is computed as follows. First, the terms in W_e with the highest purity are considered in W_p (line 10). From the subset W_p, terms that maximize the coverage are considered in W_{kw}. Note that this is done by considering D_K, which includes tweets that have been already taken from the keywords in K. Since W_{kw} may have more than one term, we randomly sample one candidate to get kw_i, i.e., the oracle keyword at iteration i (line 12). Once the keyword is selected, the subset of the covered tweets D_K is updated (line 14).

Algorithm 2: Oracle Topic Keywords Clustering

input : step n

1 Data(D_e *tweets related to the entity* e, W_e *terms occurring in tweets* D_e)

Result: $\mathcal{C}_\mathcal{O} = \{C_1, \ldots, C_k\}$

2 *Initialization*

3 **begin**

4 $\quad \mathcal{C}_\mathcal{O} = \{C_1, \ldots, C_k | d_i \in C_i,\ \forall d_i \in D_e \wedge |\mathcal{C}_\mathcal{O}| = |D_e|\}$ (one-in-one clustering)

5 $\quad K \longleftarrow \emptyset$ (oracle keywords)

6 $\quad D_K \longleftarrow \emptyset$ (tweets in D_e covered by oracle keywords K)

7 **end**

8 *Oracle Topic Keywords Algorithm*

9 **for** $i \in 1..n$ **do**

10 $\quad W_p = \left\{ \arg\max_{w \in W_e} \textit{Purity}(w) \right\}$

11 $\quad W_{kw} = \left\{ \arg\max_{w \in W_p} \textit{Coverage}(w) \right\}$

12 $\quad kw_i = sample(1, W_{kw})$

13 $\quad D_{kw_i} \longleftarrow \{d \in D_e | \text{term } kw_i \text{ occurs in } d\}$

14 $\quad D_K \longleftarrow D_K \cup D_{kw_i}$

15 $\quad W_e = W_e - W_{kw}$

16 $\quad \mathcal{C}_\mathcal{O} = \mathcal{C}_\mathcal{O} - \{C \in \mathcal{C}_\mathcal{O} | D_{kw_i} \cap C \neq \emptyset\}$

17 $\quad \mathcal{C}_\mathcal{O} = \mathcal{C}_\mathcal{O} \cup \bigcup_{C \in \mathcal{C}_\mathcal{O}} \{C | D_{kw_i} \cap C \neq \emptyset\}$

18 **end**

19 **return** $\mathcal{C}_\mathcal{O}$ at step n.

where

$$\textit{Purity}(w) = \frac{|C_w \cap G_j|}{|C_w|} \tag{6.9}$$

and

$$\textit{Coverage}(w) = \frac{|C_w - D_K|}{|D_e|} \tag{6.10}$$

being C_w the set of tweets in D_e containing the term w and D_K the tweets covered by the set of oracle keywords K already considered at each iteration i.

Then, terms in W_{kw} are removed from the vocabulary set W_e (line 15) and the clustering is updated by merging all the clusters containing the oracle keyword kw_i into one single cluster

(lines 16 and 17). When all iterations are over, the algorithm returns the clustering output resulting from considering the optimal n oracle keywords.

The oracle cluster keywords algorithm at different n steps was performed over all the entities, in both RepLab 2012 and 2013 dataset. Figures 6.7 and 6.8 show the averaged Reliability/Sensitivity curves of oracle cluster keywords in both collections, respectively. Our topic detection approaches —wikified tweet clustering and cluster keywords—are also shown, as well as the `Term Jaccard + HAC` baseline.

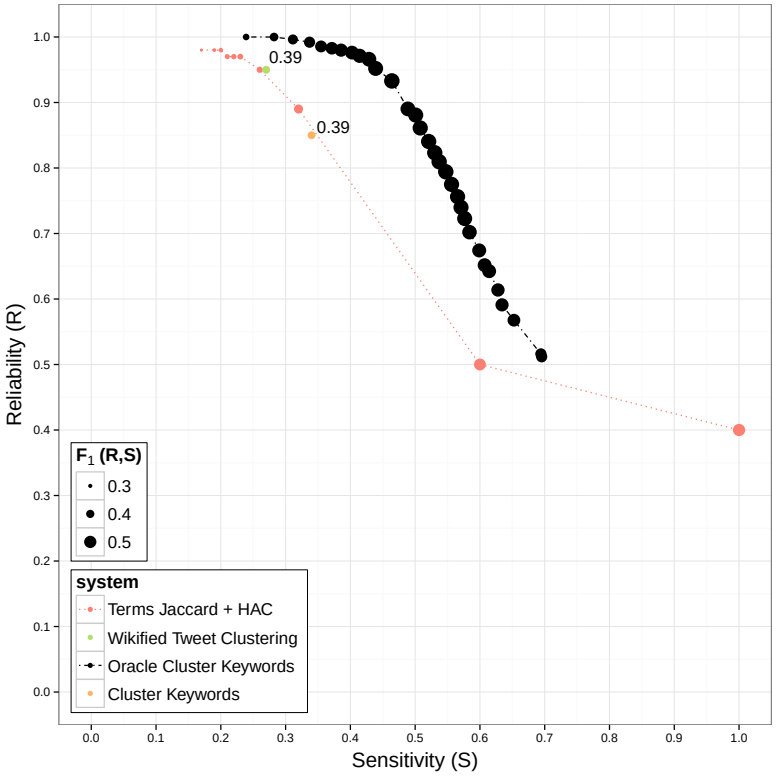


FIGURE 6.7: Oracle Cluster Keywords, proposed approaches and `Term Jaccard+HAC` baseline in the RepLab 2012 Topic Detection Task.

At RepLab 2012 (Fig.6.7), considering twelve oracle keywords, the approach obtains the highest $F_1(R, S)$ score of 0.58. At this point, the cluster are significantly pure (Reliability of 0.93), while almost half of the clustering relationships are covered (Sensitivity of 0.46). Remarkably, only considering the best five oracle keywords it is possible to cover 35% of the clustering relationships with almost perfect precision (0.99 Reliability). Note that the margin for improvement —represented by the area in between of the two curves—is not large.

At RepLab 2013 (Fig.6.8), the scenario was even more challenging. The best averaged $F_1(R, S)$, 0.61, was achieved when most of the oracle keywords were considered, with a low Reliability (0.52) but a relatively high Sensitivity (0.74). Note that being $F_1(R, S)$ the harmonic mean of Reliability and Sensitivity. Thus, it tends to aggravate the impact of small values —which is the case of considering a little number of oracle keywords, when Reliability is high but Sensitivity is very low. For instance, considering the best ten oracle keywords, clustering achieves an $F_1(R, S)$ of 0.27, given by a perfect Reliability but a low Sensitivity (0.17). The curve shows that it is

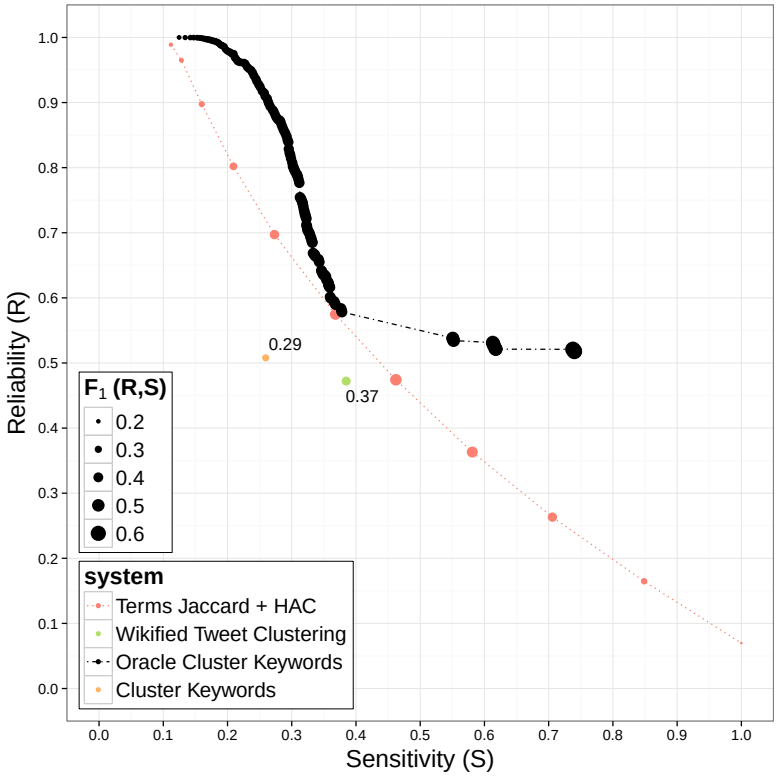


FIGURE 6.8: Oracle Cluster Keywords, proposed approaches and `Term Jaccard+HAC` baseline in the RepLab 2013 Topic Detection Task.

difficult to cover more clustering relationships without hurting precision: covering 30% of the relationships drops the precision to 0.8. Therefore, the margin for improvement is again very small.

Summing up, although cluster keyword strategy has been proposed to deal with two of the main issues of ORM topic detection (efficiency and data sparsity), results show that a simple agglomerative clustering based on text similarity between tweets is a difficult barrier to overcome. Finally, an analysis of the performance of oracle keywords reveals that the margin for improvement is not large.

6.4 Learning Similarity Functions for Topic Detection

From the results obtained so far —and, from the results of the participant systems— it is not clear whether the topic detection process could benefit from training data, and there is no clear evidence on whether Twitter-specific data (such as tweet metadata, hashtags, timestamps, etc.) could be effectively used to improve the results of term-based clustering.

In this section we focus on two related research questions:

1. *Can Twitter signals be used to improve entity-specific topic detection?* Given that tweets are very short by nature, and entity-related topics usually small, it is reasonable to think that

any extra information —and Twitter offers many potentially useful signals in addition to plain content— could be useful to solve the problem.

2. *Can previously annotated material be used to learn better topic detection models?* Usually, clustering and topic detection algorithms are unsupervised. However, in a daily reputation monitoring task, there is likely to be some amount of recently seen and (at least partially) annotated information about the entity being monitored. The question, then, is how to profit from such annotations in the topic detection task.

In order to answer these two questions, we have modeled the topic detection problem as a combination of two tasks:

1. The first is learning tweet similarity: we use all types of Twitter signals (tweet terms and concepts, hashtags, named users, timestamp, author, etc.) to learn a supervised classifier that takes two tweets as input and decides if the tweets belong to the same topic or not. Most of our experimentation is focused on this problem.
2. The second is applying a clustering algorithm that uses the confidence of the classifier above as similarity measure between tweets. For this step we simply use HAC (Hierarchical Agglomerative Clustering), which is the top performer in similar tasks [16].

6.4.1 Approach

We focus on learning similarity measures between tweets that predict whether two given tweets are about the same topic or not. We explore a wide range of similarity signals between tweets (terms, concepts, hashtags, author, timestamp, etc.) and use them as classification features to learn similarity measures. Similarity measures are, in turn, fed into a competitive clustering algorithm in order to detect topics.

Following the methodology proposed in [13] for a different clustering problem, we model the problem as a binary classification task: given a pair of tweets $\langle d_1, d_2 \rangle$, the system must decide whether the tweets belong to the same topic (`true`) or not (`false`). Each pair of tweets is represented as a set of features (for instance, term overlapping between both tweets), which are used to feed a machine learning algorithm that learns a similarity function. Once we have learned to classify tweet pairs, we take the positive classification confidence as a similarity measure, which is used by a Hierarchical Agglomerative Clustering (HAC) algorithm to identify the topics.

We now detail the learning similarity and the clustering steps. Finally, in §6.4.1.2 we describe the features used to learn the similarity function.

