Kennisextractie en populariteitsmodellering via sociale media

Knowledge Extraction and Popularity Modeling Using Social Media

Steven Van Canneyt

Promotoren: prof. dr. ir. B. Dhoedt, prof. dr. S. Schockaert, dr. ir. T. Demeester
Proefschrift ingediend tot het behalen van de graad van
Doctor in de ingenieurswetenschappen: computerwetenschappen

UNIVERSITEIT
GENT

Vakgroep Informatietechnologie
Voorzitter: prof. dr. ir. D. De Zutter
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2016 - 2017

ISBN 978-90-8578-931-4
NUR 980
Wettelijk depot: D/2016/10.500/63

UNIVERSITEIT
GENT

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Informatietechnologie

Promotoren: prof. dr. ir. Bart Dhoedt
prof. dr. Steven Schockaert
dr. ir. Thomas Demeester

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur

Vakgroep Informatietechnologie
Technologiepark-Zwijnaarde 15, B-9052 Gent, België

Tel.: +32-9-331.49.00
Fax.: +32-9-331.48.99

Dit werk kwam tot stand in het kader van een specialisatiebeurs van het IWT-Vlaanderen (Agentschap voor Innovatie door Wetenschap en Technologie in Vlaanderen).

Proefschrift tot het behalen van de graad van Doctor in de ingenieurswetenschappen: computerwetenschappen
Academiejaar 2016-2017

Dankwoord

Na 5 jaar onderzoek verrichten, papers schrijven, zwoegen, zweten, en ook af en toe genieten van het leven, ben ik toe aan het schrijven van de laatste tekst van mijn doctoraat, het dankwoord. Al wordt soms gezegd dat het afleggen van een doctoraat vaak een individueel traject is, kon ik het nooit afronden zonder de hulp van velen die ik hier wil bedanken.

Vooreerst wil ik een goeie 5 jaar terug gaan in de tijd, toen ik mijn thesis aan het afleggen was over het aanbevelen van toeristische plaatsen aan de hand van sociale media. Dank aan Olivier Van Laere en Steven Schockaert om me kennis te laten maken met machine learning in de context van sociale media, iets wat me erg in de smaak viel. Ik ging dan ook snel in op de vraag of ik wilde verder werken op dit onderwerp in het kader van een doctoraat. Ik ben mijn promotoren Steven Schockaert, Thomas Demeester en Bart Dhoedt erg dankbaar voor hun dagelijkse begeleiding, het aanleren van hoe ik onderzoek op een grondige en correcte manier kan uitvoeren, het nalezen van mijn papers en zoveel meer. Daarnaast wil ik ook graag Piet Demeester danken als drijvende kracht achter onderzoeksgroep IBCN, om voor een aangename en inspirerende omgeving te zorgen, en om onze groep te laten uitblinken tijdens de vele fusies (IBCN naar IBCN+, samenwerking tussen IBCN+ en Data Science Lab, en de fusie van iMinds met imec).

Mijn doctoraat was nooit mogelijk zonder de goede administratieve steun. Bedankt voor het plannen en boeken van de conferenties waaraan ik deelnam, de hulp bij het in orde brengen van het vele papierwerk voor het starten, verlengen en eindigen van mijn doctoraat en IWT beurs, het beheren van de vele computers en rekenclusters, en zoveel meer. Naast de administratieve steun, ben ik de financiele ondersteuning van het IWT ook heel dankbaar. Zonder deze beurs zou ik nooit de vrijheid gehad hebben tijdens mijn doctoraat zoals ik die nu heb kunnen ervaren.

Het feit is dat dit hoofdstuk waarschijnlijk de meest gelezen pagina's zullen zijn van mijn doctoraatsboek. Dit vooral door mijn collega's, om te kijken of hun naam tussen de lijst staat van de mensen die ik wil bedanken. Speciaal voor hen:
Toen ik nog maar net begon met mijn doctoraatsonderzoek en nog moest nadenken waarop ik me precies ging focussen tijdens mijn onderzoek, zag ik hoe verschillende bureaugenoten op Bureau 2.21 in de Zuiderpoort zwoegden over het finaliseren van hun publicaties, doctoraatsboek en hun carrière bij IBCN. Klaas Roobroeck, Niels Sluijs, Kristof Steurbaut en Olivier Van Laere, bedankt om me in te wijden in het leven van een doctoraatstudent. Hun bureaus werden ingevuld

door verschillende personen waarmee ik een geweldige tijd heb meegemaakt. Bedankt Steven Bohez, Maxim Claeys, Stefano Petrangeli, Jeroen Schaballie, Merlijn Sebrechts, Piet Smet en Thomas Vanhove. Ik hoop dat mijn -nu lege- bureau ook door een interessante en toffe persoon wordt opgevuld. In het bijzonder wil ik Niels Bouten bedanken. We zijn samen aan het doctoraatsavontuur begonnen, samen ons IWT aangevraagd en gehaald, ongeveer samen ons doctoraatsboek afgewerkt, onze verdedigingen voorbereid en afgelegd, en gesolliciteerd voor een job voor na ons doctoraat. Door deze gemeenschappelijke uitdagingen konden we altijd bij elkaar terecht, en met een grote dosis aan humor en relativeringsvermogen raakten we er altijd door. 3 maanden voor mijn vertrek bij IBCN zijn we verhuisd naar de iGent toren en kon ik nog even genieten van het prachtig zicht, en leerde ik nieuwe bureaugenoten beter kennen. Cedric De Boom, Elias De Coninck, Sam Leroux en Jeroen van der Hooft, hou de spirit hoog en de kwaliteit van het onderzoek nog hoger in Bureau 200.026. Als laatste bureaugenoten wil ik de echte IBCN survivals bedanken, de personen die lang voor mij begonnen werken zijn bij IBCN en die dat waarschijnlijk nog lang zullen doen. Philip Leroux, Wim Van de Meerssche, Bert Vankeirsbilck en Tim Verbelen, IBCN zou niet hetzelfde zijn zonder jullie.

Naast onderzoekswerk hoort er vaak ook projectwerk bij. Na verschillende pogingen kwam Philip Leroux af met een project, Providence, dat erg leuk en uitdagend leek om aan mee te werken. Tijdens het project kon ik voor het eerst machine learning toepassen op een ‘echte use case’, leerde ik vage vereisten omzetten naar specifieke functionaliteiten, maakte ik kennis met big data technologieën... en leerde ik veel nieuwe interessante mensen kennen. Philip Leroux, Thomas Demeester, Thomas Vanhove, en de mensen van iMinds, Newsmonkey, VRT, VUB en Massive Media, bedankt voor de aangename samenwerking. Ik heb me helemaal kunnen uitleven tijdens dit project, en het doet me dan ook veel deugd dat er een vervolg komt met Providence+.

Ten slotte wil ik natuurlijk alle andere collega’s bedanken. Jonas Anseeuw bijvoorbeeld, voor de leuke gesprekken in verband met ondernemen en het nut van bepaalde soorten onderzoek, en Pieter Bonte, voor de verhalen van al zijn vrienden die een succesvolle carrière aan het maken zijn bij onder andere Google en Apple, en de vele anderen die ik leerde kennen in de wandelgangen en tijdens de IBCN evenementen. Allemaal heel erg bedankt voor de mooie tijd bij IBCN!

Als aanloop naar mijn doctoraat was er natuurlijk de prachtige studententijd, met de feestjes en natuurlijk het veel bijleren aan de universiteit. Hierbij wil ik eerst en vooral mijn dichtste vrienden bedanken. Bedankt SPAM (Steven, Pieter De Smet, Alexander ‘den Alex’ Christiaen, en Mathieu ‘Mathew’ Desoete)! Het waren super leuke tijden op het einde van ons middelbaar, de jaarlijkse reizen naar zowat elke grote stad in Europa, het samen TV kijken bij de Mathew... en zoveel meer. Dat we nog veel mogen samen komen, samen vieren en samen op reis gaan. Daarnaast wil ik mijn studiegenoten Simon Buelens, Ewout Meyns, Mattias Putman en Thomas Roelens bedanken voor de prachtige studententijd. Na het werken tijdens de week in Gent, gingen we zaterdagavond ‘rustig’ iets drinken met Pieter

De Smet, Pieter Helewaut en Ion Dhondt. Deze ‘Bende van Brugge’ werd al vlug uitgebreid met David Bamelis, Tom Janssens, Xavier Samyn en Brecht Van Langenhove. Bedankt allemaal voor de gezellige ontspannende momenten!

Ik wil zeker mijn ouders bedanken voor de vele steun tijdens mijn studies en mijn doctoraat. De zorgende steun als ‘je kan maar je best doen’ van mama en de ondernemende visie zoals ‘het is nu dat je je moet bewijzen’ van papa zijn intussen slogans voor het leven geworden. Het doet me dan ook heel veel plezier om mijn ouders fier te zien bij het behalen van mijn doctoraat, en mijn internship bij Yahoo en werk bij Realo dat er kort op volgt. Ik bedank ook mijn oudere broers Wouter en Koen die altijd een voorbeeld voor me geweest zijn. Met een goed gevoel voor wiskunde in onze genen, werden we allemaal ingenieur. Ik zag dat Koen wel ‘een mooi leven’ had als doctoraatstudent, en ook daarin volgde ik zijn voetsporen. Helaas zal ik jullie niet volgen bij ingenieursbureau Ingenium in Brugge, maar zal ik mijn carrière verder zetten bij startup Realo in Gent. De afgelopen jaren zorgden jullie ervoor dat ik trotse nonkel, en zelf nog trotste peter, kon worden van prachtige kinderen. Bedankt Pepijn, Mien, Nel en Paula, om het kind in me terug naar boven te halen zodat jullie eens goed zot kunnen doen met jullie nonkel. Dat we nog veel leuke familiemomenten mogen beleven!

Als laatste, en meest belangrijke, wil ik mijn vriendin bedanken. Sofie, ik ben nog steeds heel erg blij dat ik die bewuste avond een stapje ging zetten in het Gentse uitgaansleven en je daar leerde kennen. Je hebt me steeds gesteund in mijn werk, ook tijdens mijn geklaag wanneer ik voor de zoveelste keer de tekst van mijn paper moest herwerken of toen de HPC servers weeral plat lagen. Ik word altijd erg vrolijk als we samen zijn, en we hebben de afgelopen 6 jaar al prachtige dingen samen beleefd. Het was super dat je mee wou naar Londen om me te steunen tijdens mijn stage bij Yahoo en om samen te genieten van de stad. Ik kijk er naar uit om in en rond Gent een prachtige toekomst samen met je uit te bouwen. Ik zie je graag!

Gent, 7 oktober 2016
Steven Van Canneyt

Table of Contents

List of Figures

List of Tables

List of Acronyms

A

ACA	Average Classification Accuracy
AdaBoost	Adaptive Boosting
AP	Average Precision
API	Application Programming Interface

B

BoW	Bag-of-Words

C

CC	Correlation Coefficient

D

DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DCG	Discounted Cumulative Gain

E

EDCoW Event Detection with Clustering Of Wavelet-based signals

ECML European Conference on Machine Learning

EM Expectation Maximization

G

GDELT Global Database of Events, Language and Tone

GMT Greenwich Mean Time

GTB Gradient Tree Boosting

GUI Graphical User Interface

H

HMM Hidden Markov Model

HTML Hyper Text Markup Language

HTTP Hyper Text Transfer Protocol

I

ID Identifier

IDCG Ideal Discounted Cumulative Gain

IEEE Institute of Electrical and Electronics Engineers

IJSWIS International Journal on Semantic Web and Information Systems

IMDb Internet Movie Database

IP Internet Protocol

IWT agency for Innovation by Science and Technology in Flanders

J

JSON JavaScript Object Notation

L

LDA Latent Dirichlet Allocation
LGD LinkedGeoData
LibLinear Library for large Linear classification

M

MAP Mean Average Precision
MLE Maximum Likelihood Estimation
MNDCG Mean Normalized Discounted Cumulative Gain
MP Mean Precision
MRRSE Mean Root Relative Squared Error
MUC Message Understanding Conferences

N

NDCG Normalized Discounted Cumulative Gain
NE Named Entity
NER Named Entity Recognition
NLP Natural Language Processing

P

PKDD Principles and Practice of Knowledge Discovery

POI Point Of Interest

R

RBF Radial Basis Functions
REST REpresentational State Transfer
RMSLE Root Mean Squared Log Error
RRSE Root Relative Squared Error

S

SNOW workshop on Social News On the Web
SVM Support Vector Machine

T

TF-IDF Term Frequency-Inverse Document Frequency
TKDE Transactions on Knowledge and Data Engineering
TweetNLP Twitter Natural Language Processing

U

UK United Kingdom
URL Uniform Resource Locator
USA United States of America

W

WEKA Waikato Environment for Knowledge Analysis

Y

YAGO2 Yet Another Great Ontology, version 2

Samenvatting
– Summary in Dutch –

Het vakgebied van de datawetenschap (data science) is de afgelopen jaren sterk in opmars. Veel bedrijven en organisaties maken tegenwoordig gebruik van datawetenschap om betere zakelijke beslissingen te kunnen nemen. Datawetenschap leidt ook in wetenschappelijke kringen tot nieuwe mogelijkheden, niet enkel om bestaande modellen te verifiëren of te weerleggen, maar ook om problemen vanuit een totaal ander perspectief en op een andere schaal te bekijken en te modelleren. Het vroegtijdig detecteren van afwijkingen in de monitoringgegevens van toestellen en software kan bijvoorbeeld het falen van machines en software voorkomen, en significante besparingen opleveren. De hoofdreden van de opkomst van datawetenschap is dat bijna elke sector van de economie momenteel toegang heeft tot meer data dan wat een decennium geleden denkbaar was. IBM schat dat 90 procent van de data in de wereld gecreëerd werd tijdens de afgelopen twee jaar. Deze heel grote verzamelingen aan data worden 'big data' genoemd en worden vaak omschreven door 4Vs: het extreme Volume van de data, de grote Variëteit aan types van data, de snelheid (Velocity) waaraan de data moet verwerkt worden, en de variërende kwaliteit (Veracity) van big data. In dit proefschrift spelen sociale media (zoals Twitter of Facebook) een belangrijke rol. Deze zijn in het bijzonder veelbelovend voor het vakgebied van datawetenschap omdat ze grote volumes aan data bevatten, over een brede gebruikersgroep beschikken en een real-time karakter hebben. In de eerste plaats kunnen sociale media gebruikt worden om nieuwe informatie te detecteren voordat deze beschikbaar wordt in gestructureerde databanken. Veel evenementen kunnen bijvoorbeeld gedetecteerd worden via de sociale media, zelfs voordat deze worden gerapporteerd in de traditionele media. Ten tweede, omdat sociale media een belangrijke bron zijn geworden om nieuwe klanten te werven, is het voor bedrijven en organisaties essentieel om de interacties op sociale media in relatie met hun merk, producten en ideeën te verzamelen, te analyseren, en te optimaliseren.

De toepassing van datawetenschappen op sociale media data brengt een aantal belangrijke uitdagingen met zich mee. In dit proefschrift beschouwen we drie grote uitdagingen. De eerste uitdaging die we beschouwen is dat bij het behandelen van sociale media data de inhoud van een individueel item vaak erg kort, grammaticaal incorrect, en divers is, en daarom heel moeilijk automatisch te interpreteren. Meer dan 50% van de berichten op Twitter bevatten bijvoorbeeld weinig nuttige informatie en zijn willekeurige gedachten, zelfpromotie, of onderhoud van

aanwezigheid zoals ‘kben terug’ of ‘net TV gekeken’. Er moeten dus methodes ontwikkeld worden die efficiënt de nuttige informatie uit de erg grote en diverse verzameling aan sociale media extraheert. Als een eerste stap om deze uitdaging aan te pakken, introduceren we een aanpak die, gebruik makend van sociale media, interessante plaatsen ontdekt en karakteriseert. In het bijzonder onderzoeken we hoe geografische geannoteerde tekstuele informatie die verzameld werd via sociale media kan gebruikt worden om nieuwe plaatsen te ontdekken. De in dit proefschrift voorgestelde methode blijkt in staat om diverse soorten van plaatsen te vinden, die nog niet aanwezig zijn in de databanken gebruikt door Foursquare, Google, LinkedGeoData, of Geonames. We breiden dit werk uit door een methode te introduceren die het semantische type (bvb. ‘conferentie’ of ‘sportevenement’) inschat van automatisch uit sociale media geëxtraheerde evenementen. De hiertoe gebruikte technieken maken gebruik van de wijze waarop het semantische evenement-type beïnvloed wordt door de tijdruimtelijke aarding van het evenement, het profiel van de aanwezigen, en het semantische type van de plaats, en andere entiteiten die geassocieerd worden met het evenement. Experimentele resultaten tonen aan dat onze methodologie kan gebruikt worden om uit sociale media evenementen van een gegeven semantisch type te ontdekken die niet worden vermeld in de Upcoming evenementen databank. Als laatste deel over gestructureerde informatie beschouwen we de extractie van onderwerpen met hoge nieuwswaarde uit sociale media. De voorgestelde methode verwerkt automatisch grote hoeveelheden van binnenkomende sociale media data om journalisten te voorzien van een uitgebreid real-time overzicht aan krantenkoppen en complementaire informatie. Onafhankelijke evaluatie toont de effectiviteit van de voorgestelde methodologie aan.

Ten tweede vereist het werken met grote hoeveelheden real-time gegevens nieuwe methodologieën en technologieën. Er moeten raamwerken worden gebouwd om grote hoeveelheden aan gegevens in real-time te verzamelen en te analyseren. In dit proefschrift stellen we een generiek raamwerk voor dat kan gebruikt worden om het consumptiegedrag van gebruikers op nieuwswebsites te verzamelen en te analyseren. Het raamwerk laat toe om de populariteit en kenmerken van online nieuwsartikels in real-time te verzamelen, en is zodanig opgebouwd dat het kan worden geschaald om miljoenen bezoekers en duizenden artikels te behandelen. Er werd een grondige evaluatie uitgevoerd op twee verschillende nieuws websites: een jong online nieuws bedrijf dat als doel heeft lezers te bereiken via sociale media (newsmonkey), en een online platform van de gevestigde openbare omroep met een meer traditionele kijk op nieuwsconsumptie (deredactie.be). We tonen aan dat het raamwerk en de voorgestelde analyse aanpak erg geschikt zijn voor beide contexten, en dat hiermee nieuwe inzichten in online nieuwsconsumptie kunnen worden bekomen.

De laatste uitdaging die we beschouwen in dit proefschrift is het voorspellen van de populariteit van media data (zoals nieuwsartikelen) over sociale media. Dit is erg uitdagend door de grote verschillen in de populariteitsdistributie (heel veel weinig populaire content en erg weinig zeer populaire content) en de grote verzameling aan factoren die de populariteit beïnvloeden. Daarom is er nood aan

technieken die toelaten de complexe afhankelijkheid tussen de kenmerken van de beschouwde media data en de finale populariteit modelleren. Om deze uitdaging aan te pakken, stellen we in dit proefschrift een nieuwe methode voor om de populariteit van online nieuws te modelleren en te voorspellen. We voeren eerst een grondige analyse uit naar de consumptiepatronen van online nieuws en hun onderliggende distributies. Deze kennis wordt dan gebruikt om de populariteit van nieuwsartikels beter te voorspellen, in vergelijking met verschillende bestaande methodes. We tonen aan dat het gebruik van eigenschappen gerelateerd aan de inhoud, metadata en temporeel gedrag van de artikels leidt tot een significante verbetering van de voorspellingen, in vergelijking met bestaande aanpakken die alleen de historische populariteit van de artikels beschouwen.

Naarmate meer en meer takken binnen de industrie sterker afhankelijk zullen worden van het analyseren van data, zal de toepasbaarheid van deze bijdragen groeien. De inzichten verworven in dit proefschrift kunnen een grondige basis vormen voor verder onderzoek. De online nieuwsanalyse en het voorspellingsraamwerk werden bijvoorbeeld reeds ontplooid bij newsmonkey en deredactie.be. Het raamwerk zal ook beschikbaar worden gesteld voor andere nieuwswebsites, om hun data te verzamelen, te analyseren en te voorspellen om zo hun publicatiestrategie te optimaliseren. In verder onderzoek kunnen de voorgestelde inzichten rond het consumptiegedrag van online nieuws en het voorspellen van hun populariteit gebruikt worden om methodes te ontwikkelen die actief bijdragen tot het optimaliseren van de publicatiestrategie.

Summary

Data science is a field that has gained a lot of interest in the last few years, and has heavily influenced research and business practices. Many companies and organizations nowadays use data science to make better business decisions. Additionally, data science leads to new opportunities in the scientific community, not only to verify or disprove existing models, but also to consider problems from a totally different perspective and to model them at a much larger scale. For instance, detecting anomalies in monitoring data of equipment and software at an early stage may prevent failure of machines and software, significantly reducing costs. The main reason underlying the rise of data science is that almost every sector of the modern economy now has access to more data than was imaginable even a decade ago. IBM estimates that 90% of the global data today has been created in the past two years. Such very large sets of data are referred to as 'big data' and are often described using 4Vs: the extreme Volume of data, the wide Variety of types of data, the Velocity at which the data must be processed, and the Veracity of big data. In this dissertation, social media such as Twitter or Facebook play an important role. Social media are particularly promising for the field of data science, due to their large data volume, broad user base, and real-time nature. In the first place, social media can be used to discover information before it is picked up and stored in structured databases. Examples include the use of social media to detect events, even before they are reported in traditional media. Secondly, as social media form an important tool to attract new customers, it becomes increasingly important for companies and organizations to monitor, analyze and optimize the interactions on social media in relation to their brand, products, and ideas.

A number of challenges arise when applying data science to social media data. In this dissertation, we address three major challenges. The first challenge we address is that social media content is often very short, noisy and diverse, and therefore very difficult to interpret automatically. For instance, more than 50% of the messages on Twitter do not contain useful information and are mostly random thoughts, self promotion or presence maintenance such as 'im back' or 'just watched TV'. In other words, methods should be constructed that efficiently extract the useful content from the very large and diverse social media data. As a first step to tackle this challenge, we propose an approach that discovers and characterizes places of interest using social media. In particular, we investigate how geographically annotated textual information obtained from Flickr photos and Twitter posts can be used to discover new places of a given type such as 'hotel' or 'school' to extend semantic databases of places. For several place types, our pro-

posed methodology finds places that are not yet contained in the databases used by Foursquare, Google, LinkedGeoData and Geonames. We have extended this work by introducing a method for discovering the semantic type of events which are extracted from social media. We have in particular focused on how the semantic type such as 'conference' or 'sport event' is influenced by the spatio-temporal grounding of the event, the profile of its attendees, and the semantic type of the venue and other entities which are associated with the event. Experimental results show that our methodology can be used to discover events of a given semantic type which are not mentioned in the Upcoming datasets, by analyzing social media data. As our last contribution on structured information extraction, we consider the extraction of newsworthy topics from social media. The proposed method allows automatically mining social media streams to provide journalists with a set of headlines and complementary information that summarizes the current newsworthy topics. Independent evaluation shows the effectiveness of the proposed methodology.

Secondly, working with a large amount of real-time data requires distinctive new techniques and technologies. Frameworks should be developed to handle and analyze a large volume of data in real-time. In this dissertation, we propose a generic framework which can be used to monitor and analyze the consumption patterns of users on news websites. The framework monitors the popularity and features of online news articles in real-time, and can be easily scaled to handle millions of visits and thousands of articles. Our framework has been thoroughly evaluated on two quite different news websites: a young online news company that focuses on accessing readers through social media (newsmonkey), and the online platform from an established national broadcaster, with a more traditional take on news consumption (deredactie.be). We show that our generic data-driven framework and analysis approach are well suited for both use cases, and lead to new insights into online news sharing and consumption behavior.

The last challenge we handle in this dissertation is to predict the popularity of media content (e.g., news articles) in social media. This is very challenging due to the high skewness in the popularity distribution and due to the very large and complex set of factors which influence the popularity. Therefore, advanced prediction methodologies should be constructed that can model the complex dependencies between the features of content and their final popularity. To address this challenge, we propose in this dissertation a novel methodology to model and predict the popularity of online news. We first conduct a thorough analysis of the view patterns of online news, and their underlying distributions. This knowledge is then used to better predict the popularity of articles, compared to various existing methods. By means of a new real-world dataset, we show that the combination of features related to content, meta-data, and the temporal behavior leads to significantly improved predictions, compared to existing approaches which only consider features based on the historical popularity of the considered articles.

As most industries will become more data-driven, the applicability of our contributions will grow and the insights gained in this dissertation can form a sound basis for further research. For example, the online news monitoring and prediction framework has been deployed at newsmonkey and deredactie.be. The framework

will also be made available for other news websites, to monitor and analyze their data and to optimize their strategy. In future research, the knowledge presented in this dissertation on consumption behavior of online news and its predicted popularity can be used to construct methodologies which actively suggest how the publishing strategy can be optimized.

1 Introduction

"It's important to remember that the primary value from big data comes not from the data in its raw form, but from the processing and analysis of it and the insights, products, and services that emerge from analysis."

– Thomas H. Davenport

1.1 Context

1.1.1 Data Science

Data science is the scientific discipline that studies the various processes and systems to extract knowledge or insights from data in various forms, either structured or unstructured. The general task of data scientists is to find and interpret rich data sources, manage large amounts of data, extract structured information from large sets of unstructured or semi-structured data, discover patterns in large data sets, build mathematical models using the data, and present and communicate the insights and findings. Data science has gained more and more interest in recent years, and affects academic as well as applied research in many domains, including speech recognition, search engines, recommendation systems, and predictive modeling. As almost every sector of our modern economy now has access to more data than was imaginable even a decade ago, data science heavily influences economics, business, and finance. It is used in many industries to allow companies and organizations to make better business decisions as well as in the scientific

community to verify or disprove existing models or theories. For instance, detecting anomalies in monitoring data of equipment and software at an early stage may prevent failure of machines and software, reducing costs. Another example, relevant for the work presented in this thesis, is the prediction of which content will be most popular on social media, allowing online news agents and marketeers to optimize their publishing strategy.

Data sets are growing rapidly because they are increasingly gathered by e.g. cheap and numerous information-sensing mobile devices, software logs, cameras, social media, and wireless sensor networks. IBM estimates that 90% of the global data today has been created in the past two years.[1] Such (very) large sets of data are often referred to as 'big data'. Working with this amount of data requires distinctive new skills and tools. The datasets are often too voluminous to fit on a single computer, to manipulate with traditional databases or to analyze with standard statistical techniques. The data is also more heterogeneous than the highly curated data of the past. Digitized text, audio, and visual content, such as social media and blog data, is typically messy, incomplete, and unstructured. Novel methods and techniques are needed to extract useful information from such data with uncertain quality. Gartner uses '3Vs' to describe big data: the extreme Volume of data, the wide Variety of types of data and the Velocity at which the data must be must processed [1]. Additionally, a new V is added by some definitions to denote the Veracity of big data. Despite the additional challenges when data scientists cope with big data versus 'small data', analyzing big data offers the opportunity to enhance insights, decision making, and process automation for very large and complex models. This data, when extracted, manipulated, stored, and analyzed can help a company to gain useful insights to increase revenues, get or retain customers, and improve operations. For instance, all websites on the world wide web can be indexed and processed to construct advanced search engines [2], or all interactions on commercial websites can be monitored and analyzed to improve user interactions and online sales [3].

1.1.2 Data Science in Social Media

The pervasive use of social media sites such as Facebook, Instagram, LinkedIn, and Twitter has led to large amounts of a new form of data, simply known as social media data. This data is mostly user-generated, informal, incomplete, and multimedia, and is often accompanied with information about time and location. The data is large in scale with quick updates taking place continuously, all over the world. Social media are particularly promising for the field of data science, because they capture a continuously growing amount of user-generated data. In the first place, social media can be used to discover information before it is picked

[1] http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html

up and stored in structured databases. The data can for instance be used to detect events, even before they are reported in traditional media [4]. In addition, social media can be used to discover places of interest not yet known by structured databases such as Google Places and Foursquare [5], or to estimate the opinions about a product or person [6]. Secondly, as social media become an important tool to attract new customers, it becomes increasingly important for companies and organizations to monitor, analyze, and optimize the interactions on social media in relation to their brand, products, and ideas [3, 7]. However, the process is not without its challenges. The stream of social media data is a prime example of 'big data', as described in Section 1.1.1. Dealing with data sets of enormous sizes which constantly grow, is a challenge on its own, but there is an additional and unique problem to social media related to the large noise-to-signal ratio. These, and related, challenges will be handled in this dissertation. Section 1.2 will describe the challenges we handle in this dissertation in more detail, and in Section 1.3 we summarize the main contributions of our research.

1.2 Problem Statement

Due to the large amount of real-time data produced by social media, these form an ideal source to discover facts not yet contained in structured databases. The most important challenge to handle social media is that its content is often very short, noisy, and diverse (i.e. veracity and variety of the data). For instance, more than 50% of the messages on Twitter do not contain generally useful information and are mostly random thoughts, self-promotion or presence maintenance such as 'im back' or 'just watched TV' [8]. In addition, the characteristics of different kinds of social media data are very diverse as e.g. Flickr photos are mostly taken at points of interest associated with structured tags, whereas the geo-locations of Twitter posts are more widespread with more noisy content [9]. Because of the potentially useful information hidden in social media data, we need methods which efficiently extract that content. This content should then be interpreted and correctly enriched to construct the structured data, e.g., to populate knowledge bases.

The volume and velocity of big social media data leads to the second challenge. Big data requires novel techniques and technologies to reveal insights from datasets that are complex and of a massive scale. Frameworks should be constructed to handle a large volume of data in real-time, and to timely analyze it. These insights can for example be used to optimize publication strategies on social media [3] or to receive knowledge about users' opinions about products [10].

The last challenge we handle in this dissertation is to predict the popularity of content in social media. This is very challenging due to the high skewness in the popularity distribution (i.e. a large amount of unpopular content and very little popular content) and the very large and complex set of factors which influ-

ence the popularity (e.g. sentiment of the content, timing, and the influence of early adopters) [11]. Therefore, advanced predictions methodologies should be constructed which can model the complex dependencies between the features of the content and their final popularity. These predictions can, together with the extracted data and insights, be used to make better decisions. For example, by predicting which content will be most popular on social media, online news agents and marketeers can optimize their publishing strategy [12].

1.3 Main Research Contributions

The main research contributions of this dissertation address the challenges which are described in Section 1.2. The following contributions are detailed in this dissertation:

1. *Methodologies for discovering and characterizing structured information using social media.*
   Databases of structured data have become increasingly popular. Event databases such as Upcoming[2] and Facebook Events[3] are used to find interesting events in the vicinity of the user. Databases of places, e.g. Google Places[4] or Geonames[5], can be used to detect the geographic location of a place, and to discover places of a given type that are close to a user-specified location. Also, users can search in the databases of news websites to find media items which they find interesting. As it is important for these systems to use an up-to-date database with a broad coverage, there is a need for techniques that are capable of expanding structured databases with new facts in an automated way. Furthermore, the entities (places, events, topics. . .) need an associated semantic type that allows for easier browsing and searching through the database.

   The main focus of these contributions is on how the often noisy and diverse data obtained from social media can be used to discover new information to extend structured databases. In this dissertation, we consider three types of structured information:

   (a) *Places of Interest*
   We investigate how geographically annotated information obtained from Flickr photos and Twitter posts can be used to discover new places of a given type such as 'hotel' or 'school' to extend semantic databases of places. For several place types, our methodology finds places that

[2] http://upcoming.org/
[3] http://events.fb.com/
[4] https://developers.google.com/places/
[5] http://geonames.org/

are not yet contained in the databases used by Foursquare, Google, LinkedGeoData and Geonames.

(b) *Events*
We introduce a method for discovering the semantic type of events which are extracted from social media, focusing in particular on how this type is influenced by the spatio-temporal grounding of the event, the profile of its attendees, and the semantic type of the venue and other entities which are associated with the event. We estimate the aforementioned characteristics from meta-data associated with social media covering the event. Experimental results show that our methodology can be used to discover events of a given semantic type from social media that are not mentioned in the Upcoming datasets.

(c) *Newsworthy Topics*
Extensive work has shown that social media can successfully be used to detect events, even before they are reported in traditional media [4]. Therefore, social media may form an excellent source for news professionals to monitor the newsworthy topics that emerge from the crowd. The task at hand is to automatically mine social media streams to provide journalists with a set of headlines and complementary information that summarize the newsworthy topics for a number of time intervals of interest. Independent evaluation shows the effectiveness of the proposed methodology.

2. *A framework to collect and analyze large amounts of online news data in real-time.*
It is often important to monitor and analyze social media data in real-time and at a large scale. For example, due to the strong competition between online news publishers in order to reach the largest possible audience, there is a need for a well thought-out online publishing strategy. This can be achieved by acquiring profound and real-time insights into the consumption and social sharing behavior of users with respect to their online content. Therefore, we propose a generic framework which can be used to monitor and analyze the consumption patterns of users on news websites. The framework monitors the popularity of online news articles in real-time, and can be easily scaled to handle millions of visits and thousands of articles. Our framework is thoroughly evaluated on two quite different news websites: A young online news company that focuses on accessing readers through social media (http://newsmonkey.be), and the online platform from an established national broadcaster, with a more traditional take on news consumption (http://deredactie.be). We show that our generic data-driven framework and analysis approach is well suited for both use cases, and use it to provide

new insights into online news sharing and consumption behavior.

3. *An approach to predict the virality of online news.*
   It is very useful to predict the future state of social media data. By predicting the popularity of advertisement or online articles on social media, for instance, marketeers and news agents can optimize their online publishing strategy. In this dissertation, we handle the use case of predicting the popularity of online news articles. We first conduct a thorough analysis of the view patterns of online news, and their underlying distributions. This knowledge is then used to better predict the popularity of articles, compared to various existing methods. By means of a new real-world dataset, we show that the combination of features related to content, meta-data, and the temporal behavior leads to significantly improved predictions, compared to existing approaches which only consider features based on the historical popularity of the considered articles.

1.4 Outline of this Dissertation

This dissertation is composed of a number of publications that were written within the scope of this Ph.D. research. The selected publications provide an integral and consistent overview of the work performed. The different research contributions are detailed in Section 1.3 and the complete list of publications that resulted from this work is presented in Section 1.5. Within this section we give an overview of the remainder of this dissertation and explain how the different chapters are linked. Table 1.1 shows the challenges that were highlighted in Section 1.2 as they apply to the different chapters.

Table 1.1: An overview of the challenges per chapter in this dissertation.

	Ch.2	Ch.3	Ch.4	Ch.5	Ch.6
Data Extraction	•	•	•		
Big Data Collection and Analysis				•	
Data Prediction					•

The research described in Chapter 2, Chapter 3 and Chapter 4 focuses on structured information extraction from social media. In particular, Chapter 2 focuses on the discovery and characterization of places of interest using social media. We discuss how geographically annotated information obtained from social media can be used to discover new places. In particular, we first determine potential places of interest by clustering the locations where Flickr photos were taken. The tags from the Flickr photos and the terms of the Twitter messages posted in the vicinity of

the obtained candidate places of interest are then used to rank them based on the likelihood that they belong to a given type.

In Chapter 3, we propose a method to discover the semantic type of events which have been extracted from social media. We first detect events which are not yet contained in existing databases using Flickr. This is done by first clustering a large set of Flickr photos based on their similarity in text, geographical location, and creation time. The obtained clusters are considered as candidate events, and events which are already known by the Upcoming event database are removed. In Chapter 2, we only use textual information of the social media to discover semantic types. In Chapter 3, however, more advanced features are discussed to better estimate the semantic type of events. In particular, the hypothesis we consider in this chapter is that in many cases the event type can be better estimated by looking at additional properties, such as timing, the type of venue, or characteristics of attendees. These properties can be readily obtained from the meta-data associated with the Flickr photos of the event. They are used to describe the events, and an ensemble learner is then used to identify their most likely semantic type.

In Chapters 2 and 3, we focus on the extraction of structured information from social media. In Chapter 5 and Chapter 6, we focus on the collection, analysis and prediction of the popularity of online news content in social media. Chapter 4 makes a bridge between these two parts as it considers the extraction of newsworthy topics from social media. In particular, Chapter 4 describes our submission for the SNOW 2014 Data Challenge. The challenge was to mine Twitter streams to provide journalists with a set of headlines and complementary information that summarizes the most newsworthy topics for a number of given time intervals. We propose a 4-step approach to solve this. First, a classifier is trained to determine whether a Twitter user is likely to post tweets about newsworthy stories. Second, tweets posted by these users during the time interval of interest are clustered into topics. For this clustering, the cosine similarity between a boosted tf-idf representation of the tweets is used. Third, we use a classifier to estimate the confidence that the obtained topics are newsworthy. Finally, for each obtained newsworthy topic, a descriptive headline is generated together with relevant keywords, tweets and pictures.

In Chapter 5 and Chapter 6, we focus on the monitoring, analysis, and prediction of the popularity of online news. In Chapter 5, we introduce a highly scalable framework which monitors and analyzes the consumption behavior of online news articles in real-time. As the optimal publishing strategy is highly publisher-specific and depends on the considered popularity metric (e.g., number of Facebook shares vs. total number of page views), our framework can be easily scaled to cover many news websites, and we discuss six popularity metrics. In addition, we introduce a number of article features used by our monitoring system, and show that these are vital for a good understanding of the news consumption and sharing behavior.

Chapter 6 presents a novel methodology to model and predict the popularity of online news. We show that well-chosen base functions can nicely model the view patterns of online news, and show how the influence of day versus night on the total view patterns can be taken into account to further increase the accuracy, without leading to more complex models. To predict the popularity of online news, we propose a method that (i) explicitly makes use of the temporal model underlying the historical view pattern of the considered article, (ii) considers, in addition to the historical popularity of the article, content-based and meta-data related features (such as author, category, emotion, etc.), and (iii) uses the gradient tree boosting algorithm instead of the more traditional linear regression techniques for making predictions.

Finally, Chapter 7 highlights the most important contributions of this dissertation and summarizes the perspectives for future research.

1.5 Publications

The research results obtained during this PhD research have been published in scientific journals and presented at a series of international conferences. The following list provides an overview of these publications.

1.5.1 Publications in international journals (listed in the Science Citation Index[6])

1. **Steven Van Canneyt**, Philip Leroux, Bart Dhoedt, and Thomas Demeester. *Predicting the Popularity of Online News based on Temporal and Content-Related Features* Submitted to Multimedia Tools and Applications, July 2016.

2. **Steven Van Canneyt**, Philip Leroux, Bart Dhoedt, and Thomas Demeester. *Towards a Data-Driven Online News Publishing Strategy.* Submitted to IEEE Transactions on Knowledge and Data Engineering, April 2016.

3. Cedric De Boom, **Steven Van Canneyt**, Thomas Demeester, and Bart Dhoedt. *Representation learning for very short texts using weighted word embedding aggregation.* Published in Pattern Recognition Letters, pages 150–159, 2016.

[6]The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: A1 publications are articles listed in the Science Citation Index Expanded, the Social Science Citation Index or the Arts and Humanities Citation Index of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper.

4. **Steven Van Canneyt**, Steven Schockaert, and Bart Dhoedt. *Categorizing events using spatio-temporal and user features from Flickr.* Published in Information Sciences, 328, pages 76–96, 2016.

5. **Steven Van Canneyt**, Steven Schockaert, and Bart Dhoedt. *Discovering and characterizing places of interest using Flickr and Twitter.* Published in the International Journal on Semantic Web and Information Systems (IJSWIS), 9(3), pages 77–104, 2013.

1.5.2 Publications in international conferences

1. Cedric De Boom, **Steven Van Canneyt**, Steven Bohez, Thomas Demeester, and Bart Dhoedt. *Learning semantic similarity for very short texts.* Published in the proceedings of the 2nd ICDM International Workshop on Representation Learning for Semantic Data, pages 1229–1234, 2015.

2. Rupert Lemahieu, **Steven Van Canneyt**, Cedric De Boom, and Bart Dhoedt. *Optimizing the popularity of Twitter messages through user categories.* Published in the proceedings of the 2nd ICDM International Workshop on Social Multimedia Data Mining, pages 1396–1401, 2015.

3. Cedric De Boom, **Steven Van Canneyt**, and Bart Dhoedt. *Semantics-driven event clustering in Twitter feeds.* Published in the proceedings of the 5th WWW International Workshop on Making Sense of Microposts, pages 2–9, 2015.

4. **Steven Van Canneyt**, Nathan Claeys, and Bart Dhoedt. *Topic-dependent sentiment classification on Twitter.* Published in the proceedings of the 37th European Conference on Information Retrieval (ECIR), pages 441–446, 2015.

5. **Steven Van Canneyt**, Steven Schockaert, and Bart Dhoedt. *Estimating the semantic type of events using location features from Flickr.* Published in the proceedings of the 8th ACM SIGSPATIAL International Workshop on Geographic Information Retrieval, pages 57–64, 2014.

6. **Steven Van Canneyt**, Matthias Feys, Steven Schockaert, Thomas Demeester, Chris Develder, and Bart Dhoedt. *Detecting newsworthy topics in Twitter.* Published in the proceedings of the SNOW 2014 Data Challenge, pages 25–32, 2014.

7. **Steven Van Canneyt**, Steven Schockaert, Olivier Van Laere, and Bart Dhoedt. *Using social media to find places of interest: A case study.* Published in the proceedings of the 1st ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information, pages 2–8, 2012.

8. **Steven Van Canneyt**, Steven Schockaert, Olivier Van Laere, and Bart Dhoedt. *Detecting places of interest using social media.* Published in the proceedings of the 2012 IEEE/WIC/ACM International Conference on Web Intelligence, pages 447–451, 2012.

9. **Steven Van Canneyt**, Steven Schockaert, Olivier Van Laere, and Bart Dhoedt. *Time-dependent recommendation of tourist attractions using Flickr.* Published in the proceedings of the 23rd Benelux Conference on Artificial Intelligence (BNAIC), pages 255–262, 2011.

1.5.3 Publications in national conferences

1. **Steven Van Canneyt**, Steven Schockaert, and Bart Dhoedt. *Categorizing events using spatio-temporal and user features from Flickr (abstract).* Published in the proceedings of the 14th Dutch-Belgian Information Retrieval Workshop (DIR), page 14, 2015.

2. **Steven Van Canneyt**, and Bart Dhoedt. *A context-aware tourism recommendation system.* Published in the proceedings of the 12th UGent-FEA PhD symposium, page 61, 2011.

References

[1] D. Laney. *3D data management: Controlling data volume, velocity, and variety*. Technical report, 2001.