6.4.1.1 Learning a Similarity Function

Our first goal is to find a classification function that takes two tweets as input and decides if they belong to the same topic or not. Once the pairwise binary classification model is built, its

confidence is used as pairwise similarity measure. Formally, let d, d' be two tweets in a set $\mathcal{T}$. We want to learn a boolean function

$$G(d, d') : \mathcal{T} \times \mathcal{T} \rightarrow \{\text{true}, \text{false}\} \tag{6.11}$$

that says if both tweets belong to the same topic or not. We define a list of features $F_{d,d'} = (f_1(d, d'), f_2(d, d') \ldots f_n(d, d')$, where each of the features is an estimation of the overlap between d, d' according to different signals. Then we estimate the similarity between d, d' as the probability that they belong to the same topic given $F_{d,d'}$:

$$sim(d, d') = P(G(d, d')|F_{d,d'}) \tag{6.12}$$

For each entity, we compute the confidence score for all the possible pairs of tweets related to it. The resulting similarity matrix is used by the Hierarchical Agglomerative Clustering (HAC) algorithm [118], with single linkage, that has been proven to perform competitively in clustering tasks such as Web People Search [37, 137]. In HAC there is no need to specify the number of clusters a priori: the first step is to create one cluster for each tweet in the similarity matrix, and then compute for each cluster the similarity to all other clusters. If the highest similarity computed is above a predefined threshold, the two clusters are merged together (agglomerated). A similarity threshold is then used as a stop criterion to get a flat clustering solution. As for "single linkage", it refers to the way in which clusters are compared during the clustering process: in single-link clustering, the similarity between two clusters is computed as the similarity of their most similar members (i.e., it focuses on the area where both clusters are closest to each other). A drawback of this method is that clusters may be merged due to single noisy elements being close to each other, but in practice it seems to be the best choice for problems related to ours [13, 116, 137].

6.4.1.2 Similarity Signals

In our study we consider a total of 13 features that capture many types of Twitter signals. Features can be divided in four families: *term features*, that take into account similarity between terms in the tweets; *semantic features*, that model tweet similarity by mapping tweets to concepts in a knowledge base, and then measuring concept overlap between tweets; *metadata*, which indicate whether the tweets have authors, named users (i.e., Twitter users mentioned in the tweets), urls and hashtags in common; and *time-aware features*, which say how close the creation time stamps are for the tweets being compared.

6.4.1.2.1 Term Features The most obvious signal to take into account is word similarity. Tweets sharing a high percentage of vocabulary are likely to talk about the same topic and hence, to belong to the same cluster. We experimented with three term features that differ in how the terms are weighted:

- `terms_jaccard`. It computes the Jaccard similarity between the set of (unweighted) terms W in the tweets.

$$f_{\text{terms\_jaccard}}(d, d') = \frac{|W_d \cap W_{d'}|}{|W_d \cup W_{d'}|} \tag{6.13}$$

- `terms_lin_cf`. Lin's similarity [109] can be seen as a weighted variation of Jaccard:

$$f_{\text{terms\_lin\_cf}}(d, d') = \frac{2 \cdot \sum_{w \in W_d \cap W_{d'}} \log \frac{1}{p(w)}}{\sum_{w \in W_d} \log \frac{1}{p(w)} + \sum_{w \in W_{d'}} \log \frac{1}{p(w)}} \tag{6.14}$$

  where $p(w) = \frac{cf(w)}{\sum_i cf(w_i)}$ and $cf(w)$ is the term frequency in the collection.

- `terms_lin_tfidf`. Similar to `terms_lin_cf`, this variant uses a tf.idf weighting function meant to capture the specificity of the term with respect to the entity of interest. To compute the tf.idf weight, all tweets related to the entity are treated as a pseudo-document D in the collection C:

$$p(w) = \frac{tf(w, D) \cdot \log \frac{N}{df(w)}}{\sum_i tf(w, D_i)} \tag{6.15}$$

  where $tf(w, D)$ denotes the term frequency of term w in pseudo-document D; $cf(t)$ denotes the term frequency in the collection C and $df(t)$ denotes the total number of pseudo-documents $D_i \in C$ in which the term t occurs at least once.

6.4.1.2.2 Semantic Features

Intuitively, representing tweets with semantics extracted from a knowledge base can be useful to group tweets that do not have words in common. For instance, tweets d_1 and d_2 about *Maroon 5* in Table 4.3 can be clustered together because the phrases *mexicanas* and *Mexico* both link to the concept `Mexico`. In some cases this relation could also be captured with stemming, but at the cost of additional false matches. In addition, it might be useful to detect salient terms when word similarity is low. For instance, the Jaccard similarity for tweets d_5 and d_6 is not high, but mapping into Wikipedia matches *Alonso*, *Ferrari* and *estrategia* in both tweets, which lead to a high concept match between them.

Based on the wikified tweet clustering approach explained in §6.2, we represent tweets as *bag-of-entities*.

Analogously to term features, we compute the semantic features `semantic_jaccard`, `semantic_lin_cf` and `semantic_lin_tfidf` over the bag-of-entities tweet representation. The feature `semantic_jaccard` is similarly defined by the Best RepLab system [161], detailed in §6.4.2.4.

6.4.1.2.3 Metadata Features

- `author`. Two tweets by the same author are more likely to be about the same topic. In Table 4.3, an example is tweets d_3 and d_4, which are both published by the same MotoGP follower.

- `namedusers`. The number of mentions of named users that co-occur in a pair of tweets also increases the probability that they are about the same issue. See for instance tweets d_7 and d_8, which are both replies to a user (@alo_oficial) which is central to the topic. Another common example are mentions of the entity's official Twitter account (@ford, @kia, @audi, @shakira, etc.).
- `urls`. Number of urls that co-occur in a pair of tweets. Tweets that belong to the same cluster may not have high word similarity but might refer to the same URL (for example, d_{11} and d_{12}). This is usually an indication of a topical relationship.
- `hashtags`. Often, hashtags denote topical co-occurrence, and it can be useful to measure their overlap separately (and in addition to) term overlap. d_9 and d_{10} are an example of two topically related tweets that share a hashtag (#F1).

6.4.1.2.4 Time-aware Features Frequently, topics reflect an ongoing event (such as a live performance of a music group) or conversation. For this reason, close timestamps increase the probability of two tweets being related. For instance, tweets d_{15} and d_{16} were both published in the hour preceding a concert by Coldplay.

We define the features to estimate temporal relation between tweets, given the timestamps t and t', as:

$$f_{\text{time}}(t, t') = \frac{1}{1 + |t - t'|} \tag{6.16}$$

which takes values between 0 and 1. We turn this equation into three different features, depending on how we represent time: in milliseconds (`time_millis`), hours (`time_hours`) or days (`time_days`).

Note that the author and the timestamp of the tweets are also considered by the Temporal Twitter-LDA system [161], described in §6.4.2.4.

6.4.2 Experiments

Before computing the features, tweets were normalized by removing punctuation, lowercasing, tokenizing by whitespaces and removing stopwords and words with less than three characters. We first analyze the results that help to answer our research questions: first, we study the impact of the different signals in the process of learning a similarity function in §6.4.2.1; then, we study the effect of embedding the similarity functions in the Clustering process to solve the ORM Topic Detection task: In §6.4.2.2, we investigate the benefits of Twitter-related signals; in §6.4.2.3, whether the learning process is effective, and in §6.4.2.4 we compare our results with state-of-the-art results on the same corpus, i.e., the best RepLab 2013 systems. Finally, we report the

results of a failure analysis that gives some insights into how reputation experts annotate and which the main challenges for automatic systems are.[7]

6.4.2.1 Learning Tweet Similarity

Before tackling the topic detection task, we analyze the effectiveness of different signals to learn a similarity function. Given the short length of tweets, our hypothesis is that Twitter-specific signals should help building better similarity functions.

We start by building a pairwise classification model using linear kernel SVM[8] [87] We randomly sample 80,000 pairs of tweets from the RepLab 2013 training dataset, keeping the `true` and `false` classes balanced. We run a 10-fold cross-validation on this sample. Table 6.8 reports results in terms of averaged accuracy (which is a suitable measure as classes are balanced) for different feature combinations.

TABLE 6.8: Learning Similarity Functions: SVM Accuracy and Maximal Pairwise Accuracy theoretical upper bound (maxPWA) for different signal combinations.

Signal Combination	**SVM Acc.**	**maxPWA**
time {milliseconds, hours, days}	0.56	0.43
metadata {authors, namedusers, urls, hashtags}	0.58	0.60
terms_jaccard	0.59	0.60
semantics {sem_jaccard, sem_lin_cf, sem_lin_tfidf}	0.59	0.70
terms {terms_jaccard, terms_lin_cf, terms_lin_tfidf}	0.59	0.78
terms + time	0.61	0.86
terms + semantics	0.60	0.87
terms + semantics + metadata	0.62	0.90
terms + semantics + time	0.61	0.91
all	0.63	0.94

We use the Student's t-test to evaluate the significance of observed differences. Significant differences are indicated using ▲ for $\alpha = 0.01$ and △ for $\alpha = 0.05$.

The relative differences seen on SVM cannot be directly extrapolated to any Machine Learning algorithm. Therefore, we also compute Maximal Pairwise Accuracy (maxPWA) [13], which is a theoretical upper bound of the effectiveness of different feature combinations, and computes the performance of an ideal Machine Learning algorithm that, for each classification instance, only listens to the features that give the right information[9].

Remarkably, the Pearson correlation between the accuracy of the linear SVM and the theoretical upper bound maxPWA is 0.93. In other words, whenever a set of features gives useful additional

[7] Code and proposed system outputs for the RepLab 2013 Topic Detection Task are publicly available at http://damiano.github.io/learning-similarity-functions-ORM/

[8] We tested other machine learning algorithms like Naïve Bayes and Decision Trees, obtaining lower absolute results but similar relative improvements; hence we report results for SVM only.

[9] Given the quadratic cost of computing maxPWA —$0(n^2)$ for n pairs— we use a balanced sample of 8,000 pairs and report the averaged scores over 10 runs.

information (as reflected in the theoretical upper bound for any learning algorithm), SVM is able to profit in direct proportion to the amount of new useful signal available. Therefore, differences seen with SVM can be generalized to other algorithms.

An inspection of the results in Table 6.8 shows that:

- In terms of absolute accuracy scores, the quality of the models is low (between 0.56 and 0.63), given that 0.50 is the performance of a random binary classifier. This indicates that the problem is challenging (see Section 6.4.2.5 for a qualitative discussion).
- *Time-aware features are useful*. Time-aware features in isolation only reach 0.56 accuracy. However, when added to content signals (`terms_jaccard`, terms, semantics), they contribute to increasing performance, with statistical significance, from 0.60 (content signals only) to $0.61^{\blacktriangle}$ (content plus time-aware features). Therefore, time features give a moderate but useful signal.
- *Semantic features are useful*. Although terms and our semantic features (links to wikipedia articles) reach the same accuracy in isolation (0.59), their combination reaches $0.60^{\blacktriangle}$ (2% relative improvement).
- *Metadata is useful*. Likewise, metadata features (0.60 accuracy) also capture additional information with respect to content only: combining both gives $0.62^{\blacktriangle}$ accuracy (3% improvement).
- *All features give best performance*. Unsurprisingly, combining all features seems to be the best choice, giving an accuracy of $0.63^{\blacktriangle}$, which has a statistically significant difference with respect to using terms (0.59, 6% relative improvement).

In summary, most signals in our study are able to improve the classification process with statistical significance over the use of term-based features only, and their combination gives the best performance. Although the absolute performance of the best learned function seems low (0.63 accuracy), we will see in the following sections that, once the classification confidence is used as similarity measure, it leads to the best topic detection performance reported on the RepLab dataset so far.

We now turn to the experiments on the Topic Detection Task. We first compare the effect of considering different Twitter signals in our similarity function (§6.4.2.2), then we study the effect of the learning process (§6.4.2.3) with respect to an unsupervised alternative, and finally we compare our results with the state-of-the-art (§6.4.2.4).

6.4.2.2 Topic Detection: Effect of Twitter Signals

We have seen that a classification model that combines all the features is the most accurate. We now use the positive classification confidence score for a pair of tweets as estimation of the similarity between them, and feed the single-link HAC clustering algorithm with this similarity score to detect the topics in the test set, for each of the 61 entities included in the dataset.

In order to answer one of our initial research questions, *Can Twitter signals be used to improve entity-specific topic detection?*, we compare the results of HAC using two learned similarity functions: a baseline using `terms_jaccard` as signal, and our best function, which uses all features.[10]

We report results using the official evaluation measures at the RepLab 2013 Topic Detection Task: Reliability & Sensitivity (R&S) [11] and its balanced F-Measure (harmonic mean), $F_1(R, S)$. Note that, in clustering tasks, R&S are equivalent to the well-known BCubed Precision and Recall measures [10].

Figure 6.9 shows results as macro-averaged R&S in the RepLab 2013 test dataset. Reliability (y-axis), Sensitivity (x-axis) and $F_1(R, S)$ (dot size and numbers) are plotted. Note that $F_1(R, S)$ is not the harmonic mean of the average R&S, but the average of the harmonic mean for each test case (the 61 entities in the test collection). Each dot in a curve represents the output of the HAC algorithm at different similarity thresholds (in percentiles). A lower similarity threshold gives larger clusters, increasing Sensitivity (BCubed Recall) at the expense of Reliability (BCubed Precision).

If we compare using all features with term similarity only (SVM(all)+HAC versus SVM(term_jaccard)+HAC), Figure 6.9 shows that they have the same maximal value ($F_1(R, S) = 0.47$), but using all features gives more Reliability at high Sensitivity scores. In order to better quantify the differences between systems, we report two measures that summarize the difference of both curves in a single score: the Area Under the R&S Curve (AUC) and the Mean Average Reliability (MAR), which is the counterpart of the standard IR evaluation measure MAP (Mean Average Precision) for our curves. Table 6.9 reports both measures for the two systems. As previously, we denote significant improvements with $^{\triangle}$ and $^{\blacktriangle}$ ($p < 0.05$ and $p < 0.01$, respectively).

TABLE 6.9: Topic Detection: Using all signals versus term co-occurrence, comparison of R&S curves with Area Under the Curve and Mean Average Reliability.

System	AUC	MAR
SVM(terms_jaccard)+HAC	0.40	0.59
SVM(all features)+HAC	0.41	$0.61^{\triangle}$

In terms of Mean Average Reliability, using all features improves over term co-occurrence with statistical significance (3% relative improvement). In terms of AUC, there is a 2% relative improvement but the difference is not statistically significant. Overall, our results suggest that the use of Twitter signals can improve the topic detection process, although the difference is not dramatic.

[10] Note that we use the expression "Twitter signals" in a broad sense —signals that go beyond terms in the tweet—, and therefore we also consider semantic features which are not, strictly-speaking, Twitter-specific signals.

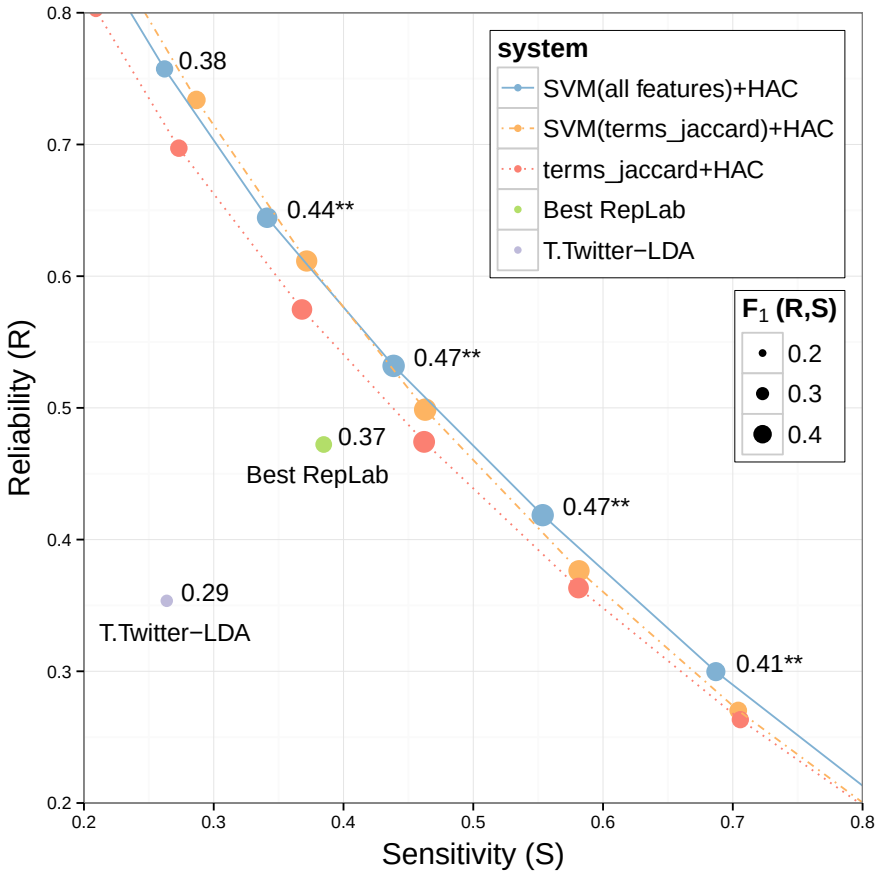


FIGURE 6.9: Reliability/Sensitivity curves for the Topic Detection Task. Size of the dot represents $\mathbf{F_1(R,S)}$ averaged over test cases. $\mathbf{F_1(R,S)}$ scores with ** indicate statistically significant improvements with respect to the best RepLab system ($\mathbf{p < 0.01}$).

6.4.2.3 Topic Detection: Effect of the Learning Process

Our second research question was: *Can previously annotated material be used to learn better topic detection models?* Although many clustering problems are unsupervised in nature, supervision in reputation monitoring makes sense: clients are monitored daily, and what has been seen before is annotated and has an effect on how fresh information is processed. Can we profit from such annotations? The case of the RepLab 2013 dataset is challenging, because tweets in the training and test sets are separated by up to six months —depending on the entity— and the issues about an entity can change dramatically in Twitter in a period of six months.

We investigated this question by comparing two approaches that use the same signal (term co-occurrence as measured by the Jaccard formula): an unsupervised system, which uses directly the Jaccard measure between two tweets as similarity measure; and a supervised system, that uses our learned similarity function using the Jaccard measure as the only feature for the classifier. In both cases, we feed the HAC algorithm with each of the similarity measures.

Figure 6.9 includes both curves (terms_jaccard+HAC and SVM(terms_jaccard)+HAC), and shows that there is a substantial difference between them. The supervised system consistently improves the performance of the unsupervised version regardless of how we set the similarity threshold.

TABLE 6.10: Supervised versus Unsupervised Topic Detection.

System	AUC	MAR
terms_jaccard+HAC	0.38	0.57
SVM(terms_jaccard)+HAC	0.40	0.59^{Δ}

Table 6.10 compares the supervised and unsupervised approaches in terms of AUC and MAR. The supervised system outperforms its unsupervised counterpart with a 2% relative improvement in terms of MAR, which is statistically significant. The difference in terms of of AUC is larger (5%), but is not statistically significant.

Overall, our results indicate that previous annotations can be used to learn better topic models, although differences are not large in our experimental setting. Probably if the time gap between tweets in the training and test sets were smaller (for instance, days instead of months), the effect of learning would be higher.

6.4.2.4 Topic Detection: Comparison with State-of-the-Art

The differences we have detected could be irrelevant or misleading if both our baseline and contrastive systems were below state-of-the-art results. Therefore, we compare our approach with two competitive systems from RepLab 2013:

- **Best RepLab** [161]. The best system in the official RepLab 2013 evaluation campaign [8]. This corresponds to the Wikified Tweet Clustering approach presented in §6.2.
- **Temporal Twitter-LDA** [161] (T.Twitter-LDA). Inspired in Twitter-LDA [201] and Topics Over Time [179], this topic model takes into account the author and the timestamp distributions, in addition to the word distribution in the tweets. In order to estimate the right number of clusters, they incorporate large amounts of additional (unlabeled) tweets to the target data to be clustered and then apply the topic model. We include this system in the comparison because T.Twitter-LDA is a good representative of generative models as compared to the HAC clustering algorithm that we have used.

Figure 6.9 compares all the systems. Numbers with ** indicate statistical improvements ($p < 0.01$) of our best system (SVM(all features)+HAC), at different similarity thresholds, with respect to the best RepLab system[11].

Note that our approach significantly outperforms both the best RepLab system and the T.Twitter-LDA approach, at any reasonable threshold. Note also that a direct application of the HAC algorithm using Jaccard as similarity metric also performs better than the two RepLab systems,

[11]Note that R,S and $F_1(R,S)$ for the two RepLab systems reported are different than the official scores [8], because we are excluding unrelated tweets from our evaluation, and we are excluding also near-duplicates. Nevertheless, all systems benefit similarly from the normalization and it does not produce any change in the official ranking

which seems to confirm that a standard clustering algorithm may be more robust when there is data sparsity, as is the case of reputation monitoring.

If we compare with inter-annotator agreement, our best system (with $F_1(R,S) = 0.47$) gets very close to the reported annotator agreement on the dataset, which is 0.48, measured as the F_1 score of one annotator vs other [8]. Inter-annotator agreement is low, but this is not surprising for a clustering task, even if annotators are reputation experts. But it may be unrealistic to look for improvements in F_1 beyond what we have reached with our learned similarity measures. It is probably more practical to do failure analysis and study where the challenges of the task lie and what the performance of our systems is on a case-per-case basis. That is what we do in the next section.

6.4.2.5 Failure Analysis

So far, we have only investigated average results of our systems across the 61 entities in the RepLab dataset. Here we perform a more detailed analysis of results.

Surprisingly – given the substantial differences between the entities in the dataset – the standard deviation of our best system is low in terms of both R, S and $F_1(R,S)$ (less than 0.09 in all cases). In particular, the F_1 values of our system trained with all features have a standard deviation of 0.06, which compares well with respect to the best RepLab 2013 system (which has a standard deviation of 0.1). Apparently, our system not only performs better on average, but is also more robust across test cases.

In terms of the effect of combining signals, we have seen that taking into account all signals has a slight —but statistically significant— improvement with respect to term matching. If we look case by case, there are only five entities (8% of the whole set) where the average Reliability of using all signals is lower —by a difference of at most 0.02— than using term co-occurrence only: *Capital One*, *Shakira*, *PSY*, *Banco Santander* and *BBVA*. In most cases, for these entities there are large topics that are easy to identify by co-occurrence. For instance, *BBVA* has a topic *Sports sponsor* in which the annotator has grouped all mentions to BBVA sports sponsoring activities. The topic covers 52% of the target tweets, and can be identified with a few keywords that have high precision and high recall and refers to the name of the Spanish soccer League. Likewise, the entity *Shakira* contains a topic *Charity*, with 92 tweets, that refers to the Barefoot Foundation and can be detected by the keyword *support* or the hashtag *#BuyABrick*.

Finally, we have manually inspected *hard topics* —those where our system either fails to cluster, leaving most tweets in single clusters, or creates just a few big noisy clusters— and *easy topics* — those that are accurately solved by any of the similarity combinations tested in our experiments.

Organizational Topics

Remarkably, we found that *hard topics* seem to be general, organizational topics that are used by the reputation manager to organize information in an abstract manner. These topics tend to be subjective, i.e., different experts may organize the information differently. Some examples are *"Concern of Customers"*, *"Bad Service"*, and *"Hate - Opinions"* for the banking domain, *"Fans Tweeting"* in the music domain or *"Looking Forward to Own a Car"*, *"Negative Opinion of an*

Owner" in the automotive domain. In these cases, the content overlap between tweets can be low; for instance, customers complain about the service of their bank in many different ways.

Event-based Topics

On the other hand, topics easy to find are fine-grained and refer to factual affairs, either specific events —*"Man Arrested for Racial Abuse during Capital One Cup Game"*, *"Cisco Hires Barclays to Sell Linksys"*, *"Barclays Fires or Disciples Staff for LS"*, *"Calls to Condemn Uganda's Politics"*, *"Qatar Selling Warrants"*, *"Dave Matthews Band at Wells Fargo Center"*— or talk about a singular dimension of the entity —*"Lexus Owners Club"*, *"Stock Analysis"*, *"Exchange Rates"*. In general, the vocabulary used in event-like topics tends to be more specific than in organizational topics such as *"Ironic Comments of Costumers"*, reducing the difficulty to identify topical relations.

The nature of hard and easy topics is, therefore, quite different. From the point of view of reputation monitoring, the second type of topics is probably more relevant, as it is where reputation alerts tend to be. Hard topics, on the other hand, seem more like a way of categorizing tweets that do not belong to any significant trending topic, and they are more likely to be used differently by different annotators. Perhaps the inter-annotator agreement in the dataset would be higher if we only looked at event-like topics. In any case, it is probably useful to make this distinction explicit both when creating test collections and when reporting results for the task.

6.4.3 Conclusions

In this section we analyzed the suitability of learning similarity functions for detecting topics about an entity in Twitter. In this context, our experimental results indicate that (i) Twitter information (authors, timestamps, etc.) can indeed be used to improve topic detection with respect to the use of textual content only. (ii) It is possible to learn a similarity metric effectively from manually annotated topics, using them to improve the estimation of pairwise tweet similarity. (iii) A conventional clustering algorithm (HAC) using our learned similarity functions performs substantially better than state-of-the-art approaches (including Temporal Twitter-LDA) on the same test collection, and gets close to the inter-annotator agreement rate. Our results seem to confirm that, when data is sparse as in our reputation monitoring scenario, conventional clustering (coupled with an effective similarity function) can be more effective than using generative models such as Temporal Twitter-LDA.