[2] B. Sergey and L. Page. *The anatomy of a large-scale hypertextual Web search engine*. Computer Networks and ISDN Systems, 30:107–117, 1998.

[3] S. Van Canneyt, P. Leroux, B. Dhoedt, and T. Demeester. *Towards a Data-Driven Online News Publishing Strategy*. submitted to IEEE Transactions on Knowledge and Data Engineering,, 2016.

[4] T. Sakaki. *Earthquake shakes Twitter users: Real-time event detection by social sensors*. In Proceedings of the 19th International Conference on World Wide Web, pages 851–860, 2010.

[5] S. Van Canneyt, S. Schockaert, and B. Dhoedt. *Discovering and characterizing places of interest using Flickr and Twitter*. International Journal on Semantic Web and Information Systems, 9(3):77–104, 2013.

[6] R. Lemahieu, S. Van Canneyt, C. De Boom, and B. Dhoedt. *Optimizing the popularity of Twitter messages through user categories*. In Proceedings of the 2nd ICDM International Workshop on Social Multimedia Data Mining, pages 1396–1401, 2015.

[7] S. Yu and S. Kak. *A survey of prediction using social media*. ArXiv e-prints, pages 1–20, 2012.

[8] M. Naaman, J. Boase, C.-h. Lai, and N. Brunswick. *Is it really about me? Message content in social awareness streams*. In Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, pages 189–192, 2010.

[9] V. Murdock. *Your mileage may vary: on the limits of social media*. SIGSPATIAL Special, 3(2):62–66, 2011.

[10] S. Van Canneyt, N. Claeys, and B. Dhoedt. *Topic-dependent sentiment classification on Twitter*. In Proceedings of the 37th European Conference on Information Retrieval, pages 441–446, 2015.

[11] I. Arapakis, B. B. Cambazoglu, and M. Lalmas. *On the feasibility of predicting news popularity at cold start*. In Proceedings of the 6th International Conference on Social Informatics, pages 290–299, 2014.

[12] S. Van Canneyt, P. Leroux, B. Dhoedt, and T. Demeester. *Predicting the popularity of online news based on temporal and content-related features*. submitted to Multimedia Tools and Applications, 2016.

2

Discovering and Characterizing Places of Interest using Flickr and Twitter

Databases of structured data have become increasingly popular. Databases of places can be used, for example, to identify places of a given type that are close to a user-specified location. As it is important for such applications to use an up-to-date database with a broad coverage, there is a need for techniques that are capable of expanding structured databases in an automated way. In this chapter we discuss how geographically annotated information obtained from social media can be used to discover places which are not yet included in existing place databases. In particular, we first determine potential places of interest by clustering the locations where Flickr photos have been taken. The tags from the Flickr photos and the terms of the Twitter messages posted in the vicinity of the obtained candidate places are then used to rank them based on the likelihood that they belong to a given type (e.g., 'school' or 'shop'). For several place types, our methodology finds places that are not yet contained in the databases used by Foursquare, Google, LinkedGeoData and Geonames. Furthermore, our experimental results show that the proposed method can successfully identify errors in existing place databases such as Foursquare.

⋆ ⋆ ⋆

S. Van Canneyt, S. Schockaert, B. Dhoedt

Published in the International Journal on Semantic Web and Information Systems (IJSWIS), 9(3), pages 77-104, 2013

2.1 Introduction

Berners-Lee's vision of the Semantic Web [1] has become increasingly popular in the last few years. The World Wide Web would evolve to a highly interconnected network of data that could be easily accessed and understood by machines. Applications could for instance use the Semantic Web to construct customized answers to a particular question. In such applications the user is no longer required to search for information or pore through results. A question can be 'What are locations of the restaurants in London?'. To answer this question, a structured dataset has to be available containing places located in London (entities), associated with their location and semantic type (properties). However, a lot of information on the Web is still unstructured or only semi-structured. Therefore, there is a need for automated methods to extend structured datasets using existing Web data. Several methods of this form have been proposed, e.g. YAGO2 [2] and BabelNet [3] are knowledge bases that are constructed using Wikipedia and Wordnet. Other research focuses on establishing structured datasets containing information of a specific type. For instance, LinkedGeoData [4] is a dataset of places constructed using OpenStreetMap, an application in which users can submit geographical data such as place semantics. In this chapter, we will focus on improving existing databases of places. More precisely, we will add new places and discover likely errors using data from the Web. Social media data is particularly promising in this respect, due to the large amounts of geographically annotated data produced by these media. For example, about 1.5% of all Twitter posts (i.e. tweets) are annotated with geographical coordinates [5]. In addition, there are currently more than 190 million geotagged Flickr photos.[1] This data has been used to e.g. automatically detect events [6–8], to find popular places [9, 10] and tourist routes [11, 12].

The main focus of this chapter is on how geographically annotated information obtained from social media can be used to discover new places of a given type such as 'hotel' or 'school' to extend semantic databases of places. Our hypothesis is that the type of a place can be derived from the tags of the Flickr photos and the terms of the Twitter posts associated with locations in the vicinity of the place. For example, if photos around a particular location contain tags such as 'food', 'dinner' and 'eating', this strongly suggests that there is a restaurant at that location. In our previous work [13], we have provided evidence for the validity of this hypothesis: given the location of various places of interest (POIs), we addressed the task of identifying those POIs that are most likely to be of a particular type. Our main conclusion was that Flickr tags are a rich source of information for deciding on the type of a place. Using Twitter terms further improved the results although this improvement was more limited. We also considered the correlation between the type of the POIs and the types of the places in the vicinity to categories the

[1] http://www.flickr.com/map, visited on February, 2013

POIs. However, this additional information led to a minimal improvement of the performance of our methodology, and in this chapter we are mainly interested in the use of social media by itself to improve databases of places. Therefore, we do not consider such correlations here. In [14] we considered the more challenging problem of finding locations where places of particular types can be found, without providing a list of candidate locations. Instead, we used a simple grid overlay to find candidate locations and compared the results against existing databases of places. This qualitative analysis demonstrated the potential of the proposed method to find POIs in London that are not yet contained in Foursquare, Google Places, Geonames and LinkedGeoData. Encouraged by these initial results, we improve the proposed methodology in this chapter and present a more detailed experimental evaluation. First, the Support Vector Machine classifier used in [13, 14] is replaced by a language modeling approach, which improves the results significantly. Second, we analyze the behavior of different feature selection techniques. We conclude that for the Flickr data correlation coefficient feature selection [15] performs significantly better than χ^2 feature selection. The performance of the proposed methodology can be further improved when names of cities and countries are removed from the considered features. Finally, we perform a large-scale evaluation on 88 different cities, where we examine the results for London in more detail. Based on this evaluation, we can conclude that our approach is able to extend and validate data sets of places. In particular, our method is able to detect new places of a particular type, even when the locations of places of interest are not given. Furthermore, our experimental results show that the proposed method can also be used to successfully identify errors in existing place databases such as Foursquare.

The remainder of this chapter is structured as follows. We start with a review of related work. The subsequent section explains how training and test data have been collected. Thereafter, we describe our methodology of discovering places of a given type. This section is followed by an experimental evaluation. Finally, we conclude our work in the last section.

2.2 Related Work

To fill in the gap between the unstructured and semi-structured data from the Web and the structured data in the Semantic Web, a number of methodologies have been proposed. Kwok [16] and Etzioni [17], for instance, extracted named entities from unstructured web pages using natural language processing. Other research [2, 3] used semi-structured data available in Wikipedia and Wordnet to construct a structured dataset in an automatic way. The used semi-structured data available in Wikipedia have been improved by applying information extraction techniques on the main text of the corresponding Wikipedia article [18, 19]. To further construct

structured data, social media have been used due to the large amount of data available. Social media are, on the one hand, used to derive ontologies which describe relations between words [20, 21]. This information can for instance be used to improve search results: Given a word as query, similar words can be used to extend the results. Schmitz [20] detected subsumption relations between Flickr tags using the co-occurrence of the tags. The methodology applied in [21] measures the similarity of tags used in the social bookmarking system BibSonomy using several statistical measures such as cosine similarity, Jaccard similarity and the mutual information metric. On the other hand, social media can be used to extract entities and their semantics. Sakaki [7] for example constructed a probabilistic model to detect the location and time of earthquake and typhoon occurrences using Twitter. The researchers in [8] described a method which discovered events by detecting unusual regional activities in Twitter. Other research [11, 12] developed methodologies that automatically constructs travel itineraries using Flickr.

In this chapter, we are focusing on automatic detection of place locations and semantics using social media. This data can e.g. be used for personalized place recommendations. Ozdikis [22] for instance developed an application which recommends places similar to a user defined place. Initial work on extending semantic datasets of places by discovering points of interest (POIs) from social media has been exclusively based on analyzing the coordinates of geotagged data. For instance, Crandall et al. [9] used the mean shift method to cluster the locations of geotagged Flickr photos to detect POIs. This method has among others been applied in [10, 23, 24] to detect and recommend popular tourist places in cities. In this chapter, mean shift clustering is used as the first step of the proposed methodology to detect candidate locations of places of a given type. The Antourage system [12] on the other hand uses a hexagonal grid overlay over a city map, and associates with each hexagon a weight based on the number of Flickr photos that have been taken within the boundaries of that cell. Given such a weighted grid, the max-min ant system meta-heuristic [25] is used to find distance constrained trips in a city covering as much as possible popular POIs. These contributions focus on using locations of geotagged photos to detect POIs. In particular, in the aforementioned works, no attempt is made to associate semantic information with places. In contrast, in this chapter we aim to discover places of a given semantic type, and we do not restrict ourselves to tourist places, by also considering e.g. schools, graveyards and libraries.

A second line of research relevant for our work analyzes text originating from social media, in order to discover places and to retrieve semantic information on these places. Rattenbury et al. [6] used multiscale burst analysis to detect place-related Flickr tags. This technique was applied in [26] to detect names for arbitrary areas in the world. They spatially clustered the locations where Flickr photos were taken using k-means clustering. For each cluster, representative tags were

searched using TF-IDF and the percentage of users in the cluster that used a given tag. To find landmarks, their names and their most representative photos, Abbasi et al. [27] proposed a further extension of this approach. To detect landmarks in a city, they first select photos containing the city name. Second, using support vector machines and the tags that have been assigned to the photos, these photos were classified either as being or not being taken of a landmark. As training data, photos were obtained from manually selected photo groups such as 'landmarks around the world'. Finally, tags and photos that describe landmarks in a given city were extracted based on the obtained photos. This research demonstrates that text from social media can be used to detect POIs and their associated name. However, the semantic type of the obtained places was not determined.

Our work is most closely related to Gazetiki [28], a gazetteer which was automatically derived from Wikipedia, Panoramio and web search using a four step approach. The method proposed in [28] first collected Wikipedia articles which contain associated geographic coordinates, and a geographical concept (i.e. a place type from Geonames) in their first sentence. From these articles, links to other Wikipedia articles with a geographical concept in their first sentence were extracted. For each obtained Wikipedia article, a candidate geographical entity was constructed with the name equal to the title of the article and the type extracted from the first sentence of the article. If the article also contains a coordinate, this coordinate was used as coordinate of the geographical entity. Second, named entity recognition was used to find additional geographical names in the titles of Panoramio photos. In addition, a candidate geographical entity was constructed for each obtained geographical name. For the candidate geographical entity obtained using Panoramio, the place type was determined by taking into account the number of search results of queries such as '<geographical name> is a <place type>' on the Alltheweb search engine. Third, the obtained candidate geographical entities without associated coordinates were geotagged by the center of gravity of the coordinates of the Panoramio photos that have been tagged with the geographical name. Finally, the obtained places were ranked using the number of results by searching for the geographical name in Alltheweb and the number of times the place was photographed.

However, to the best of our knowledge, no effort has so far been devoted to discover places of a particular type using social media, given only some examples of places of that type. In addition, none of the described work analyzed whether their approach was able to detect places which were not yet included in existing databases or is able to detect incorrect data in existing databases of places.

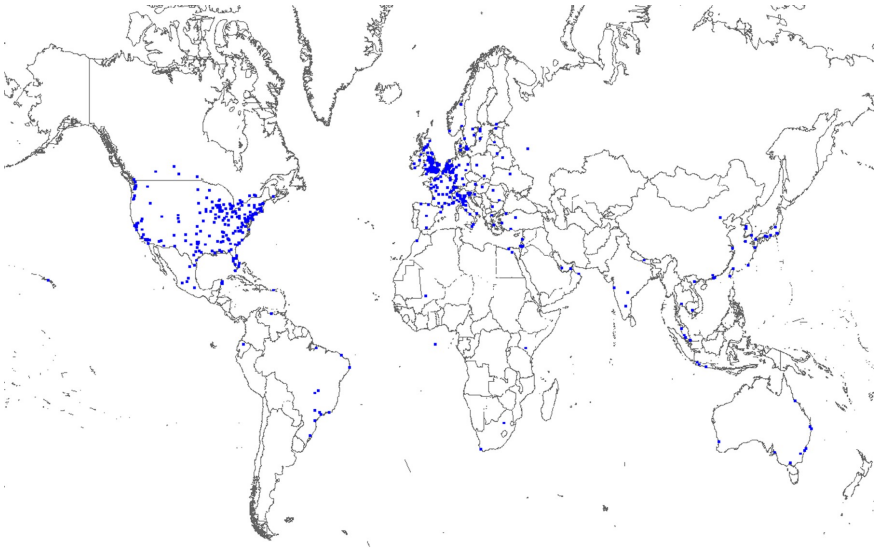


Figure 2.1: Plot of the cities in our dataset.

2.3 Data Acquisition

Our goal is to determine the locations in a city C which correspond to places of a given type t (e.g. schools, hospitals, train stations and restaurants), based on the tags of the Flickr photos taken in the city and the terms of the tweets posted in the city. To obtain training and test data, we collected a set of places with known location and type for several cities. We subsequently mined Flickr and Twitter to find metadata about these places. We now explain these steps in more detail.

2.3.1 Collecting Bounding Boxes of Cities

The considered cities were selected by first selecting the names of all cities with a population of more than 15,000 inhabitants using Geonames. For each of the obtained cities, its bounding box was determined using Yahoo! PlaceFinder. When the two bounding boxes of two different cities overlap, only the city with the largest bounding box was kept to ensure that there is no overlap in the bounding boxes of the cities in the training, test and development set. After identifying the bounding boxes of the cities, only cities where more than 1000 Flickr photos were taken and more than 1000 tweets were posted were retained. Through this process, we collected bounding boxes of 530 cities, whose locations are plotted in Figure 2.1. More detailed plots of the locations of the obtained cities in Europe and the USA are shown in Figure 2.2 and 2.3, respectively. In these figures, the radius of the circles is proportional to the number of Flickr photos and tweets posted in the city.

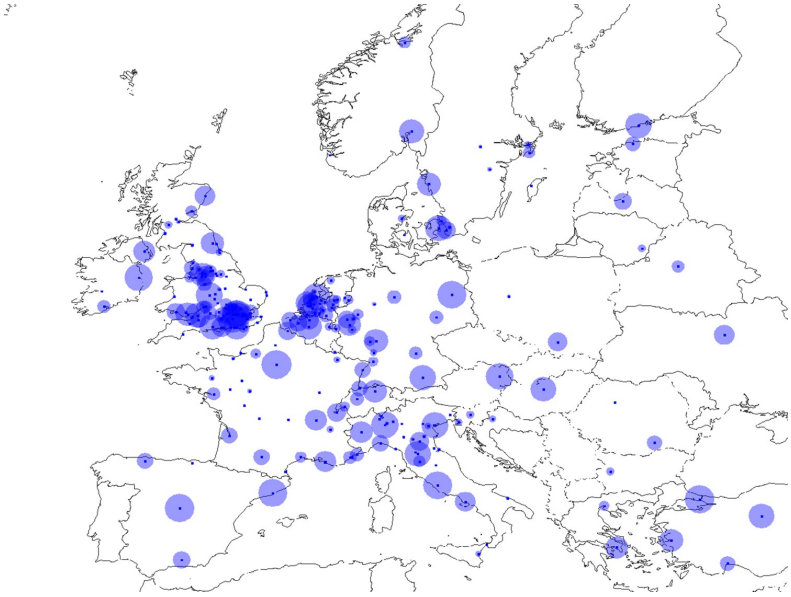


Figure 2.2: Plot of the considered cities in Europe, with the radius of the circles proportional to the number of Flickr photos and tweets posted in the city.

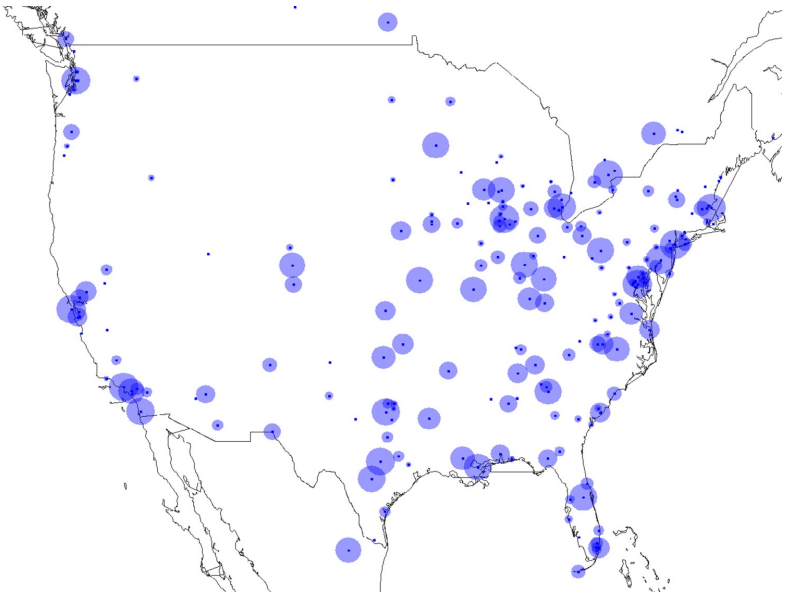


Figure 2.3: Plot of the considered cities in the USA, with the radius of the circles proportional to the number of Flickr photos and tweets posted in the city.

Table 2.1: The place types that are considered in this chapter, together with their corresponding category names in LinkedGeoData (LGD) and Geonames.

place type	LGD categories	Geonames categories
Place of Worship	PlaceOfWorship	S.CH S.MSQE
School	School University	S.SCH
Shop	Shop	S.RET
Restaurant	Restaurant FastFood	S.REST
Graveyard	GraveYard	S.CMTY S.GRVE
Hotel	TourismHotel Motel Hostel	S.HTL
Pub	Pub Bar Cafe	S.PUB S.CAFE
Station	RailwayStation TramStop	S.RSTN S.RSTP S.RSTN S.MTRO
Hospital	Hospital	S.HSP S.HSPC S.HSPD S.HSPL
Monument	Monument Memorial	S.MNMT
Airport	Airport	S.AIRP
Library	Library	S.LIBR
Museum	TourismMuseum	S.MUS
Castle	Castle	S.CSTL

Finally, the dataset has been split in three parts: two thirds of the cities were used as training data (called the training set, $S_{training}$), while one sixth of the cities were used to find optimal values of the parameters in our method (called the development set, S_{dev}). The remaining sixth was used for evaluation (called the test set, S_{test}). This was done by ranking the cities in descending order based on the number of tweets and Flickr photos taken in the city. The cities ranked 1^{st}, 7^{th}, 13^{th} ... were considered as development set, the cities ranked 2^{nd}, 8^{th}, 14^{th} ... as test set, and the remaining cities as training set. In this way we obtain three sets that contain cities with all varieties of number of Flickr photos and tweets.

2.3.2 Collecting Places of Interest

To obtain locations of known places of different types, we have used two open source databases: LinkedGeoData (LGD) and Geonames. We have in particular collected all places in these databases of the types shown in the first column of Table 2.1. These are the types with the highest number of instances in the union of the LinkedGeoData and Geonames database.

In LinkedGeoData and Geonames, some places occur multiple times. However, both the name and location of duplicate entries may be slightly different. Therefore, we used a heuristic based on the approach from [22] to detect and remove duplicates.[2] First, places are indicated as duplicates when they are located

[2]We use a simple heuristic to detect and remove duplicate places as this method is only used to clean our ground truth dataset. This method is not part of our proposed place detection and categorizing methodology. In future work, more advanced data fusion methods could be used to improve the duplication detection and merging (e.g. [29]).

Table 2.2: Statistics of the used datasets of places.

place type	LGD	Geonames	combined	in considered cities
Shop	326 388	38	316 773	64 124
Restaurant	217 145	1 315	215 613	51 647
School	284 141	241 041	349 157	46 473
Place of Worship	315 532	241 745	356 329	45 227
Pub	133 761	0	132 123	32 829
Hotel	67 563	83 210	136 174	28 567
Station	80 849	58 484	125 556	18 225
Hospital	54 363	24 281	59 599	8 400
Monument	35 110	746	32 322	4 598
Library	22 730	11 549	22 946	4 373
Graveyard	136 655	125 481	139 096	3 524
Museum	18 060	5 000	19 421	3 328
Airport	1 138	24 547	25 591	753
Castle	5 043	3 666	8 474	410
total	1 698 478	821 103	1 939 174	312 478

closer than 5 meters to each other. Second, to detect additional duplicates of a given place p all neighboring places of the same type in a range of 100 meter were selected as candidate duplicates. Each of the names of these candidates was converted to lower case, and stripped of category words such as 'restaurant', 'bar', 'tavern', etc. A place from the candidate set is assumed to be a duplicate of p if its Damerau-Levenstein distance to p is sufficiently small. For our experiments, we have used a threshold of $x/3$, with x the maximum of the lengths of both names. This way, we obtained 1,939,174 distinct places of which 312,478 are located in the considered cities. We define K as the set of known places located in the cities of $S_{training}$, which are used to train our model. The places located in S_{dev} and S_{test} are used as ground truth to respectively optimize and evaluate our methodology. An overview of the number of places per type and source can be found in Table 2.2.

2.3.3 Collecting Social Media Data

We collected data from Flickr and Twitter to obtain textual descriptions of places, which will be used to estimate their semantic type.

Collecting Flickr data We crawled the metadata of around 70% of the georeferenced photos from the photosharing site Flickr that were taken before May 2011 and which contain a geotag with street level precision (geotag accuracy of at least 15). Once retrieved, we ensured that at most one photo was retained in the collection with a given tag set and user id, in order to reduce the impact of bulk

uploads [30]. In addition, photos with invalid coordinates or without tags were removed. The dataset thus obtained contains 23,324,644 geotagged photos of which 9,516,714 are located in the considered cities.

Collecting Twitter data We used the Twitter Streaming API to collect tweets. Using the 'Gardenhose' access level, we collected about 10% of the public geotagged tweets posted between March 13, 2012 and June 23, 2012. Because we were specifically interested in the added value of using Twitter, we removed content which was automatically created by other services. More precisely, automatic generated content from Foursquare, Instagram, Path and Yahoo! Koprol was removed. Finally, the tweets were converted to lower case, and URLs and special characters such as #, & and punctuations were removed. After filtering, we ended up with a total number of 30,095,000 tweets of which 8,138,974 are located in the considered cities.

2.4 Methodology

In this section we describe various aspects of our proposed approach to discover places of interest. More precisely, we want to determine the locations in a city C which correspond with places of a given type t. Therefore, we first cluster the locations where Flickr photos have been taken to obtain the locations which potentially correspond to places of interest (POIs) in city C. We then associate with each candidate place of interest a feature vector based on the tags of the Flickr photos that are associated with locations nearby. Afterwards, a query associated with place type t is constructed based on the descriptions of known places of type t. This query is then used to rank the obtained POIs in C, based on the likelihood that they belong to type t. Finally, we discuss how Twitter can be used to further improve the results.

In the remainder of this section we describe the steps of our proposed approach in more detail. To further clarify our methodology, we explain in a running example how each step could be applied to a fictional city E. This city contains three places of interest: a museum, a church and a monument located inside the church. These places are considered as the ground truth of E.

2.4.1 Detecting Places of Interest

In the first step, we determine the locations in city $C \in S_{test}$ that potentially correspond to a place of given type t. To this end, we cluster the locations where Flickr photos have been taken using mean shift clustering [31]. Mean shift clustering is a straightforward iterative procedure that shifts each coordinate to the mean of the coordinates in its neighborhood. This algorithm is particularly suitable for

this task, as it is scalable, does not require a predefined number of clusters, and allows us to adapt the scale at which clusters should be identified. Moreover, mean shift clustering has already been successfully applied to detect POIs from Flickr photos [9].

Let L be the set of coordinates where Flickr photos have been taken in city C. The mean shift $m_b(l)$ of coordinate $l \in L$ is then given by the difference of l and the weighted mean of the coordinates nearby l:

$$m_b(l) = \frac{\sum_{l' \in L \wedge d(l,l') \leq 2b} G_b(l, l') \cdot l'}{\sum_{l' \in L \wedge d(l,l') \leq 2b} G_b(l, l')} - l \tag{2.1}$$

with b the bandwidth parameter, $d(l, l')$ the geodesic distance in meters between coordinate l and l', and $G_b(l, l')$ the kernel function which determines the weight associated with coordinate l' depending on its distance to l. We use a Gaussian kernel for a smooth density estimation:

$$G_b(l, l') = e^{-\frac{d(l,l')^2}{2b^2}} \tag{2.2}$$

The mean shift procedure then computes a sequence starting from all initial coordinates $l_1 \in L$ where

$$l_{i+1} = l_i + m_b(l_i) \tag{2.3}$$

which converges to a location that corresponds to a local maximum of the underlying distribution as $m_b(l_i)$ approaches zero. Based on the obtained clusters, we consider the center of each cluster as a candidate points of interests, called set U^F.

In our running example, two candidate points of interests corresponding to the center of the museum (poi_1) and the center of the church (poi_2) may be detected. This is formally noted by $U_E^F = \{poi_1, poi_2\}$, where poi_i is represented by a latitude and a longitude value.

2.4.2 Describing Places of Interest

We associate a feature vector V_{poi} to each candidate point of interest $poi \in U^F$ based on the tags of the Flickr photos that are associated with locations nearby poi. Let D be the dictionary containing all the tags of the Flickr photos in our dataset, the vector contains a component associated with each word $w \in D$. Formally, for feature vector V_{poi} of candidate point of interest $poi \in U^F$, the component $c_{poi,w}$ associated with word $w \in D$ is given by a Gaussian-weighted count of the number of nearby photos that have been tagged with w. For efficiency, photos whose distance to poi is more than $2\sigma_U$ are not considered:

$$c_{poi,w} = \sum_{f \in F_w \wedge d(poi,f) \leq 2\sigma_U} e^{-\frac{d(poi,f)^2}{2\sigma_U^2}} \tag{2.4}$$

with f a Flickr photo, F_w the set of Flickr photos that contain tag w, σ_U the deviation value used for the description of the candidate POIs in set U^F and $d(poi, f)$ the geodesic distance in meters between poi and the coordinates of the photo f.

The candidate points of interests poi_1 and poi_2 in the fictional city E have associated feature vectors V_{poi_1} and V_{poi_2}, respectively. Assume for instance that $V_{poi_1} = (5.99, 3.81, 0.76, 0, 0, 0)$ and $V_{poi_2} = (0, 0, 0, 7.87, 6.74, 6.63)$ where the six components of these vectors respectively refer to 'museum', 'art', 'bike', 'church', 'statue' and 'monument'.

2.4.3 Constructing a Query

To rank the candidate POIs based on the likelihood that they are associated with the given type t, we first construct an associated query q_t. Let K_t be the set of all known places of type t located in the cities of the training set $S_{training}$ and D_t the dictionary of all words which are indicative for place type t. A query q_t of type t is represented as a vector with one component $q_{t,w}$ associated with each word $w \in D_t$ given by

$$q_{t,w} = \sum_{p' \in K_t} c_{p',w} \tag{2.5}$$

where $c_{p,w}$ is defined similarly as (2.4):

$$c_{p,w} = \sum_{f \in F_w \wedge d(p,f) \leq 2\sigma_K} e^{-\frac{d(p,f)^2}{2\sigma_K^2}} \tag{2.6}$$

with f a Flickr photo, F_w the set of Flickr photos that contain tag w, σ_K the deviation value used for the descriptions of the places in the training set K and $d(poi, f)$ the geodesic distance in meters between poi and the coordinates of the photo f.

Starting from dictionary D containing all the tags of the Flickr photos in our dataset, dictionary D_t is defined as a subset of all the words that are likely to be indicative for type t. To identify such words, feature selection techniques can be used. In this chapter, we discuss chi-square (χ^2) and correlation coefficient (CC) based feature selection. The dictionary D_t is then obtained by taking the m tags with the highest χ^2, respectively CC, value. Chi-square based feature selection has been successfully applied in other research [32] and is defined as

$$\chi^2 = \frac{N \times (O_{w,t} \cdot O_{\not{w},\not{t}} - O_{\not{w},t} \cdot O_{w,\not{t}})^2}{(O_{w,t} + O_{\not{w},t}) \times (O_{w,\not{t}} + O_{\not{w},\not{t}}) \times (O_{w,t} + O_{w,\not{t}}) \times (O_{\not{w},t} + O_{\not{w},\not{t}})} \tag{2.7}$$

where the values are defined as

$$O_{w,t} = \sum_{p' \in K_t} c_{p',w} \tag{2.8}$$

with $c_{p',w}$ as defined in (2.4),

$$O_{w,\not t} = \sum_{p' \in K \setminus K_t} c_{p',w} \tag{2.9}$$

in which K is the set of all known places located in cities of $S_{training}$,

$$O_{\not w,t} = \sum_{w' \in D \setminus \{w\}} \sum_{p' \in K_t} c_{p',w'} \tag{2.10}$$

with D the dictionary of all tags of the Flickr photos in our dataset,

$$O_{\not w,\not t} = \sum_{w' \in D \setminus \{w\}} \sum_{p' \in K \setminus K_t} c_{p',w'} \tag{2.11}$$

and

$$N = \sum_{w' \in D} \sum_{p' \in K} c_{p',w'} \tag{2.12}$$

Value $O_{w,t}$ is the number of occurrences of word w in the descriptions of places of type t, $O_{w,\not t}$ the number of occurrences of w in the descriptions of places of another type than t, $O_{\not w,t}$ the number of occurrences of all words $w' \in D \setminus \{w\}$ in the descriptions of places of type t, $O_{\not w,\not t}$ the number of occurrences of all words $w' \in D \setminus \{w\}$ in the descriptions of places of a different type than t, and N the total number of occurrences of all words $w' \in D$ in the descriptions of all places in K. The correlation coefficient CC, introduced in [15], is a variant of the more popular χ^2 feature selection metric, where $CC^2 = \chi^2$:

$$CC(w,t) = \frac{\sqrt{N} \times (O_{w,t} \cdot O_{\not w,\not t} - O_{\not w,t} \cdot O_{w,\not t})}{\sqrt{(O_{w,t} + O_{\not w,t}) \times (O_{w,\not t} + O_{\not w,\not t}) \times (O_{w,t} + O_{w,\not t}) \times (O_{\not w,t} + O_{\not w,\not t})}} \tag{2.13}$$

CC can be viewed as a 'one sided' χ^2 metric. The correlation coefficient CC selects the words that are highly indicative of membership in a category, whereas the χ^2 metric will also pick out words that are indicative of non-membership in the category. In the evaluation section, we compare the results of our methodology using the CC and χ^2 metric in more detail.

As a final optimization, we exclude from D_t the names of the cities in the training set and the names of the countries in which these cities are located. Lists of alternative names of the cities and their corresponding countries were obtained using Geonames. The rationale behind filtering the names of the cities and countries is as follows: A lot of names of cities from the training set and their corresponding countries have high CC and χ^2 values because some cities have a disproportional number of places of particular types. For example, 5% of the stations in the training set are located in Tokyo leading to a high CC and χ^2 value for the word 'tokyo' when type t is equal to 'station'. This may result in a false positive observation of

a station when the word 'tokyo' is used in other cities. The impact of introducing this additional filter step is described in more detail in the evaluation section. In future work, word sense disambiguation and relatedness measures will be considered to cluster tags by meaning [33].

Note that dictionary D_t may contain indicative words for place type t in different languages. For example, for type 'graveyard' it contains words 'cemetery' (English), 'cementerio' (Spanish) and 'begraafplaats' (Dutch). The proposed approach can thus handle all those different languages without distinguishing them. However, there may be a problem when the same word has a different meaning in different languages. For instance, 'coffeeshop' means in most parts of the world 'an establishment where coffee is served', but may also mean 'a casual, popular-priced restaurant similar to a diner' in the USA, or 'a place where cannabis products are sold and consumed' in the Netherlands[3]. This problem may be solved by constructing a different dictionary and place description for each considered language and/or country. The impact of this approach on the quality of the detected places will be examined in future work.

In our running example, we consider three types of places, i.e. museums, places of worship and monuments. The query q_{museum} associated with type 'museum' contains weighted components $q_{museum,museum} = 109.92$ and $q_{museum,art} = 78.75$. We note that D_{museum} for instance could initially contain the word 'paris' which is eliminated in the final optimization step in the query constructing phase. In addition, $q_{placeofworship}$ contains components $q_{placeofworship,cathedral} = 50.81$ and $q_{placeofworship,church} = 46.80$; and $q_{monument}$ the components $q_{monument,monument} = 80.97$ and $q_{monument,statue} = 78.94$. For clarity of the example, only non-zero $q_{t,w}$ values are mentioned.

2.4.4 Ranking Places of Interest

Using the locations and descriptions of the candidate points of interest U^F in city C and a query q_t associated with place type t, we rank the points of interests based on the likelihood that they belong to type t using a language modeling approach. Other classification methods may be used, e.g., methods based on k-nearest neighbors or decision trees. However, preliminary experiments have shown that the use of language models outperforms the other methods. The probability $\mathrm{P}[poi|q_t]$ that $poi \in U^F$ belongs to type t is estimated as

$$\mathrm{P}[poi|q_t] \propto \prod_{w \in D_t} \mathrm{P}[w|poi]^{q_{t,w}} \tag{2.14}$$

[3] https://en.wikipedia.org/wiki/Coffee_shop

where $q_{t,w}$ is the weighted number of occurrences of word w in query q_t as defined in (2.5). We estimate P$[w|poi]$ using Jelinek-Mercer smoothing as

$$\text{P}[w|poi] = \lambda \cdot \frac{c_{poi,w}}{\sum_{w' \in D} c_{poi,w'}} + (1-\lambda) \cdot \text{P}[w|K] \tag{2.15}$$

with $\lambda \in [0,1]$ and the background model P$[w|K]$ is estimated using maximum likelihood:

$$\text{P}[w|K] = \frac{\sum_{poi' \in K} c_{poi',w}}{\sum_{poi' \in K} \sum_{w' \in D} c_{poi',w'}} \tag{2.16}$$

As the value of P$[poi|q_t]$ may be very small, the values are calculated in log-space to avoid significant loss of precision and underflow:

$$\text{P}[poi|q_t] \propto log \prod_{w \in D_t} \text{P}[w|poi]^{q_{t,w}} = \sum_{w \in D_t} q_{t,w} \cdot log\text{P}[w|poi] \tag{2.17}$$

We denote the right-hand side of (2.17) as $score(poi|t)$:

$$score(poi|t) = \sum_{w \in D_t} q_{t,w} \cdot log\text{P}[w|poi] \tag{2.18}$$

Finally, the candidate points of interest from set U^F are ranked based on their $score(poi|t)$ value, in descending order.

For each considered place type in the running example (i.e., museum, place of worship and monument) we rank the candidate POIs in U_E^F according to the likelihood that they belong to the given type. For museum, we get a $score(poi_1|museum)$ of -139 and a $score(poi_2|museum)$ of $-\infty$ when we set λ equal to 1. This leads to a list where poi_1 is ranked above poi_2. Note that a score of $-\infty$ indicates a likelihood of 0. In a similar way, for both the 'place of worship' and the 'monument' place type, poi_2 is ranked above poi_1. Note that when a candidate point of interest corresponds to several places of different types, it can be ranked first for different types.

2.4.5 Improving Results using Twitter

In the same way as for the Flickr data, we can obtain a ranked list of POIs only using the Twitter data. First, the locations where the tweets have been posted are clustered to find locations of candidate POIs. We will refer to this clustering as U^T. Second, these candidate POIs are ranked based on the terms of the Twitter posts that are associated with locations nearby. This is performed in a similar way as described in the previous sections, where the Flickr data is replaced by the Twitter data.

We can also use the Flickr and Twitter data together to improve the results. To this end, we again use the clustering U^F, which is only based on the Flickr data. We have also tested other clustering approaches to detect locations of candidate POIs. In one approach, the candidate POI set obtained using Twitter (U^T) was used. In a second approach, we clustered both the locations where Flickr photos have been taken and tweets have been posted, called set $U^{F\cup T}$. Finally, the sets U^F and U^T have been combined to $U^F \cup U^T$ in the last approach. Experiments have shown that these alternatives yield worse results, which is why we do not consider them in the remainder of this chapter. After the clustering step, we use the Flickr data and Twitter data separately to get two estimates which indicates the likelihood that a $poi \in U^F$ belongs to a given type t. More precisely, we first use the Flickr data to describe the POIs in U^F, to construct the queries associated with the place types, and to estimate for each $poi \in U^F$ the likelihood that poi belongs to a given type t. The log of this likelihood is indicated by $score^F(poi|t)$ as defined in (2.18). In a similar way, the Twitter data is used to describe the POIs in U^F, to construct the queries, and to estimate for each $poi \in U^F$ the log of the likelihood that poi belongs to type t, given by $score^T(poi|t)$. Afterwards, the $score^F(poi|t)$ and $score^T(poi|t)$ are combined to obtain a $score^{F,T}(poi|t)$ value which indicates the log of the likelihood that poi belongs to type t:

$$score^{F,T}(poi|t) = \eta \cdot score^F(poi|t) + (1-\eta) \cdot score^T(poi|t) \tag{2.19}$$

with $\eta \in [0,1]$. Finally, the candidate points of interest from set U^F are ranked based on their $score^{F,T}(poi|t)$ value, in descending order.

In the running example, we obtained a $score^F(poi_1|museum)$ value of -139. Recall that this value corresponds to the log of the likelihood that poi_1 belongs to place type 'museum', based on the Flickr data. Using the Twitter data, an additional feature vector $V^T_{poi_1}$ describes poi_1 using the tweets in the vicinity of this POI. The components of this vector are for instance equal to $c^T_{poi_1,exposition} = 9.03$ and $c^T_{flower} = 2.68$. The query q^T_{museum} associated with type 'museum' is constructed using the tweets in the vicinity of known museums and contains weighted components $q^T_{museum,museum} = 149.29$ and $q^T_{museum,exposition} = 132.10$. Using $V^T_{poi_1}$ and q^T_{museum} we get $score^T(poi_1|museum)$. When we set η equal to 0.75, a $score^{F,T}(poi_1|museum)$ value of -113 is obtained.

2.5 Evaluation

In this section, we describe how we optimized the parameters using the development set. Subsequently, we use the test set to examine to what extent our methodology is able to discover places which are not yet known by existing databases and to identify errors in existing databases of places.

2.5.1 Parameter Optimization

The task we consider is to discover the locations of possible POIs in a city C and to rank them according to the likelihood that they belong to a given type t. In this section, we use the development set to optimize the quality of these ranked POIs by determining the impact of different parameter values and feature selection techniques. When optimizing the parameter settings, it is useful to consider only one metric to measure the performance of our methodology. Additionally, the used metric has to summarize the performance of our methodology in one value (e.g. between 0 and 100). In particular, this metric must have an optimal value when the distances between the discovered POIs and the places of type t in our ground truth are minimal, and when the POIs which are located very close to a place of type t in our ground truth are ranked at the top. To this end, the quality of a ranked list of POIs associated with city C and type t is measured using the Normalized Discounted Cumulative Gain metric:

$$NDCG(C,t) = \frac{DCG(C,t)}{IDCG(C,t)} \times 100 \tag{2.20}$$

with $DCG(C,t)$ the Discounted Cumulative Gain of the ranking

$$DCG(C,t) = rel(C,t)@1 + \sum_{i=2}^{|U^F|} \frac{rel(C,t)@i}{log_2(i)} \tag{2.21}$$

where U^F is the set of all candidate POIs in city C, and $rel(C,t)@i$ the relevance of the POI at position i in the ranked list, defined as

$$rel(C,t)@i = e^{-\frac{d(poi_i, nn_{i,t})^2}{2h^2}} \tag{2.22}$$

with poi_i the POI at position i of the ranked lists of POIs for city C and type t, $nn_{i,t}$ the place of type t in the ground truth which is nearest to poi_i, $d(poi_i, nn_{i,t})$ the geodesic distance in meters between poi_i and $nn_{i,t}$, and h the deviation value which is set to 40. Furthermore, the Ideal Discounted Cumulative Gain, $IDCG(C,t)$, is defined as the DCG value of the optimal ranking, i.e. when the POIs located in C are ranked by relevance. Finally, we calculate the mean $NDCG$ of all cities in the development set, which is given by

$$MNDCG(t) = \frac{\sum_{C' \in S_{dev}} NDCG(C',t)}{|S_{dev}|} \tag{2.23}$$

with S_{dev} the set of all cities in the development set.