6.5 Wrap Up

ORM topic detection can be seen as the *long tail* of detecting what is being said in Twitter: except for a few aspects, the volume of information related to a specific entity (organization or company) at a given time is orders of magnitude smaller than Twitter trending topics, and that data sparsity makes the problem much more challenging than analyzing Twitter trends.

In this chapter we tackled the ORM topic detection task in different ways. We started by making preliminary experiments in more controlled scenarios: real-time summarization of scheduled events and entity aspect identification. We have seen that term specificity in time and vocabulary

are key features for identifying relevant sub-events and aspects. Lessons learned from these two scenarios were incorporated in our approaches for tackling the ORM topic detection tasks. Besides, we dealt with the following research questions:

- *As we have seen in the filtering task, Wikipedia is a knowledge base that is continuously being updated, and can be a relevant source to discover filter keywords automatically. Are the topics discussed about an entity in Twitter represented somehow in Wikipedia?*
- *Can we generalize the idea of "filter keywords" to "cluster keywords", i.e., can we use it for topic detection?*
- *Can previously annotated material be used to learn better topic detection models?*

We propose different topic detection approaches for tackling these three research questions: wikified tweet clustering, cluster keywords and learning of similarity functions. Results in the available collection for tackling this task —RepLab 2012 and RepLab 2013 datasets— show the difficulty of the topic detection task. A simple agglomerative clustering over term Jaccard similarity performs similarly or better than wikified term clustering and cluster keywords. On the other hand, it is possible to learn a similarity metric effectively from annotated material by combining different similarity functions that make use of Twitter information (authors, timestamps, etc.) besides text similarity. Remarkably, our best result —0.47 $F_1(R,S)$— gets close to the inter-annotator agreement rate of topic detection annotations in the RepLab 2013 dataset —0.48 $F_1(R,S)$.

A detailed qualitative analysis of our results has revealed that there is a special type of topics in the manual data which are harder to detect automatically. These are *organizational* topics. Rather than grouping tweets about a specific issue or event, they have a more taxonomical or structural nature: for instance, a reputation expert may group together all tweets which are *hate opinions* about a bank. Organizational topics tend to be stable across time, and have a wider vocabulary entropy. In contrast, reputation alerts, which are the key issues from a monitoring perspective (e.g., *director of the bank accused of evading taxes*) tend to be spikes in a certain period of time. Organizational topics are not only the main challenge for topic detection systems, but they may also explain the low inter-annotator agreement rates even when, as in the case of the dataset used in our experiments, manual annotations are performed by trained experts.

7 Conclusions

In this chapter, we firstly summarize the main contributions to the research field in Section 7.1. Then we review our findings with respect to our research questions in Section 7.2 and we summarize the practical consequences of our research for ORM systems in Section 7.3. Finally, in Section 7.4 we discuss some future directions for the research work initiated in this thesis.

7.1 Summary of Contributions

In this thesis, we have contributed to the formalization of the ORM problem from a scientific perspective, in order to state the main research challenges and provide a standard evaluation framework that allows the IR&NLP research communities to explore and compare solutions for the detected challenges.

ORM Tasks

We have contributed to the formal definition — in cooperation with reputation experts — of the main ORM tasks: filtering (Is a tweet related to a given entity of interest?), topic detection (What are the main topics discussed about an entity in a given tweet stream?), topic priority (What is the relevance of the topic from a reputational perspective?), polarity for reputation (Does the tweet convey positive or negative implications for the reputation of the entity?), dimension classification (What are the most interested stakeholder group for a given topic?), author profiling (What are the most influential users?) and entity aspect and opinion target identification (What is the brand identity or image of an entity in Twitter?).

Scenarios

We have distinguished two main ORM scenarios: *unknown-entity* and *known-entity* scenarios. The first refers to the situation where the entity of interest is represented as canonical name and a representative URL (e.g., the entity's homepage) but no entity-specific training data is available. Therefore, supervised models have to learn from data associated to other similar entities. On the other hand, in the *known-entity* scenario, there are manual annotated instances about the entity of interest available. Here, systems have more information to model each entity which, in some cases as in the filtering task, is paramount to build a system with an effectiveness tolerable by users.

Evaluation Campaigns

So far, three evaluation campaigns have been organized —in collaboration with UvA, Llorente&Cuenca and Yahoo! Labs, in the context of the LiMoSINe project. The campaigns cover ORM tasks in the *unknown-entity* and *known-entity* scenarios. Table 7.1 summarizes the tasks tackled in each evaluation campaign[1].

TABLE 7.1: Evaluation campaigns organized so far to study the ORM problem.

Evaluation Campaign	ORM Scenario	Filtering	Topic Detection	Topic Priority	Polarity for Reputation
WePS-3 [7]	*unknown-entity*	✓	✗	✗	✗
RepLab 2012 [9]	*unknown-entity*	✓	✓	✓	✓
RepLab 2013 [8]	*known-entity*	✓	✓	✓	✓

Test Collections

Remarkably, all of the reusable test collections built so far have been already used in further research besides the participation in evaluation campaigns [45, 68, 120, 191, 194]. To our knowledge, there is no similar dataset in the state-of-the-art in terms of volume of data and high-quality manual annotations for Online Reputation Monitoring.

Annotation Tool

As a product of the annotation of the RepLab 2013 dataset, an annotation tool has been developed and successfully used by thirteen annotators. Moreover, this tool was recently extended to be used in a semi-automatic environment [32].

Besides the formalization of the ORM problem, work specific to this thesis focused on the filtering and topic detection task in the *unknown-entity* and *known-entity* scenarios.

With respect to the *filtering* task:

- We have proposed a competitive filtering system that discovers and uses *filter keywords*: expressions that, if present in a tweet, indicate with a high probability that the tweet is related/unrelated to the entity/company.
- We defined features that characterize terms in the Twitter dataset, the company's website, ODP, Wikipedia and the searchable Web. We found that (i) term specificity in the tweet stream of each company name is a feature that discriminates between filter keywords and skip terms and (ii) the association between the term and the company website is useful to differentiate positive vs. negative filter keywords, specially when it is averaged by considering its most co-occurrent terms.
- Exploring the nature of filter keywords also led us to the conclusion that there is a gap between the vocabulary characterizing a company in Twitter and the vocabulary associated to the company in its homepage, in Wikipedia, and apparently in the Web at large.
- When entity-specific training data is available —known-entity scenario— it is more appropriate to use a simple BoW classifier than filter keywords. When enough training data is available (around 700 tweets per entity), BoW classifiers can be effectively used for the

[1] RepLab 2014 tackles the dimension classification and the author profiling tasks, which are not considered in this thesis.

filtering task. They can also be used effectively in an active learning scenario, which seems to be a useful technique for keeping the filtering system updated during the lifecycle of the ORM process. In this context, we found that, after annotating only 2% of the sampled test data, we reach 0.52 $F_1(R, S)$-score. We also found that, using active learning with margin sampling, the costs of creating an initial training set can be reduced by 90% after inspecting 10% of test data. Unlike many other applications of active learning to NLP tasks, margin sampling works better than random sampling.

With respect to the *topic detection* task:

- Our preliminary experiments indicated that terms with high specificity —terms that occur frequently in a foreground set of tweets (e.g., sub-event) and are not very frequent in a background corpus (e.g., event stream)— are useful for detecting sub-events in real-time summarization of scheduled events and for identifying entity aspects. They proved as useful features in two of our topic detection approaches: cluster keywords and learning similarity functions.
- We have seen that, as in the filtering task, linking tweets to Wikipedia concepts helps when detecting topics. Although clustering wikified tweets performs worse than a simple agglomerative clustering over term Jaccard similarity, it can be used as a similarity function that can be effectively combined with others to achieve better results. Likewise, discovering cluster keywords is a competitive approach in comparison to RepLab systems, although the term Jaccard similarity baseline is already close to inter-annotator agreement and therefore difficult to beat.
- On the other hand, it is more effective to learn from annotated material by combining different similarity functions that make use of Twitter information (authors, timestamps, etc.) in addition to text similarity. Remarkably, our best result gets close to the inter-annotator agreement rate of topic detection annotations in the RepLab 2013 dataset.
- A detailed qualitative analysis of our results revealed that reputation experts tend to group tweets in two types of topics: *event-based* topics, which correspond to conversations about a specific issue or event, and *organizational* topics, which have a more taxonomical or structural nature: for instance, a reputation expert may group together all tweets which are *hate opinions* about a bank. Organizational topics tend to be stable across time, and have a wider vocabulary entropy. This kind of topics are harder to detect automatically and may also explain the low inter-annotator agreement rates even when, as in the case of the dataset used in our experiments, manual annotations are performed by trained experts.

7.2 Answers to Research Questions

Let us now examine how our work answers the research questions that started it.

> **RQ1:** *Which challenges —when monitoring the reputation of an entity in Twitter— can be formally modeled as information access tasks? Is it possible to make reusable test beds to investigate the tasks?*

This question has been answered collectively under the umbrella of the RepLab evaluation campaigns. With the help of reputational experts we have identified and modeled different information access tasks that are involved in the ORM process. In collaboration with the University of Amsterdam (UvA), Llorente&Cuenca and Yahoo! Labs, we have organized different evaluation campaigns to tackle most of these tasks, building reusable test collections and establishing an evaluation framework that help researchers and ORM stakeholders to understand and tackle the ORM from a scientific perspective. Chapter 4 provides more details about the identified ORM tasks and the generated datasets.

In this thesis we focused on two of the most relevant ORM tasks: *filtering* and *topic detection*. In particular, we tackled these tasks analyzing three main aspects: the use of *keywords*, the use of *external resources* and the use of *training data*. These aspects were studied by the research questions listed below, along with the answers to them.

Thereby, studying the concept of *keywords*:

> **RQ2:** *Can we use the notion of filter keywords effectively to solve the filtering task?*

We have studied the use of filter keywords in the *unknown-entity* and *known-entity* scenarios. Our experiments in the unknown-entity-scenario show that automatically discovered filter keywords are able to classify a subset of 30%–60% of tweets with an accuracy range of 0.75–0.80. Tweets classified by these filter keywords can be used to feed a supervised machine learning process to obtain a complete classification of all tweets for an overall accuracy of around 0.7, similar to the best systems in the state of the art.

We also found that, on average, the best five optimal keywords can directly classify around 30% of tweets. Nevertheless, keywords defined by a human by inspecting web search results relevant to the company name only cover 15% of tweets and accuracy drops to 0.86. Therefore, finding appropriate filter keywords is challenging and Twitter-specific.

> **RQ3:** *Can we extend the idea of "filter keywords" to "cluster keywords", i.e., can we use it for topic detection?*

We have designed a two-step clustering approach that automatically groups cluster keywords by using a learned similarity function between terms and then assigns tweets to the identified keywords. Although the approach is competitive with respect to other systems that participated at RepLab 2012 and 2013, we found that a hierarchical agglomerative clustering algorithm using Jaccard word overlap to estimate tweet similarity performs better than our cluster keyword approach (and also other RepLab systems).

An empirical study of oracle cluster keywords in RepLab 2012 and 2013 datasets showed that there is only a small room for improvement (the ceiling being given by the best cluster keywords), corroborating the difficulty of solving the task with keywords.

Related to the use of Twitter signals and *external resources*:

> **RQ4:** *Where should we look for filter keywords in order to find them automatically?*

We defined features that characterize terms in the Twitter dataset, the company's website, ODP, Wikipedia and the searchable Web. We found that (i) term specificity in the tweet stream of each company name is a feature that discriminates between filter keywords and skip terms and (ii) the association between the term and the company website is useful to differentiate positive vs. negative filter keywords, specially when it is averaged by considering its most co-occurrent terms.

> **RQ5:** *Wikipedia is a knowledge base that is continuously being updated, and can be a relevant source to discover filter keywords automatically. Are the topics discussed about an entity in Twitter represented somehow in Wikipedia?*

Linking tweets to Wikipedia articles can be useful for identifying shared concepts or entities between tweets that are semantically related, even if they are written in different languages. Our wikified tweet clustering experiments show that this approach is competitive. More importantly, wikification provides useful signals that —combined to other Twitter information— obtains significantly better results than only considering term matching.