We first optimize for each considered place type the deviation value σ_K from (2.6) and σ_U from (2.4) which is used to describe the known places from the training set K and the obtained candidate POIs U^F, respectively. The optimal value

Table 2.3: Optimal parameter values.

place type	b	σ_K	σ_U	η	m *(Flickr)*	m *(Twitter)*	λ *(Flickr)*	λ *(Twitter)*
Shop	5	25	80	0.44	1000	7000	0.75	0.70
Restaurant	5	5	60	0.43	20	300	0.95	0.95
School	5	5	55	0.82	1600	2900	0.85	0.10
Place of Worship	5	10	35	0.10	400	50	0.55	0.95
Pub	5	5	45	0.87	100	5000	0.95	0.95
Hotel	5	5	50	0.23	1400	1200	0.95	0.95
Station	5	15	50	0.49	50	1500	0.85	0.35
Hospital	5	15	100	0.24	4500	100	0.80	0.45
Monument	5	5	40	0.65	400	6000	0.80	0.90
Library	5	45	45	0.99	50	50	0.95	0.80
Graveyard	25	10	75	0.32	1800	20	0.70	0.90
Museum	5	15	45	0.88	1100	6000	0.75	0.75
Airport	25	60	60	0.76	700	20	0.10	0.50
Castle	25	15	70	0.35	3000	100	0.95	0.90

Table 2.4: Optimal number of features (m) and corresponding MNDCG values for χ^2, CC and CC+filter on the place descriptions from Flickr.

place type	**Optimal number of features (*m*)**			***MNDCG***		
	χ^2	***CC***	***CC+filter***	χ^2	***CC***	***CC+filter***
Shop	2000	1000	1000	50.13	50.13	**50.33**
Restaurant	20	20	20	56.61	56.61	**56.62**
School	1600	1600	1600	35.36	35.36	**36.98**
Place of Worship	300	400	400	58.38	58.41	**58.46**
Pub	100	100	100	64.14	64.77	**67.79**
Hotel	1400	1400	1400	59.61	59.61	**60.38**
Station	50	50	50	74.08	**74.09**	**74.09**
Hospital	4500	4500	4500	42.12	42.14	**42.36**
Monument	500	500	400	63.30	63.64	**64.33**
Library	50	50	50	52.49	52.61	**54.44**
Graveyard	1700	1800	1800	74.34	74.34	**74.55**
Museum	100	1100	1100	60.96	61.24	**61.36**
Airport	700	700	700	67.55	67.56	**67.57**
Castle	2800	2800	3000	85.58	85.60	**85.63**

for each considered place type can be found in the second and third column of Table 2.3. The most informative words associated with the given place type t can be found in the tags of the Flickr photos taken close nearby the places of type t in the training set. To detect new POIs on the other hand, also tags of Flickr photos taken further away from the POI may be useful to determine its place type. For example, Flickr photos may have been taken at some distance of the actual POI. This observation leads to a smaller deviation value σ_K for the descriptions of the places in the training set than the deviation value σ_U for the description of the obtained candidate POIs.

Using these σ values, we compare the feature techniques described above, i.e.

Table 2.5: Optimal number of features (m) and corresponding MNDCG values for χ^2, CC and CC+filter on the place descriptions from Twitter.

place type	Optimal number of features (*m*)			*MNDCG*		
	χ^2	*CC*	*CC+filter*	χ^2	*CC*	*CC+filter*
Shop	300	7500	7000	**46.00**	45.48	45.58
Restaurant	300	300	300	52.14	52.14	**52.15**
School	4500	4500	2900	32.38	**32.41**	32.40
Place of Worship	50	50	50	36.14	**36.15**	36.14
Pub	6000	5000	5000	54.25	**54.30**	54.28
Hotel	1200	1200	1200	**48.25**	47.70	47.71
Station	100	100	1500	52.17	**52.63**	52.43
Hospital	100	100	100	**33.96**	**33.96**	**33.96**
Monument	4500	6000	6000	47.01	**47.18**	47.17
Library	50	50	50	**36.68**	36.45	36.61
Graveyard	20	20	20	**59.00**	**59.00**	**59.00**
Museum	6000	5500	6000	39.89	**39.90**	**39.90**
Airport	20	20	20	65.38	65.38	**65.39**
Castle	100	100	100	**81.51**	**81.51**	**81.51**

χ^2, the correlation coefficient (CC) and CC after filtering city and country names ($CC + filter$). For these experiments, we use for each place type their optimal σ values, a bandwidth value b (see Equation 2.2) of 5, and a λ value (see Equation 2.15) of 0.9. Tables 2.4 and 2.5 show for each described feature selection technique the optimal number of features m and their corresponding MNDCG value. We indicate that for some place types such as libraries, restaurants and places of worship the informative terms are located at the very top of the ranked features leading to an optimal number of features around 150. For other types, e.g. schools and hotel, the informative terms are more distributed leading to larger m values. Note that the optimal number of features may also vary depending on when Flickr data or Twitter data is used.

When only the Flickr data is used (Table 2.4), we find that using CC results in a significant improvement over χ^2 (Wilcoxon signed ranks test, $p < 0.01$). As an example, the MNDCG values of the χ^2 and CC feature selection for place type 'monument' are plotted in Figure 2.4. The top 300 ranked words according to χ^2 do not contain words that strongly characterize places of other types than monuments resulting in behavior which is similar to CC when m is smaller than 300. However, χ^2 ranks for place type 'monument' the words 'nationalcemetery' and 'food' at respectively position 359 and 384. While such terms are potentially useful to exclude particular other places types, they appear to be less effective than the terms that are directly indicative of monuments that are preferred by CC. For the other considered place types, a similar observation can be made. When we compare CC without and with the filtering step we also get a significant improvement (Wilcoxon signed ranks test, $p < 0.01$). For example, for place type

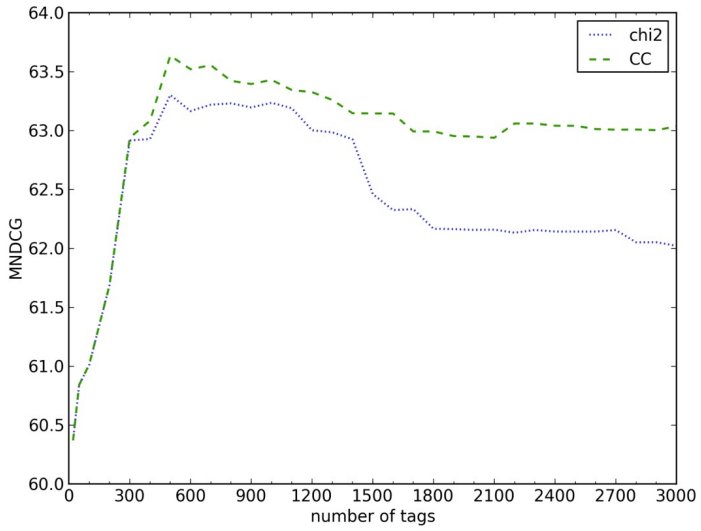


Figure 2.4: MNDCG values for different number of tags when χ^2 and CC feature selection is used on the place descriptions from Flickr, for place type 'monument'.

'hotel' the word 'italy' receives the largest CC value because a lot of hotels in the training set are located in Italy (more precisely in Venice), but this word may also refer to Italian design, cars or restaurants. By filtering such terms, the effectiveness of the method is improved. In particular, we get significant improvement of the optimal MNDCG value. Finally, by comparing the MNDCG values of the χ^2 and $CC + filter$ we conclude that latter technique performs significantly better (Wilcoxon signed ranks test, $p < 0.01$).

Surprisingly, in contrast to Flickr, there is no clear difference in the use of the different feature selection techniques for the Twitter data (Table 2.5; Wilcoxon signed ranks test, $p > 0.2$). This relates to the fact that Twitter data contains a lot of non-informative terms such as opinions, statements and personal status updates [34]. Moreover, tweets contain less geographic information such as city and country names than Flickr [35], which reduces the impact of our filtering step. We note that filtering out names of cities and countries may even decrease the performance. For schools, for instance, the word 'lacrosse' is excluded because it may refer to the city La Crosse, located in Wisconsin, United States. However, 'lacrosse' may also refer to a team sport which is played in many US colleges and the occurrence of the word 'lacrosse' may therefore indicate the presence of a school. For several place types, the result is not very sensitive to the actual number of features which is used. In such a case, the optimum number of features may

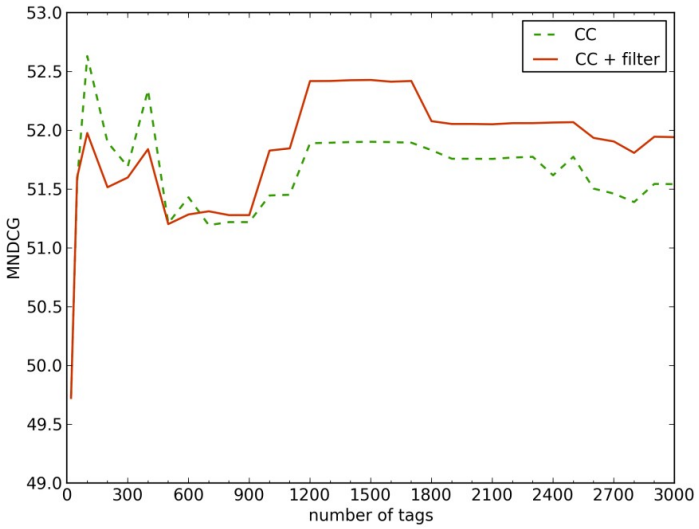


Figure 2.5: MNDCG values for different number of tags when CC and CC+filter feature selection is used on the place descriptions from Twitter, for place type 'station'.

change quite drastically between CC and $CC + filter$. This is most pronounced in the case of 'station', as shown in Figure 2.5. In the rest of this chapter, we will use $CC + filter$ to obtain a fair comparison between Flickr and Twitter.

Using the optimal settings of our methodology, we finally optimize the remaining parameters b (see Equation 2.2), λ (see Equation 2.15) and η (see Equation 2.19). The optimal values of all parameters can be found in Table 2.3.

2.5.2 Quantitative Evaluation

In this and the following section, we apply our proposed methodology to the cities from the test set, using the optimal parameters that were obtained from the cities in the development set. In this section we perform a quantitative evaluation, where we compare the difference in performance when Flickr or Twitter data is used, and demonstrate how Flickr and Twitter can be combined to optimize the results. Given a city $C \in S_{test}$ and a place type t, we evaluate the rankings of POIs using the *Average Precision* $AP(C, t, y)$ metric in addition to the NDCG metric defined in (2.20). The average precision metric is added because it can be used for a more detailed analysis than the NDCG metric by using different distance thresholds (indicated by parameter y). On the other hand, the use of NDCG is more useful for parameter optimization because it summarizes the performance of

our methodology in one metric. As for NDCG, the AP value lies between 0 and 100, in which a higher value means a better performance.

To define the average precision metric, we first define *Precision at position n*, $P(C,t,y)@n$, which is the fraction of the top n ranked POIs that are relevant to the user's information need. $P(C,t,y)@n$ is formally given by

$$P(C,t,y)@n = \frac{\sum_{i=1}^{n} relevant(C,t,y)@i}{n} \times 100 \tag{2.24}$$

For the calculation of the precision, a point of interest $poi \in U^F$ is considered as relevant if the ground truth of city C contains a place of type t within y meters of poi, where we will consider different values of y:

$$relevant(C,t,y)@i = \begin{cases} 1 & d(poi_i, nn_{i,t}) \leq y \\ 0 & otherwise \end{cases}$$

The *Mean Precision at position n*, $MP(t,y)@n$, is defined as the mean of the *Precision at position n* values of all cities in the test set:

$$MP(t,y)@n = \frac{\sum_{C' \in S_{test}} P(C',t,y)@n}{|S_{test}|} \tag{2.25}$$

with S_{test} the set of all cities in the test set. In addition, the *Recall at position n* metric, $R(C,t,y)@n$, corresponds to the fraction of POIs that are relevant which are successfully ranked in the top n POIs:

$$R(C,t,y)@n = \frac{\sum_{i=1}^{n} relevant(C,t,y)@i}{\sum_{i=1}^{|U^F|} relevant(C,t,y)@i} \times 100 \tag{2.26}$$

By computing a precision and recall for each $n \in [1, |U^F|]$ one can plot a precision-recall curve and consider the area under the curve as the relevance of the ranked list. This value can be approximated using the *Average Precision* metric $AP(C,t,y)$ [36], which is defined as

$$AP(C,t,y) = \frac{\sum_{n=1}^{|U^F|} relevant(C,t,y)@n \cdot P(C,t,y)@n}{\sum_{n=1}^{|U^F|} relevant(C,t,y)@n} \tag{2.27}$$

We finally define the *Mean Average Precision* $MAP(t,y)$ as the mean of the Average Precision values of all cities in the test set:

$$MAP(t,y) = \frac{\sum_{C' \in S_{test}} AP(C',t,y)}{|S_{test}|} \tag{2.28}$$

Table 2.6: MNDCG of the ranked points of interest when Flickr and/or Twitter data is used. The last column indicates the MNDCG values for London when both the Flickr and Twitter data is used.

place type	Flickr (SVM)	Flickr	Twitter	Flickr+Twitter	London
Shop	43.14	46.27	42.26	**46.73**	89.20
Restaurant	50.99	54.19	52.36	**56.82**	92.79
School	34.12	34.19	33.20	**35.26**	68.89
Place of Worship	49.37	49.77	33.13	**49.83**	78.76
Pub	61.87	66.17	57.77	**66.97**	95.27
Hotel	50.97	51.69	41.69	**54.08**	88.17
Station	69.70	69.89	49.98	**71.73**	87.86
Hospital	36.24	36.14	33.56	**37.68**	71.97
Monument	66.91	67.71	55.45	**67.76**	83.10
Library	53.65	**54.22**	40.37	54.07	59.08
Graveyard	74.90	**74.97**	60.60	74.42	-
Museum	60.88	64.76	41.73	**65.52**	76.17
Airport	68.39	**68.65**	60.59	68.62	-
Castle	84.79	**86.45**	79.00	86.38	95.02

with S_{test} the set of all cities in the test set. The MAP and MNDCG values of each considered approach are listed in Tables 2.6 and 2.7. In the remainder of this section, we discuss each approach in more detail.

Flickr results We start our experiments by only using the Flickr data to discover POIs in a city C and to rank them according to the likelihood that they are associated with a given place type t. First, we use the methodology from our previous work as a baseline [13, 14]. To this end, we cluster the locations where Flickr photos have been taken to obtain the locations of candidate POIs and associate a description to them as described above. We then train a multi-class support vector machine (SVM) classifier [37] for a given place type t based on the descriptions of the places in the training set. Subsequently, we use this classifier to rank the locations which potentially contain a place of interest based on the probability that they contain a place of the given type t. The performance of this approach can be found in the columns labeled with 'Flickr (SVM)' in Tables 2.6 and 2.7.

Second, we used the approach described in this chapter on the Flickr data to discover places of a given type. The main difference with our previous work is that we replaced the SVM classifier by a language model approach and introduced a feature selection technique. The performance of this approach is shown in the 'Flickr' columns of Tables 2.6 and 2.7. By comparing the performances of the two approaches, we found that our new approach significantly outperforms the SVM based baseline (Wilcoxon signed ranks test, $p < 0.01$).

Table 2.7: MAP values of the ranked points of interest Flickr and/or Twitter data is used.

place type	**Flickr (SVM)**			**Flickr**			**Twitter**			**Flickr+Twitter**		
	25m	**100m**	**1km**	**25m**	**100m**	**1km**	**25m**	**100m**	**1km**	**25m**	**100m**	**1km**
Shop	6.40	18.70	63.69	8.03	25.32	66.88	5.74	17.59	62.54	**8.14**	**26.13**	**68.18**
Restaurant	14.99	23.84	64.58	16.45	32.31	67.78	15.74	28.10	64.74	**17.31**	**36.23**	**71.09**
School	4.97	8.76	62.00	4.83	10.64	64.98	2.18	5.91	66.89	**5.82**	**11.03**	**68.64**
Place of Worship	12.12	19.86	61.66	12.10	24.22	64.98	5.15	6.57	57.33	**12.13**	**24.37**	**66.61**
Pub	29.07	37.70	69.73	31.31	46.90	75.15	26.01	36.04	70.37	**32.60**	**49.01**	**78.83**
Hotel	7.30	20.67	60.28	7.92	26.77	66.73	3.99	14.03	52.13	**9.25**	**29.82**	**70.30**
Station	35.32	54.71	63.23	35.71	58.95	65.68	21.65	31.39	58.54	**36.34**	**61.36**	**69.72**
Hospital	12.14	19.78	36.54	12.90	21.15	41.64	10.99	16.22	36.23	**13.63**	**22.44**	**42.86**
Monument	44.63	49.87	68.02	44.62	53.96	75.64	40.60	43.31	63.66	**44.67**	**54.04**	**76.92**
Library	28.52	34.02	53.90	**28.97**	**36.63**	55.27	25.31	26.41	49.58	28.96	36.40	**59.56**
Graveyard	57.75	66.20	61.96	**58.50**	66.28	**62.47**	54.55	54.69	59.67	58.10	**66.53**	61.84
Museum	30.30	40.01	62.34	32.18	46.50	71.68	24.36	25.80	50.34	**33.07**	**46.69**	**72.87**
Airport	49.78	54.08	67.96	49.91	54.51	**68.51**	49.38	53.01	54.64	**50.73**	**55.11**	68.17
Castle	76.27	81.13	80.35	**77.34**	82.84	82.10	75.58	76.04	77.65	77.28	**82.88**	**82.67**

Table 2.8: Mean Precision at 1, MP(t,500)@1, of the ranked points of interest when Flickr is used.

Shop	Restaurant	School	Place of Worship	Pub	Hotel	Station
60.23	59.09	54.55	63.64	76.14	68.18	81.48
Hospital	**Monument**	**Library**	**Graveyard**	**Museum**	**Airport**	**Castle**
51.59	69.32	57.95	55.68	64.77	67.05	78.41

To further interpret the results we calculated *Mean Precision at one*, denoted as $MP(t, 500)@1$ (see Table 2.8). Based on these metric values, we can conclude that for 81% of the cities in the test set the highest ranked POI is located within 500 meter of a known station. This is due the fact that in the most cities there are a lot of pictures taken nearby the main train station, and such pictures typically have highly indicative tags such as 'train', 'station'' and 'railway'. However, there are some challenges with using Flickr for detecting places. For instance, for 48% of the cities in the test set, the highest ranked POI is not located within 500 meter of a known hospital. One reason is that the Flickr tags can be misleading (e.g. a photo of an ill person far away from a hospital). Second, our Flickr data may contain no photos with descriptive tags taken nearby a smaller hospital of the city. Third, some hospitals are found by our methodology using Flickr, but the distance between the location of the detected POI and the hospital is larger than 500 meter. This is mainly due the fact that most of the pictures at the hospital are taken in the hospital rooms, whereas the ground truth may refer to another part of the hospital (e.g. the main entrance). Finally, it may be the case that an actual hospital is found, which is not contained in our ground truth, as LGD and Geonames are inherently incomplete. The effect of missing places in the ground truth will be investigated in more detail in the next section.

Twitter The results for the Twitter data are shown in the columns labeled with 'Twitter' in Tables 2.6 and 2.7. Although a large number of tweets are not informative [5], we observed that some tweets are very useful to recognize place types. For example tweets such as 'About to have dinner #feelsgood', '@DavidSahadi: Enjoying the first (of a few) micro beers at the outdoor bar at Big River Brewery' and 'waiting for the train...' may indicate the occurrence of restaurants, pubs and stations, respectively. However, our training data hardly contain tweets describing places of types such as places of worship and hospitals, resulting in low MNDCG and MAP values for these types.

Flickr and Twitter combined Comparing the results obtained using the Twitter data with the results obtained using the Flickr data, we find significantly better MAP and MNDCG values for Flickr (Wilcoxon signed ranks test, $p < 0.01$). Based on this observation, we may conclude that Flickr tags are more informative

for finding places of a given type than Twitter posts. However, when we use both Flickr tags and Twitter terms ('Flickr+Twitter' columns Table 2.6 and 2.7), we get a further significant improvement for the average MNDCG and MAP values over all considered place types (Wilcoxon signed ranks test, $p < 0.01$). However, no clear improvement can be observed for place types with only a few instances in our ground truth dataset such as castles and airports.

We observed the best performance for London. With over 600,000 Flickr photos and over 120,000 tweets this is the city with most social media data in our test set. The *MNDCG* values for London are shown in the column labeled with 'London' in Table 2.6. These values suggest, somewhat unsurprisingly, that the number of available photos and/or tweets substantially impacts the performance of our method. With our current dataset, the performance for London is sufficiently high to support practical applications, but this may not yet be the case for some smaller cities. As more and more geo-annotated social media becomes available, however, we could expect to see a comparable performance for a wider range of cities. Still, even for London, the performance varies substantially across different types of places. As people are less likely to tweet from a library than from a pub, it should perhaps not come as a surprise that the method works better for pubs. To further widen the applicability of the proposed method, a wider range of sources, beyond Flickr and Twitter, may need to be considered. The ability of discovering new places of interest in London will be further investigated in the next section.

2.5.3 Qualitative Evaluation

Discovering New Places of Interest In this section, we will analyze to what extent our method can discover places of type t in a city C that are not yet contained in LinkedGeoData, Geonames, Foursquare and Google Places. To find such places, we first remove from the results those places that are within distance $2\sigma_U$ from a place in the ground truth of the same type; see Table 2.3 for the values of sigma for each place type. We will focus on London to get a deeper insight in the ability of our methodology to detect new places.

Table 2.9 shows the top 10 of the resulting rankings, and indicates which places can not been found in Google Places or Foursquare when a user searches for places of a particular type (databases accessed on February 27, 2013). This may be because the places are not included in Google Places or Foursquare at all, or because they are included but classified as another type. Entries in Table 2.9 are shown in bold if they are not included in Google Places or Foursquare. Additionally, they are marked with Go and Fo is they are not included in Google Places and Foursquare, respectively, and with Go and Fo if they are only included with a different type. The place names mentioned in the table have been manually determined, as detecting place names is outside the scope of this chapter. For each

Table 2.9: Top 10 of the discovered places in London which are not yet included LGD and Geonames. Places are shown in bold if they are not included in Google Places or Foursquare. Additionally, they are marked with Go and Fo if they are not included in Google Places and Foursquare, respectively, and with Go and Fo if they are only included with a different type. Finally, errors are indicated in italic.

place type	1st place	2nd place	3rd place	4th place	5th place
Shop	Nippon and Korea Centre	New Look Oxford Street	Marks and Spencer	Selfridges and Co	**Savana (Portobello Road)**^{Go,Fo}
Restaurant	Zizzi	Otto	Pain Quotidien^{Go,Fo}	Carluccio's	Tay Do
School	University College London	Imperial College	City of Westminster College	University of the Arts	UCL Cruciform Building
Place of Worship	Westminster Cathedral	Southwark Cathedral	St Stephen Walbrook	St Sophia Greek Cathedral	St Mary Aldermary Church
Pub	Off Broadway	The Anchor	The White Horse	The Roxy	Green Man and French Horn^{Go,Fo}
Hotel	Bayswater Inn Hotel	Premier Inn Hotel	The Dorchester	Kensington Close^{Fo}	Vicarage Private
Station	Queens Grove	*train*	Waterloo	Pimlico	Fenchurchstreet station
Hospital	Royal Hospital^{Fo}	*Photo of the St Thomas' Hospital*	St Mary's Hospital	*Temperance Hospital*	*abandoned children's hospital*
Monument	Victoria Memorial	Albert Memorial^{Go}	Tower Hill Memorial^{Go,Fo}	**Webmister Abbey Lions Memorial**^{Go,Fo}	Monument of the Great Fire of London
Library	**Birkbeck Library**^{Go}	Science Museum Library	Maughan Library	**SOAS Library**^{Go}	Peckham Library
Graveyard	Brompton Cemetery	Bunhill Fields Burial Ground	Nunhead Cemetery	Saint Pancras Cemetery^{Go,Fo}	St. George's Gardens^{Go}
Museum	Science Museum	Imperial War Museum	Design Museum	Saatchi Gallery	Clink Prison Museum
Castle	*elephant and castles*	The Pirate Castle^{Go,Fo}	Buckingham Palace	Kensington Palace^{Fo}	*Castle Battersea*

place type	6th place	7th place	8th place	9th place	10th place
Shop	House of Gifts^{Go,Fo}	*shoppers*	**National Portrait Gallery Shop**^{Fo}	Rokit	Harrods
Restaurant	**Sen Nin on Islington Park St.**^{Go}	Jamie Oliver's Fifteen	Antariya Sushi Bar	*Hive Bar*	Regency^{Fo}
School	King's College	The Barlett faculty	Royal College of Surgeons	College of Communication	**Spa school**^{Fo}
Place of Worship	St Olave Church	St Martin in the Fields Church	Christ Church	St James' Church	St George's Cathedral
Pub	Daily Grind	Horse and Groom	*Hops bar*	Draft House Tower Bridge	The Elgin
Hotel	The Sanctuary^{Fo}	St Pancras Renaissance Hotel	*Chelsea Bridge*	**Great Northern Kings Cross**^{Go,Fo}	The Wellesley
Station	*railway track*	*from the train*	*central station*	*rail tracks*	*train portrait*
Hospital	*Science Museum*	The Royal Marsden	*londonroyalhospital*	St Pancras Hospital	*Old Royal Free*
Monument	**Statues on entrance County Hall**^{Go,Fo}	**The Women of World War II**^{Go,Fo}	**Statue of Richard the Lionheart**^{Fo}	**St George Statue**^{Go,Fo}	Buxton Memorial Fontain
Library	Borough Road Library	Senate House Library	**SSEES Library**^{Go}	Idea Store Whitechapel	LSE Library
Graveyard	**All Saints Church Cemetery**^{Go,Fo}	Postman's Park^{Go,Fo}	Paddington Cemetery	St John's Wood Church Gardens^{Go,Fo}	**Royal Hospital Old Burial Ground**^{Go,Fo}
Museum	**Sir John Ritblat Gallery**^{Go}	Foundling Museum	Wellcome collection	*Museum of London sign*	Kirkaldy Testing Museum
Castle	*The Castle*	Victoria Tower	*Dublin Castle*	*elephant and castles*	*elephant and castles*

of the discovered places, we manually assessed whether they were of the correct type. The erroneously detected places are those shown in italic.

In London, our method is able to find places of worship, schools, shops, restaurants, graveyards, castles, hotels, pubs, stations, libraries, museums and monuments that are not yet included in our LinkedGeoData and Geonames. Our method was not able to find new airports because the considered region of London contains no airports. Several of these places are not yet included in the Google Places and Foursquare database. As shown in Table 2.9, places not present in Google Places are for instance shops, restaurants, hotels, monuments, libraries, graveyards and museums. Additionally, our method is able to extend Foursquare with shops, schools, monuments and graveyards. Finally, some places such as the Savanna shop at Portobello Road, the Women of World War II monument, and All Saints Church Cemetery are neither included in Foursquare nor Google Places. Furthermore, several places were detected that were already present in Foursquare and Google Places, but without the desired type associated.

Our proposed method is thus able to detect places which are not available in LinkedGeoData, Geonames, Google Places and Foursquare. However, the detected places should be manually checked before adding them to existing databases to obtain a 100% of accuracy. Closer examination of the detected places revealed some challenges with using social media. The first challenge is that Flickr photos may be taken at a far distance from the place of interest (e.g. a photo taken from the St. Thomas' Hospital taken at Leathermarket Gardens more than 500 meter away). Second, the used Flickr and Twitter data may be out-of-date (e.g. a photo of the Hops bar which closed down). Third, the Twitter term and Flickr tags corresponding to a name of a place or region may incorrectly suggest the presence of a place of a particular type (e.g. the tag 'Elephant and Castle', corresponding to a major Junction in London, incorrectly suggest the presence of a castle). Finally, the Flickr tags and Twitter terms may not be related to the place nearby the location of the user (e.g. the tweet 'waiting for a taxi to go to the hospital').

Validation of Known Places of Interest In the previous sections, we have described how social media can be used to extend databases of places. Another way of improving databases of places is to identify and remove incorrect information in these databases. The presence of incorrect place information may be due to various reasons: places of interest may have been closed, their type may have been changed (e.g. a shop converted into a pub), or the places may even have been incorrectly added to the database. However, it is very time-consuming to manually check the correctness of the data in existing databases. We describe in this section how our methodology can be used to facilitate this data validation process. In particular, given a type t and the locations of the places in the databases which are associated with this type, we indicate which places are most likely incorrect.

Table 2.10: Top 5 of the Foursquare places in London which are most likely incorrect. Places are marked with 1 if the place type is incorrect, with 2 if the place is incorrect located and with 3 if the Foursquare place is no place of interest at all. Finally, errors are indicated in italic.

place type	**1st place**	**2nd place**	**3rd place**	**4th place**	**5th place**
Shop	William Hill [1 2]	*Londis*	The Pantry [2]	Specsavers [2]	nternational Food Centre [2]
Restaurant	BananaTree [2]	Favourite Chicken [2]	Quality Caf [1 2]	*5 Skehans Thai*	Sticky Fingers [2]
School	IBAM London [2]	London Studio Centre [2]	*School of Pharmacy*	*SAE Institute*	*London Knowledge Lab*
Place of Worship	Tasmin [3]	St Anne's Church [2]	*MRBC*	Jon Bon Jovi's Dressing Roo [3]	*Church on the Corner*
Pub	V.V. Coffee Bar [2]	*Charlton*	The Clarrence [2]	The Asylum [2]	Home Of Morris [3]
Hotel	*Quality Maitrise Hotel*	5 Doughty Street [3]	Alternative Urban Residence [3]	Jury's Inn [2]	Dylan Apartments [2]
Station	Clapham High [2]	Euston Station [2]	Platform 13 - Gatwick Express [2]	London Field [2]	Brondesbury [2]
Hospital	*London Chest Hospital*	*Tavistock Centre*	*Brondesbury medical center*	*BMI The London Independent Hospital*	Ruskin Wing [2]
Monument	Harley Street and Cavendish Street [3]	New River Walk [2]	The Hellotxtplex [3]	Carlyle's House [1]	Tower Hamlets Labour Party [1]
Library	Clapham Library [2]	Nunhead Library [2]	*Regents Park Library*	CLR James Library [2]	*Paddington Library*
Graveyard	Kensal Green [2]	Behind you [3]	Bunhill fields [2]	*St Paul's Churchyard*	The Stylenoir Lair [3]
Museum	18 Stafford Terrace [3]	*The Jewish Museum*	The Pill Box [1 2]	*Royal Mews*	Epio HQ [1]
Airport	TfL Bus 12 [3]	Admirals Club T3 [1]	Biggin Hill Airport [2]	Heathrow Airport [2]	*The London Heliport*

Places are considered as incorrect when there is no place of type t at their location.

For this case study we use the database of Foursquare, a platform on which users can freely add places to the database. The database contains a lot of unverified places, indicating that the owners of the places of interest have not claimed and did not verify the place information. For instance, about 95% of the places we collected from London were not verified. For these unverified places in particular, a method to automatically assess the likelihood that they are accurate would be useful. We first collected 21,436 Foursquare places from London with a type corresponding to one of the considered place types in this chapter. The task we consider is to determine which of the collected places are most likely incorrect. In particular, given a type t and the locations of the places in the Foursquare database which are associated with this type, we used the Flickr and Twitter data posted nearby the locations to rank them based on the likelihood that there is no place of type t located nearby.

The results are shown in Table 2.10. Places are marked with ^{1} if the type of the place is incorrect, with ^{2} if the place is incorrectly located and with ^{3} if the Foursquare place is no place of interest at all. Finally, detected Foursquare places that are wrongly indicated as incorrect are indicated in italic. Most of the places which are considered most likely to be incorrect are indeed incorrect, most often because they have an incorrect location. Additionally, some places have a wrong associated type. For instance Epio HQ is a software company which is incorrectly categorized as museum. Finally, some places in the Foursquare databases do not correspond with a general place of interest. Examples are the 'pub' labeled 'Home Of Morris' which corresponds to someone's home.These results confirm that our method is able to facilitate the detection of incorrect information in databases of places.

2.6 Conclusion

In this chapter, we demonstrated how social media can be used to improve existing databases of places. We first used mean shift clustering on the locations of a set of Flickr photos to obtain the locations which potentially correspond to places of interest (POIs) in a given city C. We then associated with each candidate POI a feature vector based on the tags of the Flickr photos that are associated with locations nearby. Afterwards, we associated a query with each place type t based on the descriptions of known places of that type. The obtained query is used to rank the candidate POIs based on the likelihood that they belong to type t. To produce this ranking, we relied on a language modeling approach, which performed significantly better than the Support Vector Machine classifier used in our previous work [13, 14]. Finally, we discussed how Twitter can be used to improve the results.

In the optimization phase of our proposed methodology, we analyzed the behaviour of different feature selection techniques. We concluded that for the Flickr data, correlation coefficient feature selection [15] performs significantly better than χ^2. The performance of the proposed methodology was further significantly improved when names of the cities in the training set and the names of the countries in which these cities are located were removed from the features. Surprisingly, in contrast to Flickr, we did not find a clear difference in performance between the use of the different feature selection techniques for the Twitter data.

We performed a large-scale evaluation on 88 different cities. Using Flickr, our methodology was for instance able to find a location which is within 500 meter of a known station for 81% of the cities in the test set. We concluded that Flickr tags are more informative for finding places of a given type than Twitter posts. However, as we have demonstrated in this chapter, using tweets in addition to Flickr photos can still be used to improve the quality of the results. We further examined the results for London in more detail to analyze to what extent our approach can discover new places of a particular type. Based on this evaluation, we could conclude that our method is able to detect places which were not yet included in LinkedGeoData, Geonames, Google Places and Foursquare. Additionally, we explained how our methodology can be used to identify errors in existing databases of places such as Foursquare.

Acknowledgment

Steven Van Canneyt is funded by a Ph.D. grant of the Agency for Innovation by Science and Technology in Flanders (IWT). We are grateful to Olivier Van Laere for his help with collecting and processing some of the data we have used in this chapter.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. *The Semantic Web*. Scientific American Magazine, 284(5):34–43, 2001.

[2] J. Hoffart, F. Suchanek, K. Berberich, and G. Weikum. *YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia*. Artificial Intelligence, 194(1):28–61, 2013.

[3] R. Navigli, D. Informatica, and S. P. Ponzetto. *BabelNet : Building a Very Large Multilingual Semantic Network*. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, number July, pages 216–225, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[4] S. Auer, J. Lehmann, and S. Hellmann. *LinkedGeoData: Adding a spatial dimension to the web of data*. In A. Bernstein, D. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, editors, Proceedings of the 8th International Semantic Web Conference, volume 5823, pages 731–746, Chantilly, VA, USA, 2009. Springer Berlin Heidelberg.

[5] V. Murdock. *Your mileage may vary: on the limits of social media*. SIGSPATIAL Special, 3(2):62–66, 2011.

[6] T. Rattenbury, N. Good, and M. Naaman. *Towards automatic extraction of event and place semantics from Flickr tags*. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 103–110, New York, NY, USA, 2007. ACM.

[7] T. Sakaki. *Earthquake shakes Twitter users: Real-time event detection by social sensors*. In Proceedings of the 19th International Conference on World Wide Web, pages 851–860, 2010.

[8] R. Lee, S. Wakamiya, and K. Sumiya. *Discovery of unusual regional social activities using geo-tagged microblogs*. World Wide Web, 14(4):321–349, 2011.

[9] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. *Mapping the world's photos*. In Proceedings of the 18th International Conference on World Wide Web, pages 761–770, New York, NY, USA, 2009. ACM.

[10] S. Van Canneyt, S. Schockaert, O. Van Laere, and B. Dhoedt. *Time-dependent recommendation of tourist attractions using Flickr*. In Proceedings of the 23rd Benelux Conference on Artificial Intelligence, pages 255–262, 2011.

[11] M. D. Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. *Automatic construction of travel itineraries using social breadcrumbs*. In Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, pages 35–44, New York, NY, USA, 2010. ACM.

[12] S. Jain, S. Seufert, and B. Srikanta. *Antourage: mining distance-constrained trips from Flickr*. In Proceedings of the 19th International Conference on World Wide Web, pages 1121–1122, New York, NY, USA, 2010. ACM.

[13] S. Van Canneyt, B. Dhoedt, and T. Demeester. *Predicting the popularity of online news using curve fitting and gradient tree boosting*. in preparation.

[14] S. Van Canneyt, S. Schockaert, O. Van Laere, and B. Dhoedt. *Using social media to find places of interest: A case study*. In Proceedings of the 2012 ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information, pages 2–8, 2012.

[15] H. T. Ng, W. B. Goh, and K. L. Low. *Feature selection, perceptron learning, and a usability case study for text categorization*. In Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 67–73, 1997. Available from: http://dl.acm.org/citation.cfm?id=258537.

[16] C. Kwok, O. Etzioni, and D. S. Weld. *Scaling question answering to the web*. ACM Transactions on Information Systems, 19(3):242–262, 2001.

[17] O. Etzioni, M. Cafarella, D. Downey, A.-m. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. *Unsupervised Named-Entity Extraction from the Web : An Experimental Study*. Artificial Intelligence Journal, 165(1):1–42, 2005.

[18] F. Wu and D. S. Weld. *Autonomously semantifying Wikipedia*. In Proceedings of the 16th ACM Conference on Information and Knowledge Management, pages 41–50, New York, New York, USA, 2007. ACM Press. Available from: http://portal.acm.org/citation.cfm?doid=1321440.1321449, doi:10.1145/1321440.1321449.

[19] F. Wu and D. S. Weld. *Automatically refining the wikipedia infobox ontology*. In Proceedings of the 17th International Conference on World Wide Web, pages 635–644, New York, NY, USA, 2008. ACM. Available from: http://dl.acm.org/citation.cfm?id=1367583.

[20] P. Schmitz. *Inducing ontology from Flickr tags*. In Proceeding of the Collaborative Web Tagging Workshop at the World Wide Web Conference, volume 50, pages 3–6, New York, NY, USA, 2006. ACM.

Available from: http://wwwconference.org/proceedings/www2006/www.rawsugar.com/www2006/22.pdf.

[21] B. Markines, C. Cattuto, F. Menczer, D. Benz, A. Hotho, and G. Stumme. *Evaluating similarity measures for emergent semantics of social tagging*. In Proceedings of the 18th International Conference on World Wide Web, pages 641–650, New York, NY, USA, 2009. ACM.

[22] O. Ozdikis, F. Orhan, and F. Danismaz. *Ontology-based recommendation for points of interest retrieved from multiple data sources*. In Proceedings of the International Workshop on Semantic Web Information Management, New York, NY, USA, 2011. ACM.

[23] L. Cao, J. Luo, A. Gallagher, X. Jin, J. Han, and T. S. Huang. *A worldwide tourism recommendation system based on geotagged web photo*. In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing, pages 2274–2277, Dallas, TX, 2010. IEEE.

[24] M. Clements, P. Serdyukov, and A. de Vries. *Using Flickr geotags to predict user travel behaviour*. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 851–852, 2010.

[25] T. Stützle and H. H. Hoos. *Max-min ant system*. Future Generation Computer Systems, 16(8):889–914, 2000.

[26] S. Ahern, M. Naaman, and R. Nair. *World explorer: visualizing aggregate data from unstructured text in geo-referenced collections*. In Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries, pages 1–10, New York, NY, USA, 2007. ACM.

[27] R. Abbasi, S. Chernov, W. Nejdl, and R. Paiu. *Exploiting flickr tags and groups for finding landmark photos*. In M. Boughanem, C. Berrut, J. Mothe, and C. Soule-Dupuy, editors, Advances in Information Retrieval, pages 654–661. Springer Berlin / Heidelberg, 2009.

[28] A. Popescu and G. Grefenstette. *Gazetiki: automatic creation of a geographical gazetteer*. In Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries, pages 85–93, 2008.

[29] A. Bronselaer, S. Marcin, S. Zadrożny, and G. De Tré. *Dynamical order construction in data fusion*. Information Fusion, 27:1–18, 2016.

[30] P. Serdyukov, V. Murdock, and R. van Zwol. *Placing Flickr photos on a map*. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 484–491, 2009.

[31] Y. Cheng. *Mean shift, mode seeking, and clustering*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(8):790–799, 1995.

[32] O. Van Laere, S. Schockaert, and B. Dhoedt. *Finding locations of Flickr resources using language models and similarity search*. In Proceedings of the 1st ACM International Conference on Multimedia Retrieval, pages 48–55, New York, NY, USA, 2011. ACM.

[33] J. Gracia and E. Mena. *Multiontology semantic disambiguation in unstructured web contexts*. In Proceedings of the 2009 K-CAP Workshop on Collective Knowledge Capturing and Representation, pages 1–9, 2009.

[34] M. Naaman, J. Boase, C.-h. Lai, and N. Brunswick. *Is it really about me? Message content in social awareness streams*. In Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, pages 189–192, 2010.

[35] O. Van Laere, S. Schockaert, and B. Dhoedt. *Georeferencing Flickr resources based on textual meta-data*. Information Sciences, 238:52–74, 2013.

[36] M. Zhu. *Recall, precision and average precision*. Technical report, University of Waterloo, 2004.

[37] K. Crammer and Y. Singer. *On the algorithmic implementation of multiclass kernel-based vector machines*. Journal of Machine Learning Research, 2:265–292, 2002.

3

Categorizing Events using Spatio-Temporal and User Features from Flickr

In Chapter 2, we demonstrated that places could be discovered and characterized using social media. In Chapter 3, we now focus on the detection and categorization of events, again using social media. Based on Flickr, we show that it is possible to detect events not yet contained in existing databases. In Chapter 2, we only used textual information of the social media to discover semantic types. In Chapter 3, however, more advanced features are discussed to better estimate the semantic type of events. In particular, we introduce a method for discovering the semantic type of extracted events, focusing on how this type is influenced by the spatio-temporal grounding of the event, the profile of its attendees, and the semantic type of the venue and other entities which are associated with the event. We estimate the aforementioned characteristics from metadata associated with Flickr photos of the event and then use an ensemble learner to identify its most likely semantic type. Experimental results based on an event dataset from Upcoming.org and Last.fm show a marked improvement over bag-of-words based methods.

⋆ ⋆ ⋆

S. Van Canneyt, S. Schockaert, B. Dhoedt
Published in Information Sciences, 328, pages 76-96, 2016

3.1 Introduction

Several authors have shown that social media can successfully be used to detect events [1–6], even before they have been reported in traditional media [7]. However, it is difficult to evaluate queries such as 'In which countries did U2 perform during 2013?' against a set of events that have been detected in this way. Answering such queries requires access to a structured representation of events. The absence of such structured representations limits the applicability of current methods for event extraction from social media. In particular, it is of interest to learn structured representations of the kind that have traditionally been considered in template-based information extraction [8, 9]. The most relevant template for a given event is often based on the semantic type of that event. For instance, for a football match we want to encode the final score. In contrast, we want to know the magnitude and number of casualties of an earthquake. In this chapter, we study how the semantic type of events can be extracted from social media, as a first step towards automatically extending and creating structured event databases.

Evidence about the semantic type of an event can be obtained by analyzing social media documents, such as Flickr photos taken at the event, which we consider in this chapter, or tweets that have been sent about the event. In particular, similar to e.g. [1, 2, 5], we represent an event as a set of social media documents related to that event, together with its associated characteristics. A set of social media documents related to an event may for instance be automatically extracted from social media [1–3, 5] or may be extracted from existing event databases such as Upcoming.[1] Most initial work about discovering the semantic types of events only used textual information [10–12], which may lead to poor performance when the text is noisy (e.g. in some Twitter posts) or absent (e.g. in some Flickr photos). However, social media documents also contain metadata which provide an indication about the spatio-temporal and attendees features of an event. The hypothesis we consider in this chapter is that in many cases the event type can be discovered by looking at properties, such as timing, the type of venue or characteristics of attendees, which can be readily obtained from social media sources. For example, when an event occurs on a Saturday inside a sport complex and it has basketball players as main actors, it is very likely that this event is of type 'basketball game'.

Even though our methods can be applied more generally, we will restrict ourselves in this chapter to experiments on Flickr photos. In particular, the considered characteristics of an event are estimated using its associated Flickr photos, and these characteristics are then used to describe the event. To estimate the type of a given event, we use an ensemble of classifiers, one for each of the considered descriptors. Subsequently, we consider two use cases. First, these trained classifiers are used to analyze in detail to what extent our methodology is able to discover

[1] http://upcoming.org/

the semantic type of known events that have no associated semantic type. This is useful, for example, to improve existing event databases such as Upcoming, for which we found that about 10% had no known type. Second, the model is used to estimate the semantic type of events which have been automatically detected from Flickr, which could substantially increase the applicability of existing methods for automated event detection.

The remainder of this chapter is structured as follows. We start with a review of related work in Section 3.2. Next, in Section 3.3, we describe our methodology for classifying events based on their characteristics. Subsequently, Section 3.4 presents the experimental results. Finally, we discuss and conclude our work in Section 3.5 and Section 3.6.

3.2 Related Work

Early work on extracting structured data from text focused largely on news articles. The Message Understanding Conferences (MUC) were organized during the 1990s [9] to encourage the development of new and better methods to extract information from documents. The main task of these conferences was to automatically fill in a template with information about the event described in a given news article. For each event type considered, a template was constructed by the organizers with characteristics specific to it. For example, the template of the 'airplane crash' event type contained characteristics such as the place and the consequences of the event. The standard methodology to handle this task consisted of two major parts. First, the system extracted facts, i.e. entities and actions, from the text through local text analysis. Second, global text analysis was used to merge the discovered facts or to produce new facts through inference. The obtained knowledge was finally used to fill in the event templates. More details on this method are described in [8]. An interesting project related to event detection using news media is the GDELT project[2]. GDELT monitors the world's broadcast, print and web news in real-time. It identifies and connects people, locations, organizations, themes, emotions, quotes and news-oriented events which are stored in a structured event database. This information gives a global perspective on what is happening, its context, who is involved, and how the world is feeling about it. This data was for instance used to visualize the protests and unrest around the world on a map in real-time.

In the last few years, the focus has shifted somewhat from news articles to social media due to the latter's large data volume, the broad user base and its real-time aspect. However, social media documents tend to be noisy and are often very short compared to news articles, which has led to new challenges.

[2]http://gdeltproject.org/

There has been a lot of interest in detecting events and their associated documents using social media. In [3], for example, the authors analyzed the temporal and locational distributions of Flickr tags to detect bursty tags in a given time window, employing a wavelet transform to suppress noise. Afterwards, the tags were clustered into events such that each cluster consists of tags with similar geographical distribution patterns and with mostly the same associated photos. Finally, photos corresponding to each detected event were extracted by considering their related tags, time and location. EDCoW [6] used wavelet transformations to measure the bursty energy of each word used in Twitter posts. It then filtered words with low energy in a given time window t. Finally, the remaining words were clustered using modularity-based graph partitioning to detect events in t. Twevent [4] improved the approach of EDCoW by first splitting the incoming tweets in n-grams. An n-gram was then considered as an event segment in a given time window when the occurency of that n-gram was significantly higher than its expected occurency. The obtained event segments were finally clustered into events and ranked based on the importance of their event segments in Wikipedia.

Becker et al. [1] represented an event as a cluster of social media documents related to that event. To detect events, they clustered social media documents based on their textual, time and location similarity features. They used a classifier with these similarity scores as features to predict whether a pair of documents belongs to the same cluster. To train the classifier, known clusters of social media documents were used, which were constructed manually and by using the Upcoming database. When the probability that a document belongs to an existing cluster is larger than a threshold, a new cluster is generated for this document. Becker et al. [2] introduced an additional step which classifies the clusters corresponding to candidate events as 'event' or 'non-event' based on e.g. the burstiness of the most important words in the clusters and the coherence of the content of the social media documents in the cluster. Using the methodology described in [1, 2], the authors were able to detect events using Flickr and Twitter data. Their methodology was evaluated in [1] by comparing the detected photo clusters and the photo clusters collected from the Upcoming dataset. The approach from [5] added two steps to the approach of Becker et al. First, the methodology from [5] only used the k nearest clusters as candidate clusters of a given document. Second, they used a classifier to determine if a document belongs to an existing cluster or a new cluster, instead of using a threshold. This improved the approach of Becker et al. both in terms of scalability and accuracy. A work similar to [1] is proposed by Petkos et al. [13]. As in [1], the probability of whether a pair of Flickr photos belongs to the same cluster was determined using a classifier. In addition to textual, time and location similarities as input for this classifier, they also considered visual similarities. Given the probabilities of all document pairs, the photos were clustered using k-means clustering. Instead of combining the textual, temporal and location

similarity feature into one similarity metric as in [1, 5, 13], Li et al. [14] proposed a method that clusters two Flickr photos into one event if all three similarities are smaller than given thresholds. Subsequently, the obtained events are ranked based on the importance of the events at a given moment. To estimate the importance of an event, they consider the number of photos associated with the event, the number of users who created those photos, the area and period covered by all photos, and the time between the creation time of the photos and the given time. The aim of clustering Flickr photos into events was also considered in the yearly social event detection challenge at MediaEval [15], first organized in 2011. The challenge in 2011 and 2012 consisted in returning sets of photos from a given collection that represent social events satisfying some criteria (e.g. soccer matches in Barcelona). For the challenges held in 2013 and 2014, the task was to first cluster a photo collection into events, and then select events of interests (e.g. events of a particular type or matching a query). The most common approach was to cluster by location and time. Additionally, external sources such as the Google Geocoding API and Freebase were often used to better determine whether an event matches the given criteria. A more comprehensive summary of the challenges, pursued approaches and results can be found in [15].

Some initial research has been performed to discover the semantic type and other characteristics of an event using social media. The methodology introduced in [11], for instance, consists of classifying Flickr photos into different event types using their tags, description and title. For this purpose, a Naive Bayes classifier was trained on photos associated with events of known types. The authors also experimented with adding the creation date of the photos as a feature, but no clear improvement was observed. Yao et al. [16] detected events using the tagging history of the social bookmarking webservice Del.icio.us. The authors organized the detected events by mapping them to a hierarchy of semantic types, i.e. an automatically generated taxonomy extracted from the same tag space from which bursty events were detected [10]. The detected events were then mapped to an appropriate type at a suitable level based on the coverage of tags of the event in the subtree of the type. The task of classifying photos was also considered in the social event detection challenge at MediaEval 2013 [17]. In particular, photos had to be classified into ‘event’ and ‘non-event’ and into event types. Training data was collected using the Instagram API and retrieved photos were manually labeled into event types such as conferences, protests and sport events. Participants mainly used textual features such as the tags, title and description of the photos. Some participants enriched these textual features using e.g. a mapping to Wordnet or by extracting latent topics.

Some related work focused on visual features of photos to estimate the semantic type of the event shown in a photo. Li et al. [18], for instance, classified the

type of an event in a photo using texture and geometry features of the image. In addition, their proposed method provided semantic labels to the image scene and object components in the photo. All these photo properties were estimated simultaneously using a generative model. The approach described in [19] used a set of photos labeled with the type of the event shown in the photos to train a support vector machine. In addition, the satellite photos corresponding to the photo locations were obtained, and were used to train an AdaBoost classifier. The input features used for both classifiers were obtained using the color and texture properties of the photos. Given an image with an unknown event type, both classifiers were used to estimate the event type. Finally, these two estimations were fused using a meta-classifier to obtain a final estimation. The objective of the approach introduced by [20] was to detect the semantic type of an event described by a collection of photos. They first extracted color and texture features for each image associated to the event. The k-means algorithm was used to cluster the features resulting in 1000 visual words. Each visual word corresponds to a component of the event feature vector. The component value of the vector is set to the number of images associated with the event that contain the visual word related to that component. The obtained feature vector was finally normalized. Subsequently, the most discriminative compositional features were extracted and used to train an AdaBoost classifier. Their approach was tested on Flickr photos of frequently occurring events with distinctive visual characteristics such as a road trip, skiing and a wedding. In contrast to use the visual features of the images which occur in social media, we are focusing on how the metadata of social media documents can be used to detect the semantic type of events.

Benson et al. [21] introduced a structured graphical model which simultaneously analyzes individual tweets, clusters them according to an event, and induces a canonical value for each event characteristic. In the evaluation of their approach, they focused on concerts and considered the artist and venue as characteristics of these events. The artists and venues mentioned in tweets are extracted using a conditional random field approach together with an approach which matched words to a dataset of known artists and venue names. The approach described in [22] also used text analysis techniques on tweets to extract characteristics of events. They first extracted a ‘snapshot’ of tweets mentioning a given entity during a given time interval. The snapshot was then classified as ‘event’ or ‘non-event’ using different features such as the entity burstiness in Twitter during the given time interval. For snapshots related to events, the associated entities, actions performed by these entities and audience opinions about these events were extracted using regular expressions. The focus of the methodology described in [23] was to automatically determine when an event started and ended. In particular, tweets about an event were first classified as posted ‘before’, ‘during’ or ‘after’ the event. For this step, a classifier was trained using only textual features of the tweet such as the tense

of the verbs. These estimated labels were then used as input for a Hidden Markov model to estimate the temporal boundaries of the event. The authors of [7] developed a methodology which estimates in real time the location of earthquakes using Twitter data. For earthquakes, tweets were first collected using a keyword search to extract tweets which are potentionally related to real earthquakes. Examples of such keywords used in this paper are 'earthquake' and 'shaking'. Second, a classifier was used to estimate if these tweets are truly referring to an actual earthquake occurrence. Third, the locations of the tweets were estimated using their associated geo-coordinates or based on the home location of their user. Finally, a Twitter user is considered as a 'social sensor' which reports about an earthquake occurrence by tweeting about it. By regarding a tweet as a sensor value, the earthquake location estimation problem was then reduced to detecting an object and its location from sensor readings. To solve this problem, Kalman filtering and particle filtering were considered.

The work most related to our approach is described by Ritter et al. [12]. The approach uses the estimated date, actor and action of an event to improve the prediction of its semantic type. In particular, the words corresponding to the actor and the action of the event are extracted from its associated tweets. In addition, they mapped temporal expressions in the tweets associated with the event to calendar dates, which are used to estimate the date of the event. The date and the words corresponding to the actor and the action are then used to cluster the events, where a kind of regularization constraint is imposed that makes it more likely that two events which occurred on the same date are assigned to the same cluster. The obtained clusters were finally manually labeled with event types. However, 53.5% of their detected events were allocated to a cluster containing events with incoherent types or with types which are not of general interest. Note that [12] considers a clustering problem in which they first cluster the events and then manually label these clusters with event types. In contrast, we consider a classification problem in which classifiers are trained to recognize events of a particular type. Furthermore, the approach does not consider other characteristics such as the event location and participants to further improve the event classifications.

In our previous work [24], we focused on how location features extracted from Flickr photos of an event can be used to estimate the semantic type of the event. In this chapter, we extend our previous work in the following way: First, we also investigate how the event type is influenced by the temporal grounding of the event, the profile of its attendees, and the semantic type of the entities which are associated with the event. Second, the evaluation of the proposed methodology has been extended thoroughly. In particular, a dataset from Last.fm is now used to examine how well the approach is able to discover fine-grained event types. Finally, we evaluate our method's ability to discover events of a given semantic type that are not mentioned in the Upcoming database.

3.3 Methodology

The objective of this chapter is to discover the semantic type of events based on characteristics of the event that can be derived from the metadata of Flickr photos. In particular, the metadata we consider are the creation time and date of the photos, the user who created the photo, and the geographic location of the photos, where available. In the following, we assume that a training set K is available, containing events with a known semantic type, together with a list of associated Flickr photos. Additionally, we consider a set U containing events whose semantic type our method will try to estimate. This set may contain known events with an unknown semantic type, or events which have been automatically extracted from social media and therefore have no associated type. Both cases will be considered in the evaluation section. As mentioned, an event is represented as a set of Flickr photos related to that event and an associated semantic type, which is similar to the representation used in e.g. [1, 2, 5]. The set of all events is called $E = K \cup U$, the set of photos that are associated with event $e \in E$ is denoted by D_e, and the set of event types associated with e is denoted by $T_e \subseteq T$ where T is the set of all considered event types. Note that an event may have more than one type, e.g. an event where a person gives a lecture about art can be classified as both 'education' and 'art'.

In Section 3.3.1, we explain how the social media documents related to an event can be used to estimate characteristics such as its actors, participants, time and location. These characteristics are used to describe the event. To estimate the type of a given event, we use an ensemble of classifiers, one for each of the considered descriptors. More details about the classification framework we used can be found in Section 3.3.2.

3.3.1 Descriptions of Events

In this section, we describe how several characteristics of an event are estimated using their associated Flickr photos, and how they are used to construct feature vectors for the event.

3.3.1.1 Bag-of-Words (baseline)

A baseline approach to describe the events is to use the textual content of the Flickr photos associated with them. The textual content associated with a photo consists of a set of tags, a title and a description. In previous work, the textual content of social media documents has already been used to classify events [11, 16]. In this 'bag-of-words' approach, a vector describing an event $e \in E$ is constructed, whose components are associated with a word that appears in dictionary W. This dictionary W is the set of all terms from the textual content of the photos associated with

the events in the training set K. For feature vector V_e^b of event e, the component $comp_w^b$ associated with word $w \in W$ is given by its number of occurrences in D_e:

$$comp_w^b = \sum_{d \in D_e} |d_w| \tag{3.1}$$

with $|d_w|$ the number of times photo $d \in D_e$ contains word w. We use the Euclidean norm to normalize these feature vectors. We also tested TF-IDF weighting, but as this yielded worse results, we do not consider it in the remainder of the chapter. The set of all non-zero bag-of-words feature vectors corresponding to the events in K is denoted by $V^b(K)$. Note that we do not consider vectors with all components equal to zero for training. A zero vector can for instance be obtained when there is no textual information available for the event. We denote a zero vector by $\mathbf{0}$.

3.3.1.2 Entities Associated with Events

The semantic type of the entities associated with an event can provide valuable information about the type of the event. For example, if the event is associated with a musician it is more likely to be a music event, and if it is located in a football stadium it is more likely to be a sport event. To determine the entities which are related to the events, we map mentions of names in the text associated with the event onto canonical entities registered in the YAGO2 knowledge base [25]. The text of an event consists of the tags, descriptions and titles of its associated photos. For this process, we use the AIDA entity detection framework [26]. Given a natural language text, it maps mentions of ambiguous names onto entities (e.g. persons or places) registered in the YAGO2 knowledge base. The similarity between a detected mention and its associated entity is given by a *mention-entity similarity score* between 0 and 1, denoted by s_m. In addition, it maps these entities to a semantic type from the YAGO2 taxonomy. For example, the mention 'La Tasca' in sentence 'Our team at La Tasca!' is mapped to YAGO2 entity La_Tasca of type 'restaurant' with similarity score s_m of 0.6. The confidence p_m that a detected mention m corresponds to the recognized entity is based on the mention-entity similarity s_m. This confidence score is conservative and we noted that several mappings with $s_m = 0$ were actually correct. For instance, the AIDA framework correctly maps the mention 'SSV Markranstadt' in sentence 'Budissa Bautzen - SSV Markranstadt' to football club SVV Markranstädt, but the associated s_m is set to zero. Therefore, we use the following, smoothed confidence score:

$$p_m = 0.9 \cdot s_m + 0.1 \tag{3.2}$$

Note that by smoothing s_m, we ensure that each detected ambiguous entity name receives a non-zero confidence score.

Based on this semantic information, a feature vector V_e^r for each event $e \in E$ is constructed. Each component of this vector is associated with a semantic type from S, the set of all semantic types of the entities associated with the events in training set K. Formally, for feature vector V_e^r of event e, the component $comp_s^r$ associated with semantic type $s \in S$ is given by the weighted number of mentions in the text of e that are linked to an entity of semantic type s:

$$comp_s^r = \sum_{m \in M_{e,s}} p_m \tag{3.3}$$

with $M_{e,s}$ the list of all mentions of names in the description of event e which are linked to an entity of semantic type s. Finally, we use the Euclidean norm to normalize these feature vectors. We write $V^r(K)$ for $\{V_e^r \mid e \in K, V_e^r \neq \mathbf{0}\}$.

3.3.1.3 Event Participants

Social media contain valuable information about the behaviour of users, which can be valuable for estimating their interests. For example, it has been shown that the social media behaviour of users is useful for recommending places of interest [27]. Taking inspiration from this, for an event e, we use the types of the events in $K \backslash \{e\}$ that have been visited by the participants of e as evidence about the type of e. For example, when a lot of the participants of an event visited comedy shows in the past, it is more likely that this event is also a comedy show. We make the assumption that the creators of Flickr photos from an event are the participants of that event; we denote this set of users by U_e. Formally, the component $comp_t^u$ associated with event type $t \in T$ of vector V_e^u is given by the number of events with type t that have been visited by the participants of e:

$$comp_t^u = \sum_{u \in U_e} |\{e' \mid e' \in K_{t,u}, e' \neq e\}| \tag{3.4}$$

with $K_{t,u} \subseteq K$ the events of type t which have been visited by user u. Finally, we use the Euclidean norm to normalize these feature vectors. We write $V^u(K)$ for $\{V_e^u \mid e \in K, V_e^u \neq \mathbf{0}\}$.

3.3.1.4 Event Time and Date

The type of an event may be correlated with its time and date. For instance, an event occuring on a Saturday night is more likely to be a music related event than a family event. Time and date information extracted from social media data has already been successfully used for point of interest recommendation [28]. We consider a feature vector based on the creation time and date of the Flickr photos in D_e. The vector V_e^i contains one component for each hour of the day, day of the

week, week of the month and month of the year. The component $comp_p^i$ is given by the number of photos $d \in D_e$ that have been created during time period p:

$$comp_p^i = |D_{e,p}| \tag{3.5}$$

with $D_{e,p}$ the photos in D_e which have been taken during time period p. Finally, we use the Euclidean norm to normalize these feature vectors. We write $V^i(K)$ for $\{V_e^i \mid e \in K, V_e^i \neq \mathbf{0}\}$.

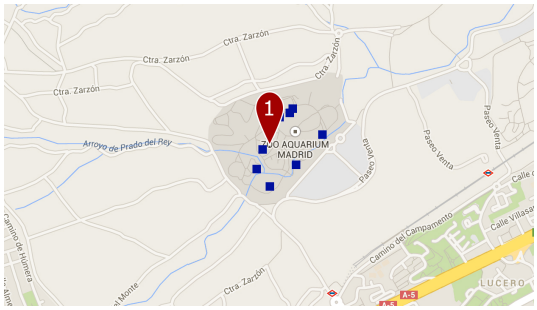
3.3.1.5 Event Location

If we know the type of some events which have taken place near the location of the considered event e, then this might be used as further evidence about the type of e. For example, when a lot of music events were organized nearby e, it is more likely that e is also a music event (e.g. because the location of e corresponds to a concert hall). Furthermore, the photos taken nearby the event may contain words which relate to the place type of the venue of the event, the types of the events organized in the past at that place, etc. This information can then be used to discover the semantic type of the event. We first describe how the locations of the events were estimated using their associated Flickr photos. Second, we give formal descriptions of event feature vectors based on their locations.

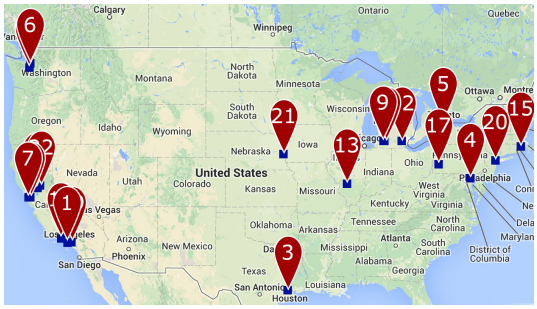
Estimating Event Locations To estimate the locations of an event $e \in E$, we use the geographic coordinates of the photos in D_e, where available; we denote this set of coordinates by O_e. We consider three approaches to estimate the location of a given event e from the set O_e. When O_e is empty, we consider the location of the event as unknown.

The first approach considers the geometric median of the coordinates in O_e as the location of the event e, denoted by $L_e = \{l\}$. In this approach, we assume that an event has only one location. Therefore, the weight $w(l)$ of the location $l \in L_e$ is set to 1. For instance, the Madrid Flickr meet was held on July 10, 2008 at the zoo of Madrid (Upcoming id 865742). Figure 3.1(a) shows the photos and estimated location of this event. The dots indicate the geographic coordinates of its associated photos in D_e. The marker indicate the estimated locations of the event, sorted by their weight. This approach is called 'median location'.

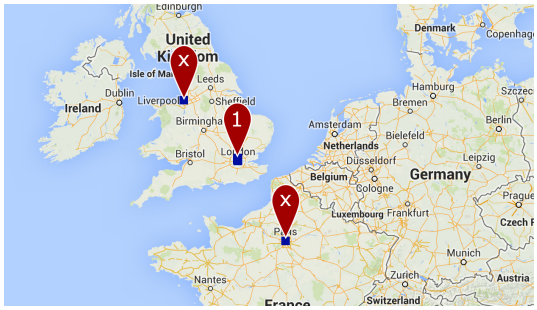
However, photos which are associated with an event may have been taken at different locations. For instance, 'the day of the donut' (Upcoming id 472136) which was held on April 16, 2008 took place at different locations, of which 20 are shown in Figure 3.1(b). On this day, people came together at different restaurants, pubs, bakeries and shops to share and eat donuts. To estimate different locations for the same event, we apply meanshift clustering [29] to the coordinates in O_e. The mean shift $m_b(o)$ of coordinate $o \in O_e$ is given by the difference of coordinate



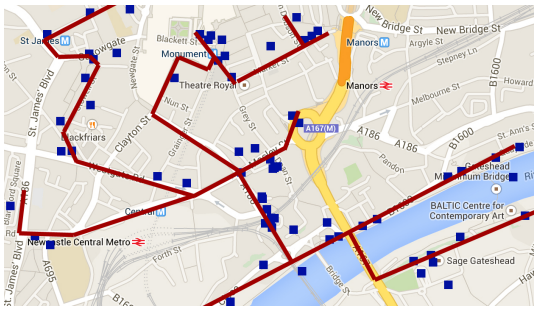
(a) 'median location'



(b) 'meanshift all'



(c) 'meanshift top'



(d) trajectory

Figure 3.1: Estimated event locations using different approaches. The dots indicate the geographic coordinates of the photos associated with the event. The markers and lines indicate the estimated locations, sorted by importance.

o and the weighted mean of the coordinates in O_e nearby o :

$$m_b(o) = \frac{\sum_{dist(o,o') \leq 2 \cdot b} G_b(o, o') \cdot o'}{\sum_{dist(o,o') \leq 2 \cdot b} G_b(o, o')} - o \tag{3.6}$$

with b the bandwidth parameter which is set to 2.5, $dist(o, o')$ the geodesic distance in kilometers between coordinate o and o', and $G_b(o, o')$ the kernel function which determines the weight associated with coordinate o' depending on its distance to o. We use a Gaussian kernel for a smooth density estimation:

$$G_b(o, o') = e^{-\frac{dist(o,o')^2}{2 \cdot b^2}} \tag{3.7}$$

The mean shift procedure then computes a sequence starting from all initial coordinates $o_1 \in O_e$ where

$$o_{i+1} = o_i + m_b(o_i) \tag{3.8}$$

which converges to a location that corresponds to a local maximum of the underlying distribution as $m_b(o_i)$ approaches zero. In this second approach, the coordinates of the center of all the clusters are considered as the locations of event e. We denote the set of these locations by $L_e = \{l_1, l_2 \ldots l_k\}$. The weight $w(l_i)$ of location $l_i \in L_e$ is taken as the percentage of coordinates from O_e that are clustered to location l_i. This approach is called 'meanshift all'.

In the third approach, called 'meanshift top', we assume that an event only takes place at one location and that photos which were taken far from this location are noise. For example, the Yahoo! BBC Hackday 2007 event (Upcoming id 173371) was held at London (see Figure 3.1(c)). 33 out of the 35 associated photos are indeed taken at the venue of the event (number 1). However, some participants took photos of event items at their home location. Thus, for this event, the estimated location with most associated photos is the real venue of the event. Therefore, in this approach, the coordinates of the center of the cluster containing most coordinates from O_e is considered as the location of the event e. This location is denoted by $L_e = \{l_1\}$ and the weight $w(l_1)$ of $l_1 \in L_e$ is set to 1.

Finally, note that we assume in this chapter that events are held at one or more points of interest. However, some events in the Upcoming and Last.fm database are not held at a fixed point of interest. Figure 3.1(d), for instance, shows the locations where the photos of the UK Flickr Meet were taken (Upcoming id 1827864). A group of photography enthusiasts took a walk in the Tyne and Wear county of England, and took photos at different locations during that walk. In this case, the location of the event takes the form of a trajectory, rather than a point or a fixed set of (disjoint) points. An approach which estimates the location of an event as a trajectory may be considered in future work.

Nearest Events The nearest events feature vector is based on the types of the events which have taken place nearby the location of the considered event e. Formally, for feature vector V_e^n of event e, the component $comp_t^n$ associated with event type $t \in T$ is given by the Gaussian-weighted number of nearby events of type t:

$$comp_t^n = \sum_{l \in L_e} \sum_{\substack{l' \in L_t \setminus L_e \\ dist(l,l') \leq 2 \cdot \sigma}} w(l) \cdot w(l') \cdot e^{-\frac{dist(l,l')^2}{2 \cdot \sigma^2}} \tag{3.9}$$

with $\sigma > 0$ determining the geographic scale, $dist(l, l')$ the geodesic distance in kilometers between location l and l', and L_t the locations from L which are associated to an event of type t. Set L_e contains the locations of event e and are obtained using the 'median location', 'meanshift top' or 'meanshift all' approach described above. Instead of using a Gaussian weighting, we also consider the following alternative, in which the k nearest events are considered for a fixed k, each being weighted based on their distance to the event:

$$comp_t^n = \sum_{l \in L_e} \sum_{l' \in N_{k,l,t}} w(l) \cdot w(l') \cdot \frac{1}{1 + dist(l, l')} \tag{3.10}$$

with $dist(l, l')$ the geodesic distance in kilometers between location l and l'. The set of all detected locations associated with events in the training set K is denoted by L. The set $N_{k,l}$ containing the k locations from $L \setminus L_e$ which are closest to l, and $N_{k,l,t}$ contains the locations from $N_{k,l}$ which are associated to an event of type t. Finally, we use the Euclidean norm to normalize these feature vectors. We write $V^n(K)$ for $\{V_e^n \mid e \in K, V_e^n \neq \mathbf{0}\}$.

Nearest Documents This type of feature vector is inspired by the approach described in [30], which uses the tags of Flickr photos taken nearby a place to discover its semantic type. Our assumption is that the textual content of all Flickr photos taken in the vicinity of an event may provide evidence about its type. In contrast to the photos in D_e, there is no guarantee that these nearby photos are associated with the event itself. For instance, the photos may even have been created years before the event took place. However, these nearby photos may contain words which relate to the place type of the venue of the event, the types of the events organized in the past at that place, etc. This information can then be used to discover the semantic type of the event.

We consider F as a large set of Flickr photos. Using the textual content of the photos in F which have been created nearby the location of the events, we describe an event e as a feature vector V_e^f. Similar as for the 'nearest events' approach, we consider the 'median location', 'meanshift top' and 'meanshift all' approaches to estimate the location of the event. Each component of this vector is associated with a term from the dictionary W^f, containing all tags of the photos which have

been taken nearby events in the training set. In the first representation, component $comp_w^f$ associated with term $w \in W^f$ is given by the Gaussian-weighted number of times a nearby photo contains w:

$$comp_w^f = \sum_{l \in L_e} \sum_{\substack{d \in F \\ dist(l,d) \leq 2 \cdot \sigma'}} w(l) \cdot |d_w| \cdot e^{-\frac{dist(l,d)^2}{2 \cdot \sigma'^2}} \tag{3.11}$$

with $dist(l, d)$ the geodesic distance in kilometers between location l and the coordinates of the photo $d \in F$, and $|d_w|$ the number of times photo $d \in F$ contains term w. For the second representation, the component $comp_w^f$ is given by:

$$comp_w^f = \sum_{l \in L_e} \sum_{d \in N'_{k',l}} w(l) \cdot |d_w| \cdot \frac{1}{1 + dist(l, d)} \tag{3.12}$$

with set $N'_{k',l}$ containing the k' photos from F which are closest to l. Finally, we use the Euclidean norm to normalize these feature vectors. We write $V^f(K)$ for $\{V_e^f \mid e \in K, V_e^f \neq \mathbf{0}\}$.

3.3.2 Classification Framework

For each type of feature vector described above, we learn a separate classifier. Each type of feature vector is used to classify the events in U. The output of these classifiers is then combined to estimate the semantic types of the events in U. To achieve this, we use a method which is based on the stacking framework introduced by Wolpert [31] and Ting and Witten [32], which we describe in detail in the following paragraphs. How an event with an unknown type is classified using this method is visualized in Figure 3.2.

Our classification framework consists of two phases. In the first phase, a set of learning algorithms L^b, L^r, L^u, L^i, L^n, L^f is selected, one for each described feature vector. A learning algorithm is a function which maps a set of training items (i.e. feature vectors) to a classifier. The optimal learning algorithm for each vector is selected using 5-fold cross-validation on the training set K (see Section 3.4.1.2). For each type of feature vector $x \in \{b, r, u, i, n, f\}$, a base classifier C^x is trained on $V^x(K)$ using learning algorithm L^x, i.e. $C^x = L^x(V^x(K))$. Using this classifier, we can classify each event e from set U using its associated feature vector V_e^x. We denote the resulting classification for event e by $pred^x(e)$, and the confidence that e belongs to type $t \in T$ is denoted by $conf^x(t|e)$. Note that it may be the case that vector $V_e^x = \mathbf{0}$, e.g. the nearest-documents vector V_e^f contains only zero values if the location of the event is unknown. For these zero vectors, the confidence $conf^x(t|e)$ is set to the same value for each considered type $t \in T$. In addition, $pred^x(e)$ is set to the type with most associated events in the training set.

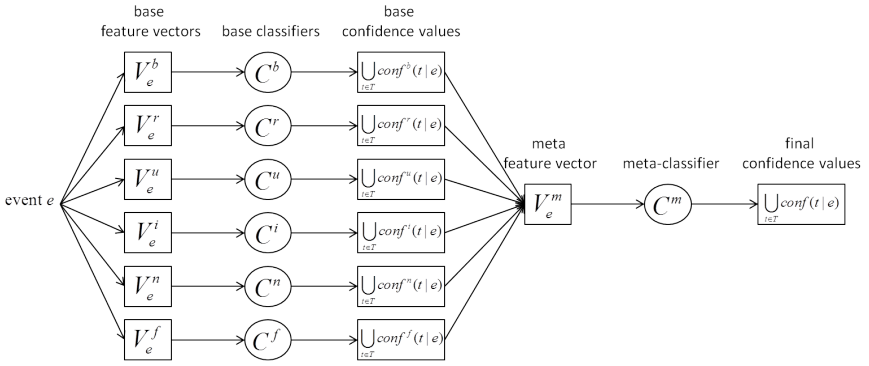


Figure 3.2: Schematic overview of our approach for classifying an event e. For each base feature vector type, the confidence that event e belongs to each considered type $t \in T$ is determined, denoted by conf$^x(t|e)$ for each $x \in \{b, r, u, i, n, f\}$.The meta feature vector V_e^m is then constructed by combining these confidence values. Finally, this meta feature vector is used to estimate the confidence conf$(t|e)$ that event e belongs to type $t \in T$. The predicated type pred(e) of event e is set to the type t with largest confidence value conf$(t|e)$.

In the second phase, a meta-classifier is learned that combines the outputs of the base classifiers. To generate a training set for learning the meta-classifier, a k-fold cross-validation procedure on the training set K was used, with k set to 5. We train each of the base classifiers using 80% of the training set K. We then use the learned classifiers to classify the remaining 20% of the training data. Repeating this process five times results in predictions *pred*$^x(e)$ and *conf*$^x(t|e)$ for each event e in K, each type of vector x and each event type $t \in T$. We also tested other values for k, yielding similar results, but for clarity we limit discussion to the case of $k = 5$. Similar as proposed in [32], the meta feature vector V_e^m is then constructed by combining the *conf*$^x(t|e)$ values for each $x \in \{b, r, u, i, n, f\}$ and $t \in T$. We can also use the *pred*$^x(e)$ values as described in [31] or both the *pred*$^x(e)$ and *conf*$^x(t|e)$ values, or just the combination of all the base features $\{V_e^x | x \in \{b, r, u, i, n, f\}\}$. Initial experiments have shown that these alternatives yield worse results, which is why we do not consider them in the remainder of the chapter. Finally, a classifier C^m is trained on vector set $V^m(K)$ using a learning algorithm L^m, i.e. $C^m = L^m(V^m(K))$. For each event $e \in U$, this classifier is then used to estimate its type *pred*(e) and the confidence that it belongs to semantic type $t \in T$, denoted by *conf*$(t|e)$.

3.4 Experimental Results and Discussion

In this section, we first use a dataset collected from Upcoming to examine the performance of each considered feature. In the Upcoming dataset, high level event types are considered such as 'sport', 'music' and 'conferences'. Subsequently, a dataset from Last.fm is used to examine how the proposed methodology performs for more fine-grained event types (i.e. subtypes of music events). Finally, we evaluate our method's ability to discover events of a given semantic type that are not mentioned in the Upcoming database.

3.4.1 Assigning General Types to Known Events

The first part of this section explains how the ground truth data is collected using the Upcoming event database. Second, we describe how we determine the optimal learning algorithms and event representations using 5-fold cross-validation on training set K. Finally, we examine to what extent the proposed characteristics of events are helpful for discovering their type.

3.4.1.1 Data Acquisition

Similar to [1], in this section we use ground truth data from the Upcoming event database. This database contains information about a large set of events. For each event, it stores an ID, an event type and references to a set of Flickr photos associated with the event. In addition, these Flickr photos contain the ID of their associated Upcoming event as one of their tags. Using the Flickr API, we first collected all photos which are tagged with an event ID from the Upcoming database. In this way we obtained 373 494 Flickr photos which were taken between January 1, 2000 and April 30, 2013 and which are associated with 22 290 events. Note that one photo may be associated with more than one event, e.g. a photo may be associated with an event such as a conference and one of its subevents such as the social dinner. Second, we retrieved the semantic types of the collected events from the Upcoming database. The 2 670 events (12%) with an unknown semantic type were removed. Finally, events with the same set of associated documents were considered as duplicates and only one of these events was retained in our dataset. As a result of this process, we obtained 16 469 events with a known type and 347 320 Flickr photos which are associated with at least one of these events. We collected the tags, title, description, user, creation date and geographic location of the photos, where available. In particular, for 40% of the photos in our dataset, geographic coordinates were available, and that 35% of these events have at least one associated photo which contains geo-coordinates. The considered types and the number of examples of each type in our dataset can be found in Table 3.1. Note that the sum of the number of events per type (20 647) is larger than the

Table 3.1: Upcoming dataset: number of events per type.

event type	#events	event type	#events
Music	6401	Family	600
Social	4571	Comedy	544
Performing Arts	2412	Commercial	543
Education	1726	Media	540
Festivals	1149	Conferences	209
Community	886	Technology	171
Sports	767	Politics	128

Table 3.2: Optimal learning algorithms for each type of feature vector.

feature vector	learning algorithm
Bag-of-Words	L2-regularized L2-loss SVM (dual) [34]
Entities	L2-regularized L2-loss SVC (dual) [34]
Participants	L1-regularized logistic regression [34]
Time and Date	L2-regularized logistic regression (primal) [34]
Nearest Events (3.9)	L2-regularized logistic regression (primal) [34]
Nearest Events (3.10)	L2-regularized logistic regression (primal) [34]
Nearest Documents (3.11)	L2-regularized L1-loss SVC (dual) [34]
Nearest Documents (3.12)	L2-regularized L1-loss SVC (dual) [34]
Meta-Classifier	L2-regularized logistic regression (primal) [34]

total number of obtained events (16 469) because one event may have more than one type. Finally, the dataset of events has been split in two parts: 5/6th of the dataset was used as training data (called the training set, K) and 1/6th was used for testing (called the test set, X). This test set is used to examine to what extent the proposed methodology is able to discover the semantic type of known events. For a fair evaluation, we ensured that no Flickr photos were associated with both an event in the training set and an event in the test set.

We crawled an additional set of Flickr photos, called set F, using the Flickr crawling tool[3] developed by Van Laere et al. [33]. As stated in [33], about 70% of the georeferenced photos from the photo-sharing site Flickr can be collected using their tool. In particular, we crawled the tags, description, title, user, creation date and geographic coordinates of the photos that were taken between May 2011 and April 2014 which contain a geotag with street level precision (geotag accuracy of at least 15). The dataset thus obtained contains 60 235 552 geotagged photos. This dataset is used to calculate the 'Nearest Documents' features of the events.

Table 3.3: Number of events per number of locations found by the meanshift clustering approach.

#locations	#events	#locations	#events
0	10776	7	3
1	5441	9	2
2	190	10	1
3	30	11	1
4	9	14	1
5	6	17	1
6	7	23	1

3.4.1.2 Optimal Learning Algorithms

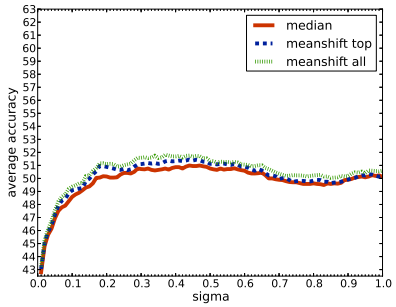
We used 5-fold cross-validation on the training set to find the learning algorithms that optimize the classification accuracy. In particular, the training dataset K was randomly partitioned in five equally sized subsets. The following process was repeated 5 times. Each time, one of the five subsets was used as validation (set K_v) and the remaining four sets were formed to form training set K_t. We trained a classifier using set K_t, which was then used to classify the events of K_v and to calculate its classification accuracy. The settings that optimized the average accuracy of the five folds were found by repeating this cross-validation approach for several learning algorithms. As candidate learning algorithms, we considered all methods implemented in WEKA [35] as well as the Support Vector Machine (SVM) implementations of LibLinear [34]. We used the standard configurations of the learning algorithms, both for WEKA and LibLinear.[4] The learning algorithms that were obtained from the training set can be found in Table 3.2.

3.4.1.3 Optimal Event Location Representation

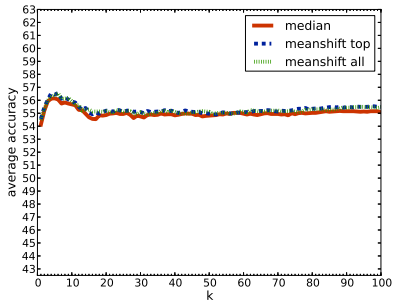
We also used the 5-fold cross-validation process to determine the optimal nearest-events and nearest-documents representations, as described in Section 3.3.1.5. This cross-validation process is performed on the events in training set K which contain at least one associated photo with geographic coordinates. As mentioned, we consider three approaches to estimate the location of an event, called 'median location', 'meanshift top' and 'meanshift all'. Table 3.3 shows a histogram of how many locations were found for the events in the collected Upcoming dataset. Additionally, two types of feature vector representations have been considered, one based on a Gaussian distribution (3.9) (3.11) and another based on the k nearest neighbours of the event (3.10) (3.12). The average accuracies for different param-

[3]https://github.com/ovlaere/flickr-crawler

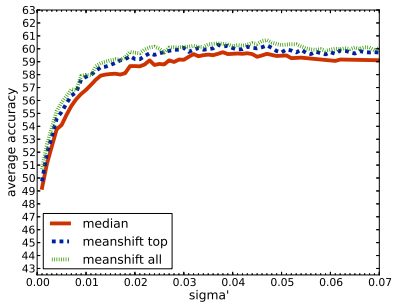
[4]We also did experiments with tuned parameters, yielding similar results, but for clarity we limit our discussion to the standard configurations.



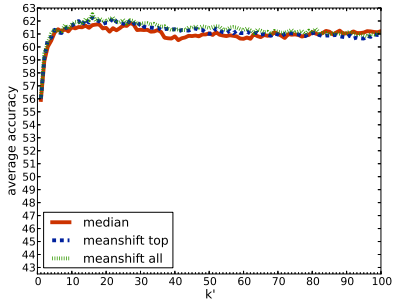
(a) Nearest Events (3.9)



(b) Nearest Events (3.10)



(c) Nearest Documents (3.11)



(d) Nearest Documents (3.12)

Figure 3.3: The average accuracy for different nearest-events and nearest-documents representations.

Table 3.4: Optimal parameters (par) and related average classification accuracy in percentage (ACA) for different nearest-events and nearest-documents representations using cross-validation on the training set.

	median location		**meanshift top**		**meanshift all**	
	par	**ACA**	**par**	**ACA**	**par**	**ACA**
Nearest Events (3.9, σ)	0.440	51.15	0.460	51.62	0.440	52.01
Nearest Events (3.10, k)	5	56.15	6	**56.55**	6	**56.55**
Nearest Documents (3.11, σ')	0.038	59.74	0.037	60.31	.047	60.67
Nearest Documents (3.12, k')	27	61.85	16	62.27	16	**62.53**

eter values of the six considered nearest-events representations can be found in Figure 3.3(a,b), and for the nearest-documents representations in Figure 3.3(c,d). The optimal parameter values and their associated average accuracy values can be found in Table 3.4.

We first discuss the performance of the different nearest-events representations. Figure 3.3(a) shows the average accuracies for different σ values when the Gaussian-weighted features are used (3.9). We can observe that the average accuracy increases when σ increases from 0.010 to 0.200, and stagnates when a larger σ value is used. As (3.9) only considers nearby events located at a maximum of $2 \cdot \sigma$ kilometers of the given event e, this means that events up to 400 meters of e tend to be relevant for determining its semantic type. The average accuracies when using different k values for the k nearest neighbours of an event used in (3.10) are shown in Figure 3.3(b). The average accuracy increases between $k = 1$ and $k = 6$, then decreases until $k = 12$ and then stagnates. To further compare the performance of each considered nearest-events representation, we look at their average classification accuracy when their optimal parameters are used (Table 3.4). For each location estimation approach, the representation from (3.10) significantly outperforms the representation from (3.9) (sign test, $p < 0.001$). The average accuracy for 'meanshift top' and 'meanshift all' is significantly higher than for 'median location' (sign test, $p < 0.001$) when (3.9) is used, but the difference in accuracy is not significant in the case of (3.10). Note that one reason why the use of a clustering method instead of the median location shows only limited improvement is because only for 1.5% of the events in the training data more than one cluster is found by the meanshift method (see Table 3.3). As 65% of the collected Upcoming events have no associated photos with geo-coordinates, we also experimented with automated methods for estimating the coordinates of Flickr photos in D_e for each considered event e based on their tags [33]. However, initial experiments did not yield better results. We find no significant difference when 'meanshift top' or 'meanshift all' are used for both (3.9) and (3.10) (sign test, $p > 0.05$). As the 'meanshift all' location estimation in combination with (3.10) gives the best average accuracy, we will use this representation in the rest of the chapter.