> **RQ6:** *Can Twitter signals be used to improve entity-specific topic detection?*

Twitter information (authors, timestamps, etc.) can indeed be used to improve topic detection with respect to the use of textual content only. Our study of learning similarity functions reveals that taking into account all signals has a slight—but statistically significant—improvement with respect to term matching.

Related to the use of *training data*:

> **RQ7:** *When entity-specific training data is available, is it worth looking for filter keywords in external resources or is it better to learn them automatically from the training data?*

When entity-specific training data is available —as in the known-entity scenario— it is wiser to use a simple BoW classifier than filter keywords. When enough training data is available (around 700 tweets per entity), BoW classifiers can be effectively used for the filtering task.

> **RQ8:** *In an active learning scenario, what is the impact in terms of effectiveness of an informative sampling over a random sampling? How much of the (initial) annotation effort can be reduced by using active learning?*

BoW classifiers can be used in an active learning scenario, which seems to be a useful technique for keeping the filtering system updated during the lifecycle of the ORM process. In this context, we found that the effectiveness reached by randomly sampling 10% of the test data can be achieved by an informative active learning process (margin sampling) considering only 2% of the data (i.e., around 30 tweets in our experiments). This amounts to an 80% reduction in annotation costs during the active learning process. We also found that using active learning with margin sampling can reduce the effort of making the initial training set: the costs of creating a bulk training set can be reduced by 90% after inspecting 10% of test data.

RQ9: *Can previously annotated material be used to learn better topic detection models?*

Yes, similarity functions can be learned effectively from manually annotated topics, using them to improve the estimation of pairwise tweet similarity. We have found a statistically significant difference in a challenging scenario, where training data includes tweets produced six months earlier than the test tweets. In a real monitoring scenario, annotated data would be much more recent, and it is very likely that the learning effect would be substantially higher.

7.3 Practical Outcome

The contributions provided by our research may have an impact on the process of building better Online Reputation Monitoring systems for reputation experts. This is a set of *best practice* recommendations that derive from our work:

- *Filtering unrelated tweets is crucial.* It is the main step in which an automatic process can reduce the effort of the manual expert significantly. Note that having an effective filtering component reduces the *error propagation* effect when tackling other tasks such as topic detection, polarity for reputation or entity aspect identification.
- *Annotating entity-specific data is cost-effective.* If possible, move to the *known-entity* scenario, i.e., annotate some entity-specific training data ($500 - -700$ tweets). It will allow to train significantly more effective filtering system. If the life-cycle of the monitoring process is long-lived, use active learning to keep the model updated.
- *Topic Detection: previous training data is useful.* Learning similarity functions —which combines different Twitter signals— seems the best way to tackle the problem. Here, precision-oriented systems are preferred. Although they might produce small but pure clusters, it is more useful to the user than big but noisy clusters. Intuitively, automatically detecting event-like topics might be more useful than organizational topics. So, it seems reasonable to differentiate between these two topic categories, and let the user organize not-event related tweets.

7.4 Future Directions

Interactive Evaluation with Real Users

There are different research questions generated by the work presented in this thesis. So far, we analyzed filtering and topic detection systems by evaluating them with respect to manually annotated gold standards —on which the margin in comparison to inter-annotator agreement is low. Next step would be to evaluate our systems with real users, in an interactive evaluation framework. A possible way to do this is analyzing the performance of semi-automatic systems for ORM [32]. Moreover, the use of *active learning* techniques —which we have analyzed in the context of the filtering task, but which can be also explored for topic detection and other ORM tasks— fits perfectly in this scenario.

Event-based vs. Organizational Topics

We have found that ORM experts tend to use two types of topics (event-based and organizational). We believe that some signals that seem useless for the whole set of topics are more likely to perform well for a subclass of topics. For instance, time signals will be more useful for event-based topics than for organizational topics. Therefore, it is worth exploring these two families of topics independently in the future. More precisely, automatic detection efforts should probably focus on event-like topics, because they are more likely to become reputation alerts and because they are less dependent on the organizational preferences of a particular expert.

Topic Priority

Besides the filtering and topic detection tasks, there are other ORM tasks, as the ones tackled at RepLab, which still have a large room for improvement. In particular, the experiments done so far for topic priority at RepLab show the difficulty of the task. An in-depth analysis of the factors and signals that are used by experts to decide whether a topic is a reputation alert or not is still needed.

The work presented in this thesis has been developed under the umbrella of a collective effort that —for the first time— enabled a systematic and comparative evaluation of the information access challenges behind the ORM process. Our results not only contributed with competitive algorithms, but —more importantly— with a better understanding of the difficulties and issues that are intrinsic to the filtering and topic detection tasks. We hope that our outcomes will help to build more effective ORM tools, as well as to refine the evaluation methodology. There is still a long way to go, and we believe that the future will shed more light about the ORM challenges that are still open.

List of Publications

[7] E. Amigó, J. Artiles, J. Gonzalo, **D. Spina**, B. Liu, and A. Corujo. WePS-3 Evaluation Campaign: Overview of the Online Reputation Management Task. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[8] E. Amigó, J. Carrillo de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martín, E. Meij, M. de Rijke, and **D. Spina**. Overview of RepLab 2013: Evaluating online reputation monitoring systems. In *CLEF '13*, LNCS, pages 333–352. Springer, 2013.

[32] J. Carrillo de Albornoz, E. Amigó, **D. Spina**, and J. Gonzalo. ORMA: A Semi-Automatic Tool for Online Reputation Monitoring in Twitter. In *Proceedings of the 36th European Conference on Information Retrieval (ECIR)*, 2014.

[119] T. Martín, **D. Spina**, E. Amigó, and J. Gonzalo. UNED at RepLab 2012: Monitoring Task. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[141] M.H. Peetz, **D. Spina**, J. Gonzalo, and M. de Rijke. Towards an Active Learning System for Company Name Disambiguation in Microblog Streams. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*. CLEF, 2013.

[160] **D. Spina**, E. Amigó, and J. Gonzalo. Filter Keywords and Majority Class Strategies for Company Name Disambiguation in Twitter. In *CLEF 2011 Conference on Multilingual and Multimodal Information Access Evaluation*, pages 50–61, 2011.

[161] **D. Spina**, J. Carrillo de Albornoz, T. Martín, E. Amigó, J. Gonzalo, and F. Giner. UNED Online Reputation Monitoring Team at RepLab 2013. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*. CLEF, 2013.

[162] **D. Spina**, J. Gonzalo, and E. Amigó. Discovering filter keywords for company name disambiguation in twitter. *Expert Systems with Applications*, 40(12):4986 – 5003, 2013.

[163] **D. Spina**, Julio Gonzalo, and Enrique Amigó. Learning similarity functions for topic detection in online reputation monitoring. In *SIGIR '14: 37th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2014.

[164] **D. Spina**, E. Meij, M. de Rijke, A. Oghina, M.T. Bui, and M. Breuss. Identifying Entity Aspects in Microblog Posts. In *SIGIR'12 Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1089–1090. ACM, 2012.

[165] **D. Spina**, E. Meij, A. Oghina, M.T. Bui, M. Breuss, and M. de Rijke. A Corpus for Entity Profiling in Microblog Posts. In *LREC Workshop on Language Engineering for Online Reputation Management*, 2012.

[202] A. Zubiaga, **D. Spina**, E. Amigó, and J. Gonzalo. Towards Real-Time Summarization of Scheduled Events from Twitter Streams. In *Hypertext'12 Proceedings of the 23rd ACM conference on Hypertext and Social Media*, pages 319–320. ACM, 2012.

A

The WePS-3 ORM Task Dataset

TABLE A.1: Training entities in the WePS-3 Online Reputation Management Task dataset.

Query	**Entity Name**	|**Related**|/ |**Unrelated**| **Tweets**	**Total Tweets**
alcatel	alcatel	460/17	477
amadeus	Amadeus IT Group	64/417	481
apollo	Apollo Hospitals	9/436	445
armani	Giorgio Armani S.p.A.	329/119	448
barclays	barclays	335/133	468
bart	BART (Bay Area Rapid Transit)	216/248	464
bayer	bayer	292/148	440
blockbuster	Blockbuster Inc.	314/143	457
boingo	Boingo Wireless	185/255	440
bulldog	Bulldog Solutions	2/448	450
cadillac	cadillac	295/172	467
craft	Craft Magazine	9/456	465
delta	Delta Holding	0/479	479
dunlop	Dunlop Tyres	109/348	457
edmunds	Edmunds.com	156/280	436
elf	Elf corporation	14/468	482
emperor	Emperor Entertainment Group	8/482	490
fender	Fender Musical Instruments Corporation	302/152	454
folio	Folio Corporation	6/449	455
foxtel	Foxtel	479/6	485
fujitsu	Fujitsu	413/37	450
harpers	Harper's Magazine	181/297	478
impulse	Impulse (Records)	15/439	454
lamborghini	lamborghini	376/22	398
linux	linux	378/6	384
liquid	Liquid Entertainment	3/457	460
lufthansa	Lufthansa	460/10	470
luxor	Luxor Las Vegas	209/220	429
lynx	LYNX Express	3/472	475
mack	Mack Group	1/469	470
magnum	Magnum Research	26/427	453
mandalay	Mandalay Bay Resort and Casino	349/116	465
marriott	Marriott International	416/36	452
marvel	Marvel comics	404/55	459
mdm	mdm (Marketing and Design Matters)	21/450	471
mep	MEP	20/451	471
mercedes	Mercedes-Benz	356/103	459
mercer	Mercer consulting firm	16/442	458
mgm	MGM Grand Hotel and Casino	179/259	438
mta	Metropolitan Transportation Authority (New York)	163/282	445
nikon	nikon	439/5	444
nordic	Nordic Airways	9/467	476
philips	Royal Philips Electronics Inc.	360/79	439
pierce	pierce manufacturing	1/464	465
pioneer	Pioneer Company	33/425	458
renaissance	Renaissance Technologies	1/467	468
renault	Renault	404/10	414
rover	Land Rover	273/165	438
shin	shin corporation	2/471	473
smarter	Smarter Travel	5/446	451
southwest	Southwest Airlines	132/329	461
yamaha	Yamaha Corporation	451/19	470

TABLE A.2: Test entities in the WePS-3 Online Reputation Management Task dataset.

Query	**Entity Name**	**|Related|/ |Unrelated| Tweets**	**Total Tweets**
amazon	Amazon.com	404/20	424
apache	apache foundation	182/219	401
apple	Apple Inc.	372/74	446
blizzard	Blizzard Entertainment	145/282	427
camel	camel	21/420	441
canon	Canon inc.	379/46	425
cisco	Cisco Systems	318/76	394
cvs	CVS pharmacy	354/75	429
denver	Denver Nuggets	10/447	457
deutsche	Deutsche Bank	209/228	437
emory	Emory University	159/209	368
ford	Ford Motor Company	262/150	412
fox	Fox Entertainment Group	161/251	412
friday's	T.G.I. Friday's	56/377	433
gibson	Gibson Guitar Corporation	75/358	433
gm	General Motors	179/221	400
jaguar	Jaguar Cars Ltd.	221/194	415
jfk	John F. Kennedy International Airport	248/173	421
johnnie	Johnnie Walker	70/339	409
kiss	kiss band	20/412	432
lexus	Lexus	367/11	378
liverpool	Liverpool FC	139/284	423
lloyd	Lloyds Banking Group	23/408	431
mac	macintosh	187/135	322
mcdonald's	McDonald's	324/42	366
mclaren	McLaren Group	106/332	438
metro	Metro supermarket	2/368	370
milan	A.C. Milan	97/313	410
mtv	MTV	404/18	422
muse	muse band	277/113	390
oracle	Oracle Corporation	296/85	381
orange	Orange	17/441	458
paramount	Paramount Group	24/384	408
roma	A.S. Roma	54/386	440
scorpions	scorpions	294/115	409
seat	seat S.A.	1/414	415
sharp	Sharp Corporation	33/400	433
sonic	sonic.net	11/425	436
sony	sony	344/9	353
stanford	Stanford Junior University	237/130	367
starbucks	Starbucks	391/12	403
subway	subway	183/214	397
tesla	Tesla Motors	125/282	407
us	US Airways	1/438	439
virgin	Virgin Media	39/390	429
yale	Yale University	261/121	382
zoo	Zoo Entertainment	4/458	462

B

The RepLab 2012 Dataset

TABLE B.1: Trial entities in the RepLab 2012 dataset.