The average accuracies for the considered nearest documents representations are shown in Figure 3.3(c,d). Similar to the nearest-events vectors, the average classification accuracy first increases when the parameter σ' increases, after which it stagnates. However, the sigma value for which the average accuracy starts to stagnate for the nearest-documents representations ($\sigma' = 0.025$) is much smaller than for the nearest-events representations ($\sigma = 0.200$). One of the reasons is that the set of potential nearest documents (set F, $|F| = 56.7$ million) is much larger than the set of potential nearest events (set K, $|K|$ =13 725), which means that even with a small σ' value enough information can usually be obtained. Figure 3.3(d) shows the average accuracies when (3.12) is used. The average accuracy increases substantially between $k' = 1$ and $k' = 5$, and is optimal for $k' = 16$. The average classification accuracy for each considered nearest-documents representation is shown in the last two columns of Table 3.4. In each case, we assume that the optimal parameters are used. Similar to the nearest-event representations, the representation from (3.12) significantly outperforms the representation from (3.11) (sign test, $p < 0.001$). For both (3.12) and (3.11), 'meanshift all' and 'meanshift top' performs significantly better than 'median location'. However, there is no significant difference between 'median location' and 'meanshift top' (sign test, $p > 0.05$). As the 'meanshift all' location estimation in combination with (3.12) gives the best average classification accuracy, we will use this representation in the rest of the chapter.

3.4.1.4 Experimental Results

The task we consider in this section is to estimate the semantic type of the known events in test set X. Tables 3.5 and 3.6 summarize the result of our evaluation on test set X. The precision-recall curves for each semantic type are shown in Figure 3.4. The differences in accuracy are sometimes limited partly because the test set is imbalanced. Even a naive classifier returning the most occuring category ('Music') achieves 38% of accuracy, for instance. Therefore, the average precisions of the events from X which are ranked based on the confidence *conf*$(t|e)$ that they belong to type t are also considered.

When using all the proposed characteristics, we observe that the average precision is always higher than when the baseline is used and that the mean average precision significantly increases from 30% to 53% and the accuracy from 65% to 73% (sign test, $p < 0.001$). Furthermore, the average precision substantially improves for relatively rare event types (e.g. 'family', 'comedy', 'commercial', 'conferences' and 'technology'). For instance, the average precision for conferences increases from 4% to 52%. The types with a lot of associated events in the training set (see Table 3.1) have in general a higher classification performance than relatively rare event types. However, there are some exceptions, for instance the average precision for sport and comedy is much higher than for the types with a

Table 3.5: Average precision per event type and feature vector type.

event type	bag-of-words	entities	participants	time and date	location: nearest events	location: nearest docs	all characteristics
Music	86.00	62.74	73.22	48.39	49.57	62.40	**89.46**
Social	68.08	41.18	58.28	33.46	38.02	41.52	**77.72**
Performing Arts	42.55	20.10	48.70	18.56	23.07	24.87	**61.83**
Education	40.83	15.93	46.12	17.88	17.79	15.10	**58.19**
Festivals	23.17	8.35	21.52	9.99	12.21	6.96	**42.43**
Community	17.91	11.09	37.67	14.36	11.94	10.98	**41.16**
Sports	55.32	15.98	33.57	14.86	14.34	17.60	**76.58**
Family	10.59	11.01	33.27	4.78	13.21	9.18	**36.75**
Comedy	23.10	6.63	65.23	6.02	7.49	5.94	**70.77**
Commercial	14.26	3.33	25.34	3.74	13.41	4.31	**42.87**
Media	23.30	4.13	29.68	4.73	5.33	4.56	**46.10**
Conferences	4.34	3.18	41.61	3.42	3.03	3.00	**52.43**
Technology	9.23	2.50	36.52	3.25	6.35	1.82	**38.79**
Politics	2.45	1.05	5.80	1.79	5.43	1.12	**10.67**

similar number of associated events in the training set. The photos associated with sport events have a lot of indicative associated tags, leading to a high performance when using the bag-of-words features. The comedy shows and related photos are mostly uploaded by the organizers of these events as a form of advertisement. This boosts the performance of the participant features for the comedy event type. To better understand when the proposed approach is most useful, we partition the events in the test set based on the number of associated photos and the number of words in their associated text fields (Table 3.7).

For all considered sets of events, except for the set of events with more than 100 associated words, the accuracy of our approach is significantly higher than that of the baseline (sign test, $p < 0.001$). As expected, these results confirm the assumption that using additional features is mostly useful for events that have few associated photos or only photos for which a limited amount of text has been provided.

In the remainder of this section, we will discuss the performance of each event characteristic in more detail. Based on the classification accuracies in Table 3.6, we can conclude that the bag-of-words representation leads to the best classification accuracy if only one feature vector type is used. As described in Section 3.3.1.1, the bag-of-words feature vectors are based on the tags, titles and descriptions of the Flickr photos. In accordance with the findings from [11], we can conclude

Table 3.6: Classification accuracy and mean average precision (MAP) per feature vector type.

event characteristic	accuracy (%)	MAP (%)
Bag-of-Words (baseline)	65.20	30.08
Entities	47.16	14.80
Participants	64.43	39.75
Time and Date	41.84	13.23
Location: Nearest Events	46.68	15.80
Location: Nearest Documents	46.98	14.95
Bag-of-Words + Entities	65.78	40.52
Bag-of-Words + Participants	72.38	52.87
Bag-of-Words + Time and Date	66.80	43.84
Bag-of-Words + Nearest Events	66.95	43.70
Bag-of-Words + Nearest Documents	65.78	40.38
All Features	**72.78**	**53.27**

Table 3.7: Influence of the number of photos and the number of words on the improvement in classification accuracy.

#photos	#events	accuracy (%)		
		baseline	all	difference
1	1 113	58.58	72.15	13.57
2 - 5	564	73.58	78.19	4.61
6 - 20	531	68.17	70.81	2.64
> 20	536	67.16	69.59	2.43

#words	#events	accuracy (%)		
		baseline	all	difference
0-1	282	35.46	64.54	29.08
2-5	316	56.96	74.68	17.72
6-10	204	67.65	75.00	7.35
11-35	510	69.41	74.12	4.71
36-100	470	71.70	73.83	2.13
> 100	962	70.69	72.45	1.76

that the Flickr tags provide the best individual classification accuracy (61.55%), in comparison to only using the titles (53.83%) or only using the descriptions (52.44%). In contrast to the findings from [11], however, combining all types of textual information outperforms using only the tags (sign test, $p < 0.001$). For instance, the photos associated with the national robotic week event in our test set (Upcoming id 7965146) has no tags. However, an associated photo contains the description '@ National Robotics Week at Stanford Law School' which leads the classifier to correctly derive that this event is of type 'education'.

We observe no significant improvement of the baseline when combining the entities representation and the bag-of-words representation (sign test, $p > 0.05$).

This seems related to the fact that the entity features are extracted from the text fields. In particular, the entities vector can only perform well when enough textual information is available, in which case the bag-of-words representation is also likely to perform well. Additionally, we use the AIDA framework to determine the types of the entities which are mentioned in the text of the events. The accuracy of this framework is about 80% [26] which will lead to the introduction of some noise in the feature representations. For example, AIDA maps the word 'SFAC' in text 'Image courtesy of the Artist and SFAC Galleries' incorrectly to YAGO2 entity 'San Francisco Italian Athletic Club' of type 'athletic club'. Therefore, the classifier considers the event with this associated text (Upcoming id 10906905) as a sport event instead of an event of type 'performing arts'. Had the words 'SFAC Galleries' been correctly mapped to 'San Francisco Arts Commission Gallery' the discovered event type would have been correct.

The characteristic with the best individual MAP score is the 'event participants', which by itself already yields a MAP score which is higher than the baseline. It is particularly interesting to note that the features based on event participants yield a higher average precision on event types for which little training data is available, such as 'family', 'comedy', 'conferences', 'technology' and 'politics'. For instance, the photos associated to conferences hardly contain informative words leading to poor performance of the baseline. On the other hand, there is a strong correlation between the users who have taken a photo at a conference and the semantic types of the other events they have visited. For example, a lot of conference organizers take photos of all their events. Therefore, the average precision for conferences increases from 4% to 57% when adding the user information to the textual data. Combining 'event participants' with the bag-of-words representation improves the MAP score with 23 percentage points and significantly increases the classification accuracy (sign test, $p < 0.001$). However, it should be noted that the user overlap between the events in the training set K and test set X may be unusually large because both sets are obtained from the Upcoming database. For instance, a user who created a photo of an event in $E = K \cup X$ has on average created photos associated with 2.5 other events in E. One of the main reasons is that event organizers often add information and photos of all their events to Upcoming as a form of publicity. This leads to a high percentage of events with non-empty 'event participant' feature vectors. In particular, 86% of the events in X have at least one associated user who also created a photo of an event in the training set K. The added value of the user features is further examined in the last section of the evaluation where the test set consists of events which are automatically extracted using Flickr data and which are not included in existing databases such as Upcoming.

The features based on the event time provide the lowest MAP score and classification accuracy. However, jointly considering the bag-of-words and the time-

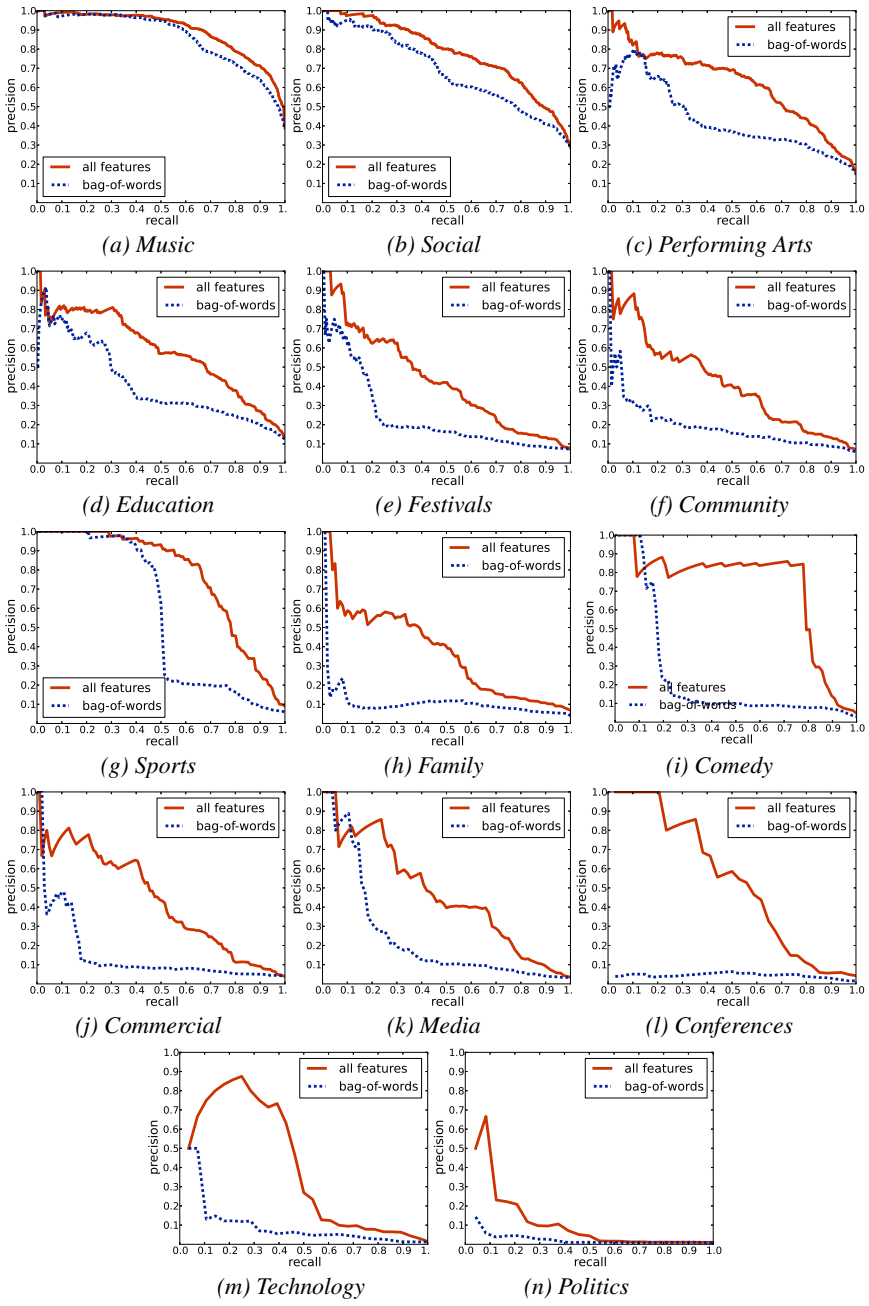


Figure 3.4: The precision-recall curves for the baseline (bag-of-words) and our approach (all features).

and-date vectors does outperform the bag-of-words representation in a statistically significant way (sign test, $p < 0.001$). For example, for an event in our test set (Upcoming id 108479) with associated words such as 'show' and 'fantastic', it is not clear whether it belongs to semantic type 'comedy', 'performing arts' (e.g. a theater show) or 'music'. However, when we know that the associated photos are taken between Friday 10 p.m. and Saturday 3 a.m., it is more likely that this is a music event.

The use of the known type of the nearest events improves the MAP score of the baseline with more then 10 percentage points and significantly improves the classification accuracy (sign test, $p < 0.001$). A similar observation can be made when the text of the nearest photos is used. For instance, the baseline approach was unable to discover that a skateboard race event in our test set (Upcoming id 318498) was of type 'sport' because its associated tags were not sufficiently informative. However, the photos taken close to the event contain words such as 'ferrari' and 'race' which may indicate that the event was held on a race track. Together with the information that all known nearby events are of type 'sport', the ensemble learner was able to discover the correct type. We also experimented with automated methods for estimating the coordinates of Flickr photos in D_e for each considered event E based on their tags with the aim of increasing the number of events whose location can be estimated [33]. However, initial experiments did not yield better results. We did not find a significant difference between the classification accuracy when the nearest-events or the nearest-documents vectors are used (sign test, $p > 0.05$). However, the classification accuracy is significantly better when the bag-of-words and both location features are used (67.27%) compared to only using the bag-of-words and nearest-events vectors or only using the bag-of-words and nearest-documents vectors (sign test, $p < 0.001$).

3.4.2 Assigning Fine-Grained Types to Known Events

The dataset from Upcoming used in the previous section contains events of general types such as 'festivals', 'politics' and 'social'. In this section, we want to examine if the proposed methodology also works for more fine-grained types such as 'rock music', 'folk music' and 'pop music'. Therefore, a dataset was collected using the Last.fm database containing events with an associated semantic type which is a subtype of 'music event'.

3.4.2.1 Data Acquisition

The Last.fm database contains information about a large set of music events. For each event, it stores an ID, the artists performing at the event, and references to a set of Flickr photos associated with the event. Similar to the Upcoming database, these Flickr photos contain the ID of their associated Last.fm event as one of their

tags. First, we collected all photos which are tagged with an event ID from the Last.fm database using the Flickr API. In particular, we obtained 2 271 172 Flickr photos which were taken between January 1, 2000 and August 31, 2014 and which are associated with 88 057 events.

In contrast to the Upcoming database, we can not extract the semantic types of the Last.fm events directly from their API. However, Last.fm artists have associated tags which are generated by the users. These tags often indicate the music genre to which the artists belong, such as 'rock', 'alternative' and 'electronic'. Therefore, we use the tags of the artists performing at an event as indication of the semantic type of that event. In particular, we used the Last.fm API to determine the most popular artist tags and manually discarded those tags that did not correspond to music genres. This process resulted in a total of 30 music genres, called set T. To determine the semantic types of the collected events based on the tags of their artists, we first extracted the artists performing at the events using the Last.fm API. For 87 265 events at least one artist was collected, resulting in a set of 106 779 unique artists. Second, the tags of these artists were collected using the Last.fm API, and the tags which are not element of T were discarded. The tags are weighted by Last.fm (between 0 and 100) based on the number of users which associated the tag to the artist. Third, we associated with each event a set of weighted tags, defined as the average tag weight of the artists associated with that event. An event is associated with a semantic type if the corresponding tag has a weight of at least 50. This resulted in a set of 62 095 events with an associated type. In addition, 1 434 569 Flickr photos associated with these events were obtained. Note that one photo can be associated with more than one event, e.g. a photo may be associated with an event such as a music festival and one of its subevents such as one of the performances at the festival, both of which may have a different ID in Last.fm. For 33% of these photos geographic coordinates are available, this means that 37% of the Last.fm events have at least one associated photo with geo-coordinates. The considered types and the number of events per type are shown in Table 3.8. Similar to the Upcoming dataset, the Last.fm dataset was split in two parts: 5/6ths of the dataset was used as training data (called training set K') and 1/6th was used for testing (called the test set, Y).

3.4.2.2 Experimental Results

The results of our evaluation on the Last.fm dataset is summarized in Tables 3.9 (a), 3.9 (b) and 3.10. The entities features were not used in this 'all characteristics' approach because we observed in Section 3.4.1 that they do not significantly improve the classification accuracy. Similar as for the Upcoming dataset, we observe that the proposed model substantially outperforms the bag-of-words representation in terms of average precision. Furthermore, the mean average precision increases from 27% to 52% and the classification accuracy increases in a statistically signifi-

Table 3.8: Last.fm dataset: number of events per type.

event type	#events	event type	#events
Rock	27416	Soul	1088
Electronic	12652	Rap	1059
Folk	10075	Country	718
Pop	7810	Techno	688
Punk	4766	Reggae	646
Metal	4746	House	581
Acoustic	3677	World	500
Hardcore	2926	90s	489
Ambient	2232	Latin	465
Jazz	1877	Classical	357
Hip Hop	1858	RnB	356
Dance	1805	Disco	228
Emo	1530	70s	217
Blues	1352	00s	214
80s	1302	60s	148

Table 3.9 (a): Average precision per event type and feature vector type.

event type	bag-of-words	participants	time and date	location: nearest events	location: nearest docs	all characteristics
Rock	84.79	53.10	47.23	46.53	45.98	**86.24**
Electronic	72.74	43.64	27.23	25.77	23.09	**80.42**
Folk	67.72	36.68	18.80	20.12	18.54	**78.65**
Pop	47.45	32.69	14.57	16.64	13.34	**65.31**
Punk	46.32	29.89	9.21	13.34	12.17	**67.48**
Metal	73.04	46.02	9.12	13.80	10.36	**84.67**
Acoustic	31.39	16.52	7.09	7.83	6.01	**60.12**
Hardcore	52.89	38.16	5.74	15.19	6.94	**73.60**
Ambient	36.92	16.65	4.31	5.82	4.46	**60.02**
Hip Hop	24.09	14.35	3.40	4.10	3.32	**60.63**
Dance	14.99	13.66	3.92	3.76	2.76	**50.55**
Jazz	37.47	18.02	4.33	6.16	5.55	**63.31**
Emo	15.02	15.88	3.29	3.85	2.84	**50.97**
Blues	12.32	13.12	3.04	4.99	3.16	**47.00**
80s	15.14	8.54	2.82	2.39	1.95	**53.29**

Table 3.9 (b): Average precision per event type and feature vector type.

event type	**bag-of-words**	**participants**	**time and date**	**location: nearest events**	**location: nearest docs**	**all characteristics**
Soul	12.94	8.15	2.02	2.81	1.80	**48.78**
Rap	15.35	12.93	1.92	2.66	1.91	**54.15**
Country	14.97	9.85	1.62	5.13	2.33	**51.86**
Reggae	29.68	5.33	1.06	1.45	1.34	**50.81**
Techno	8.91	25.49	6.12	6.19	1.21	**47.40**
90s	5.88	6.53	0.89	0.64	0.64	**32.43**
House	5.01	10.59	5.09	0.97	0.98	**22.93**
World	12.93	3.38	0.89	0.96	0.87	**46.26**
Latin	12.33	6.87	0.61	4.46	2.12	**37.71**
RnB	5.06	5.19	0.83	0.67	0.66	**31.11**
Classical	19.85	17.81	0.88	2.70	4.10	**54.85**
Disco	8.67	9.18	0.67	3.05	3.05	**20.35**
70s	1.77	4.46	0.59	0.48	0.47	**32.03**
00s	7.92	3.31	0.82	0.57	0.57	**15.76**
60s	5.36	0.35	0.44	0.61	0.62	**24.07**

Table 3.10: Classification accuracy and mean average precision (MAP) per feature vector type.

event characteristic	**accuracy (%)**	**MAP (%)**
Bag-of-Words (baseline)	77.35	26.63
Participants	53.40	17.54
Time and Date	44.25	6.28
Location: Nearest Events	45.20	7.45
Location: Nearest Documents	43.04	6.11
Bag-of-Words + Participants	77.67	51.43
Bag-of-Words + Time and Date	77.26	44.40
Bag-of-Words + Nearest Events	77.21	43.76
Bag-of-Words + Nearest Documents	77.36	40.14
All Characteristics	**77.69**	**51.76**

cant way (sign test, $p < 0.001$). Again we see the most dramatic increase for event types with the least amount of training data. For instance, the average precision for techno events increases from 9% to 47% by adding the proposed features to the bag-of-words representation.

Similar as for the Upcoming dataset, we can conclude that the bag-of-words features lead to the best individual classification accuracy. Generally, the more instances of an event type the training set K' contains, the better the associated average precision gets. However, there are a number of exceptions as can be seen in Tables 3.9 (a) and 3.9 (b). For instance, the average precision of 'classical' (20%) with 297 associated events in the training set is higher than the average precision of 'dance' (15%) with 1500 associated events in the training set. One of the underlying reasons is that a lot of classical music events have associated photos containing tags which indicate the type of the event such as 'johnwilliams', 'operafestival' and 'classical', whereas such informative tags tend to be rarer for dance events.

Based on the performance values shown in Table 3.10, we note that the use of the participants features leads to the second best individual classification performance. Combining the participants features with the bag-of-words representations improves the MAP score with 25 percentage points and significantly increase the classification accuracy (sign test, $p < 0.001$). This is similar to the observations when the general types of the Upcoming dataset are used.

In contrast to the Upcoming dataset results, we get no statistically significant improvement in classification accuracy when the 'time and date', 'nearest events' or 'nearest documents' features are added to the bag-of-words features (sign test, $p > 0.05$). The main reason for this is that these features are very similar for the considered subtypes of the 'music' type. For instance, a lot of different types of music events take place at very similar locations (e.g. a club) and time (e.g. in the evening). However, these features may still be useful for fine-grained event types as they slightly improve the mean average precision.

3.4.3 Assigning Types to Detected Events

So far we have focused on estimating the type of a given set of events. In practice, on the other hand, we will often be more interested in using social media to discover new events of a given type. To assess the usefulness of our method in such a context, in this section, we will analyze how it performs on the output of a standard method for event detection from social media.

3.4.3.1 Data Acquisition

We used the approach from [1] to obtain a set Z of photo clusters from Flickr, each assumed to represent an event. In particular, we clustered the Flickr pho-

tos in F based on their similarity in text, geographical location and creation time. We used a logistic regression model [35] with these similarity scores as features to predict whether a pair of Flickr photos should belong to the same cluster. To train the model, the Flickr photos associated to events in Upcoming training set K were used. When the probability that a photo $d \in F$ belongs to an existing cluster is smaller than threshold τ, a new cluster is generated for this photo. For our experiments, we have used a threshold of 0.5. Finally, the clusters containing photos associated with an Upcoming event were removed, as we want to specifically examine if the proposed methodology can be used to detect new events of a given type from social media. More details about this event detection method can be found in [1]. As a result of this process, 13 680 365 photo clusters (set Z) with an average of 4.4 associated photos were retrieved. These obtained photo clusters are considered as 'candidate events'.

3.4.3.2 Experimental Results

For a particular event type t, we rank the candidate events from Z based on the confidence *conf*$(t|e)$ that they are of type t. The highest ranked events can then be used to automatically extend or construct a structured database of events, possibly after a manual verification of their correctness. For evaluation purposes, we manually evaluated the top 100 discovered events for each type $t \in T$ and calculated its precision at position 10, 50 and 100. The results can be found in Table 3.11.

The performance of the baseline (bag-of-words) is highly dependent on the considered event type. For some types, the events in the training set have a lot of associated text which is indicative of the type, resulting in high P@n values. The baseline classifier has learned, for example, that words such as 'convention' and 'comiccon' may indicate the occurence of a commercial event (i.e. a fan convention), whereas music events are associated with words such as 'concert' and 'gig'. However, the training set hardly contains informative words for community events, technology events and conferences. One of the reasons of the poor performance of the detected community events is that only 36% of these events in the training set have associated photos with tags, in comparison to an average of 74% for all events in K.

In Section 3.4.1.4, we found that the participants features lead to the best individual average precision for Upcoming events. Again we find that incorporating the participants features substantially improves the results. One of the reasons for this improvement is that 1.5% of the events in Z are associated with a photo from a user who also has photos in the training set K. For the 'technology' type, for instance, the classifier prefers events which have associated photos created by users who also took photos at technology events in K. This leads to a better quality of the top ranked events. One of these Flickr users is a PhD student who is also involved in the OpenStreetMap project and who therefore often visits technological

Table 3.11: Precision at n (P@n) per event type for the automatically extracted events.

event type	bag-of-words			bag-of-words + participants			all characteristics		
	P@10	**P@50**	**P@100**	**P@10**	**P@50**	**P@100**	**P@10**	**P@50**	**P@100**
Music	100	98	98	100	98	99	100	100	100
Social	90	68	56	90	72	64	90	78	64
Performing Arts	100	100	76	100	100	100	100	100	100
Education	90	82	73	90	94	95	90	94	94
Festivals	100	100	100	100	100	100	100	100	100
Community	0	0	4	30	20	23	50	26	25
Sports	90	80	84	100	100	100	100	100	100
Family	80	64	41	100	90	85	100	90	83
Comedy	100	50	25	20	8	21	20	8	17
Commercial	90	94	89	100	100	100	100	100	94
Media	90	36	20	100	100	98	100	96	95
Conferences	0	0	7	30	8	6	30	8	7
Technology	0	4	5	90	30	15	90	30	23
Politics	50	62	53	100	64	80	100	90	85
average	70.00	59.86	52.21	82.14	70.29	70.43	83.57	72.86	70.50

events.

However, we found that some users who only post photos of comedy events in K also post photos of other types of events in Z. As a result of this, taking user interests into account actually leads to a worse performance for this event type. These events and related photos are mostly uploaded to the Upcoming database by organisers of comedy shows as a form of advertising, but these users also take photos at other events which are not in the Upcoming database. In particular, when both the participants and bag-of-words features are used, most top-ranked events of type comedy are from a single Flickr user. Accordingly, all photos taken by this user in K are from comedy events. However, this user also took many photos at events of other semantic types, leading to a decrease of the considered P@n values in comparison to the classifier which only uses bag-of-words features. When we remove the events visited by this user from set Z, however, we obtain a P@10 of 90%, a P@50 of 80%, and a P@100 of 83%.

The performance of the classifier when all the considered feature vectors are used can be found in the last three columns of Table 3.11. Similar as for the Last.fm dataset, the entities features were not used in the 'all characteristics' approach because their computation time is relatively high in comparison to the increase of the classification accuracy. Similar as for test set X, we observe that the best performance is obtained when all the considered features are combined. However, the improvement compared to the case when only the bag-of-words and participants features are used is limited.

3.5 Discussion

In future work, we would like to investigate how our proposed methodology can be applied to other social media platforms such as Twitter. The advantage of using Twitter posts instead of Flickr photos is that Twitter generates a lot more data. In particular, about 500 million tweets are sent per day[5] and an estimated 1.5% of these tweets are geotagged [36]. This gives a rough estimation of 7.5 million geotagged tweets per day. In contrast, around 1 million photos per day are uploaded to Flickr[6] of which about 3.3% contain geo-data[7], i.e. there are about 0.33 million new geotagged Flickr photos per day. However, using Twitter instead of Flickr to discover and categorize events would lead to some challenges. First, the content of Twitter posts are often far less informative than the text associated with Flickr photos [36]. Naaman et al. [37] analyzed the content of tweets, and determined that roughly 65% of tweets are personal status updates, presence maintenances or

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

[6] http://techcrunch.com/2014/02/10/flickr-at-10-1m-photos-shared-per-day-170-increase-since-making-1tb-free/

[7] http://code.flickr.net/2009/02/04/100000000-geotagged-photos-plus/

statements, such as 'Im happy', 'good morning twitter' and 'the sky is blue', respectively. As these types of tweets use the same vocabulary across a wide geography and time, they will not be useful to discover and categorize events. Other types of tweets, such as information sharing and opinions may not contain information about the event the user is attending. Moreover, in contrast to the text associated with Flickr photos, the content of tweets may be not relevant to the location of the user and the time the tweet was posted. For instance, a tweet may contain information about an event the user attended the previous day. However, Van Canneyt et al. [30] have shown that using the content of tweets posted nearby places of interest, in addition to the tags of the Flickr photos taken nearby these places, significantly improves the automatic categorization of these places. Therefore, it would be interesting to investigate how Twitter can be used in our methodology, as an addition to Flickr data, to further improve the categorization of events. The second challenge of using Twitter is to retrieve training data of tweets related to structured event datasets. For instance, Upcoming and Last.fm events are not associated with tweets. However, training data could be determined using Twitter hashtags. For example, Twubs contains a database of hashtags related to events[8]. These hashtags are categorized into event types such as 'concert', 'conference' and 'meetup'. Using the tweets containing these hashtags and the semantic type of the hashtags, our classifiers can be trained on Twitter data.

Instead of using the stacking framework described in Section 3.3.2, we could combine the features described in Section 3.3.1 into one vector to train a single classifier. Similar to the approach described in Section 3.4.1.2, we used 5-fold cross-validation on the training set to find the learning algorithms that optimize the classification accuracy. This resulted in the L2-regularized L2-loss support vector classification implementation of LibLinear. When using the Upcoming dataset, we obtained a classification accuracy 68.04% and a mean average precision of 45.40%. These results are significantly worse than our proposed stacking ensemble (see 'all features' in Table 3.6, sign test, $p < 0.001$). One of the reasons is that in the stacking ensemble, the learning algorithm is optimized for each considered feature type as described in Section 3.4.1.2. For example, for the small participants feature vectors with length equal to the number of considered event types, logistic regression works best. On the other hand, support vector machines give the best performance for the long bag-of-words feature vectors, with a length equal to the number of words in the dictionary. First optimizing the learning algorithm for each feature type and then combining the predictions using a meta-classifier gives thus a better performance than using one vector which combines all features.

[8] http://twubs.com/p/hashtag-directory/event/1064592_179

3.6 Conclusions

The problem of event detection from social media has been widely studied. To maximize the potential of event detection, however, there is a need to learn structured representations of events. As a first step towards automatically extending and creating structured event databases, we have proposed a methodology to discover the semantic type of events, using the relationship between an event type and other event characteristics. These characteristics have been estimated using the metadata of the Flickr photos associated with the event. We first used a dataset collected from Upcoming to examine the performance of each considered characteristic. In the Upcoming dataset, high level event types are considered such as 'sport', 'music' and 'conferences'. When using our methodology instead of the baseline which only uses the text of the Flickr photos related to an event to estimate its semantic type, the classification accuracy increased significantly. We observed that considering the type of the events visited in the past by the participants of the event led to the most substantial improvement over the baseline approach. The classification performance was further improved when the types of known events organized nearby the event, the textual content of the photos taken in the vicinity of the event, and the time and date of the event were considered. The classification accuracy did not change statistically significantly when the semantic types of the entities associated to the event were also considered. Examining the specific results, this seemed to be related to the fact that the entity features are extracted from the text fields. Second, a dataset from Last.fm was used to demonstrate that the proposed methodology also works for more fine-grained event types (viz. subtypes of music events). Finally, we showed how our methodology can be used to discover events of a given semantic type from Flickr that are not mentioned in existing event datasets.

Acknowledgment

Steven Van Canneyt is funded by a Ph.D. grant of the Agency for Innovation by Science and Technology in Flanders (IWT).

References

[1] H. Becker, M. Naaman, and L. Gravano. *Learning similarity metrics for event identification in social media*. In Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, pages 291–300, 2010.

[2] H. Becker, M. Naaman, and L. Gravano. *Beyond trending topics: Real-world event identification on Twitter*. In Proceedings of the 5th International AAAI Conference on Weblogs and Social Media, pages 438–441, 2011.

[3] L. Chen and A. Roy. *Event detection from Flickr data through wavelet-based spatial analysis*. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, pages 523–532, 2009.

[4] C. Li, A. Sun, and A. Datta. *Twevent: Segment-based event detection from tweets*. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pages 155–164, 2012.

[5] T. Reuter and P. Cimiano. *Event-based classification of social media streams*. In Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, page 22, 2012.

[6] J. Weng, Y. Yao, E. Leonardi, and F. Lee. *Event detection in Twitter*. In Proceedings of the 5th International AAAI Conference on Weblogs and Social Media, pages 401–408, 2011.

[7] T. Sakaki. *Earthquake shakes Twitter users: Real-time event detection by social sensors*. In Proceedings of the 19th International Conference on World Wide Web, pages 851–860, 2010.

[8] R. Grishman. *Information extraction: Techniques and challenges*. In Information Extraction A Multidisciplinary Approach to an Emerging Information Technology, pages 10–27. 1997.

[9] R. Grishman and B. Sundheim. *Message Understanding Conference-6: A brief history*. In Proceedings of the 16th Conference on Computational Linguistics, pages 466–471, 1996.

[10] B. Cui, J. Yao, G. Cong, and Y. Huang. *Evolutionary taxonomy construction from dynamic tag space*. World Wide Web, 15(5):581–602, 2012.

[11] C. Firan, M. Georgescu, and W. Nejdl. *Bringing order to your photos: Event-driven classification of Flickr images based on social knowledge*. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pages 189–198, 2010.

[12] A. Ritter, O. Etzioni, and S. Clark. *Open domain event extraction from Twitter*. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1104–1112, 2012.

[13] G. Petkos, S. Papadopoulos, and Y. Kompatsiaris. *Social event detection using multimodal clustering and integrating supervisory signals categories and subject descriptors*. In Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, pages 23–30, 2012.

[14] X. Li, H. Cai, Z. Huang, Y. Yang, and X. Zhou. *Spatio-temporal event modeling and ranking*. Web Information Systems Engineering, 8181:361–374, 2013.

[15] G. Petkos, S. Papadopoulos, V. Mezaris, R. Troncy, P. Cimiano, T. Reuter, and Y. Kompatsiaris. *Social event detection at MediaEval : a three-year retrospect of tasks and results*. In Proceedings of the ACM ICMR 2014 Workshop on Social Events in Web Multimedia, pages 1–8, 2014.

[16] J. Yao, B. Cui, Y. Huang, and Y. Zhou. *Bursty event detection from collaborative tags*. World Wide Web, 15(2):171–195, 2012.

[17] T. Reuter, S. Papadopoulos, G. Petkos, V. Mezaris, Y. Kampatsiaris, P. Cimiano, C. de Vries, and S. Geva. *Social event detection at MediaEval 2013: Challenges, datasets, and evaluation*. In Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop, pages 89–90, 2013.

[18] L. J. Li and L. Fei-Fei. *What, where and who? Classifying events by scene and object recognition*. In Proceedings of the IEEE International Conference on Computer Vision, pages 1–8, 2007.

[19] J. Luo, J. Yu, D. Joshi, and W. Hao. *Event recognition: Viewing the world with a third eye*. In Proceeding of the 16th ACM International Conference on Multimedia, pages 1071–1080, 2008.

[20] J. Yuan, J. Luo, and Y. Wu. *Mining compositional features from gps and visual cues for event recognition in photo collections*. IEEE Transactions on Multimedia, 12(7):705–716, 2010.

[21] E. Benson, A. Haghighi, and R. Barzilay. *Event discovery in social media feeds*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 389–398, 2011.

[22] A.-M. Popescu, M. Pennacchiotti, and D. Paranjpe. *Extracting events and event descriptions from Twitter*. In Proceedings of the 20th International Conference on World Wide Web, pages 105–106, 2011.

[23] A. Iyengar, T. Finin, and A. Joshi. *Content-based rediction of temporal boundaries for events in Twitter*. In Proceedings of the 3rd IEEE International Conference on Social Computing, pages 186–191. IEEE, 2011.

[24] S. Van Canneyt, S. Schockaert, and B. Dhoedt. *Estimating the semantic type of events using location features from Flickr*. In Proceedings of the 8th ACM SIGSPATIAL International Workshop on Geographic Information Retrieval, pages 57–64, 2014.

[25] J. Hoffart, F. Suchanek, K. Berberich, and G. Weikum. *YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia*. Artificial Intelligence, 194(1):28–61, 2013.

[26] J. Hoffart, M. A. Yosef, I. Bordino, H. Furstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. *Robust disambiguation of named entities in text*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 782–792, 2011.

[27] M. Clements, P. Serdyukov, and A. de Vries. *Using Flickr geotags to predict user travel behaviour*. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 851–852, 2010.

[28] S. Van Canneyt, S. Schockaert, O. Van Laere, and B. Dhoedt. *Time-dependent recommendation of tourist attractions using Flickr*. In Proceedings of the 23rd Benelux Conference on Artificial Intelligence, pages 255–262, 2011.

[29] Y. Cheng. *Mean shift, mode seeking, and clustering*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(8):790–799, 1995.

[30] S. Van Canneyt, S. Schockaert, and B. Dhoedt. *Discovering and characterizing places of interest using Flickr and Twitter*. International Journal on Semantic Web and Information Systems, 9(3):77–104, 2013.

[31] D. H. Wolpert. *Stacked generalization*. Neural Networks, 5(2):241–260, 1992.

[32] K. M. Ting and I. H. Witten. *Issues in stacked generalization*. Journal of Artificial Intelligence Research, 10:271–289, 1999.

[33] O. Van Laere, S. Schockaert, and B. Dhoedt. *Georeferencing Flickr resources based on textual meta-data*. Information Sciences, 238:52–74, 2013.

[34] S. Keerthi, S. Sundararajan, and K. Chang. *A sequential dual method for large scale multi-class linear SVMs*. In Proceedings of the 14th ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 408–416, 2008.

[35] M. Hall, H. National, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. *The WEKA data mining software : An update*. SIGKDD Explorations, 11(1), 2009.

[36] V. Murdock. *Your mileage may vary: on the limits of social media*. SIGSPATIAL Special, 3(2):62–66, 2011.

[37] M. Naaman. *Geographic information from georeferenced social media data*. SIGSPATIAL Special, 3(2):54–61, 2011.

4

Detecting newsworthy topics in Twitter

In the previous two chapters, we focused on the extraction of structured information from social media. In Chapter 5 and Chapter 6, we will focus on monitoring, analyzing, and making predictions in the context of online news content in social media. The current chapter makes a bridge between these two parts as it considers the extraction of newsworthy topics from social media. In particular, we discuss a methodology to solve the SNOW 2014 Data Challenge. The task was to mine Twitter streams to provide journalists with a set of headlines and complementary information, summarizing the most newsworthy topics for a number of given time intervals. We propose a 4-step approach to solve this. First, a classifier was trained to determine whether a Twitter user is likely to post tweets about newsworthy stories. Second, tweets posted by these users during the time interval of interest were clustered into topics. Third, we used a classifier to estimate the confidence that the obtained topics are newsworthy. Finally, for each obtained newsworthy topic, a descriptive headline was generated together with relevant keywords, tweets, and pictures. We ended at place 4 out of 11 participants, e.g. our method detected 34 out of 59 ground truth topics whereas the winner detected 39 of them.

⋆ ⋆ ⋆

S. Van Canneyt, M. Feys, S. Schockaert, T. Demeester, C. Develder, B. Dhoedt

Published in the Proceedings of the SNOW 2014 Data Challenge, pages 25-32, 2014

4.1 Introduction

Social media is an excellent source to detect events due to their large data volume, broad user base and real-time nature. Extensive work has shown that social media can successfully detect events [1–5], even before they are reported in traditional media [6, 7]. Therefore, social media may be an excellent source for news professionals to monitor the newsworthy topics that emerge from the crowd. However, we have to deal with noisy text fragments which are in addition often very short (e.g. Twitter posts).

In this chapter, we propose our methodology for a solution to the SNOW 2014 Data Challenge. The task of this challenge is to automatically mine social streams to provide journalists with a set of headlines and complementary information that summarize the newsworthy topics for a number of timeslots (time intervals) of interest. For an overview of the details of this challenge, we refer to [8]. Given a stream of tweets and a time interval of interest, we first determine the users who posted the tweets during that time interval which are most likely to post about newsworthy stories. This is accomplished by a classifier trained on profile features of the users. Second, the tweets posted by these users are clustered into topics based on the cosine similarity of their boosted *tf-idf* representations. This boosting is considered, on the one hand, to raise the importance of bursty words. On the other hand, proper nouns and verbs are boosted as they are essential keywords in most discussed topics (e.g. topic subjects and actions). Third, several features of the obtained topics are determined which are used to classify them as 'newsworthy' or 'not newsworthy'. Finally, for each detected newsworthy topic, a headline that summarizes the topic, accompanied by a set of relevant tweets, pictures and keywords are determined. The quality of the extracted newsworthy topics will be evaluated by a panel of news professionals selected by the challenge organizers. However, initial observations show the effectiveness of our methodology.

The remainder of this chapter is structured as follows. We start with a review of related work in Section 4.2. Next, in Section 4.3, we describe our methodology for discovering newsworthy topics. Subsequently, Section 4.4 presents the experimental results. Finally, we conclude our work in Section 4.5.

4.2 Related work

There has been a lot of interest in detecting events and trending topics in social media. This research can be divided in two types of approaches. In the first type, social media documents (e.g. tweets) are clustered. This is referred to as *document-pivot*. An event or topic is thus represented by a cluster of documents. The second line of work first selects the most important words, which are then clustered. In

this approach, referred to as *feature-pivot*, an event or topic is represented by a cluster of words.

Document-pivot approaches cluster social media documents by leveraging some similarity metric between them. TwitterStand [7], for instance, only uses the tweets of Twitter users who usually post news related tweets. They however did not use a classifier to determine these users, but manually constructed an initial set of these users. This set is updated based on the number of times the tweets of a user is associated with a newsworthy topic. Subsequently, an online clustering algorithm is used, which assigns the news related tweets to the closest cluster if the distance to this cluster is smaller than a given threshold. Otherwise, a new cluster with this tweet as the only member is created. The distance between a cluster and a tweet is based on the words in the tweet and the time at which the tweet was posted. The obtained clusters are considered as newsworthy topics. Finally, for each obtained topic, additional relevant tweets are searched using the hashtags present in the tweets of its corresponding cluster. Becker et al. [1] clustered social media documents based on their textual, time and location similarity features. They used a classifier with these similarity scores as features to predict whether a pair of documents belongs to the same cluster. To train the classifier, known clusters of social media documents were used which were constructed manually and by using the Upcoming database. When the probability that a document belongs to an existing cluster is smaller than a threshold, a new cluster is generated for this document. Becker et al. [9] introduced an additional step which classifies the clusters corresponding to candidate events as 'event' or 'non-event' based on e.g. the burstiness of the most important words in the clusters. Using the methodology described in [1, 9], the authors were able to detect events using Flickr and Twitter data.

Feature-pivot methods use statistical models to extract sets of words that are representative for the most important topics and events described in a corpus of documents. In [2], for example, the authors analyze the temporal and locational distributions of Flickr tag usage to detect bursty tags in a given time window, employing a wavelet transform to suppress noise. Afterwards, the tags are clustered into events such that each cluster consists of tags with similar locational distribution patterns and with similar associated photos. Finally, photos corresponding to each detected event are extracted. EDCoW [5] uses wavelet transformations to measure the bursty energy of each word used in Twitter posts, and then filters words with low energy in a given time window. Finally, the remaining words are clustered using modularity-based graph partitioning to detect events. Twevent [3] improved the approach of EDCoW by first splitting the incoming tweets in n-grams. An n-gram was then considered as an event segment in a given time window when the occurency of that n-gram was significantly higher than its expected occurency. The obtained event segments were finally clustered into events using Jarvis-Patrick clustering and ranked based on the importance of their event seg-

ments in Wikipedia. SocialSensor [10] selects the most bursty n-grams in a time window t based on their *df-idf*$_t$ score. This score is an adapted version of the *tf-idf* metric, penalizing n-grams whose popularity began in the past and which are still popular in the present. In addition, a boost factor is considered to raise the importance of proper nouns. The top ranked n-grams are then clustered using a hierarchical clustering algorithm and the co-occurences of the n-grams in the tweets. Finally, the clusters are ranked according to the highest *df-idf*$_t$ score of the n-grams contained by the cluster. They compared their approach with a standard feature-pivot, a standard document-pivot, and a Latent Dirichlet Allocation (LDA) approach. The document-pivot approach outperformed the feature-pivot and LDA approach. However, the quality of the top ranked topics was higher for their proposed approach than for the document-pivot approach. The authors also introduced two approaches which are based on Frequent Pattern Mining with similar or worse performance.

4.3 Methodology

For a stream of tweets (called test set, T^n), we want to determine the most newsworthy topics. In particular, for each time interval of interest $i \in I$, $m \geq 1$ newsworthy topics will be automatically extracted. To easily interpret the extracted topics, each topic will be in the form of a short headline that summarizes the topic, accompanied by a set of tweets, URLs of relevant pictures, and a set of keywords. To optimize the proposed methodology, we use a training set T^k of tweets with known newsworthy topics.

For a given stream of tweets T^n and a time interval $i \in I$, we first determine the users who posted the tweets during time interval i who are most likely to post about newsworthy stories. The tweets of these users are then clustered into topics. Thereafter, the obtained topics are ranked based on the confidence that they are newsworthy. Finally, for each detected newsworthy topic, the headline, most relevant tweets, tags and pictures are determined. The implementation of our methodology has been made publicly available to the research community.[1] In the rest of this section, we will explain each step in more detail.

4.3.1 News Publisher Detection

The first step of the proposed methodology is to estimate the likelihood that a Twitter user will post tweets about newsworthy topics. We indicate Twitter users who almost always publish newsworthy tweets as 'news publishers'. Examples are official twitter accounts of news papers, news programs or news websites. Given a set of tweets, the corresponding authors can then be ranked based on the probability

[1] https://github.com/svcanney/twittertopics

Table 4.1: Features used to detect Twitter accounts of news publishers.

Textual features	
username bag-of-words	term frequencies of the words in the user name
description bag-of-words	term frequencies of the words in the user description
Meta-data features	
#followers	number of followers
#following	number of following
$\frac{\#\text{follower}}{\#\text{following}+1}$	number of followers in comparison to the number of following
#tweets	number of tweets the user posted
#favorites	number of tweets the user favorited
#lists	number of lists the user follows
verified?	is the user account verified or not?
URL?	contains the user profile an URL or not?

that they are news publishers. Only tweets of the top ranked users will be used to detect newsworthy topics.

We first manually annotate 10,000 Twitter users as 'news publisher' or 'other'. We call this set of user U. Second, we use 5-fold cross-validation on the set U to find relevant user features and to train a classifier that optimizes the average precision of the users, which are sorted based on the likelihood that they are news publishers. As candidate classifiers, we consider all methods implemented in WEKA [11] as well as the Support Vector Machine (SVM) implementations of LibLinear [12]. The obtained features are shown in Table 4.1. The classifier which led to the largest average precision is a Bayesian belief network that uses a local K2 search algorithm [13].

Finally, user set U is used to train a Bayesian belief network which estimates the probability that the users which posted the tweets in test set T^n during time interval i are news publishers. The users with probability larger than α are considered as 'news publishers', noted as set P_i^n. Similarly, for each $i' \in I'$, the news publishers who posted tweets in the training set T^k during time i' are contained in the set $P_{i'}^k$. Set I' contains the considered time intervals corresponding to the training set T^k.

4.3.2 Topic Detection

In the second step of our methodology we cluster the tweets posted by users in P_i^n. Using only the tweets of news publishers, we significantly reduce the noisy tweets leading to 'junk'-topics. The clustering is performed using the DBSCAN [14] algorithm with parameters ϵ and minimum number of points required to form a cluster *minPts*.

As distance measure we use the cosine distance between the boosted *tf-idf* representations of the tweets. The boosted *tf-idf* value of a word w in tweet t

posted during time interval i is given by

$$\textit{tf-idf}_i^w = \textit{tf-idf}^w \cdot E\textit{-boost}^w \cdot T\textit{-boost}_i^w \tag{4.1}$$

Factor $\textit{tf-idf}^w$ is the standard term frequency-inverse document frequency for word w in tweet t. The document frequencies used for this $\textit{tf-idf}^w$ value are obtained from a set of tweets T^e which is unrelated to T^k and T^n. As T^k and T^n may contain tweets which are related to a specific event (see Section 4.4.1), we would have much lower $\textit{tf-idf}^w$ values for the event-specific words when these sets were used to calculate the document frequencies. Nonetheless, these words can be very relevant in the detected topics. By using an unrelated set of tweets, we are thus able to use more general event-independent document frequencies.

The first boosting factor $E\textit{-boost}^w$ is the boosting of proper nouns and verbs, similar as in [15], since they are typically more important than other words. The authors of [15] discovered that a boosting value of 1.5 for this kind of words and 1 for other words led to the best clustering results. Therefore, we use the same boosting values in this chapter. We use TweetNLP [16] to tokenize the tweets and determine the grammatical function of the words.

The second boosting factor $T\textit{-boost}_i^w$ is temporal boosting, in which we boost the words based on their relative document frequency in this time interval *i* versus the previous time intervals, thus the burstiness of the words. More concretely, we define

$$p_i^w = \frac{df_i^w}{N_i} \tag{4.2}$$

as the relative frequency of word w in time interval i, with df_i^w the document frequency of the word in the time interval and N_i the total number of tweets posted during i. We boost each term with the following temporal boosting factor:

$$T\textit{-boost}_i^w = \frac{p_i^w}{p_{0,i-1}^w} \tag{4.3}$$

with $p_{0,i-1}^w$ the exponential moving average of the relative frequencies of the word w for the time intervals 0 until $i-1$, using a smoothing factor λ.

Finally, we define the center of a cluster $c \in C_i^n$ as vector $\textit{center}_c$, obtained by averaging out all boosted *tf-idf* representations of the tweets in cluster c.

The detected topics from tweet test set T^n during time interval i are given by C_i^n. Similarly, the detected topics of training set T^k during interval $i' \in I'$ are given by $C_{i'}^k$. Additionally, we define set $C^k = \bigcup_{i'} C_{i'}^k$.

4.3.3 Topic Ranking

We explore different features to describe the detected clusters of C_i^n in order to identify newsworthy topics. A classifier trained on C^k is then used to detect newsworthy topics in the set of clusters C_i^n during interval i, indicated by the set S_i^n.

The training set of detected topics C^k is used to find the optimal features and classifier. Similar to the approach described in Section 4.3.1, we consider all methods implemented in WEKA [11] as well as the Support Vector Machine (SVM) implementations of LibLinear [12] as candidate classifiers. We first manually label the topics in training set C^k as 'newsworthy' or 'not newsworthy'. Second, C^k is partitioned into two disjoint subsets of topics, based on their time intervals: development set C^d comprises the first two thirds, the validation set C^v the last third. The topics of the development set C^d are used to train a classifier. This classifier is then used to estimate the likelihood that a topic $c \in C^v$ is newsworthy. For a particular time interval, the corresponding topics can then be ranked based on this likelihood. The objective is thus to optimize the mean average precision of these rankings. The obtained features are shown in Table 4.2. These features are divided in four categories. The first category takes the number of tweets in the clusters and their type into account. For instance, a cluster with just a few associated tweets may not be related to a newsworthy topic. The second category considers the features of the users. If the users who posted the tweets in the clusters are very likely to be news publishers (e.g. with probability higher than 0.9), the cluster probably corresponds to a newsworthy topic. The third category of features describes the topical coherence of the cluster, based on the hypothesis that newsworthy clusters tend to address a central topic, whereas noisy non-newsworthy topics cover more heterogeneous topics. The last category of features is used to exclude clusters corresponding to a topic that was already detected in a previous time interval, as we consider topics only as newsworthy when they occur for the first time. The classifier that leads to the highest mean average precision is Support Vector Machines (SVM) trained using sequential minimal optimization [17].

4.3.4 Topic Enrichment

The final step in our methodology is the topic enrichment. This step starts from each obtained newsworthy topic $s \in S_i^n$ and generates a headline, extracts keywords, a list of associated tweets and a list of pictures. These steps are mostly handled individually and are discussed in the following subsections.

4.3.4.1 Headline Creation

The headline of newsworthy topic s is constructed as a cleaned up version of the most representative tweet sentence in the set of tweets related to s. These tweet sentences are obtained by splitting each tweet in tweet sentences based on the presence of punctuation marks, and only retaining sentences containing at least one verb. To retrieve the most representative tweet sentence, we select the sentence with maximum cosine similarity between its boosted *tf-idf* representation and the vector associated with the topic center center_s. Subsequently, we apply a

Table 4.2: Features used to detect newsworthy topics.

Tweet features	
#tweets	number of tweets in the cluster
%original tweets	percentage of tweets in the cluster which are original tweets
%retweets	percentage of tweets in the cluster which are retweets
%replies	percentage of tweets in the cluster which are replies
%mentions	percentage of tweets in the cluster which contains user mentions
User features	
#users	number of users who posted the tweets in the cluster
%news publishers	percentage of users whose probability that they are news publishers is larger than x, with $x \in \{0.6, 0.7, 0.8, 0.9\}$
Topical coherence features	
%topic tweets (1)	percentage of tweets in the cluster containing the word of $center_c$ with highest *tf-idf*$_i^w$ value
%topic tweets (2)	percentage of tweets in the cluster containing the word of $center_c$ with second highest *tf-idf*$_i^w$ value
%topic tweets (3)	percentage of tweets in the cluster containing the word of $center_c$ with third highest *tf-idf*$_i^w$ value
Non duplicates features	
max similarity (1)	highest cosine similarity between the cluster center and the center of previous detected newsworthy clusters
max similarity (2)	second highest cosine similarity between the cluster center and the center of previous detected newsworthy clusters
max similarity (3)	third highest cosine similarity between the cluster center and the center of previous detected newsworthy clusters
max similarity (4)	fourth highest cosine similarity between the cluster center and the center of previous detected newsworthy clusters

set of rules to clean the obtained sentence: (1) Removing the mentions of users if they are part of a retweet mention. (2) Removing all URLS and emoticons. (3) Removing hashtags if they do not syntactically belong in the sentence. (4) Removing the '@' and '#'-symbols from the remaining hashtags and user mentions. (5) Removing parts of sentences inside parentheses. (6) Splitting the camel case words into different words. (7) End the headline with a punctuation mark.

4.3.4.2 Keywords Extraction

The keywords are chosen as the words present in the headline which are in the top 50% of the most important words associated to topic s. This importance of a word w is given by its *tf-idf*$_i^w$ value in *center*$_s$.

4.3.4.3 Representative Tweets Extraction

To extract a representative set of tweets, we first expand the list of tweets related to our topic by including tweets from users which are not indicated as 'news publishers'. In particular, we consider all tweets in T^n posted during i with a cosine similarity between their boosted *tf-idf* representation and the center of the topic which is higher than ω. Next, these tweets are ordered based on their relevance to the topic, denoted as *relevance*$_s^t$. The *relevance*$_s^t$ value of tweet t is defined as the cosine similarity between its boosted *tf-idf* representation and the center of the topic *center*$_s$, multiplied by the *user_factor*. This factor is $\upsilon \geq 1$ if the user who posted tweet t is indicated as a 'news publisher' and 1 otherwise. This ordered list of tweets related to topic s is denoted by T_s^n.

The tweets associated with a single topic should be sufficiently different from each other, therefore we discard tweets in T_s^n which are near-duplicates of tweets that are ranked higher in the list. To measure the similarity between the tweets in T_s^n, we use the cosine similarity between the non-boosted version of the *tf-idf* representations of the tweets. In particular, tweets are considered as 'near-duplicates' if their similarity is higher than φ. We discard boosting in this step, since the goal of boosting was to increase the impact of the topic-related words, thereby diminishing the impact of the other words in the tweet. However, the tweets in T_s^n are all related to the same topic, and all contain these topic-related words leading to a high cosine similarity of their boosted *tf-idf* representations, mainly caused by the presence of these topic-related words. As we want to obtain a coherent diverse set of tweets describing this topic, we want tweets that contain these topic-related words, but have a significant number of different non-topic-related words. If we had used the boosted *tf-idf*, the cosine similarity would almost only be impacted by the number of matching topic-related words. Finally, the top 5 tweets of this filtered T_s^n list are considered as representative for topic s.

4.3.4.4 List of pictures

In order to obtain a full list of pictures related to topic s, the tweets of T_s^n containing the same picture URL are grouped. Picture URLs are obtained by using the media entities associated with the tweets. The picture URLs are then sorted based on the sum of the *relevance*$_s^t$ values of the tweets containing the URL. Finally, the top 5 picture URLs are considered as relevant to topic s.

4.4 Evaluation

4.4.1 Data Acquisition and Settings

In order to evaluate our approach, we crawled the Twitter posts meta-data of the given Twitter id's related to the 2012 US elections event posted on Twitter between November 6, 2012 23:30 GMT and November 7, 2012 7:00 GMT (training set, T^k). The test set T^n contains tweets related to the Syria, Ukraine, terror and bitcoin-problems mentioned on Twitter between February 25, 2014 18:00 GMT and February 26, 2014 18:00 GMT. More details about the training and test set can be found in [8]. Additionally, an unrelated set tweets was obtained from the sample-stream of the Twitter Streaming API from November 29, 2013 until February 5, 2014 (external set, T^e). Non-English tweets were removed using LDIG[2]. To calculate the term frequencies in the obtained tweets, TweetNLP [16] was used to tokenize the tweets and to remove words related to punctuations, URLs, determiners, etc. The obtained words were then transformed to lower case and words with fewer than three characters were removed. Finally, the words were Porter stemmed [18]. As a result of this process, we obtained 928 791 tweets for training our methodology (training set, T^k), 973 658 tweets for evaluating our methodology (test set, T^n), and 77,741,801 tweets which have been used as external set T^e. User set U contains 10,000 Twitter users who are randomly selected from the users who posted the tweets in T^e. The time intervals of interest for the test set and training set are given by the challenge organizers and are respectively 15 minutes and 10 minutes long. We empirically set $\alpha = 0.04$, $\epsilon = 0.4$, *minPts* $= 3$, $\lambda = 0.5$, $\omega = 0.6$, $\upsilon = 1.5$ and $\varphi = 0.7$.

4.4.2 Experimental Results

4.4.2.1 News Publisher Detection

As described in Section 4.3.1, we use 5-fold cross-validation on the user set U to optimize and evaluate the methodology which detects news publisher. User set U contains 10 000 Twitter users who are manually annotated as 'news publisher'

[2]https://github.com/shuyo/ldig

or 'other'. As a result of this process, 1.64% of the users were labeled as 'news publisher'. The proposed methodology to rank users based on the likelihood that they are news publishers resulted in an average precision of 88.83%. In general, 99.41% of the users in U were correctly classified, which is significantly higher than the 98.36% accuracy when all users are classified as 'other' (sign test, $p < 0.001$).

4.4.2.2 Topic Ranking

The training set of detected topics C^k is used to optimize and evaluate the topic ranking methodology, as described in Section 4.3.3. Set C^k contains 116 manually annotated clusters, of which 54 are labeled as 'newsworthy'. For each considered time interval i' corresponding to clusters in validation set C^v, the clusters of C^v associated with i' are ranked based on the confidence that they are related to a newsworthy topic. The mean average precision of these rankings is 99.17%. In general, 82.05% of the clusters in the validation set were classified correctly.

4.4.2.3 Methodology Performance

Our methodology extracted 433 newsworthy topics from the test set, given by set $S^n = \bigcup_i S_i^n$. The newsworthy topics of time intervals February 26, 2014 09:15 until 10:15 GMT are shown in Table 4.3. These results show the effectiveness of our methodology to discover newsworthy topics in Twitter. As we only use tweets posted by 'news publishers' to detect topics, most of the discovered topics are indeed newsworthy. However, we observe that some duplicates are not removed mainly because users sometimes discuss one topic in different words, i.e. the high similarity of these topics can not be detected using cosine similarity on their associated words (e.g. topic 7 and 10). In addition, some non-newsworthy topics were incorrectly extracted due to users who are classified as 'news publisher' who post non-newsworthy content (e.g. topic 15). Finally, we observe that the obtained headlines are informative and constructed in a syntactically correct way.

The extensive summary of the newsworthy topics extracted during time interval February 26, 2014 09:15 can be found in Table 4.4. We observe that the representative tweets for a particular topic are sufficiently different from each other, i.e. no near-duplicates or retweets are given. Additionally, we note that the coherence of the tweets associated with topic 2 is higher than the coherence of the tweets associated with topic 1. In particular, topic 2 covers one clear topic (i.e. about Sofia monument's makeover), in contrast, topic 1 covers very similar, but different, topics (i.e. about a military vehicle in Kiev, Ukraine; and about a military vehicle in Sevastopol, Ukraine). The discovered pictures related to these newsworthy topics are shown in Figure 4.1. In total, 24% of the discovered newsworthy topics contains at least one related picture.

Table 4.3: Automatically extracted newsworthy topics from Twitter.

nr	time interval	headline
1	26-02-14 09:15	Jubilant protesters driving military vehicle from a Kiev Museum around Parliament building.
2	26-02-14 09:15	Sofia monument's latest makeover provokes protest from Russia.
3	26-02-14 09:30	I'm in Charge of Miltary Now, Ukraine's Interim President Says.
4	26-02-14 09:30	GDP grew 0.7% in Q4, unrevised from preliminary estimate.
5	26-02-14 09:30	Russia urges OSCE to condemn "neo-fascist" sentiment in west Ukraine.
6	26-02-14 09:30	The price of Bitcoin on MT. Gox is US $135.0000.
7	26-02-14 09:30	Bitcoin Has Made A Really Impressive Recovery.
8	26-02-14 09:45	Ukraine minister disbands Berkut riot police blamed for violence.
9	26-02-14 09:45	Japanese Authorities Probing Collapsed Bitcoin Exchange.
10	26-02-14 09:45	How bitcoin can turn it around.
11	26-02-14 09:45	Hezbollah says Israel bombed its positions near Syrian border 2 days ago, vows response.
12	26-02-14 10:00	Russia's deputy finance minister says no multilateral talks on financial aid to Ukraine are taking place.
13	26-02-14 10:00	Japan donates $14 mil. for Syria weapons disposal.
14	26-02-14 10:15	This is Beijing, less than three weeks apart.
15	26-02-14 10:15	Your spring tweet has appeared in our latest Edition mag.
16	26-02-14 10:15	Ukraine 'set to unveil new government'.

Table 4.4: Summaries of extracted newsworthy topics during time interval 26-02-14 09:15.

nr	tags	representative tweets
1	Jubilant,protesters,driving, vehicle,Museum,Parliament	Jubilant protesters driving military vehicle from a Kiev Museum around Parliament building #Kiev #Ukraine
		Another #Russia—n armored vehicles spotted in #Sevastopol in #Crimea. #Ukraine http://qn.quotidiano.net/esteri/2014...
2	Sofia,monument, makeover,provokes	Pro-Ukraine paint job - Sofia monument's latest makeover provokes protest from Russia http://bbc.in/1frf9UN
		Kijw w Sofii. RT: @BBCWorld Pro-Ukraine paint job in Sofia provokes protest from Russia http://bbc.in/1frf9UN
		Pro-#Ukraine paint job-Sofia monument's latest makeover provokes #protest from R http://bbc.in/1frf9UN via @BBCWorld

(a)

(b)

(c)

Figure 4.1: Pictures related to newsworthy topic number 1 (a,b) and number 2 (c).

4.4.2.4 Evaluation by the SNOW organizers

The newsworthy topics in S_n and their summaries are evaluated across a mixture of quantitative and qualitative dimensions by a panel of news professionals selected by the SNOW 2014 Data Challenge organizers [8]. The evaluation was conducted by three independent evaluators, located in different countries and organizations. The evaluation was done on a set of five timeslots (starting at 18:00, 22:00, 23:15 on 25/2, and on 1:00, 1:30 on 26/2), and was blind, i.e. the evaluators did not know which participant produced which topic they evaluated.

The organizers first constructed two ground truth sets of topics. The first, T_{ref}, comprised the 59 topics manually created by the organizers based on mainstream media stories in UK news outlets (BBC and NewsWhip) during the 24 hours of the Twitter crawl. The second, denoted as T_{ext}, was created in a pooled way based on the submissions of participants during the five selected timeslots. More specifically, the evaluators assessed all submitted topics during those five timeslots as being newsworthy or not. Topics that received at least two votes by evaluators were included in a list. After removing duplicates, a set of $|T_{ext}| = 70$ participant-pooled topics was defined. To evaluate the performance of the extracted topics by the participants, four evaluation criteria were used: precision-recall, readability, coherence/relevance and diversity. The recall score, R_{ref}, was calculated on the T_{ref} dataset, and the F1 score, F_{ext}, was calculated based on the T_{ext} dataset. Evaluators were instructed to assign a score between 1 and 5 indicating the readability of the headline, the coherence and the diversity of the tweets. The readability score Q, coherence score C and diversity score D were computed only on the basis of the newsworthy topics, and by averaging over the three evaluators. For each of the scores, the organizers first identified the maximum attained scores, and then normalized the scores of each participant with respect to the latter. In the end, the aggregate score for each participant was derived by the following equation:

$$AS = 0.25 \cdot R_{ref} \cdot F_{ext} + 0.25 \cdot Q + 0.25 \cdot C + 0.25 \cdot D \tag{4.4}$$

The results can be found in Table 4.5. Our team name was 'IBCN' and we ended at place 4 out of 11 participants. We, for instance, obtained 34 out of the 59 topics in ground truth set T_{ref}, whereas the winner Insight [19] detected 39 of them. More details about the evaluation process can be found in [8].

Insight [19] performed an approach similar to ours. Their first step was also aggressive tweet filtering. The authors filtered the tweets based on their content (number of hashtag...), whereas we filtered based on the user. They also clustered and ranked the obtained tweets to obtain newsworthy topics. However, they made some different choices to implement those steps. For example, they used hierarchical clustering instead of DBSCAN, and ranked the clusters based on the maximum boosted tf-idf term value in the cluster instead of using a classifier. Their performed topic enrichment step was very similar to ours. The Insight approach

Table 4.5: Overview of normalized scores and aggregate results.

Team	R_{ref}	F_{ext}	Q	C	D	AS	Rank
UKON	0.667	0.615	0.870	0.885	0.611	0.694	7
IBCN	**0.879**	**0.592**	**0.998**	**0.821**	**0.680**	**0.755**	**4**
ITI	0.485	0.661	0.911	0.942	0.666	0.710	5
math-dyn	0.955	0.640	0.931	0.988	0.608	0.785	3
Insight	1.000	1.000	0.961	1.000	0.608	0.892	1
FUB-TORV	0.591	0.119	0.848	0.962	0.576	0.614	10
PILOTS	0.364	0.227	1.000	0.972	0.553	0.652	9
RGU	0.909	0.686	0.955	0.849	0.942	0.842	2
UoGMIR	0.258	0.775	0.974	0.795	0.680	0.662	8
EURECOM	0.364	0.062	0.686	0.755	0.720	0.546	11
SNOWBITS	0.212	0.408	0.876	0.877	1.000	0.710	6
std. deviation	0.292	0.290	0.090	0.084	0.282	0.100	

thus did not use a significantly different approach, but their implementation and parameter settings resulted in better performance.

Note that there are some limitation to the evaluation approach of the SNOW organizers. As the evaluation process is conducted manually by evaluators, we are not able to optimize the used methods and parameters automatically. Therefore, it is hard to conclude whether the performance is due to the used approach or due to the used implementation and parameter settings. Furthermore, the evaluation was only conducted on 5 timeslots. Finally, our method removed topics which were already found in previous timeslots, whereas the winner for instance did no duplication removal. As a result of our duplication removal step, we could find topics of the ground truth in other timeslots than the evaluated timeslots. These topics were not used to calculate the recall and F1-score, leading to worse results in the SNOW evaluation process. On the other hand, detection of duplicated topics was not penalized.

4.5 Conclusions

We proposed a methodology which automatically mines Twitter streams to provide journalists with a set of headlines and complementary information that summarizes the most important topics for a number of time intervals of interest. As we are only interested in newsworthy topics, we only use tweets of users who are classified as 'news publishers'. These tweets are then grouped into topics using a DBSCAN clustering algorithm, whereby the similarity between the tweets is determined using the cosine similarity on their boosted *tf-idf* representations. Thereafter, a classifier is trained to estimate which of the detected topics is newsworthy. Finally, for each obtained newsworthy topic, a descriptive headline, together with

relevant tweets, keywords and pictures is determined. Experimental results show the effectiveness of the proposed methodology.

4.6 Acknowledgments

Steven Van Canneyt is funded by a Ph.D. grant of the Agency for Innovation by Science and Technology in Flanders (IWT).

References

[1] H. Becker, M. Naaman, and L. Gravano. *Learning similarity metrics for event identification in social media*. In Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, pages 291–300, 2010.

[2] L. Chen and A. Roy. *Event detection from Flickr data through wavelet-based spatial analysis*. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, pages 523–532, 2009.

[3] C. Li, A. Sun, and A. Datta. *Twevent: Segment-based event detection from tweets*. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pages 155–164, 2012.

[4] T. Reuter and P. Cimiano. *Event-based classification of social media streams*. In Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, page 22, 2012.

[5] J. Weng, Y. Yao, E. Leonardi, and F. Lee. *Event detection in Twitter*. In Proceedings of the 5th International AAAI Conference on Weblogs and Social Media, pages 401–408, 2011.

[6] T. Sakaki. *Earthquake shakes Twitter users: Real-time event detection by social sensors*. In Proceedings of the 19th International Conference on World Wide Web, pages 851–860, 2010.

[7] J. Sankaranarayanan, B. E. Teitler, H. Samet, M. D. Lieberman, and J. Sperling. *TwitterStand: News in tweets*. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 42–51, 2009.

[8] S. Papadopoulos, D. Corney, and L. Aiello. *SNOW 2014 Data Challenge: Assessing the Performance of News Topic Detection Methods in Social Media*. In Proceedings of the SNOW 2014 Data Challenge, 2014.

[9] H. Becker, M. Naaman, and L. Gravano. *Beyond trending topics: Real-world event identification on Twitter*. In Proceedings of the 5th International AAAI Conference on Weblogs and Social Media, pages 438–441, 2011.

[10] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Goker, I. Kompatsiaris, and A. Jaimes. *Sensing Trending Topics in Twitter*. IEEE Transactions on Multimedia, 15(6):1268–1282, 2013.

[11] M. Hall, H. National, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. *The WEKA data mining software : An update*. SIGKDD Explorations, 11(1), 2009.

[12] S. Keerthi, S. Sundararajan, and K. Chang. *A sequential dual method for large scale multi-class linear SVMs*. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 408–416, 2008.

[13] G. Cooper and E. Herskovits. *A Bayesian method for the induction of probabilistic networks from data*. Machine Learning, 9(4):309–347, 1992.

[14] M. Ester, H. Kriegel, J. Sander, and X. Xu. *A density-based algorithm for discovering clusters in large spatial databases with noise*. In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, pages 226–231, 1996. Available from: http://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf.

[15] S. Phuvipadawat and T. Murata. *Breaking news detection and tracking in Twitter*. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pages 120–123, aug 2010.

[16] K. Gimpel, N. Schneider, B. O. Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. *Part-of-speech tagging for twitter: Annotation, features, and experiments*. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 42–47, 2010. Available from: http://oai.dtic.mil/oai/oai?verb=getRecord{&}metadataPrefix=html{&}identifier=ADA547371.

[17] J. Platt. *Fast training of Support Vector Machines using Sequential Minimal Optimization*. In Advances in Kernel Methods - Support Vector Learning, pages 185–208. 1998.

[18] M. Porter. *An algorithm for suffix stripping*. Program: Electronic Library and Information Systems, 14(3):130–137, 1980.

[19] G. Ifrim, S. Bichen, and I. Brigadir. *Event detection in Twitter using aggressive filtering and hierarchical tweet clustering*. In Proceedings of the SNOW 2014 Data Challenge, pages 1–7, 2014.

5

Towards a Data-Driven Online News Publishing Strategy

Due to the global availability of online news, there is a strong competition between online publishers in order to reach the highest possible audience. This is why an intelligent online publishing strategy is of the highest importance to news publishers. This can be achieved by acquiring profound and real-time insights into the consumption and social sharing behavior of users with respect to their online content. In this chapter, we propose a highly scalable framework that monitors and analyzes the consumption behavior of online news articles in real-time. As the optimal publishing strategy is highly publisher-specific and depends on the considered popularity metric (e.g. number of Facebook shares), our framework can be easily scaled to cover many news websites and it considers six popularity metrics. In addition, we introduce a number of article features used by our monitoring system, and show that these are vital for a good understanding of the news consumption and sharing behavior. Our framework is thoroughly evaluated on two quite different news websites. We show that our generic data-driven framework and the analysis approach are well suited for real-world use cases, and use them to provide new insights into online news sharing and consumption behavior.

⋆ ⋆ ⋆

S. Van Canneyt, P. Leroux, B. Dhoedt, T. Demeester

Submitted to IEEE Transactions on Knowledge and Data Engineering, Apr. 2016

5.1 Introduction

Online news constitutes a large and still growing market. A higher popularity of content generally means more revenues, for any underlying business model. However, for a news article to become popular, it is essential that it reaches a large audience within a short time. The popularity of an article can be improved by promoting the article on the front page of the news website, or by publishing a link to the article on different social media platforms such as Facebook and Twitter. However, there is a complex interaction between different articles, restricting the possible actions publishers can take to promote individual articles. For instance, the number of articles journalists can write and put on their home page is limited, as is the number of Twitter and Facebook messages publishers can afford to send out per day, in order to retain the interest of their audience. Especially via social media channels, the competition is harsh, because different sources compete to generate content which is potentially relevant to a large subset of the population. Therefore, nowadays a well thought out online publishing strategy is of the highest importance to publishers. This can be achieved by acquiring profound insights into the consumption and social sharing behavior of users with respect to online news. These insights can help to provide answers to questions such as 'What story works best on which medium and when?' and 'What is best joint strategy for both social media and the own news portal for a given set of news articles?'.

A primary analysis was performed for some specific online news agencies and a limited set of popularity metrics [1]. However, we noticed that the analysis can be highly dependent on the considered news website and popularity metric. For instance, the strategy to optimize the number of Facebook shares for website A may be very different from the strategy to optimize the number of visits for website B. Therefore, we propose a generic framework which can be used to monitor and analyze the consumption and social sharing behavior of users on news websites. The framework monitors the popularity of online news articles in real-time, and can be easily scaled to handle millions of visits and thousands of articles. As the publishing strategy depends on the popularity metric that the agency wants to optimize, the framework handles six different metrics including the contributions of Facebook and Twitter shares and the direct browsing behavior of readers. Additionally, we consider a large set of article features such as the article's title and category. Some new article features are introduced, such as the emotion that expresses why users are expected to share the article, which appears to be highly indicative for the sharing and visiting behavior of the articles. The proposed framework can be used by any news website to acquire profound insights into the consumption and social sharing behavior of users with respect to their online news articles.

The proposed framework is evaluated thoroughly on two Belgian news websites which have very different publishing strategies. The first considered website

is newsmonkey[1]. This Buzzfeed-like private news agency website was launched in 2013 and publishes a wide variety of content, ranging from local and global news to viral entertainment articles. The website mainly focuses on optimizing their content in terms of the number of Facebook shares and Facebook visits. The second website is deredactie.be[2], owned by the national public-service broadcaster for the Flemish Region Belgium (VRT). The website was launched in 2003 and mainly covers traditional news items and movie fragments of programs broadcasted by their public television channel. This website focuses on the number of article views by optimizing their websites front-end.

We verified that our framework is able to collect data in real-time for a data-driven online news publishing strategy. The framework has been running in a stable way since April 2015, monitoring tens of thousands of articles and capturing millions of visits. The framework can easily be scaled to handle a lot more views, articles, and websites. During the evaluation of the framework, we noticed that the type of content that is popular, depends on the considered popularity metric. For instance, news about taboo, malicious pleasure and video fragments are much more popular in term of social visits than in term of social shares. On the other hand, articles covering opinions are more likely to be shared than to be visited. We also observed a strong difference between the average consumption and sharing behavior between the articles of newsmonkey en deredactie.be. For example, the difference between the Facebook and Twitter behavior is much more clear for newsmonkey articles. In addition, deredactie.be covers other article categories than newsmonkey, leading to other observations. These insights are used by the journalists to proactively adapt and optimize their news publishing and social sharing strategies.

The remainder of this chapter is structured as follows. We start with a review of related work in Section 5.2. Next, in Section 5.3, we describe the proposed framework to collect and analyze the popularity behavior of articles in real-time. Subsequently, in Section 5.4 and 5.5, we evaluate our framework thoroughly on two news websites. Finally, we conclude our work in Section 5.6.

5.2 Related Work

A small number of research contributions discuss the relationship between news article features and their popularity, and this for a limited set of specific news websites and popularity metrics. Berger et al. [1] investigated the impact of the article features on the likelihood that they will be included in the 'most e-mailed list'. The data was collected from the New York Times, which continually reports which articles have been e-mailed the most during the past 24 hours. They focused

[1] http://newsmonkey.be
[2] http://deredactie.be

on the impact of emotion in the article, and concluded that content that evokes high arousal emotions (awe, anger, anxiety) are more likely to be popular than other emotions (e.g., sadness) or low emotions. In addition, articles that are featured longer in more prominent positions on the New York Times home page, articles by more famous authors, and longer articles are more likely to make it to the most e-mailed list.

Other research focuses on the predictivity of article features towards its popularity. Tsagkias et al., for example, [2] collected articles from Dutch news websites and used the number of comments as popularity metric. They investigated the predictivity toward the popularity of five different groups of article features. They concluded that textual features (e.g., tf-idf values of the article text) and semantic features (e.g., number of person-type entities) are most predictive, and that the predictivity of meta-data features (e.g., publication time, category), cumulative features (e.g., number of near-duplicates), and real world features (temperature and publication time) is more limited. The authors of [3] investigated methods to predict the number of tweets that will mention a given news article, based on its features. The most predictive feature they considered was the article source. The contribution of the category of the article, the subjectivity of its content, and named entities appeared to be limited. The research described in [4] investigated the prediction models constructed in [3] more thoroughly and added content features such as the length of the title and the popularity of the named entities on Twitter, Wikipedia, and web search. As popularity metrics, they considered both the number of tweets and the number of page views one week after publishing. The focus of the paper was not to investigate the relationship between the features and the popularity of the article, but the feasibility of predicting news popularity before its actual publication. They concluded that it is hard to accurately estimate an article's popularity, solely on the basis of content features, without incorporating any early-stage popularity information.

In this chapter, we propose a generic framework that can be applied to any news website and that considers a more extensive set of popularity metrics and article features than previously reported. We evaluate the framework on two news websites with very different publishing strategies over a nine-month period.

5.3 Framework

In order to collect profound insights into the consumption and social sharing behavior of users, we propose a generic framework containing three main parts, as visualized in Figure 5.1. First, the most prominent part of the framework is the monitoring system. This system monitors the popularity of news articles in real-time and collects a set of features related to the articles. Second, this monitored data is analyzed to acquire insights. Finally, we demonstrate an example Graph-

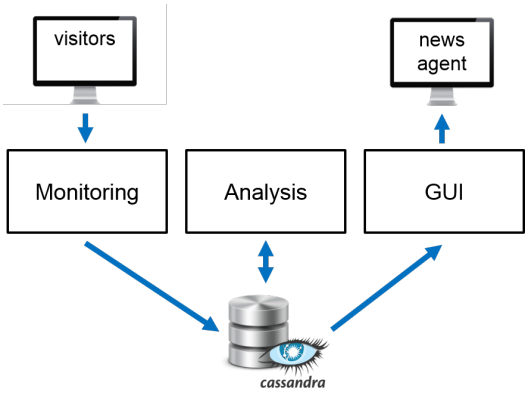


Figure 5.1: High-level visualization of proposed framework.

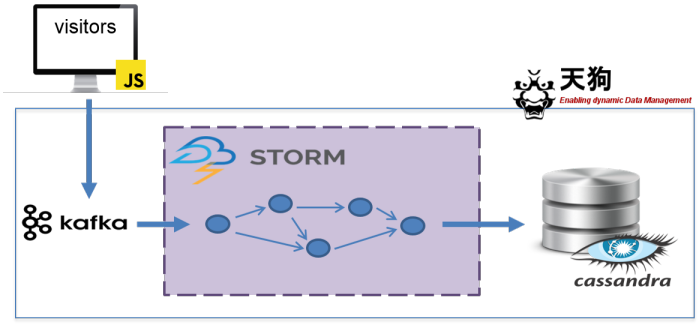


Figure 5.2: Monitoring architecture.

ical User Interface (GUI) which displays the monitored article data in real-time, together with the analysis. This GUI can be used by news agents to improve their insights into the popularity behavior of their articles and to optimize their publishing strategy. The framework is explained in more detail in the rest of this section.

5.3.1 Monitoring System

The real-time monitoring of online news websites demands a solid scalable architecture, both in terms of storage and speed. The overview of the architecture we developed to collect features and to monitor the number of views of articles in real-time is shown in Figure 5.2. We embedded a Javascript snippet in all the web pages of the news articles we wanted to track. Each time a web page is visited, this tracking code sends information about the user to our platform. This information includes the web page URL, the HTTP referer, and the user IP. The information is

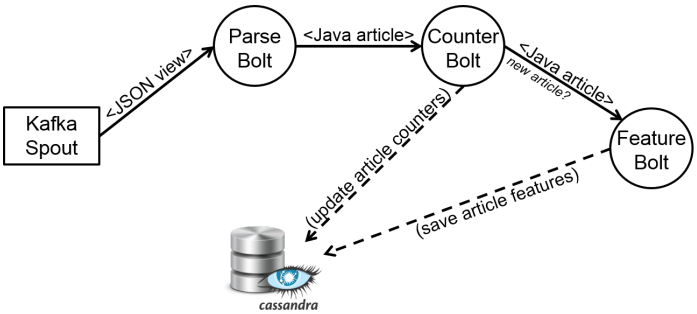


Figure 5.3: Storm topology used to monitor the data.

sent in JSON format to an Apache Kafka[3] message broker. Kafka is used because it is optimized to be fast and scalable, and messages are persisted on disk and replicated within the cluster to prevent data loss when Kafka or other parts of our monitoring system fails. An Apache Storm[4] computation system reads the messages from Kafka and in real-time computes a series of features and the number of views of the news articles. Storm is a distributed real-time computation system that is designed to be scalable, fast, and fault-tolerant. Finally, the article features and number of views are saved in an Apache Cassandra[5] database. This type of database can handle large amounts of data across many servers, providing high availability with no single point of failure. The used technologies are designed to be fast and fault-tolerant. Additionally, they can be easily distributed over several servers, making the framework highly scalable in terms of the number of news websites, processed articles, and monitored visits. These big data streaming and storage technologies have been set up using Tengu [5].

As mentioned, a Storm system was constructed to determine the articles features and views in real-time. An application in Storm is described through a directed acyclic graph with spouts and bolts acting as the graph vertices. Edges on the graph direct data from one node to another. Together, the topology acts as a real-time data transformation pipeline. Spouts and bolts can be distributed over different threads and machines. Our used topology is visualized in Figure 5.3. The spout reads the JSON messages from Kafka. These messages are sent to the Parse bolt which parses the JSON messages into Java article objects and determines if the view originates from Facebook, Twitter, or elsewhere. The obtained article objects are then sent to the Counter bolt. This bolt detects if the viewed article is already known by our system. If the article is known, the view counters of the article are updated in the Cassandra database. If the article is not known, a record

[3]http://kafka.apache.org/
[4]http://storm.apache.org/
[5]http://cassandra.apache.org/

for the article is added to the Cassandra database. This record contains basic article information such as its URL and number of views. The objects of unknown articles are sent to the Feature bolt. This bolt determines additional features of the article. In particular, the article title, authors, categories, and potential sources are parsed from the HTML page of the news article. Named entities are extracted from the article title using [6], and the Facebook shares and number of tweets containing the article sources are collected. The features are finally added to the database. See Section 5.3.3 for an overview of the used features.