Entity Id	Query	Entity Name	|Related|/ |Unrelated| Tweets	Total Topics	Total Tweets
RL2012E00	apple	Apple Inc.	281/19	36	300
RL2012E01	lufthansa	Lufthansa Airlines	299/1	47	300
RL2012E02	alcatel	Alcatel-Lucent	289/11	36	300
RL2012E03	armani	Giorgio Armani S.P.A.	179/121	27	300
RL2012E04	barclays	Barclays PLC	298/2	39	300
RL2012E05	marriott	Marriott International	294/6	13	300

TABLE B.2: Test entities in the RepLab 2012 dataset.

Entity Id	**Query**	**Entity Name**	**|Related|/ |Unrelated| Tweets**	**Total Topics**	**Total Tweets**
RL2012E06	"gas natural"	Gas Natural SDG, S.A.	127/266	20	393
RL2012E07	telefonica	Telefónica, S.A.	129/64	30	193
RL2012E08	bbva	Banco Bilbao Vizcaya Argentaria, S.A.	199/0	17	199
RL2012E09	yahoo	Yahoo! Inc.	193/7	12	200
RL2012E10	repsol	Repsol S. A.	389/9	16	398
RL2012E11	bing	Bing	97/103	26	200
RL2012E12	indra	Indra Sistemas, S. A.	2/284	2	286
RL2012E13	google	Google Inc.	136/64	21	200
RL2012E14	endesa	Endesa, S.A.	182/17	26	199
RL2012E15	ING	ING Group	2/198	1	200
RL2012E16	bme	Bolsas y Mercados Españoles	0/216	0	216
RL2012E17	bankia	Bankia	211/0	12	211
RL2012E18	iberdrola	Iberdrola	200/0	42	200
RL2012E19	banco santander	Banco Santander, S.A.	203/5	19	208
RL2012E20	mediaset	Mediaset S.p.A.	184/0	18	184
RL2012E21	iag	International Consolidated Airlines Group, S.A.	125/75	14	200
RL2012E22	inditex	Industria de Diseño Textil, S.A.	204/0	26	204
RL2012E23	mapfre	MAPFRE	295/4	42	299
RL2012E24	bank of america	Bank of America Corporation	176/0	22	176
RL2012E25	blackberry	BlackBerry	182/17	25	199
RL2012E26	bmw	Bayerische Motoren Werke AG (BMW)	165/19	33	184
RL2012E27	bp	BP p.l.c.	5/190	4	195
RL2012E28	chevrolet	Chevrolet	192/7	20	199
RL2012E29	ferrari	Ferrari S.p.A.	171/29	14	200
RL2012E30	fiat	Fiat S.p.A.	164/33	34	197
RL2012E31	vw	Volkswagen	85/113	20	198
RL2012E32	wilkinson	Wilkinson Sword	9/196	2	205
RL2012E33	gillette	Gillette	139/57	22	196
RL2012E34	nivea	Nivea	141/60	9	201
RL2012E35	microsoft	Microsoft Corporation	197/3	56	200
RL2012E36	caixabank	CaixaBank	242/0	15	242

C

The RepLab 2013 Dataset

TABLE C.1: Entities in the RepLab 2013 dataset.

Entity Id	Query	Entity Name	Domain	Training			Test		
				|Rel.|/ |Unrel.| Tweets	Total Topics	Total Tweets	|Rel.|/ |Unrel.| Tweets	Total Topics	Total Tweets
RL2013D01E001	BMW	Bayerische Motoren Werke AG	automotive	683/61	60	744	1255/265	51	1520
RL2013D01E002	Audi	AUDI Aktiengesellschaft	automotive	633/66	63	699	1326/168	75	1494
RL2013D01E003	Volvo	AB Volvo	automotive	681/33	84	714	1452/78	139	1530
RL2013D01E005	Toyota	Toyota Motor Corporation	automotive	690/12	82	702	1515/14	135	1529
RL2013D01E008	Volkswagen	Volkswagen	automotive	707/0	84	707	1500/7	115	1507
RL2013D01E009	Honda	Honda Motor Company, Ltd.	automotive	716/74	109	790	1466/190	187	1656
RL2013D01E012	Nissan	Nissan Motor Company Ltd	automotive	691/13	42	704	1481/56	68	1537
RL2013D01E013	Fiat	Fiat S.p.A	automotive	653/99	96	752	1364/146	135	1510
RL2013D01E014	Suzuki	Suzuki Motor Corporation	automotive	462/394	45	856	1178/474	98	1652
RL2013D01E015	Mazda	Mazda Motor Corporation	automotive	671/31	56	702	1400/77	73	1477
RL2013D01E016	Chrysler	Chrysler Group LLC	automotive	714/3	97	717	1558/1	117	1559

Continued on next page

Table C.1 – continued from previous page

Entity Id	Query	Entity Name	Domain	Training			Test		
				|Rel.|/ |Unrel.| Tweets	Total Topics	Total Tweets	|Rel.|/ |Unrel.| Tweets	Total Topics	Total Tweets
RL2013D01E019	Subaru	Subaru	automotive	635/51	92	686	1314/117	73	1431
RL2013D01E022	Ferrari	Ferrari S.p.A	automotive	699/101	78	800	1505/299	133	1804
RL2013D01E025	Bentley	Bentley Motors Limited	automotive	221/568	32	789	562/1084	84	1646
RL2013D01E033	Porsche	Porsche Automobil Holding SE	automotive	642/137	39	779	1339/256	52	1595
RL2013D01E035	Yamaha	Yamaha Motor Company	automotive	453/431	61	884	1036/859	65	1895
RL2013D01E040	Kia	Kia Motors	automotive	259/545	63	804	536/1124	100	1660
RL2013D01E041	Ford	Ford Motor Company	automotive	391/391	51	782	898/816	66	1714
RL2013D01E043	Jaguar	Jaguar Cars Ltd.	automotive	347/356	47	703	750/740	66	1490
RL2013D01E044	Lexus	Lexus	automotive	408/401	108	809	980/599	127	1579
RL2013D02E051	RBS bank	The Royal Bank of Scotland plc	banking	309/394	103	703	532/969	143	1501
RL2013D02E054	Barclays	Barclays PLC	banking	746/1	62	747	1562/4	114	1566
RL2013D02E055	HSBC	HSBC Holdings plc	banking	791/6	154	797	1649/7	109	1656
RL2013D02E056	Bank of America	Bank of America Corporation	banking	696/1	66	697	1543/0	88	1543
RL2013D02E057	Wells Fargo	Wells Fargo & Company	banking	170/604	44	774	239/1325	64	1564

Continued on next page

Table C.1 – continued from previous page

Entity Id	Query	Entity Name	Domain	Training			Test		
				|Rel.|/ |Unrel.| Tweets	Total Topics	Total Tweets	|Rel.|/ |Unrel.| Tweets	Total Topics	Total Tweets
RL2013D02E060	PNC	PNC Financial Services Group, Inc.	banking	185/174	38	359	175/481	68	656
RL2013D02E063	Capital One	Capital One Financial Corporation	banking	711/8	71	719	1580/10	69	1590
RL2013D02E067	Santander	Banco Santander, S.A.	banking	275/496	61	771	584/1177	85	1761
RL2013D02E068	Bankia	Bankia S.A.	banking	741/19	29	760	1614/26	30	1640
RL2013D02E070	BBVA	Banco Bilbao Vizcaya Argentaria, S.A.	banking	715/1	103	716	1527/0	131	1527
RL2013D02E076	Goldman Sachs	The Goldman Sachs Group, Inc	banking	414/317	100	731	1048/569	220	1617
RL2013D03E086	harvard	Harvard University	university	629/94	53	723	1462/123	172	1585
RL2013D03E087	stanford	Stanford University	university	512/154	72	666	1142/273	213	1415
RL2013D03E088	berkeley	The University of California, Berkeley	university	189/525	77	714	419/1111	131	1530
RL2013D03E089	MIT	Massachusetts Institute of Technology	university	399/267	51	666	841/611	72	1452
RL2013D03E090	princeton	Princeton University	university	99/584	35	683	622/838	63	1460
RL2013D03E091	columbia	Columbia University	university	44/663	36	707	122/1331	62	1453
RL2013D03E093	yale	Yale University	university	547/165	98	712	1177/302	166	1479

Continued on next page

Table C.1 – continued from previous page

Entity Id	Query	Entity Name	Domain	Training			Test		
				|Rel.|/ |Unrel.| Tweets	Total Topics	Total Tweets	|Rel.|/ |Unrel.| Tweets	Total Topics	Total Tweets
RL2013D03E096	hopkins	The Johns Hopkins University	university	95/608	18	703	191/1353	24	1544
RL2013D03E097	nyu	New York University	university	665/23	36	688	1403/138	86	1541
RL2013D03E124	oxford	University of Oxford	university	233/465	27	698	336/1149	46	1485
RL2013D04E145	Adele	Adele	music	674/20	54	694	1449/94	67	1543
RL2013D04E146	Alicia Keys	Alicia Keys	music	728/2	69	730	1527/0	114	1527
RL2013D04E149	beatles	The Beatles	music	696/5	26	701	1503/28	44	1531
RL2013D04E151	Led Zeppelin	Led Zeppelin	music	908/0	52	908	1887/3	62	1890
RL2013D04E152	Aerosmith	Aerosmith	music	724/6	80	730	1561/8	147	1569
RL2013D04E153	Bon Jovi	Bon Jovi	music	796/20	60	816	1604/4	106	1608
RL2013D04E155	U2	U2	music	641/205	52	846	1331/518	67	1849
RL2013D04E159	AC/DC	AC/DC	music	592/306	80	898	1520/379	125	1899
RL2013D04E161	The wanted	The Wanted	music	750/20	48	770	1526/29	89	1555
RL2013D04E162	Maroon 5	Maroon 5	music	738/0	47	738	1682/2	87	1684
RL2013D04E164	Coldplay	Coldplay	music	707/2	32	709	1530/8	48	1538
RL2013D04E166	Lady Gaga	Lady Gaga	music	837/0	60	837	1893/1	87	1894
RL2013D04E167	madonna	Madonna	music	774/18	45	792	1525/78	56	1603

Continued on next page

Table C.1 – continued from previous page

Entity Id	Query	Entity Name	Domain	Training			Test		
				|Rel.|/ |Unrel.| Tweets	Total Topics	Total Tweets	|Rel.|/ |Unrel.| Tweets	Total Topics	Total Tweets
RL2013D04E169	Jennifer Lopez	Jennifer Lopez	music	858/4	36	862	1930/8	48	1938
RL2013D04E175	Justin Bieber	Justin Bieber	music	833/1	22	834	1779/11	31	1790
RL2013D04E185	Shakira	Shakira	music	890/54	55	944	1392/116	47	1508
RL2013D04E194	PSY	PSY	music	512/346	96	858	1835/137	105	1972
RL2013D04E198	The Script	The Script	music	400/343	45	743	987/542	82	1529
RL2013D04E206	Whitney Houston	Whitney Houston	music	733/0	52	733	1572/0	54	1572
RL2013D04E207	Britney Spears	Britney Spears	music	570/109	79	679	1254/245	176	1499

Bibliography

[1] N. Agarwal and H. Liu. Blogosphere: Research issues, tools, and applications. *SIGKDD Explor. Newsl.*, 10(1):18–31, 2008.

[2] E. Agirre and P. Edmonds. *Word sense disambiguation: Algorithms and applications*. Springer, 2006.

[3] R. Al-Kamha and D. Embley. Grouping search-engine returned citations for person-name queries. In *Proceedings of the 6th annual ACM international workshop on Web information and data management*, pages 96–103, 2004.

[4] D. Albakour, C. Macdonald, and I. Ounis. Identifying local events by using microblogs as social sensors. In *Proceedings of OAIR 2013, the 10th Conference on Open Research Areas in Information Retrieval*, 2013.

[5] J. Allan. Introduction to Topic Detection and Tracking. In *Topic Detection and Tracking*, volume 12 of *The Information Retrieval Series*, pages 1–16. Springer US, 2002.

[6] J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.

[7] E. Amigó, J. Artiles, J. Gonzalo, D. Spina, B. Liu, and A. Corujo. WePS-3 Evaluation Campaign: Overview of the Online Reputation Management Task. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[8] E. Amigó, J. Carrillo de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martín, E. Meij, M. de Rijke, and D. Spina. Overview of RepLab 2013: Evaluating online reputation monitoring systems. In *Proceedings of CLEF '13*, pages 333–352, 2013.