We launched a separate Storm process to construct popularity series of the news articles. Each hour, the popularity of the news articles that are published no longer than one week ago are extracted and stored in the Cassandra database. The current number of Facebook, Twitter and total views of an article are retrieved from our Cassandra database. The Facebook Graph API is used to collect the so-called Facebook buzz of messages containing the URL of the news article (see Section 5.3.2). We used the Twitter Rest API to retrieve the number of tweets containing the URL of the news article as an additional measure of popularity. In addition, each hour, the messages posted during the last 6 hours on the official Facebook pages and Twitter accounts of the news websites were collected. This information was used to determine for each article when it was posted on the news agency's social media pages.

5.3.2 Popularity Metrics

The framework considers six different popularity metrics. In particular, we consider the number of total views, Facebook views, Twitter views, direct views, Facebook buzz, and Twitter posts. Facebook views and Twitter views are the page visits which come directly from, respectively, Facebook and Twitter. The direct views, mostly from users directly browsing through the news agency's web page or through search engines, are constructed as the total views minus the Facebook and Twitter views. The Facebook buzz is the sum of the number of Facebook likes, comments, and shares related to the article. Finally, the Twitter posts or 'tweets' are the number of posts on Twitter that contain the article's URL.

An article's popularity is measured as the value of the previously introduced metrics 7 days after publishing on the website. We observed that most articles have a lifetime considerably shorter than that, such that the previously introduced metrics become stable well before a week after their initial publication.

5.3.3 Article Features

Both automatically retrieved features and manually annotated features are considered by our framework. The automatically constructed features are based on the

title and the publication time of the article. In particular, we determine if the article contains a number and we also use named entity recognition [6] to extract the named entities and their types from the title. In addition, the day of week and the hour of day the article is published on the news website is determined. The manual features are constructed by the journalist and are dependent on the news website. A manual feature used by most websites contains the article's topics. The journalists working for these websites have tagged each article with one or more categories (e.g., politics and economy). The journalists of news websites considered in this chapter also label their articles with other features such as emotion and target audience (see Section 5.4.1).

5.3.4 Analysis Module

In the analysis module, we investigate the impact of the various article features on the different popularity metrics. This module is implemented in Java and is performed on all data currently in the database. In contrast to the monitoring of the articles' popularity and features, the analysis module is not in real-time. The module can for instance be rerun every couple of days. The analysis is performed in batches because we want some general insights into the consumption behavior of the articles, which mostly do not change very fast. For example, the best publication time based on the articles of last year, the best performing category on Facebook for the last week, or the author with receiving most retweets for each of the past 24 hours.

To evaluate the impact of the article features on their popularity, we determine for each considered feature the average popularity of the articles containing this feature. For a fair evaluation, we only consider articles which are published on Facebook for the Facebook related popularity metrics (Facebook views and Facebook buzz). The same approach is applied for the Twitter related metrics. This filtering is conducted to eliminate the bias of the publishing strategy of the news website.

We also investigate the predictivity of the article features towards the popularity metrics. In other words, we study to what extent can we predict the final popularity of an article given its features. For this task, we split the data in two parts. The first two thirds are used as training set K, and the last third is considered as test set U. We train a linear regression model[6] on our training set K with a subset of the article features as input variables and the total popularity after 7 days as output variable. This regression model is then used to predict the total popularity of the articles in the test set U. The root mean squared log error (RMSLE) is used to evaluate the performance of these predictions. This evaluation metric is

[6]We use the WEKA data mining library [7].

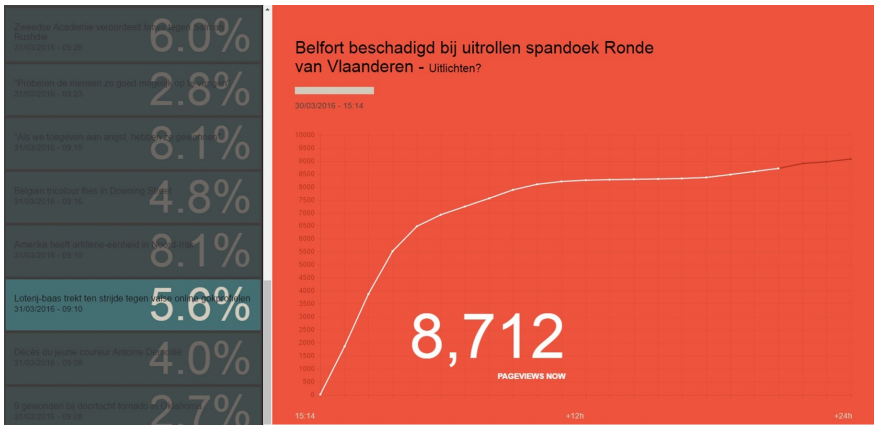


Figure 5.4: Screenshot of a part of the GUI showing the number of views of an article received over time.

also used in the ECML/PKDD 2014 Predictive Analytics challenge[7]. In general we can state that article features which lead to a smaller RMSLE have a higher predictive power.

5.3.5 Graphical User Interface

The Graphical User Interface (GUI) visualizes the news articles with their features and popularity in real-time. Together with the insights calculated by the analysis module, this information can be used to better make decisions on the publishing and sharing strategy. A screenshot of the number of views an article received over time is shown in Figure 5.4.

The back-end is implemented using the Play Framework[8], which is a lightweight web framework optimized for highly scalable applications. It retrieves data from the Cassandra database and represents it as JSON data. The front-end uses css, html and angularJS to nicely represent the monitored and analyzed data.

5.3.6 Evaluation

The framework has been running in a stable way starting from April 2015, and monitored about 40,000 articles, 15 million visits and 4 million social shares before February, 2016. At the moment it monitors two news websites, i.e., newsmonkey (Dutch and French website) and deredactie.be, and handles on average about 4,000 views per minute, with peaks up to 10,000 views per minute. The

[7]https://sites.google.com/site/predictivechallenge2014/
[8]https://www.playframework.com/

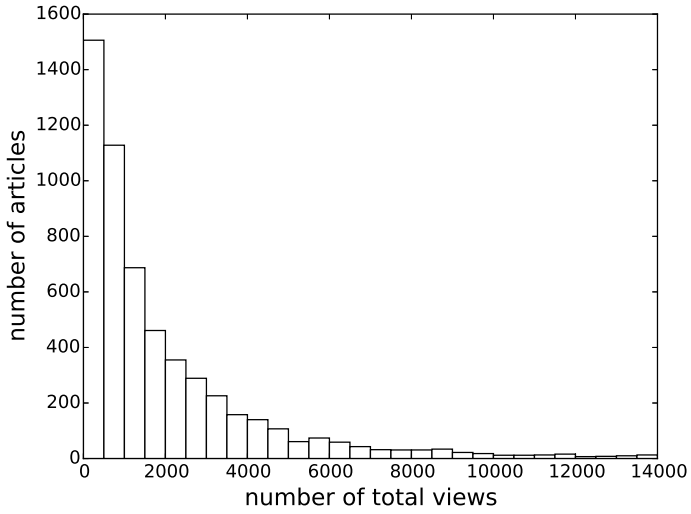


Figure 5.5: Newsmonkey: Histogram of the total number of views.

framework can easily be scaled to handle a lot more views and websites. The insights (see next sections) and real-time monitored data are used by the journalists of newsmonkey and deredactie.be to improve their publishing strategy.

5.4 Use Case: newsmonkey

In this section and Section 5.5, we apply the proposed framework on existing news websites and discuss the obtained insights. A collaboration with these news sites was set up to be able to track their news articles. In the first use case, we consider the news website of newsmonkey. Newsmonkey is a private Buzzfeed-like news website, currently focusing on the Belgian market. Similar to Buzzfeed, they combine breaking news with highly shareable stories. The website mainly focuses on retrieving views and shares from Facebook.

5.4.1 Monitoring: Data Statistics

The monitoring system collected 5,652 articles between April 27, 2015 and January 20, 2016. The average number of views per article is around 2,500, of which 65% come directly from Facebook and only 1% from Twitter. The average Facebook buzz and Twitter posts per article are respectively about 200 and 5. This is in line with the strategy of newsmonkey, with the main focus on optimizing Facebook buzz and Facebook views. An article receives on average 6 Twitter views per Twitter post, and 8 Facebook views per Facebook buzz unit. The histogram for the

Table 5.1: Newsmonkey: Top articles for each considered popularity metric.

popularity	*article title*
+93,000 views	What happened in Cologne has a name: taharrus jama'i. This is what you should know about it:
+92,000 views	13 reasons why a West-Flemish guy is the perfect love
+46,000 direct views	Popcorn Time is back
+15,000 direct views	Call for media silence about #BrusselsLockdown is massively followed: that's the power of social media
+81,000 Facebook views	What happened in Cologne has a name: taharrus jama'i. This is what you should know about it:
+80,000 Facebook views	"World War III might start this summer. If we are lucky, it won't be a nuclear one"
+1000 Twitter views	10 things we did not know our iPhones are able to do
+900 Twitter views	Hi-laaaa-rious: the NMBS-Twitter remains deadly calm with idiotic questions
+25,000 Facebook buzz	13 reasons why a man who can cook is super sexy
+24,000 Facebook buzz	13 reasons why a West-Flemish guy is the perfect love
+140 Twitter posts	"Unfortunately for us we have only one politician blessed with the Churchill-factor: Bart De Wever"
+100 Twitter posts	There are certainties: the socialist rail union is striking again

Table 5.2: Newsmonkey: Emotional reasons to engage with news article on Facebook, together with an example article's title.

emotion	*article title*
recognizability	21 issues only people with a tattoo understand
identity	19 reasons for why nurses are real superheroes
awe	14 hidden gems: affordable holidays that aren't obvious
humor	13 things men say when they are drunk
pride	19 reasons for why it is great to have a big sister
malicious pleasure	It's the time of the wedding parties: here are 37 pictures of thing that can go wrong
altruism	29 ingenius tricks that make your parents' lives less tiring
taboo	13 sex questions that men do not dare to ask women
outrage	28 sentences that ICT professionals never want to hear again
nostalgia	Yummie! Here are 21 candies from our childhood that we terribly miss
softening	21 great reasons why rabbits are the most awesome and cutest pets in the world

total number of views is shown in Figure 5.5. We see that there is a high skewness in the popularity distribution, i.e., a large number of unpopular articles and a very few popular articles. In other words, the popularity of the most popular article is very high in comparison to the average popularity. The top two articles for each metric are shown in Table 5.1. The original content is in Dutch, but we translated the titles into English for convenience. As can be observed based on these titles, the type of articles that are popular depends on the considered metric. This will be investigated thoroughly in Section 5.4.2.

Prior to publication, the journalists of newsmonkey manually label all articles with one or more category (e.g. politics and economy) from a set of 16 categories (see Table 5.3). They also indicate whether, according to their experience, they estimate those articles are likely to go viral on Facebook. For the articles assigned the 'will go viral' label, they also indicate the target audience and expected emotion. The target audience is given by the target gender (female, male, or both) and target age range (18-24, 25-34 years, or 18-34 years old). The annotated emotion label of a particular article is not the direct emotion the article content is expected to provoke in the readers, as considered in previous research [1, 4]. Instead, it is the emotion users expect to solicit by their sharing of the content. The considered emotion labels are recognizability, identity, awe, humor, pride, malicious pleasure, altruism, taboo, outrage, nostalgia, and softening. Example titles for each emotion can be found in Table 5.2. In the considered dataset, only 336 articles were manually labeled by the journalists as 'will go viral' and were annotated with the target audience and emotion.

5.4.2 Analysis

In this section, we investigate the impact which features impact the different popularity metrics for the newsmonkey articles most. The average article popularity for the considered features and metrics can be found in the Tables 5.3 (direct and total views), 5.4 (Twitter) and 5.5 (Facebook). The popularities are given relative to the average article popularity (in percentage). The RMSLE values for the considered article features and popularity metrics is listed in Table 5.6. Note that the insights gained from our analysis only reflect the behavior of the target audience of the considered publisher, i.e., in this case Flemish readers. Unlike the proposed framework and analysis method, the reported insights cannot be directly generalized to any other publisher or target audience.

5.4.2.1 Category

The articles' category is the most predictive feature (see Table 5.6). This is in contrast to the observations of [3], where the authors concluded that the predictivity of the category feature was very limited towards the number of Twitter posts. The

Table 5.3: Newsmonkey: The average relative popularity for the articles containing the given feature (in percentage), considering all 5,652 articles. The ranks of the features for each feature type are given between brackets.

	Total views		*Direct views*		*% articles*
all articles	100		100		100
category: society	91	(9)	108	(5)	23
category: politics	55	(14)	90	(10)	10
category: tv	175	(3)	**176**	(1)	9
category: music	75	(10)	83	(12)	9
category: life and style	**189**	(1)	112	(4)	8
category: cyberspace	99	(6)	103	(8)	8
category: tech and gadgets	93	(8)	103	(7)	6
category: planet	67	(11)	82	(13)	6
category: travel	95	(7)	85	(11)	6
category: movies	62	(13)	67	(14)	5
category: stars	120	(4)	116	(3)	4
category: economy	65	(12)	93	(9)	4
category: body and soul	188	(2)	116	(2)	4
category: science	100	(5)	107	(6)	3
category: pets	45	(15)	42	(16)	2
category: games	37	(16)	48	(15)	1
no number in title	81	(2)	92	(2)	71
number in title	**147**	(1)	**119**	(1)	29
no named entity in title	**117**	(1)	96	(4)	37
named entity in title: person	87	(4)	105	(2)	27
named entity in title: organization	93	(3)	**110**	(1)	23
named entity in title: location	79	(5)	94	(5)	18
named entity in title: other	99	(2)	102	(3)	14
publication on website: week	99	(2)	98	(2)	81
publication on website: weekend	**106**	(1)	**107**	(1)	19

Table 5.4: Newsmonkey: The average relative popularity for the articles containing the given feature (in percentage). For a fair evaluation, we only consider articles that are published on Twitter (3747 articles). The ranks of the features for each feature type are given between brackets.

	Twitter views	*Twitter posts*	*% articles*
articles published on Twitter	100	100	100
category: society	98 (8)	121 (3)	28
category: politics	135 (6)	**177** (1)	14
category: tv	57 (13)	62 (13)	12
category: music	48 (14)	58 (14)	10
category: life and style	159 (2)	77 (8)	2
category: cyberspace	152 (3)	89 (6)	8
category: tech and gadgets	**169** (1)	109 (4)	7
category: planet	82 (11)	99 (5)	7
category: travel	114 (7)	72 (10)	5
category: movies	16 (15)	37 (16)	7
category: stars	71 (12)	43 (15)	4
category: economy	148 (5)	135 (2)	5
category: body and soul	153 (4)	78 (7)	2
category: science	90 (9)	70 (11)	4
category: pets	7 (16)	75 (9)	1
category: games	84 (10)	65 (12)	1
no number in title	**104** (1)	**100** (1)	76
number in title	87 (2)	99 (2)	24
no named entity in title	**118** (1)	**108** (1)	28
named entity in title: person	95 (3)	96 (4)	32
named entity in title: organization	98 (2)	107 (2)	28
named entity in title: location	87 (4)	104 (3)	22
named entity in title: other	75 (5)	72 (5)	16
publication on Twitter: week	98 (2)	100 (2)	77
publication on Twitter: weekend	**108** (1)	**102** (1)	23

Table 5.5: Newsmonkey: The average relative popularity for the articles containing the given feature (in percentage). For a fair evaluation, we only consider articles that are published on Facebook (4366 articles). The ranks of the features for each feature type are given between brackets.

	Facebook views	*Facebook buzz*	*% articles*
articles published on Facebook	100	100	100
category: society	92 (8)	85 (4)	20
category: politics	34 (16)	57 (12)	10
category: tv	149 (3)	138 (3)	11
category: music	64 (10)	64 (10)	10
category: life and style	202 (2)	**289** (1)	9
category: cyberspace	99 (5)	46 (14)	8
category: tech and gadgets	89 (9)	38 (16)	6
category: planet	58 (11)	82 (5)	6
category: travel	96 (6)	78 (6)	6
category: movies	56 (12)	53 (13)	6
category: stars	108 (4)	57 (11)	5
category: economy	52 (13)	42 (15)	4
category: body and soul	**223** (1)	238 (2)	4
category: science	92 (7)	68 (9)	3
category: pets	42 (15)	74 (7)	2
category: games	48 (14)	71 (8)	1
no number in title	76 (2)	64 (2)	71
number in title	**161** (1)	**190** (1)	29
no named entity in title	**129** (1)	**140** (1)	36
named entity in title: person	72 (5)	63 (5)	28
named entity in title: organization	79 (3)	77 (4)	23
named entity in title: location	78 (4)	89 (2)	17
named entity in title: other	97 (2)	87 (3)	15
publication on Facebook: week	99 (2)	**101** (1)	72
publication on Facebook: weekend	**107** (1)	88 (2)	38
not labeled as 'will go viral'	83 (2)	70 (2)	92.3
labeled as 'will go viral'	**307** (1)	**463** (1)	7.7
target audience: 18-34 years	296 (2)	450 (2)	6.3
target audience: 18-24 years	246 (3)	226 (3)	0.6
target audience: 25-34 years	**440** (1)	**761** (1)	0.8
target audience: women and men	306 (2)	395 (2)	5.7
target audience: women	304 (3)	**727** (1)	1.6
target audience: men	**322** (1)	358 (3)	0.4
emotion: recognizability	334 (4)	314 (5)	1.7
emotion: identity	256 (8)	590 (2)	1.6
emotion: awe	179 (9)	235 (8)	1.1
emotion: humor	322 (6)	307 (6)	0.9
emotion: pride	**581** (1)	**1694** (1)	0.6
emotion: malicious pleasure	289 (7)	103 (11)	0.5
emotion: altruism	212 (10)	198 (9)	0.5
emotion: taboo	322 (5)	186 (10)	0.5
emotion: outrage	458 (2)	587 (3)	0.4
emotion: nostalgia	363 (3)	497 (4)	0.4
emotion: softening	140 (11)	261 (7)	0.0

Table 5.6: Newsmonkey: Root mean squared log error (RMSLE) of linear regression predictions if feature is considered.

	Total views	*Direct views*	*Facebook views*	*Twitter views*	*Facebook buzz*	*Twitter posts*
all features	1.164	0.915	2.202	2.474	1.772	0.859
category	1.177	0.936	2.228	2.530	1.777	0.861
number in title	1.267	1.003	2.357	2.487	1.854	0.975
named entities in title	1.273	1.006	2.355	2.512	1.863	0.964
'will go viral' label	1.271	1.013	2.362	2.518	1.854	0.972
audience: age	1.271	1.013	2.362	2.518	1.855	0.972
audience: gender	1.272	1.012	2.362	2.521	1.855	0.973
emotion	1.268	1.011	2.358	2.519	1.851	0.972
time of day	1.282	1.003	2.389	2.472	1.886	0.966
week/weekend	1.276	1.004	2.366	2.492	1.869	0.973

reason may be that the authors of [3] used Feedzilla news categories which are less indicative toward an article's popularity than the newsmonkey categories.

If we consider Facebook buzz (Table 5.5) versus Twitter posts (Table 5.4), we observe on the one hand that the top 3 categories for Facebook are life and style, body and soul, and tv; and for Twitter politics, economy and society. In other words, the type of content most shared on Facebook is light-weighted and with emotional content, lists, and funny pictures. On the other hand, news related facts such as breaking news about politics and thorough analysis about the economy are more likely to be shared on Twitter.

Categories such as politics and planet are higher ranked for Facebook buzz and Twitter posts than for Facebook views and Twitter views. Examples are 'Sp.a demands one police district for Brussels and more resources for State security' and 'Daring plan to clear away plastic from the sea using an ocean vacuum cleaner starts in Japan next year'. This leads to a low average number of views per share (about 5 views per share, versus an average of about 7.5 views per share). Users want to share the message and opinion of the article with their friends, but are less interested in the content details of the article. On the other hand, categories such as cyberspace and tech-and-gadgets are higher ranked for Facebook views and Twitter views than for Facebook buzz en Twitter buzz. These articles receive on average more than 10 views per share. The titles of these article are for instance 'Apple does it again and baffles the entire audience: Microsoft is called on stage' and '12 things we didn't know are lethal to our smartphone's battery'. Typical users are more interested in the details of those technical subjects than the desire to share the articles with their friends.

5.4.2.2 Title

We first determine if the article title contains a number. Those articles are mostly articles containing lists and their title often starts with 'n reasons why...', with n a number. These 'list' articles are constructed with the main objective that they

are very shareable on Facebook. As can be seen in Table 5.5, articles containing a number in their title receive indeed on average 2 to 3 times as much Facebook engagement than articles without a number in their title. However, articles containing a number do not perform well on Twitter (see Table 5.4). Articles with a number in their title even receive on average less Twitter engagement than articles that do not have a number in their title.

The assumption is that articles containing a named entity in their title are mostly factual news items, e.g. 'IS claims attacks in Paris'. In contrast, for light weight news, we assume that the title often will not contain a named entity, for instance '9 reasons for why journalists are the best sweethearts'. As expected based on the previous observations, we notice articles without a named entity in their title obtain a higher Facebook engagement than articles with named entities (Table 5.5). On average, the Facebook engagement is more than 60% higher for articles without an associated named entity than with. More surprisingly, the Twitter engagement is also higher for articles without than with associated named entities (Table 5.4). This is in contrast with the observations in Section 5.4.2.1, where we noticed that fact-related news performs better on Twitter than light-weight news. A reason for this may be that having a named entity in the title is not optimal to decide whether the article is purely factual. For instance, article titles such as 'Buffalooo! 13 reasons why AA Ghent really deserves the title' contain named entities and are light-weighted. The category of the article may be a more informative feature to indicate the kind of news. However, we note that the difference in popularity between articles with and without named entities in their title is lower in terms of Twitter engagement than in terms of Facebook engagement.

5.4.2.3 Label 'will go viral'

The journalists of newsmonkey manually annotated a news article as 'will go viral' if they estimate that it will be very popular on Facebook. Indeed, those articles receive on average 4 times as much Facebook views and 7 times as much Facebook buzz than articles that are not labeled as 'will go viral' (see Table 5.5). Based on the input from the publishers, we define articles as 'viral' on Facebook if they belong to the top 15% of all articles in terms of Facebook views or in terms of Facebook buzz. Using this definition, we observe that 66% of the articles which are labeled as 'will go viral' are indeed going viral on Facebook (precision). In addition, 20% of the articles that did go viral were labeled as 'will go viral' (recall). In other words, with the proposed cut-off of virality as the top 15% most popular articles, we can say that most of the articles which are labeled as 'will go viral' will indeed go viral. However, the low recall indicates that most articles that turn out to go viral, are not identified as such before publication.

5.4.2.4 Target Audience

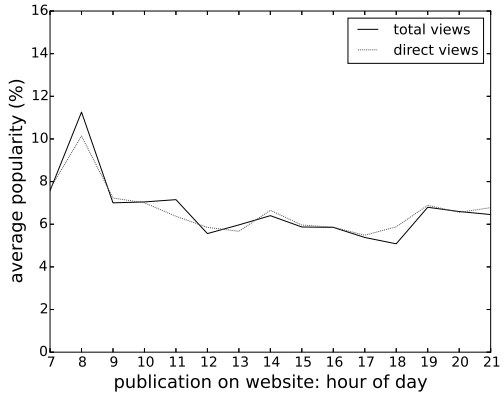
The most popular content on Facebook is targeted to the older half of the target audience of newsmonkey (25 - 34 years), see Table 5.5. Articles targeted to an audience between 18 and 24 years are about half as popular in terms of Facebook views and less than a third as popular in terms of Facebook buzz. Articles targeted to men receive on average more Facebook views per Facebook buzz (on average 7 views/buzz) than articles targeted to women (on average 3 views/buzz). This may indicate a different sharing behavior for men and women. However, a more thorough analysis is needed in order to make clear conclusions.

5.4.2.5 Emotion

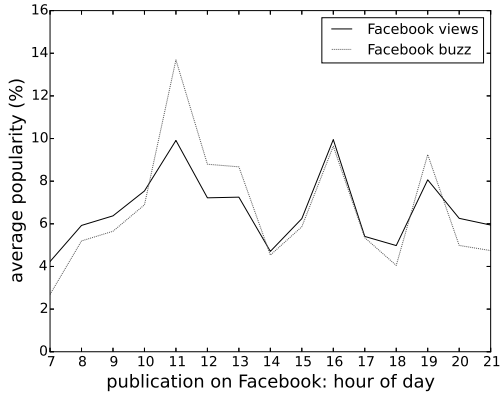
The most important emotional reason to engage on Facebook with news articles is pride (see Table 5.5). Example articles are '19 reasons for why it is great to have a big sister' and '15 reasons for why Star Wars fans are the best lovers'. Identity is an important reason to share an article with friends or to like or comment on an article. However, those articles receive a relatively small number of click-troughs (rank 2 versus rank 7, with an average of less than 4 Facebook views per Facebook buzz). The users identify themselves with the title of the article and want to share it with their friends, they are less interested in the actual content of the article. The opposite behavior is observed for articles containing malicious pleasure or taboo content, which receive a high number of Facebook views per Facebook buzz (rank 6 vs. 10, with on average more than 14 Facebook views per Facebook buzz). The users are interested in the content of those articles, but do not like to talk about it. This is common behavior associated with taboo topics. The rank for the softening emotion is also very different for Facebook views and Facebook buzz. Yet, as the dataset contains only two articles associated with this emotion, we can make no clear conclusions, except that most likely the journalists think this kind of content is not likely to go viral and hence avoid spending time writing them. The importance of the other emotions is similar for both the Facebook views and buzz. These emotions are outrage, nostalgia, recognizability, humor, awe, and altruism (sorted by importance).

5.4.2.6 Publication time

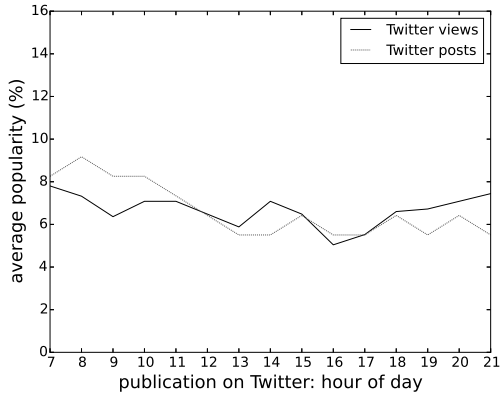
The average article popularity as a function of publication time is plotted in Figure 5.6. Note that in this figure the publication time for total and direct views corresponds to the moment that the article is published on the newsmonkey website. In contrast, the considered publication time for the Facebook views and buzz, and Twitter views and posts, corresponds to the moment that the article is published on Facebook and Twitter, respectively.



(a)



(b)



(c)

Figure 5.6: Newsmonkey: The average relative popularity as a function of the publication hour. The publication time is set to the moment the article is published on (a) the website, (b) Facebook, and (c) Twitter, respectively.

As can be seen in Figure 5.6a, the best time to publish an article on the website is at 8am. The reason for this may be that a lot of people go in the morning to news websites for a general news update. The best moments to publish articles on Facebook are around 11am, 4pm and 7pm (Figure 5.6b). The first two hours correspond to the hour before the noon break and the end of the work day. At the end of those work blocks, people probably tend to go on Facebook for some relaxations and news updates. Additionally, the start of the evening, at 7pm, is a moment that a lot of people consume articles on Facebook. When posting articles on Facebook at those times, more people will see the news article and will engage with them. The moment when an article is published on Twitter has no clear correlation with its popularity (Figure 5.6c). This may be explained by the low Twitter engagement of the articles in our dataset receive.

5.5 Use Case: deredactie.be

The second website we consider is deredactie.be[9]. The website is owned by the Flemish public broadcaster in Belgium (VRT). It mainly covers traditional news items and movie fragments of programs broadcasted by their public television channel. This website mainly focuses on the total number of article views by optimizing their website's front-end, but is starting to focus more on social media such as Facebook.

5.5.1 Monitoring: Data Statistics

Our monitoring system collected 30,674 articles between April 27, 2015 and January 20, 2016. 4,760 of those articles are not in the main language of the website (Dutch), and are not considered in this chapter. The average number of views per article is about 2,500, of which 10% come directly from Facebook and only 0.4% comes from Twitter. In other words, about 90% of the visits are direct views originated by users directly browsing through the news agency's web page or through search engines. This is a lot more than for newsmonkey articles (about 34%) and is in line with the strategy of deredactie.be, which mainly focuses on users directly visiting the deredactie.be website. The average Facebook buzz and Twitter posts per article are respectively about 120 and 0.5. An article retrieves on average 20 Twitter views per Twitter post, and 2 Facebook views per Facebook buzz unit.

The journalists of deredactie.be manually label all articles with a category (e.g., politics or culture) from a tree of categories. In this chapter, we consider the 9 main categories, which are shown in Table 5.7. Note that the TV programs category contains movie fragments of programs broadcasted by their public television channel

[9] http://deredactie.be

Table 5.7: Deredactie.be: The average relative popularity for the articles containing the given feature (in percentage), considering all 25,914 articles.

	Total views		*Direct views*		*% articles*
all articles	100		100		100
category: TV programs	66	(10)	65	(10)	39
category: abroad	100	(8)	104	(8)	18
category: domestic	103	(7)	106	(7)	16
category: culture and media	112	(5)	110	(6)	12
category: politics	112	(6)	117	(5)	4
category: economy	69	(9)	74	(9)	3
category: entertainment	**332**	(1)	**306**	(1)	3
category: opinion	202	(3)	204	(3)	2
category: science	216	(2)	211	(2)	1
category: other	119	(4)	120	(4)	1
no number in title	**101**	(1)	**101**	(1)	83
number in title	96	(2)	97	(2)	17
no named entity in title	113	(2)	111	(2)	35
named entity in title: person	110	(3)	109	(3)	26
named entity in title: organization	72	(5)	73	(5)	15
named entity in title: location	77	(4)	79	(4)	28
named entity in title: other	**115**	(1)	**117**	(1)	8
publication on website: week	99	(2)	99	(2)	82
publication on website: weekend	**103**	(1)	**105**	(1)	18

(mainly news and current affairs programs), whereas the other categories cover more traditional news articles.

5.5.2 Analysis

The difference between the visit and share behavior can be clearly seen in Table 5.8 and Table 5.9. Articles containing opinions or political content are more popular in terms of social shares then in terms of visits. Opinions provoke reactions, leading to shares, reactions, and likes, but it appears that the users are less interested in the details of the article. On the other hand, entertainment and TV program articles are much more popular in terms of social visits than in number of shares. These articles contains video fragments that can only be watched when visiting the article on the news website. We can not observe a large difference between Facebook and Twitter behavior. However, we notice that entertainment and TV program articles are even more unpopular in terms of Twitter posts than in terms of Facebook buzz.

There is no clear difference between the popularity of articles containing a number in their title and other articles. The main reason for that is that the presence or absence of a number in the title has no indication of the type of content the article contains. This is in contrast to newsmonkey articles, where articles

Table 5.8: Deredactie.be: The average relative popularity for the articles containing the given feature (in percentage). For a fair evaluation, we only consider the articles that are published on Twitter (10,276 articles).

	Twitter views		*Twitter posts*		*% articles*
all articles	100		100		100
category: TV programs	90	(8)	6	(10)	7
category: abroad	69	(9)	105	(5)	25
category: domestic	107	(7)	119	(3)	27
category: culture and media	111	(6)	103	(6)	18
category: politics	114	(5)	120	(2)	6
category: economy	63	(10)	93	(7)	6
category: entertainment	**187**	(1)	48	(8)	4
category: opinion	129	(3)	**145**	(1)	3
category: science	123	(4)	106	(4)	2
category: other	168	(2)	43	(9)	1
no number in title	**103**	(1)	98	(2)	83
number in title	88	(2)	**112**	(1)	17
no named entity in title	**114**	(1)	98	(4)	34
named entity in title: person	101	(3)	105	(2)	26
named entity in title: organization	91	(4)	**110**	(1)	17
named entity in title: location	79	(5)	100	(3)	30
named entity in title: other	113	(2)	88	(5)	8
publication on Twitter: week	**102**	(1)	88	(2)	84
publication on Twitter: weekend	100	(2)	**151**	(1)	16

Table 5.9: Deredactie.be: The average relative popularity for the articles containing the given feature (in percentage). For a fair evaluation, we only consider the articles that are published on Facebook (3,008 articles).

	Facebook views		*Facebook buzz*		*% articles*
all articles	100		100		100
category: TV programs	127	(2)	84	(8)	15
category: abroad	87	(6)	97	(6)	17
category: domestic	72	(7)	102	(4)	17
category: culture and media	89	(5)	95	(7)	20
category: politics	54	(9)	115	(3)	4
category: economy	49	(10)	74	(10)	1
category: entertainment	**160**	(1)	98	(5)	15
category: opinion	125	(3)	**202**	(1)	3
category: science	71	(8)	120	(2)	5
category: other	101	(4)	80	(9)	2
no number in title	**102**	(1)	**100**	(1)	85
number in title	88	(2)	97	(2)	15
no named entity in title	**117**	(1)	103	(2)	42
named entity in title: person	98	(2)	**109**	(1)	30
named entity in title: organization	73	(5)	89	(5)	12
named entity in title: location	80	(3)	100	(3)	19
named entity in title: other	79	(4)	96	(4)	10
publication on Facebook: week	**101**	(1)	99	(2)	84
publication on Facebook: weekend	97	(2)	**105**	(1)	16

containing a number in their title mostly contain lists which are very shareable on Facebook. This type of content is not covered by deredactie.be.

Note that a number of insights are similar to the observations made in the newsmonkey use case. For example, articles covering (political) opinions perform much better in term of social shares than in term of visits. However, some observations are different between the two considered news websites. For instance, the difference between Twitter and Facebook behavior is less clear for deredactie.be articles then for the newsmonkey use case. In addition, deredactie.be covers types of content (e.g. video fragments), which are covered by newsmonkey during the monitored period, and vice versa (e.g. lists). This confirms our hypothesis that the analysis into news consumption and sharing behavior is different for different websites, and that we need a generic framework which can be applied to any news website.

5.6 Conclusion

To help optimizing publishing strategies of news agencies, we proposed a framework to monitor and analyze the consumption and social sharing behavior of users on news websites. To test the potential of this framework, we evaluated it thoroughly on two major news websites. We concluded that it is able to monitor the popularity of online news articles in real-time, as it has been running for more than 11 months now[10] having collected more than 40,000 articles, 15 million visits and 4 million social shares. The framework is constructed so that it can scale up to handle many more articles, visits, shares and websites in the future. During the evaluation, we observed that the optimal publishing strategy depends on the considered popularity metric, and differs for both discussed news sites. For example, there is a clear difference in Twitter share behavior and Facebook share behavior for newsmonkey articles. In particular, light-weighted newsmonkey articles with emotional content, funny pictures or lists perform best on Facebook. On the other hand, newsmonkey articles containing breaking news and a thorough analysis about politics or economy perform much better on Twitter than on Facebook. The difference in behavior between Facebook and Twitter was less clear for the deredactie.be articles. Studying new article features led to better understanding of the sharing behavior. For instance, the feature that indicates the emotion that captures why users want to share the article, led to the insight that taboo may be a good indicator to read an article, but not to share it with friends. These observations confirm our assumption of a need for a framework which can be used for a data-driven online news publishing strategy, covering several popularity metrics and features, and which can be applied on any news website.

[10] March, 2016

This real-time monitoring and analysis framework has been deployed at newsmonkey and deredactie.be, and will be used to optimize their publishing strategy. The framework will also made available for other news websites, to monitor and analyze their data and to optimize their strategy. In future work, we will introduce novel methodologies to predict the popularity of online articles. This information can then be used to better decide what article to publish on what social platform at what time.

Acknowledgment

Steven Van Canneyt is funded by a Ph.D. grant of the Agency for Innovation by Science and Technology in Flanders (IWT). The presented research was performed within the MIX-ICON project PROVIDENCE, facilitated by iMinds-Media and funded by the IWT.

References

[1] J. Berger and K. L. Milkman. *What makes online content viral?* Journal of Marketing Research, 49(2):192–205, 2012.

[2] M. Tsagkias, W. Weerkamp, and M. De Rijke. *Predicting the volume of comments on online news stories*. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, pages 1765–1768, 2009.

[3] R. Bandari, S. Asur, and B. a. Huberman. *The pulse of news in social media: Forecasting popularity*. In Proceedings of the 6th International Conference on Weblogs and Social Media, pages 26–33, 2012.

[4] I. Arapakis, B. B. Cambazoglu, and M. Lalmas. *On the feasibility of predicting news popularity at cold start*. In Proceedings of the 6th Internation Conference on Social Informatics, pages 290–299, 2014.

[5] T. Vanhove, G. Van Seghbroeck, T. Wauters, F. De Turck, B. Vermeulen, and P. Demeester. *Tengu: An experimentation platform for big data applications*. In Proceedings of the International Workshop on Computer and Networking Experimental Research Using Testbeds, pages 42–47, 2015.

[6] J. Deleu and A. D. Moor. *Named entity recognition on Flemish audio-visual and news-paper archives*. In Proceedings of the 12th Dutch-Belgian Information Retrieval Workshop, pages 38–41, 2012.

[7] M. Hall, H. National, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. *The WEKA data mining software : An update*. SIGKDD Explorations, 11(1), 2009.

6

Predicting the Popularity of Online News based on Temporal and Content-Related Features

In Chapter 5, we proposed a framework to monitor and analyze the consumption behavior of online news articles in real-time. In this chapter, we introduce novel methodologies to model and predict the popularity of online articles. This information can then be used by news agents to optimize their online publishing strategy. We first conduct a thorough analysis of the view patterns of online news, and their underlying distributions. We show that well-chosen basic functions lead to suitable models, and show how the influence of day versus night on the total view patterns can be taken into account to further increase the accuracy, without leading to more complex models. Second, we turn to the prediction of future popularity, given recently published content. By means of a new real-world dataset, we show that the combination of features related to content, meta-data, and the temporal behavior leads to significantly improved predictions, compared to existing approaches which only consider features based on the historical popularity of the considered articles.

⋆ ⋆ ⋆

S. Van Canneyt, P. Leroux, B. Dhoedt, T. Demeester
Submitted to Multimedia Tools and Applications, July 2016

6.1 Introduction

The online consumption of news content, a large and still growing market with respect to the traditional printed media, is undergoing major changes. The original paradigm of users consuming content that was pre-selected by news agents, shifts towards a setting where users themselves decide on which content is relevant to them and their circles, and whom they share it with over social media. As there is a strong competition between online publishers in order to reach the highest possible audience, it is becoming very important to decide which articles to promote on the front page of a news website, and which articles to publish on different social media platforms such as Twitter and Facebook. Therefore, in this chapter, we propose a novel methodology to model and predict the popularity of online news articles. These popularity models and predictions can then be used by news agents to optimize their online publishing strategy (which falls outside the scope of the current work).

We first conduct a thorough study to identify the distributions which underlie the view patterns of articles, i.e., the number of visits articles receive over time. This is important in order to understand how the popularity changes over time. This study is performed on the articles published by the Belgian Buzzfeed-like website newsmonkey[1] between April and September 2015. We observe that a view pattern in general consists of several components. The contribution that we refer to as the *direct views*, becomes visible as soon as the article is published on the news publisher's website. However, when the article is additionally published on social media channels, clear additional components in the view patterns start to appear. In this chapter, besides the direct views, we will focus on the *Facebook views* and the *Twitter views*. We introduce a model that closely fits these components and demonstrate that this model performs better than baseline log-normal fits [1–3]. Additionally, we take the influence of the diurnal cycle on the view patterns into account to further increase the accuracy, without obtaining more complex models.

As a second contribution, we propose a novel methodology to predict the final popularity of online news articles. As the total number of views consists of easily identifiable components related to the origin of the views (i.e., direct views, Facebook views, Twitter views), we train different regressors to respectively predict the behavior for each of these components. Existing approaches train linear regressors using features based on historical popularity values of the articles [4–7]. We investigate three ways to improve upon these baseline methods: (a) We explicitly make use of our proposed temporal model underlying the historical view pattern of the considered article, and use its parameters as additional features for the regressors. (b) In addition to using the historical popularity of the articles, we show that a variety of content-based and meta-data related features (such as author, category,

[1] http://newsmonkey.be

emotion, etc.) significantly contribute to improving the popularity predictions. (c) Finally, we show that more complex regression algorithms, as compared to the standard linear regression approach, can further improve the prediction effectiveness.

The remainder of this chapter is structured as follows. We start with a review of related work in Section 6.2. In Section 6.3 we describe the data acquisition process. Subsequently, in Section 6.4, we investigate the dynamics of the views received by articles, and propose a simple and effective model to model the view patterns. Our methodology to better predict the final popularity of articles using novel features and advanced regression algorithms is described in Section 6.5. Finally, we conclude our work in Section 6.6.

6.2 Related Work

The prediction of the popularity of online content has recently attracted a considerable amount of research. Some authors tackled the problem of predicting the popularity of an item before its publication [2, 8, 9]. Pre-publication predictions are particularly useful for web content characterized by a short lifespan such as online news articles. The researchers in [2, 8, 9] built classifiers to classify news articles into different classes, such as 'low popularity', 'medium popularity', and 'high popularity'. As quantitative indicators of popularity, they considered the number of comments on an article, the number of associated tweets, and the number of views. However, the researchers in [2] and [9] concluded that it is hard to accurately estimate an article's popularity without incorporating any early-stage popularity information. We did similar pre-publication experiments on our dataset which led to the same conclusion. Therefore, we will focus on post-publication predictions in this chapter.

Post-publication prediction methods predict an item's popularity based on the users' attention received early after publication. Kaltenbrunner et al. [1] analyzed the popularity of news articles, and found that the long term target popularity of online content is strongly correlated with its early reference popularity. Based on that observation, they proposed a linear popularity prediction model with the early popularity and a constant multiplication factor as input. The multiplication factor was set to the average growth in the training set. The authors of [4] improved that prediction model by optimizing the multiplication factor specifically for the considered performance metric. Their method showed good predictive performance on several data sets: votes on Digg stories [4], views of Youtube videos [4], views of blog posts [10], and comments on articles published on a French [3] and Dutch news platform [3, 11].