[9] E. Amigó, A. Corujo, J. Gonzalo, E. Meij, and M. de Rijke. Overview of RepLab 2012: Evaluating online reputation management systems. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[10] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009.

[11] E. Amigó, J. Gonzalo, and F. Verdejo. A General Evaluation Measure for Document Organization Tasks. In *Proceedings of SIGIR'13*, 2013.

[12] J. Artiles. *Web People Search*. PhD thesis, UNED University, 2009.

[13] J. Artiles, E. Amigó, and J. Gonzalo. The role of named entities in web people search. In *Proceedings of EMNLP'09*, 2009.

[14] J. Artiles, A. Borthwick, J. Gonzalo, S. Sekine, and E. Amigó. Weps-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[15] J. Artiles, J. Gonzalo, and S. Sekine. The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. *Proceedings of Semeval-2007*, 2007.

[16] J. Artiles, J. Gonzalo, and S. Sekine. Weps 2 evaluation campaign: overview of the web people search clustering task. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[17] S. Asur, B. A. Huberman, G. Szabó, and C. Wang. Trends in social media : Persistence and decay. *Arxiv preprint arXiv:1102.1402*, 2011.

[18] J. Atserias, G. Attardi, M. Simi, and H. Zaragoza. Active learning for building a corpus of questions for parsing. In *Proceedings of LREC'10*. ELRA, 2010.

[19] L. Azzopardi and K. Balog. Towards a living lab for information retrieval research and development. In *Multilingual and Multimodal Information Access Evaluation*, pages 26–37. Springer, 2011.

[20] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (CoLing-ACL '98)*, pages 79–85, 1998.

[21] Balahur, Alexandra and Tanev, Hristo. Detecting Entity-Related Events and Sentiments from Tweets Using Multilingual Resources. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[22] H. Becker, D. Iter, M. Naaman, and L. Gravano. Identifying content for planned events across social media sites. In *Proceedings of the fifth ACM international conference on Web search and data mining (WSDM '12)*, pages 533–542, 2012.

[23] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of WSDM'10*, pages 291–300, 2010.

[24] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. In *Proceedings of ICWSM-11*, volume 11, pages 438–441, 2011.

[25] J. Benhardus and J. Kalita. Streaming Trend Detection in Twitter. *Int. J. Web Based Communities*, 9(1):122–139, 2013.

[26] J. L. A. Berrocal, C. G. Figuerola, and Á. Zazo Rodríguez. REINA at RepLab2013 Topic Detection Task: Community Detection. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[27] D. M. Blei and J. D. Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10:71, 2009.

[28] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3, 2003.

[29] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences (HICSS'10)*, pages 1–10, 2010.

[30] Brandchats. Brandchats, 2014. http://www.brandchats.com. [Accessed 22th April 2014].

[31] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL'06*, volume 6, pages 9–16, 2006.

[32] J. Carrillo-de Albornoz, E. Amigó, D. Spina, and J. Gonzalo. ORMA: A Semi-Automatic Tool for Online Reputation Monitoring in Twitter. In *Proceedings of the 36th European Conference on Information Retrieval (ECIR)*, 2014.

[33] J. Carrillo-de Albornoz, I. Chugur, and E. Amigó. Using an emotion-based model and sentiment analysis techniques to classify polarity for reputation. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[34] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM-10)*, 2010.

[35] D. Chakrabarti and K. Punera. Event summarization using tweets. In *Proceedings of the fifth International AAAI Conference on Weblogs and Social Media (ICWSM-11)*, pages 66–73, 2011.

[36] Y. Chen, H. Amiri, Z. Li, and T.-S. Chua. Emerging Topic Detection for Organizations from Microblogs. In *Proceedings of SIGIR'13*, 2013.

[37] Y. Chen, S. Y. M. Lee, and C.-R. Huang. PolyUHK: A robust information extraction system for web personal names. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS), WWW'09*, 2009.

[38] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM'10)*, pages 759–768, 2010.

[39] J. M. Chenlo, J. Atserias, C. Rodriguez, and R. Blanco. Fbm-yahoo! at replab 2012. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[40] R. L. Cilibrasi and P. M. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 2007.

[41] G. Cormack and T. Lynam. Trec 2005 spam track overview. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, 2005.

[42] J.-V. Cossu, B. Bigot, L. Bonnefoy, M. Morchid, X. Bost, G. Senay, R. Dufour, V. Bouvier, J.-M. Torres-Moreno, and M. El-Beze. LIA@RepLab 2013. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[43] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL'07*, pages 708–716, 2007.

[44] C. Dellarocas, N. Awad, and X. Zhang. Exploring the value of online reviews to organizations: Implications for revenue forecasting and planning. In *Proceedings of the International Conference on Information Systems*, 2004.

[45] L. Derczynski, D. Maynard, N. Aswani, and K. Bontcheva. Microblog-genre noise and impact on semantic annotation accuracy. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 21–30, 2013.

[46] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (CoLing'10)*, pages 277–285, 2010.

[47] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19, 1993.

[48] P. Edmonds and S. Cotton. Senseval-2: Overview. In *Proceedings of The Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, pages 1–6, 2001.

[49] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP'10)*, 2010.

[50] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[51] P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM'10)*, pages 1625–1628, 2010.

[52] R. L. Figueroa, Q. Zeng-Treitler, L. H. Ngo, S. Goryachev, and E. P. Wiechmann. Active learning for clinical text classification: is it better than random sampling? *JAMIA*, 19(5):809–816, 2012.

[53] J. G. Fiscus and G. R. Doddington. Topic Detection and Tracking Evaluation Overview. In *Topic Detection and Tracking*, volume 12 of *The Information Retrieval Series*, pages 17–31. 2002.

[54] C. Fombrun. *Reputation. Realizing Value from the Corporate Image*. Harvard Business School Press, 1996.

[55] C. Fombrun and M. Shanley. What's in a name? Reputation building and corporate strategy. *Academy of Management Journal*, 33(2):233–258, 1990.

[56] E. Frank, G. Paynter, I. Witten, C. Gutwin, and C. Nevill-Manning. Domain-specific keyphrase extraction. In *International joint conference on artificial intelligence*, volume 16, pages 668–673, 1999.

[57] M. A. García-Cumbreras, M. García-Vega, F. Martínez-Santiago, and J. M. Peréa-Ortega. SINAI at WePS-3: Online Reputation Management. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[58] A. Gentile, Z. Zhang, L. Xia, and J. Iria. Semantic relatedness approach for named entity disambiguation. In *Digital Libraries*, pages 137–148, 2010.

[59] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo. Deriving Marketing Intelligence from Online Discussion. In *Proceedings of 11th ACM International Conference on Knowledge Discovery and Data Mining (KDD'05).*, 2005.

[60] N. S. Glance, M. Hurst, and T. Tomokiyo. Blogpulse: Automated trend discovery for weblogs. In *Proceedings of the WWW'04 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2004.

[61] C. Gooi and J. Allan. Cross-document coreference on a large scale corpus. In *Proceedings of HLT/NAACL'04*, volume 4, 2004.

[62] S. Gouws, D. Metzler, C. Cai, E. Hovy, and C. Marina del Rey. Contextual bearing on linguistic variation in social media. In *ACL Workshop on Language in Social Media (LSM 2011)*, 2011.

[63] M. A. Greenwood, N. Aswani, and K. Bontcheva. Reputation profiling with gate. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[64] R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th conference on Computational linguistics (CoLing'96)*, pages 466–471, 1996.

[65] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information Diffusion Through Blogspace. In *Proceedings of WWW'04*, pages 491–501, 2004.

[66] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh. *Feature extraction: foundations and applications*, chapter 2.2, pages 67–78. Springer Verlag, 2006.

[67] V. Hangya and R. Farkas. Filtering and polarity detection for reputation management on tweets. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[68] V. Hangya and R. Farkas. Target-oriented opinion mining from tweets. In *Proceedings of the IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom '13)*, pages 251–254. IEEE, 2013.

[69] J. Hannon, K. McCarthy, J. Lynch, and B. Smyth. Personalized and automatic social summarization of events in video. In *Proceedings of the 16th international conference on Intelligent user interfaces (IUI '11)*, pages 335–338, 2011.

[70] D. Harman. Overview of the Fourth Text REtrieval Conference (TREC-4). In *TREC-4*, 1995.

[71] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings of the 27th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '04)*, 2004.

[72] T. Hoffman. Online reputation management is hot?but is it ethical? *Computerworld, February*, 2008.

[73] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of SIGIR'99*, 1999.

[74] Holopedia. Holopedia Project, 2014. http://nlp.uned.es/holopedia/. [Accessed 22th April 2014].

[75] L. Hong and B. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88, 2010.

[76] R. Hu. *Active learning for text classification*. PhD thesis, Dublin Institute of Technology, 2011.

[77] D. Hull et al. The trec-7 filtering track: description and analysis. *NIST SPECIAL PUBLICATION SP*, pages 45–68, 1998.

[78] D. Hull and S. Robertson. The trec-8 filtering track final report. In *Proceeding of eighth Text Retrieval Conference (TREC-8)*, 1999.

[79] D. Inouye and J. Kalita. Comparing twitter summarization algorithms for multiple post summaries. In *Proceedings of the IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing (PASSAT/SocialCom'11)*, pages 298–306, 2011.

[80] B. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188, 2009.

[81] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[82] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65, 2007.

[83] H. Ji and R. Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL'11)*, volume 1, pages 1148–1158, 2011.

[84] H. Ji, R. Grishman, H. Dang, K. Griffitt, and J. Ellis. Overview of the tac 2010 knowledge base population track. In *TAC (Text Analysis Conference) 2010 Workshop*, 2010.

[85] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160, 2011.

[86] V. Jijkoun, M. de Rijke, and W. Weerkamp. Generating focused topic-specific sentiment lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, 2010.

[87] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98*, 1998.

[88] D. Jurgens and K. Stevens. The s-space package: an open source package for word space models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 30–35, 2010.

[89] D. Kalashnikov, S. Mehrotra, Z. Chen, R. Nuray-Turan, and N. Ashish. Disambiguation algorithm for people search on the web. In *Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE'07)*, pages 1258–1260, 2007.

[90] P. Kalmar. Bootstrapping Websites for Classification of Organization Names on Twitter. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[91] R. Kaptein. Learning to analyze relevancy and polarity of tweets. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[92] J. Karlgren, M. Sahlgren, F. Olsson, F. Espinoza, and O. Hamfors. Profiling reputation of corporate entities in semantic space: Notebook for replab at clef 2012. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[93] M. Kaufmann and J. Kalita. Syntactic normalization of twitter messages. In *International Conference on Natural Language Processing, Kharagpur, India*, 2010.

[94] M. Khalid, V. Jijkoun, and M. de Rijke. The Impact of Named Entity Normalization on Information Retrieval for Question Answering. In *Proceedings of the 30th European Conference on Information Retrieval (ECIR 2008)*, pages 705–710, 2008.

[95] S. Kim, O. Medelyan, M. Kan, and T. Baldwin. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, 2010.

[96] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, 2003.

[97] D. L. Kreher and D. R. Stinson. *Combinatorial algorithms: generation, enumeration, and search*, volume 7. CRC press, 1998.

[98] B. Krishnamurthy, P. Gill, and M. Arlitt. A few chirps about twitter. In *Proceedings of the first workshop on Online social networks (WOSP'08)*, pages 19–24, 2008.

[99] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466, 2009.

[100] S. Kullback and R. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[101] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web (WWW '10)*, pages 591–600, 2010.

[102] G. Laboreiro, L. Sarmento, J. Teixeira, and E. Oliveira. Tokenizing micro-blogging messages using a text classification approach. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pages 81–88, 2010.

[103] A. Lamb, M. J. Paul, and M. Dredze. Separating Fact from Fear: Tracking Flu Infections on Twitter. In *Proceedings of the North American Chapter of the Association for Computational Linguistics, Human Language Technologies (NAACL-HLT 2013)*, 2013.

[104] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):pp. 159–174, 1977.

[105] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33, 1977.

[106] K. Lerman and R. Ghosh. Information Contagion: an Empiric Study of the Spread of News on Digg and Twitter Social Networks. In *Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM-10)*, 2010.

[107] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*, pages 497–506, 2009.

[108] LiMoSINe. LiMoSINe EU Project, 2014. http://www.limosine-project.eu/. [Accessed 22th April 2014].

[109] D. Lin. An information-theoretic definition of similarity. *Proceedings of ICML'98*, 1998.

[110] B. Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[111] W. Liu and T. Wang. Active learning for online spam filtering. In *Proceedings of AIRS '08*, pages 555–560, 2008.

[112] X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu. Entity linking for tweets. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL'13)*, 2013.

[113] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: Joint models of topic and author community. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09)*, pages 665–672, 2009.

[114] G. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of HLT-NAACL'03*, pages 33–40, 2003.

[115] G. S. Mann. *Multi-document Statistical Fact Extraction and Fusion*. PhD thesis, Johns Hopkins University, 2006.

[116] G. S. Mann. *Multi-document Statistical Fact Extraction and Fusion*. PhD thesis, 2006.

[117] H. Mann and D. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947.

[118] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[119] T. Martín, D. Spina, E. Amigó, and J. Gonzalo. UNED at RepLab 2012: Monitoring Task. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[120] T. Martín-Wanton, J. Gonzalo, and E. Amigó. An unsupervised transfer learning approach to discover topics for online reputation management. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1565–1568, 2013.

[121] J. Martínez-Romo and L. Araujo. Web people search disambiguation using language model techniques. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[122] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of SIGMOD'10*, pages 1155–1158, 2010.

[123] P. McNamee and H. Dang. Overview of the TAC 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, 2009.

[124] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*, pages 171–180, 2007.

[125] Mei, Qiaozhu and Liu, Chao and Su, Hang and Zhai, ChengXiang. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th international conference on World Wide Web (WWW'06)*, pages 533–542, 2006.

[126] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining (WSDM'12)*, 2012.

[127] M. Michelson and S. Macskassy. Discovering users' topics of interest on twitter: a first look. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pages 73–80, 2010.

[128] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. YALE: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD*, 2006.

[129] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of CIKM'07*, volume 7, pages 233–242, 2007.

[130] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proceedings of EMNLP'04*, volume 4, pages 404–411, 2004.

[131] E. Milios, Y. Zhang, B. He, and L. Dong. Automatic term extraction and document similarity in special text corpora. In *Proceedings of the Sixth Conference of the Pacific Association for Computational Linguistics*, pages 275–284, 2003.

[132] D. Milne and I. Witten. Learning to link with wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM'08)*, pages 509–518, 2008.

[133] S. Milstein, A. Chowdhury, G. Hochmuth, B. Lorica, and R. Magoulas. *Twitter and the Micro-Messaging Revolution: Communication, Connections, and Immediacy–140 Characters at a Time*. O'Reilly Media / Radar, 2008.

[134] S. Moghaddam and M. Ester. On the Design of LDA Models for Aspect-based Opinion Mining. In *Proceedings of CIKM'12*, 2012.

[135] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[136] Nielsen-Online. Nielsen BuzzMetrics, 2014. http://www.nielsen-online.com/products_buzz.jsp?section=pro_buzz. [Accessed 22th April 2014].

[137] R. Nuray-Turan, Z. Chen, D. V. Kalashnikov, and S. Mehrotra. Exploiting Web Querying for Web People Search in WePS2. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS), WWW'09*, 2009.

[138] B. O'Connor, M. Krieger, and D. Ahn. Tweetmotif: Exploratory search and topic summarization for Twitter. pages 2–3, 2010.

[139] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[140] M. J. Paul and M. Dredze. You are what you Tweet: Analyzing Twitter for Public Health. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM-11)*, 2011.

[141] M. Peetz, D. Spina, J. Gonzalo, and M. de Rijke. Towards an Active Learning System for Company Name Disambiguation in Microblog Streams. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[142] M.-H. Peetz, M. de Rijke, and A. Schuth. From sentiment to reputation. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[143] S. Petrovic, M. Osborne, and V. Lavrenko. The Edinburgh Twitter corpus. In *Proceedings of the HLT-NAACL 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 25–26, 2010.

[144] I. Pollach. Electronic word of mouth: A genre analysis of product reviews on consumer opinion web sites. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, volume 3, 2006.

[145] L. Ponzi, C. Fombrun, and N. Gardberg. RepTrak Pulse: Conceptualizing and Validating a Short-Form Measure of Corporate Reputation. *Corporate Reputation Review*, 14(1), 2011.

[146] J. Quinlan. *C4. 5: programs for machine learning*. Morgan Kaufmann, 1993.

[147] Qureshi, Muhammad Atif and O'Riordan,Colm and Pasi, Gabriella . Concept Term Expansion Approach for Monitoring Reputation of Companies on Twitter. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[148] D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *Proceedings of the Fourth International Conference on Weblogs and Social Media (ICWSM-10)*, 2010.

[149] Repustate. Repustate, 2014. https://www.repustate.com. [Accessed 22th April 2014].

[150] 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 (WWW'10)*, pages 851–860, 2010.

[151] P. Saleiro, L. Rei, A. Pasquali, C. Soares, J. Teixeira, F. Pinto, M. Nozari, C. Felix, and P. Strecht. POPSTAR at RepLab 2013: Name ambiguity resolution on Twitter. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[152] Salesforce. Salesforce ExactTarget Marketing Cloud (previous Radian6), 2014. http://www.salesforcemarketingcloud.com/. [Accessed 22th April 2014].

[153] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513 – 523, 1988.

[154] C. Sanchez-Sanchez, H. Jimenez-Salazar, and W. A. Luna-Ramirez. UAMCLyR at Replab2013: Monitoring task. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[155] M. Sassano. An empirical study of active learning with support vector machines for japanese word segmentation. In *Proceedings of the ACL '02*, pages 505–512, 2002.

[156] Semantria. Semantria, 2014. https://semantria.com. [Accessed 22th April 2014].

[157] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[158] B. Sharifi, M.-A. Hutton, and J. Kalita. Summarizing microblogs automatically. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '10)*, pages 685–688, 2010.

[159] SocialMention. SocialMention, 2014. http://www.socialmention.com. [Accessed 22th April 2014].

[160] D. Spina, E. Amigó, and J. Gonzalo. Filter Keywords and Majority Class Strategies for Company Name Disambiguation in Twitter. In *CLEF 2011 Conference on Multilingual and Multimodal Information Access Evaluation*, pages 50–61, 2011.

[161] D. Spina, J. Carrillo de Albornoz, T. Martín, E. Amigó, J. Gonzalo, and F. Giner. UNED Online Reputation Monitoring Team at RepLab 2013. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[162] D. Spina, J. Gonzalo, and E. Amigó. Discovering filter keywords for company name disambiguation in twitter. *Expert Systems with Applications*, 40(12):4986 – 5003, 2013.

[163] D. Spina, J. Gonzalo, and E. Amigó. Learning Similarity Functions for Topic Detection in Online Reputation Monitoring. In *Proceedings of SIGIR'14*, 2014.

[164] D. Spina, E. Meij, M. de Rijke, A. Oghina, M. T. Bui, and M. Breuss. Identifying entity aspects in microblog posts. In *SIGIR*, pages 1089–1090, 2012.

[165] D. Spina, E. Meij, A. Oghina, M. T. Bui, M. Breuss, and M. de Rijke. A Corpus for Entity Profiling in Microblog Posts. In *LREC Workshop on Language Engineering for Online Reputation Management*, 2012.

[166] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'10)*, pages 841–842, 2010.

[167] M. Strube and S. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1419, 2006.

[168] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, Mar. 2002.

[169] Trackur. Trackur, 2014. http://www.trackur.com. [Accessed 22th April 2014].

[170] M. Tsagkias. *Mining Social Media: Tracking Content and Predicting Behavior*. PhD thesis, University of Amsterdam, 2012.

[171] M. Tsagkias and K. Balog. The University of Amsterdam at WePS3. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[172] M. Tsagkias, M. de Rijke, and W. Weerkamp. Linking online news and social media. In *Proceedings of the fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 565–574, 2011.

[173] P. Turney. Learning to extract keyphrases from text, national research council. *Institute for Information Technology, Technical Report ERB-1057*, 1999.

[174] J. Villena-Román, S. Lana-Serrano, C. Moreno, J. García-Morera, and J. C. G. Cristóbal. Daedalus at replab 2012: Polarity classification and filtering on twitter data. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[175] Viralheat. Viralheat, 2014. https://www.viralheat.com. [Accessed 22th April 2014].

[176] E. M. Voorhees and D. K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. 2005.

[177] S. Wagner and D. Wagner. Comparing clusterings: an overview. Technical report, Faculty of Informatics, Universität Karlsruhe (TH), 2007.

[178] X. Wan, J. Gao, M. Li, and B. Ding. Person resolution in person search results: Webhawk. In *Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM'05)*, pages 163–170, 2005.

[179] X. Wang and A. McCallum. Topics over Time: A non-Markov Continuous-time Model of Topical Trends. In *Proceedings of KDD'06*, 2006.

[180] J. Weng, Y. Yao, E. Leonardi, and F. Lee. Event detection in twitter. Technical Report HPL-2011-98, HP Laboratories, 2011.

[181] J. Wiebe, T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2):165–210, 2005.

[182] R. Wilson. Keeping a watch on corporate reputation. *Strategic Communications Management*, 7(2), 2003.

[183] T. Wilson, Z. Kozareva, P. Nakov, S. Rosenthal, V. Stoyanov, and A. Ritter. SemEval-2013 task 2: Sentiment analysis in twitter. *Proceedings of the International Workshop on Semantic Evaluation, SemEval*, 13, 2013.

[184] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[185] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. Kea: practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries (DL '99)*, pages 254–255, 1999.

[186] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *Proceedings of ECIR'07*, pages 246–257, 2007.

[187] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. Representative sampling for text classification using support vector machines. In *Proceedings of ECIR '03*, pages 393–407, 2003.

[188] C. Yang, S. Bhattacharya, and P. Srinivasan. Lexical and machine learning approaches toward online reputation management. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[189] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. *Proceedings of the IEEE International Conference on Data Mining (ICDM'10)*, 0:599–608, 2010.

[190] S. R. Yerva, Z. Miklós, and K. Aberer. It was easy when apples and blackberries were only fruits. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[191] S. R. Yerva, Z. Miklós, and K. Aberer. Entity-based classification of twitter messages. *IJCSA*, 9(1):88–115, 2012.

[192] M. Yoshida, S. Matsushima, S. Ono, I. Sato, and H. Nakagawa. ITC-UT: Tweet Categorization by Query Categorization for On-line Reputation Management. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[193] A. Younus, C. O'Riordan, and G. Pasi. Cirgdisco at replab2012 filtering task: A two-pass approach for company name disambiguation in tweets. In *CLEF 2012 Eval. Labs and Workshop Notebook Papers*, 2012.

[194] S. Zhang, J. Wu, D. Zheng, Y. Meng, Y. Xia, and H. Yu. Two stages based organization name disambiguity. *Computational Linguistics and Intelligent Text Processing*, pages 249–257, 2012.

[195] Y. Zhang, E. Milios, and N. Zincir-Heywood. A comparative study on key phrase extraction methods in automatic web site summarization. *Journal of Digital Information Management*, 5(5):323, 2007.

[196] Y. Zhang, N. Zincir-Heywood, and E. Milios. Term-based clustering and summarization of web page collections. *Advances in Artificial Intelligence*, pages 60–74, 2004.

[197] Y. Zhang, N. Zincir-Heywood, and E. Milios. World wide web site summarization. *Web Intelligence and Agent Systems*, 2:39–54, 2004.

[198] Y. Zhang, N. Zincir-Heywood, and E. Milios. Narrative text classification for automatic key phrase extraction in web document corpora. In *Proceedings of the 7th annual ACM international workshop on Web information and data management*, pages 51–58, 2005.

[199] S. Zhao, L. Zhong, J. Wickramasuriya, and V. Vasudevan. Human as real-time sensors of social and physical events: A case study of twitter and sports games. *Arxiv preprint arXiv:1106.4300*, 2011.

[200] 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 (ACL'11)*, pages 379–388, 2011.

[201] 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 *Proceedings of the 33rd European conference on Advances in Information Retrieval (ECIR'11)*, 2011.

[202] A. Zubiaga, D. Spina, E. Amigó, and J. Gonzalo. Towards Real-Time Summarization of Scheduled Events from Twitter Streams. In *Hypertext'12 Proceedings of the 23rd ACM conference on Hypertext and Social Media*, pages 319–320, 2012.

[203] A. Zubiaga, D. Spina, R. Martínez, and V. Fresno. Real-time classification of twitter trends. *Journal of the Association for Information Science and Technology (JASIST)*, 2014.