While the model of Szabo and Huberman [4] seems reasonably accurate, especially given its simplicity, it does have shortcomings. In particular, different

pieces of content may display a very similar popularity at an early stage, yet exhibit a diverse popularity behavior afterwards. In other words, despite the observations in [1], online content may experience very different popularity evolution patterns [5, 6]. Therefore, the authors of [5, 12] investigated whether the use of the historical popularity values of online content between the publication time and an early reference time leads to more accurate predictions of the total popularity at a future target time. Pinto et al. [5] divided the time between publication of the article and the reference time into different intervals, and trained a linear regression model using the number of observed views the articles received during each time interval. This model was further improved by incorporating features constructed from the similarities between the considered view pattern and the training instances. The model proposed in [12] used the retweet pattern of a tweet during its first hour to predict the number of retweets three days after publication. The authors partitioned the first hour into five equally sized time intervals, and then recorded the number of retweets during each time interval. This information was used to describe each tweet by a set of features (such as retweet time series, retweet acceleration, and author). These features were used to determine the most similar tweets in the training set of the given tweet. The predicted popularity was then set to the weighted average of the number of retweets among these similar tweets.

The last category of post-publication prediction methods uses data from one domain (e.g. social media) and transforms it into knowledge to predict content popularity in another domain (e.g. the site where the content was published). Oghina et al. [13] trained a linear regression model based on several textual features extracted from Twitter, as well as various statistics from Youtube, to predict movie ratings on IMDb. The authors of [6] proposed a second-order multiple linear regression model to predict the number of views of online news articles after 7 days. For a given reference time, the model used the total number of views, Facebook shares and Twitter posts of the article, in addition to Twitter statistics such as the average number of followers of people sharing on Twitter and the entropy of the tweets.

The objective of the ECML/PKDD 2014 Predictive Analytics challenge[2] was to predict the number of views, Facebook shares, and Twitter posts of web pages after their first 48 hours online. As input, the popularity trends during the first hour were given. The winner [7] of the challenge combined different ideas of the models proposed in [6] and [5]. Similar to [6], they used second-order multiple linear regression models based on several popularity metrics to predict the number of views. For the given reference time (i.e., one hour after publication), the model considered the number of views, Facebook shares and Twitter posts per time interval (i.e., 5 minutes), starting from the publication time until the reference time. Additional features were formed using the publication weekday and hour of the

[2]https://sites.google.com/site/predictivechallenge2014/

article. The authors of [7] further improved their model by using the ideas presented in [5]. In particular, they also used the similarity of the view pattern to canonical patterns extracted from the training set, in order to improve the model performance. These canonical patterns were constructed by normalizing and clustering all view patterns in the training set.

Our proposed method differs from these post-publication approaches in multiple aspects. We explicitly model the temporal behavior underlying the historical popularity of the articles, and use the resulting parameters as additional features for the regressors. We also consider features related to the content and meta-data of the articles. Finally, we propose the use of a more advanced regression algorithm.

6.3 Experimental Data

In this study we use data from the newsmonkey online news platform. Newsmonkey is a Buzzfeed-like news website, currently focusing on the Belgian market. Similar to Buzzfeed, newsmonkey combines breaking news with highly shareable stories. Our dataset consists of 2614 articles and the detailed associated click data, which we collected between April 27, 2015 and September 10, 2015. The first three quarters (until July 25, 2015) are used as a training set K, and the last quarter is considered as the test set U (with content from August 6, 2015 onwards, in order to limit the immediate correlation between both). An article's final popularity is measured as the number of views 120 hours (5 days) after publication on the website. The boxplot of the total number of views received per day after publication is plotted in Figure 6.1. We observed that most articles have a lifetime considerably shorter than this 5 day period, such that the number of views becomes stable well before 5 days after their initial publication. The average number of views per article is about 2000, of which 62% directly come from Facebook and only 3% from Twitter. This is in line with the strategy of newsmonkey, mainly focusing on optimizing their popularity on Facebook.

Few articles reach a very high number of views, whereas the majority of articles only get a low reach. As an illustration, Figure 6.2 shows the number of views for all articles as a function of the rank of each article, when sorted by decreasing number of views. Except for the least popular articles, the observed behavior is approximately linear on logarithmic axes. This Zipfian behavior means that the number of views per article follows a power law.

Figure 6.3 shows the normalized popularity for the articles in the training set as the fraction of views for each hour of the day, considered separately for the views originating from Facebook (Facebook views), from Twitter (Twitter views), and all remaining views (direct views). We notice that the users are much less active at night than during the day. Also, there is some difference in behavior between the three considered types of views.

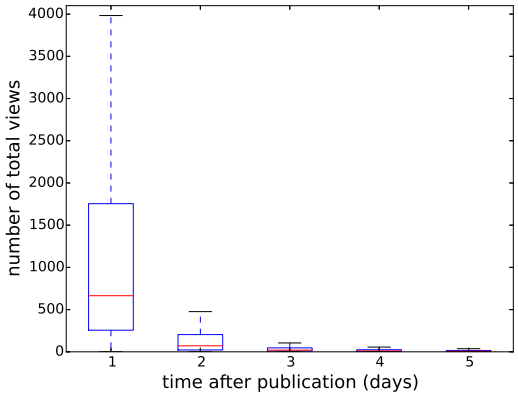


Figure 6.1: Boxplot of number of total views received per day after publication, for all articles in the dataset.

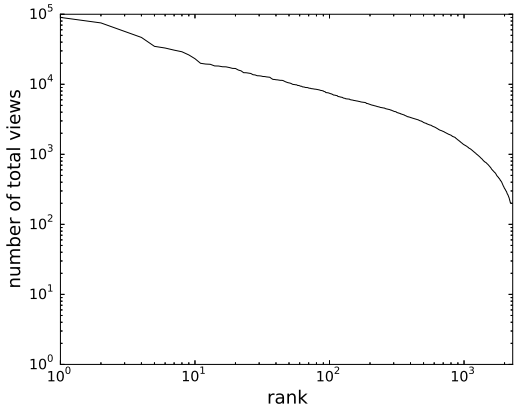


Figure 6.2: Zipfian distribution of the number of views for all articles in the dataset, ranked in decreasing order.

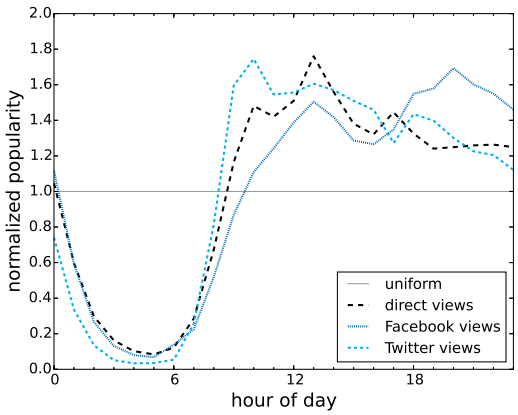


Figure 6.3: Normalized number of direct views, Facebook views, and Twitter views for each hour of the day, averaged over the articles in the training set.

6.4 Popularity Pattern Modeling

A good understanding of how the popularity changes over time and which external elements have the largest impact, is essential in order to create a suitable model, or to design appropriate features for popularity predictions. Therefore in this section we propose a new temporal popularity model. Despite its simplicity, we show that this model is able to accurately capture the temporal behavior of a particular popularity measure for a given article, and compare it with a number of existing models. In this chapter, we consider the evolution of the total number of views of each article, measured at hourly intervals. However, the methodology can be easily extended towards other popularity metrics (e.g. Facebook shares) and more fine-grained time intervals. As an illustration throughout this section, we will use the total number of views of a typical article, shown in Figure 6.4. This particular article was published on Twitter immediately after its publication online, and on Facebook 25 hours later. In Section 6.5, a prediction model is introduced that makes use of the insights obtained from the proposed temporal model and explicitly uses its parameters as features.

6.4.1 Log-normal Baseline

In previous work, the popularity of online news articles is often modeled using a log-normal distribution [1–3, 11]. In particular, its cumulative distribution can be used to model the total number of views at a particular time:

$$v_t^i \approx s^i \cdot \text{clogn}(t; \mu^i, \sigma^i) \tag{6.1}$$

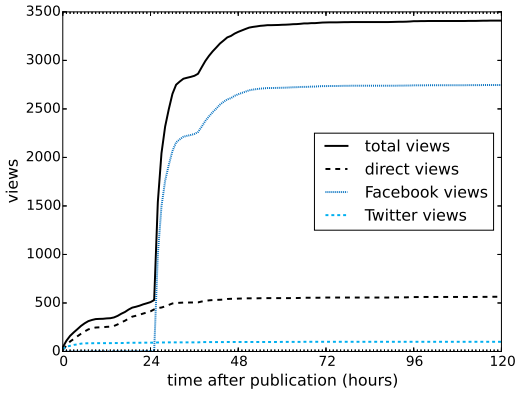
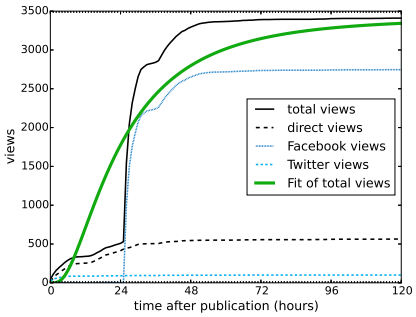
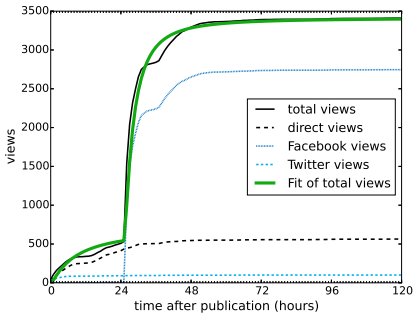


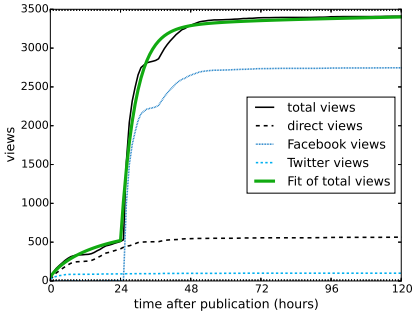
Figure 6.4: Number of views of an example article as a function of time.



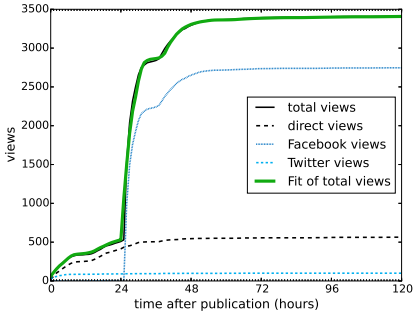
(a) Log-normal fit of total views, Eq. 6.1. Total views RRSE=0.342.



(b) Log-normal fit per component, Eq. 6.3. Total views RRSE=0.077.



(c) LinExp fit per component, Eq. 6.6. Total views RRSE=0.067.



(d) Transformed-time fit per component. Total views RRSE=0.045.

Figure 6.5: Example curve fit with different models, for the example in Figure 6.4, with indication of the root relative squared error (RRSE) of the total views fit.

with v_t^i the observed total number of views of article i at time t, s the scale factor that corresponds to the number of views at infinity, and clogn the cumulative log-normal distribution given by

$$\mathrm{clogn}(t;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^t e^{-\frac{(\ln(\xi)-\mu)^2}{2\sigma^2}} d\xi \tag{6.2}$$

with μ and σ the parameters of the distribution. The log-normal fit of the view pattern of Figure 6.4 is shown in Figure 6.5a.

The authors of [1, 3] used user comments as popularity metric. However, the number of view patterns may be more complex. As can be seen in Figure 6.4, the curve of the total number of views consists of multiple components which do not necessarily start at the publication time. As could be anticipated, it appears to be a good approximation to assume that the direct views start to arrive at the moment the article is published on the website ($t = 0$), the Facebook views at the moment the article is published on Facebook, and the Twitter views at the moment it is posted on Twitter. Since we can measure which views originate from Facebook, Twitter, or from elsewhere, we can explicitly model these different components. A better model for the total number of views therefore consists of the sum of the separate log-normal fits of the different components:

$$\begin{aligned} v_t^i \approx\ & s_d^i \cdot \mathrm{clogn}_d(t;\mu_d^i,\sigma_d^i) \\ & + b_F \cdot s_F^i \cdot \mathrm{clogn}_F(t-t_F;\mu_F^i,\sigma_F^i) \\ & + b_T \cdot s_T^i \cdot \mathrm{clogn}_T(t-t_T;\mu_T^i,\sigma_T^i) \end{aligned} \tag{6.3}$$

with $\mathrm{clogn}_d(\cdot)$, $\mathrm{clogn}_F(\cdot)$ and $\mathrm{clogn}_T(\cdot)$ the log-normal distribution associated with respectively the direct views, Facebook views, and Twitter views as defined in Equation 6.1. Parameter b_F (resp. b_T) is a known binary parameter that indicates whether the article is published on Facebook (resp. Twitter). Parameter t_F is the number of time units after the original publication ($t = 0$) that the article is posted on Facebook, and similarly for t_T on Twitter. Strictly speaking, $\mathrm{clogn}(t;\mu,\sigma)$ is not defined for $t < 0$, but in Equation 6.3 we simple assume the contributions from the Facebook and Twitter components to be zero before their respective publication moments. The fit of the example article of Figure 6.4 according to this strategy can been seen in Figure 6.5b.

6.4.2 Linear-Exponential Popularity Model (LinExp)

In this section, we investigate alternative models, accurately capturing the observed behavior, preferably having parameters that are intuitively interpretable. When inspecting the data, measured by the hour, we rarely observed the typical log-normal behavior of an initial slow uptake, which increases and then again

slows down towards the asymptotic value. We noticed that most often there simply is an initial uptake speed, that immediately starts to relax in a gradual way. Also, sometimes we noticed a small and constant uptake, independent from the large initial uptake directly after publication. A simple model for the uptake speed ν (or the number of views per time unit) corresponding to these observations and starting at time $t = 0$ is

$$\nu(t; c_1, c_2, T) = \frac{c_1}{T} e^{-\frac{t}{T}} + c_2 \tag{6.4}$$

The first term of the right-hand side represents an exponential relaxation that reflects the gradual decrease of the uptake speed. The second term is the small constant uptake that sometimes becomes visible. It can be explained intuitively by assuming a small constant chance that a random user clicks the considered article, e.g. when browsing the news site, and which is independent of the article's publication time.

By integrating Equation 6.4 up to the current time t, the cumulative behavior becomes

$$V(t; c_1, c_2, T) = c_1(1 - e^{-\frac{t}{T}}) + c_2 t. \tag{6.5}$$

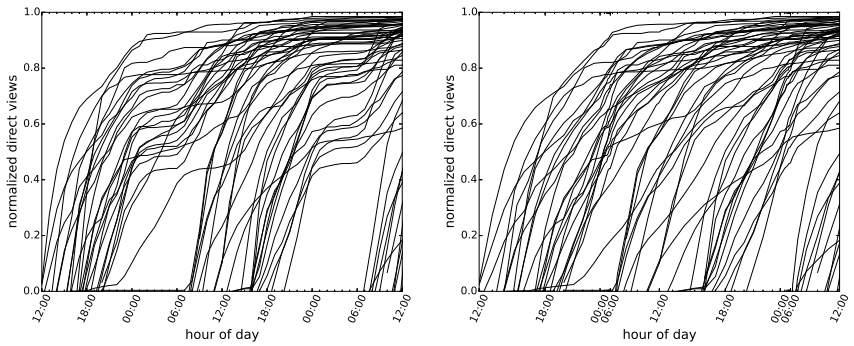
The total number of views v_t^i for article i at time t can thus be modeled by adding different components of this form, for direct views, Facebook views, and Twitter views.

$$\begin{aligned} v_t^i \approx\ & V_d(t; c_{1,d}^i, c_{2,d}^i, T_d^i) \\ & + b_F \cdot V_F(t - t_F; c_{1,F}^i, c_{2,F}^i, T_F^i) \\ & + b_T \cdot V_T(t - t_T; c_{1,T}^i, c_{2,T}^i, T_T^i) \end{aligned} \tag{6.6}$$

We will call this the *Linear-Exponential model*, abbreviated as the *LinExp model*. The fit of the example article of Figure 6.4 according to this proposed popularity model can be seen in Figure 6.5c.

6.4.3 Time Transformation

As can be seen in Figure 6.4, the number of visits retrieved between 7 and 14 hours and between 31 and 38 hours after publication is almost zero. This corresponds more or less to the period between 1 am and 8 am. As most people in the target audience sleep during that period, the articles do not retrieve a lot of additional visits and the view pattern also 'sleeps'. This is reflected by the average number of views per hour of the day or night as shown in Figure 6.3 for direct views, Facebook views, and Twitter views. We would like to include this behavior into the model, without adding more degrees of freedom than necessary. In order to give a better qualitative idea of the problem, in Figure 6.6a we show all direct views for two randomly chosen consecutive days (June 10-12, 2015), as a function



(a) Uniform hours *(b) Normalized hours*

Figure 6.6: Number of direct views in function of time for articles published between June 10, 2015 12:00 and June 12, 2015 12:00.

of time. The x-axis denotes the time starting on June 10, 2015 at noon, up to 2 days later. Along the y-axis, the direct views are shown, normalized for convenience by their stabilized value after 5 days. We clearly see that the night has a similar effect on most articles. This effect is stronger with later publication times. Also, during the second night that articles have been online, this effect is less pronounced but still present.

There are several ways to model this effect. Directly replacing the functions from Equation 6.5 by a more complex mathematical expression that depends on the publication time and models the observed behavior, would come with additional parameters and lead to a more complex model. This can be avoided by noticing that the day/night effect seems to be article-independent. We therefore propose the following heuristic: we replace each uniform time interval by the corresponding normalized value of the average number of reads during that hour, i.e., the values shown in Figure 6.3 for the respective components. As a result, nightly hours have a shorter normalized duration or effectively go faster, whereas during the day the effective time goes slower than on average. By applying this time transformation, the average number of reads per unit of normalized time would become uniform throughout the day. If indeed this time effect is completely article independent, we can expect that the day/night effect in the individual article view patterns disappears as well. Figure 6.6b shows the same view patterns as Figure 6.6a, but with the transformed time axis, and we can conclude qualitatively that the day/night effect is no longer clearly visible. Note that the time transformation needs to be applied for each of the components (direct, Facebook, Twitter) separately, as they are subject to a different reading behavior as already shown in Figure 6.3.

The proposed time transformation seems to be a suitable heuristic, and has an important advantage: we need to calculate the transformation only once per

component type (direct views, Facebook, or Twitter), after which we can apply the original model of Equation 6.5 on the transformed time axis, without adding any model parameters. Even more, this transformation could be adapted to the day of week or weekend, or to the seasons, just by suitably averaging the number of views per hour. While evaluating the model, the inverse transformation needs to be made. For example, the predicted popularity at transformed time $\tilde{t}$ corresponds to the predicted popularity at the actual time t, in which $\tilde{t}$ was obtained by transforming t as described above. The transformed-time fit of the example article of Figure 6.4 can be seen in Figure 6.5d.

6.4.4 Parameter Estimation

For the log-normal baseline, the parameters s, μ, and σ are estimated using maximum likelihood estimation (MLE), as described in [14].

For the LinExp popularity model of Section 6.4.2, the parameters are also estimated with MLE. More in particular, with $\boldsymbol{c} := [c_1, c_2]^\top$ and $\boldsymbol{\phi}(t) := [(1 - e^{-\frac{t}{T}}), t]^\top$, we can write $V(t; \boldsymbol{c}, T) = \boldsymbol{c}^\top \boldsymbol{\phi}(t)$. Note that in line with Section 6.4.3, t denotes the transformed time with respect to the start of the considered component.

Minimizing the sum of squared errors, or equivalently, maximizing the likelihood under the assumption of additive Gaussian noise, leads to the following estimate $\hat{\boldsymbol{c}}$ for the coefficients:

$$\hat{\boldsymbol{c}} = \Big(\sum_t \boldsymbol{\phi}_t \boldsymbol{\phi}_t^\top \Big)^{-1} \Big(\sum_t v_t \boldsymbol{\phi}_t \Big) \tag{6.7}$$

in which v_t denotes the observed popularity value at time t, and we shortly write $\boldsymbol{\phi}_t := \boldsymbol{\phi}(t)$. A detailed treatment of this linear regression problem is given in [15].

The time constant T is also an unknown parameter. It can be determined by applying the expectation maximization (EM) algorithm, in which the expectation step, given by Equation 6.7, is followed iteratively by the maximization step

$$T = \text{argmax}_T \Big(- \sum_t (v_t - \hat{\boldsymbol{c}}^\top \boldsymbol{\phi}_t)^2 \Big) \tag{6.8}$$

6.4.5 Evaluation

To evaluate the quality of the curve fitting for article i, we use the root relative squared error (RRSE):

$$RRSE_i = \sqrt{\frac{\sum_t (\hat{v}_t^i - v_t^i)^2}{\sum_t (\overline{v}^i - v_t^i)^2}} \tag{6.9}$$

with v_t^i the observed number of total views at time t of article i, and $\hat{v}_t^i$ the value approximated by the model. We denote the average of the v_t^i observations for the

Table 6.1: MRRSE of the temporal popularity models. All differences between models are significant ($p < 0.001$).

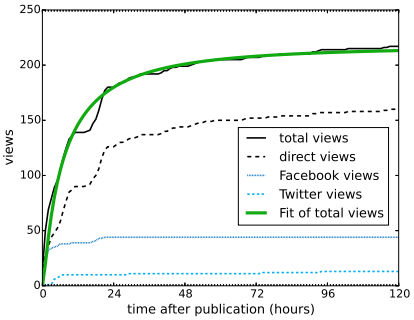
model	MRRSE
log-normal fit, Eq. 6.1	0.233
log-normal fit per component, Eq. 6.3	0.211
linexp fit per component, Eq. 6.6	0.151
transformed-time fit per component	0.124

considered article as $\overline{v}^i$. For our experiments we have an hourly observation of the total views, starting from the moment of publication, up to 5 days (120 hours) later, or $t = 0, \ldots, 120$. The RRSE is calculated over the articles in the training set K and its mean value (written MRRSE) is used to evaluate the different models:

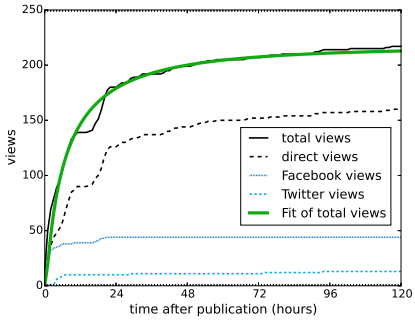
$$MRRSE = \frac{1}{|K|} \sum_{i=1}^{|K|} RRSE_i \tag{6.10}$$

The MRRSE values for the considered temporal popularity models are shown in Table 6.1. We notice that the curve fitting performance is improved by explicitly modeling the different components (Equation 6.3) instead of directly fitting the total number of views (Equation 6.1). The proposed LinExp model as defined in Equation 6.6 leads to further improvements. We can hence conclude that the functions in Equation 6.5 better describe the separate components than the log-normal model. The time transformation leads to a further decrease in the average error. All mentioned improvements appeared significant up to the level $p = 0.001$, using a one-sided bootstrap significance test [16].

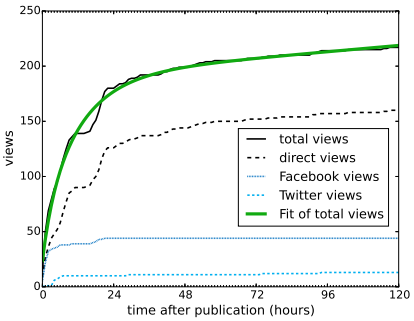
Figure 6.5 clearly shows the added value of modeling the direct views, Twitter views and Facebook views separately instead of directly fitting the total number of views (Figure 6.5a vs. 6.5b). The error further decreased when using our proposed model (Figure 6.5c). However, the error reduction is small because the popularity totally stagnates after two days, leading to a small linear component $c_2 t$ in Equation 5. Finally, the use of the time transformation leads to further improvements (Figure 6.5d). Figure 6.7 provides another visual illustration. It shows the various model fits for a less popular article as compared to Figure 6.4, with indication of the RRSE. In this example, the article was published on Facebook and Twitter together with its initial online publication, such that modeling all three components separately does not contribute much with respect to directly modeling the total number of views (Figure 6.7a vs. 6.7b). However, we notice that while the Twitter and Facebook views have become stable after one day, there is a slight linear increase of the direct views, which continues during the subsequent days. The linear component $c_2 t$ in Equation 6.5, which we introduced as a constant (i.e., publication time independent) rate of users browsing to the article, accurately models that



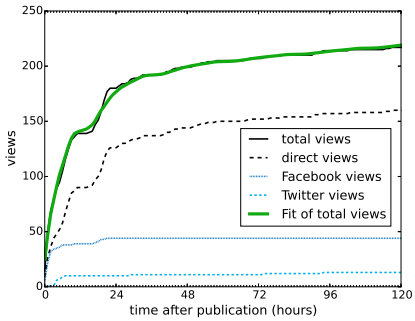
(a) Log-normal fit of total views, Eq. 6.1. Total views RRSE=0.160.



(b) Log-normal fit per component, Eq. 6.3. Total views RRSE=0.156.



(c) LinExp fit per component, Eq. 6.6. Total views RRSE=0.101.



(d) Transformed-time fit per component. Total views RRSE=0.065.

Figure 6.7: Another example curve fit with different models, with indication of the root relative squared error (RRSE) of the total views fit.

behavior. This leads to a lower error, as seen from Figure 6.7b with the log-normal model vs. Figure 6.7c with the proposed LinExp popularity model of Equation 6.6. The time-transformed model as described in Section 6.4.3 further reduces the error, confirming the added value of taking into account the variation in popularity throughout the day.

6.5 Popularity Prediction

In this section, we show how the total popularity of news articles can be predicted. An important insight from the previous section, is the need to model the different components separately, which we follow for the prediction task as well. In particular, we train three different regressors to respectively predict the direct views, Facebook views, and Twitter views. The articles' final popularity is measured for each of these components, at target time τ after publication on the website, on Facebook, or on Twitter, respectively. The objective is thus to predict for each article at a particular *reference time* r its final popularity at a future point in time, which we will refer to as the *target time* τ (with $0 \leq r \leq \tau$). Note that we use the general term 'views' to indicate any of the previously introduced popularity metrics. It may refer to direct views, Facebook views, or Twitter views, but also to other popularity metrics like Facebook shares, which we do not explicitly treat in this chapter.

In Section 6.5.1, we give an extensive overview of existing approaches, which we implemented as baseline methods. Our proposed prediction methodology is described in Section 6.5.2. Finally, we evaluate the baselines and the proposed methodology in Section 6.5.3.

6.5.1 Baselines

This section provides an overview of baseline methods based on linear regression models. Similar to the approaches described in [4, 6, 7] we log-transform the popularity values as there is a better correlation between the log-transformed popularities at reference time r and target time τ than between the untransformed popularities. The regression model thus takes the form

$$log(1 + \hat{\boldsymbol{v}}_\tau) = log(1 + \boldsymbol{X}_r)\boldsymbol{\beta} \tag{6.11}$$

in which we assume a component-wise logarithmic transformation of the vector $\hat{\boldsymbol{v}}_\tau$ of predicted views at target time τ for the considered articles, and of the article features in matrix $\boldsymbol{X}_r$ constructed at reference time r. Each row $\boldsymbol{x}_r^i$ in matrix $\boldsymbol{X}_r$ corresponds to the vector of feature values of article i and

$$log(1 + \hat{v}_\tau^i) = log(1 + \boldsymbol{x}_r^i)\boldsymbol{\beta} \tag{6.12}$$

with vector $\hat{v}_\tau^i$ the predicted number of views at target time τ of article i. The parameters $\boldsymbol{\beta}$ are estimated using ordinary least squares on the training set K:

$$\boldsymbol{\beta} = \text{argmin}_{\boldsymbol{\beta}} ||log(1 + \boldsymbol{v}_\tau) - log(1 + \boldsymbol{X}_r)\boldsymbol{\beta}||_2^2 \tag{6.13}$$

with $\boldsymbol{v}_\tau$ the observed views at target time τ and $|| \cdot ||_2^2$ the squared L_2-norm. The goal of this objective function is to minimize the sum of the squared errors on the log-transformed data. We consider several baseline methods that are based on this linear regression model and describe them in the following paragraphs.

Szabo and Huberman model (SH model) The simplest model, introduced by Szabo and Huberman [4], only considers the number of visits measured at reference time r,

$$\boldsymbol{x}_r^i = [v_r^i] \tag{6.14}$$

with v_r^i the number of visits for article i at reference time r.

Multivariate Linear model (ML model) Pinto et al. [5] extended the SH model by considering the whole history of the number of visits, or

$$\boldsymbol{x}_r^i = [v_1^i, v_2^i \dots v_r^i] \tag{6.15}$$

with v_t^i the number of visits for article i, observed t time units after publication.

Radial Basis Functions model (RBF model) The authors of [5] extended their ML model by indirectly incorporating the different possible popularity patterns. In particular, they proposed to take into account the similarity in terms of early popularity between the article and n randomly selected examples from the training set, called subset S. Gaussian Radial Basis Functions (RBF) were used for measuring the similarity between articles i and $a \in S$:

$$RBF_a(i) = e^{-\frac{||\boldsymbol{x}_r^i - \boldsymbol{x}_r^a||_2^2}{2 \cdot \sigma^2}} \tag{6.16}$$

with $\boldsymbol{x}_r^i$ the ML feature vector as defined in Equation 6.15, and parameter $\sigma > 0$. Equation 6.12 can then be rewritten as

$$log(1 + \hat{v}_\tau^i) = log(1 + \boldsymbol{x}_r^i)\boldsymbol{\beta} + \sum_{a \in S} w_a \cdot RBF_a(i) \tag{6.17}$$

with $\boldsymbol{x}_r^i$ as defined in Equation 6.15. The ML model and RBF model were originally optimized and evaluated using the mean relative squared error, instead of the sum of the squared logarithmic errors as used in this chapter.

First-Order Social Media model (FOSM model) The fourth baseline is based on the model introduced by Castillo et al. [6]. The authors proposed a multiple linear regression model which uses the number of visits at reference time, together with metrics retrieved from social media. The first-order model is given by Equation 6.12, whereby

$$\boldsymbol{x}_r^i = [v_r^i,\ v_{r,F}^i,\ v_{r,T}^i,\ m_{r,F}^i,\ m_{r,T}^i] \tag{6.18}$$

with $v^i_{r,F}$ the number of views originating from Facebook article i received at reference time r, $v^i_{r,T}$ the number of article i views originating from Twitter at reference time r, and $m^i_{r,F}$ and $m^i_{r,T}$ the number of respectively Facebook shares and tweets related to article i at time r. To be precise, the original model described in [6] does not consider Facebook or Twitter views. Instead, their model includes the number of visits from link referrals, direct traffic from e-mail, and some Twitter statistics such as the entropy of the tweets and number of unique tweets. However, since these features are not available in our dataset, we replace them by the features listed in Equation 6.18.

Second-Order Social Media model (SOSM model) The paper of [6] also describes a second-order variant of their first-order social media model. In addition to the first-order features described in Equation 6.18, they also include the second-order interactions of these features. These features are included to model the interdependency of the variables.

Mixed model Our last baselines are based on the models proposed by Figueiredo et al. [7], winner of the ECML/PKDD 2014 Predictive Analytics Challenge. Their models are based on the ideas of the RBF and SOSM model. The first model considers both the whole history of the popularity metric values and the metrics retrieved from social media. The vector representing the whole history of the number of visits is defined as

$$\boldsymbol{v}^i_r = [v^i_1, v^i_2 \dots v^i_r] \tag{6.19}$$

Similarly, the history of Facebook views, Twitter views, Facebook shares and Twitter posts are represented by vectors $\boldsymbol{v}^i_{r,F}$, $\boldsymbol{v}^i_{r,T}$, $\boldsymbol{m}^i_{r,F}$, $\boldsymbol{m}^i_{r,T}$, respectively. The binary vector $\boldsymbol{d}^i$ is a one-hot feature vector to represent the week day, and similarly, $\boldsymbol{h}^i$ represents the publication hour of article i. The feature vector $\boldsymbol{x}^i_r$ representing article i in Equation 6.12 is then constructed by concatenating $\boldsymbol{d}^i$, $\boldsymbol{h}^i$, $\boldsymbol{v}^i_r$, $\boldsymbol{v}^i_{r,F}$, $\boldsymbol{v}^i_{r,T}$, $\boldsymbol{m}^i_{r,F}$, and $\boldsymbol{m}^i_{r,T}$, and all of their pairwise interactions, represented by the elementwise products. Again, the original model does not consider the Facebook views and Twitter views. It considers the time series of the average time each user spends on the page, which we have to leave out as it is unavailable in our dataset.

Mixed-Trend model Similar to the RBF model, the authors of [7] extended their Mixed model by indirectly incorporating the different possible popularity patterns. In particular, they proposed to take into account the similarity in terms of early popularity between the article and k cluster centers. The early popularity of article i can be represented by the vector

$$\boldsymbol{p}^i_r = [log(1+\delta^i_1), log(1+\delta^i_2) \dots log(1+\delta^i_r)]^\top \tag{6.20}$$

with δ^i_t the number of visits gained in time interval t, i.e. $\delta^i_t = v^i_t - v^i_{t-1}$. The similarity between two articles a and i is then quantified using the euclidean distance:

$$dist_a(i) = ||\overline{\boldsymbol{p}}^i_r - \overline{\boldsymbol{p}}^a_r||_2 \tag{6.21}$$

with $\overline{\boldsymbol{p}}_r^i$ the z-normalized vector of $\boldsymbol{p}_r^i$. This distance function is used to determine k cluster centers (set C) using the k-means algorithm on the training set. Equation 6.12 can then be modified to

$$log(1 + \hat{v}_\tau^i) = log(1 + \boldsymbol{x}_r^i)\boldsymbol{\beta} + \sum_{a \in C} w_a \cdot dist_a(i). \tag{6.22}$$

6.5.2 Proposed Methodology and Features

We will evaluate the popularity predictions based on five different models, besides the baselines described above. These five proposed models differ in terms of the considered regression algorithm, and the different types of included features. The features, discussed below, are listed in Table 6.2. For the models, we distinguish between a linear regression model (similar to the baselines) and the gradient tree boosting (GTB) algorithm. The latter is often used in winning methodologies for Kaggle competitions[3], because it can handle non-linearities in the data and interactions between the features. We use the regression implementations available in the Python scikit-learn package.[4] The models can be characterized as follows

- **LM history**: linear regression model, based on the 'history' features described in Table 6.2,
- **LM history+curve**: linear regression model, based on the 'history' and 'curve' features described in Table 6.2,
- **RIDGE history+curve**: linear regression model with L2 regularization (ridge regression, $\alpha = 1.0$), based on the 'history' and 'curve' features,
- **GTB history+curve**: GTB regression, with the 'history' and 'curve' features,
- **GTB all**: GTB regression, with all described features.

We provide a short description for each feature groups listed in Table 6.2:

History These features capture the popularity pattern of the article. Similar to [6], we use the popularity expressed by other metrics (e.g., Facebook views and Twitter views) to better predict the considered popularity metric (e.g., direct views). In particular, the total views, direct views, Facebook views, Twitter views, and Facebook shares are considered. Similar to the prediction method described in Section 6.5.1, all popularity values are log-transformed.

Curve Features We incorporate our knowledge of the distributions which underlie the article popularity pattern, as discussed in Section 6.4. In particular, we estimate the parameters of Equation 6.5 for the known historical popularity values

[3] https://www.kaggle.com/
[4] http://scikit-learn.org

domain	*name*	*description*
History	views	number of views for article a at reference time r, i.e., v_r^a
	viewsHistory	$\forall h \in [1,5]$ number of views for article a received between reference time r and h hours earlier, i.e., $v_r^a - v_{r-h}^a$
	directViews	number of direct views for article a at reference time r, i.e., $v_{r,d}^a$
	directViewsHistory	$\forall h \in [1,5]$ number of direct views for article a received between reference time r and h hours earlier, i.e., $v_{r,d}^a - v_{r-h,d}^a$
	facebookViews	number of Facebook views for article a at reference time r, i.e., $v_{r,F}^a$
	facebookViewsHistory	$\forall h \in [1,5]$ number of Facebook views for article a received between reference time r and h hours earlier, i.e., $v_{r,F}^a - v_{r-h,F}^a$
	twitterViews	number of Twitter views for article i at reference time r, i.e., $v_{r,T}^a$
	twitterViewsHistory	$\forall h \in [1,5]$ number of Twitter views for article a received between reference time r and h hours earlier, i.e., $v_{r,T}^a - v_{r-h,T}^a$
	facebookShares	number of Facebook shares for article i at reference time r, i.e., $m_{r,F}^a$
	facebookSharesHistory	$\forall h \in [1,5]$ number of Facebook shares for article a received between reference time r and h hours earlier, i.e., $m_{r,F}^a - m_{r-h,F}^a$
Curve Features	relaxation amplitude	parameter c_1 in Equation 6.4, after curve fitting of Equation 6.4 on the view pattern between hour 0 and reference time r
	linear amplitude	parameter c_2 in Equation 6.4, after curve fitting of Equation 6.4 on the view pattern between hour 0 and reference time r
	relaxation constant	parameter T in Equation 6.4, after curve fitting of Equation 6.4 on the view pattern between hour 0 and reference time r
Author	authorAverage	average popularity of articles in the training set published by the author of a
	authorStd	standard deviation of the popularity of the articles in the training set published by the author of a
	authorCount	number articles in the training set published by the author of a
	authorBinary	binary vector representing the author of article a
Category	categoryAverage	average popularity of articles in the training set with the same category as a
	categoryStd	standard deviation of the popularity of the articles in the training set with the same category as a
	categoryCount	number of articles in the training set with the same category as a
	categoryBinary	binary vector representing the category of article a
Publication Time and Date	hourOfDayAverage	average popularity of the articles in the training set published during the same hour of day as a
	hourOfDayStd	standard deviation of the popularity of the articles in the training set published during the same hour of day as a
	hourOfDayCount	number of the articles in the training set published during the same hour of day as a
	hourOfDayBinary	binary feature indicating the hour of day the article is published
	dayOfWeekAverage	average popularity of the articles in the training set published during the same day of week as a
	dayOfWeekStd	standard deviation of the popularity of the articles in the training set published during the same day of week as a
	dayOfWeekCount	number of the articles in the training set published during the same day of week as a
	dayOfWeekBinary	binary feature indicating the day of week the article is published
Title	numberInTitle	binary feature indicating if the title of a contains a number
	entityInTitle	binary vector indicating the type of the named entity in the title of a (if present)
Source Article	hasSourceArticle	binary feature indicating if a has a source article
	sourceArticleShares	number of Facebook shares the source article of a received at publication time of a
Virality	willGoViral	binary feature indicating if objective of the article is to go viral on Facebook
Target Audience	genderAverage	average popularity of articles in the training set with the same target gender as a
	genderStd	standard deviation of the popularity of the articles in the training set with the same target gender as a
	genderCount	number of articles in the training set with the same target gender as a
	genderBinary	binary vector representing the target gender of article a
	ageAverage	average popularity of articles in the training set with the same target age as a
	ageStd	standard deviation of the popularity of the articles in the training set with the same target age as a
	ageCount	number of articles in the training set with the same target age as a
	ageBinary	binary vector representing the target age of article a
Emotion	emotionAverage	average popularity of articles in the training set with the same target share emotion as a
	emotionStd	standard deviation of the popularity of the articles in the training set with the same target share emotion as a
	emotionCount	number of articles in the training set with the same target share emotion as a
	emotionBinary	binary vector representing the target share emotion of article a

Table 6.2: Features considered in this chapter for training regressors to predict the popularity of article a.

as described in Section 6.4.4. These parameters are then used as features for the regression model.

Author We include the average popularity of the articles in the training set published by the same author of the considered article, its standard deviation, and also the number of training articles by that author. In addition, one-hot feature vectors are used to indicate the specific author.

Category The journalists of newsmonkey manually labeled all articles with one or more category from a set of 16 categories (society, politics, tv, music, life and style, cyberspace, tech and gadgets, planet, travel, movies, starts, economy, body and soul, science, pets, and games). Similar to the author features, we represent the categories of the article by an average popularity, standard deviation, number of articles, and a binary vector to indicate the categories.

Publication Time and Date Similar to [7], we include the publication hour and week day as features.

Title We determine whether the title of the considered article contains a number (binary feature). Articles containing a number in their title are mostly articles containing lists, and their title often starts with a phrase like 'n reasons why ...', with n a number. These 'list' articles are constructed with the main objective to be very shareable on Facebook, which makes the described feature very informative. Additionally, we use named entity recognition [17] to extract the named entities and their type from the title. The possible entity types, for which binary features are introduced, are organization, location, person, or miscellaneous.

Source Article With a binary feature, we indicate whether or not the article refers to a source article, i.e., an article from another news website which is cited by the article (for example from Business Insider or Mashable). We also include a feature with the number of Facebook shares the source article already received at publication time of the considered article.

Virality The journalists of newsmonkey annotated some articles with labels reflecting their experience on which articles will go viral on Facebook and why. In particular, they manually labeled their articles as 'will go viral' if they estimated that the article would be very popular on Facebook.

Target Audience For the articles labeled as 'will go viral', the authors also indicated the target audience. The target audience is given by the target gender (female, male, or both) and target age range (18-24, 25-34, or 18-34 years old).

Emotion In addition to the target audience, the authors labeled 'will go viral' articles with an emotion label, which is included as a binary feature vector. The annotated emotion label of a particular article is not the direct emotion it is expected to provoke in the readers, as considered in previous research [9, 18]. Instead, it is the emotion of why users are expected to *share* the article. The considered emotion labels are recognizability, identity, awe, humor, pride, malicious pleasure, altruism, taboo, outrage, nostalgia, and softening.

6.5.3 Evaluation

In this section, we evaluate the proposed prediction methodologies. The models are trained using training set K and evaluated on the articles in a separate test set U. The parameters of the models are optimized using 5-fold cross-validation on the training set. As evaluation metric indicating the performance of the predictions, we use the root mean squared log error (RMSLE):

$$RMSLE = \sqrt{\frac{1}{|U|}\sum_{i \in U}(log(\hat{v}_\tau^i + 1) - log(v_\tau^i + 1))^2} \tag{6.23}$$

with v_τ^i the observed number of views of article i at target time τ, and $\hat{v}_\tau^i$ the predicted number of views. In other words, the RMSLE indicates how well the popularity at target time τ is predicted for all articles in the test set. This evaluation metric is also used in the ECML/PKDD 2014 Predictive Analytics challenge[5]. To determine whether the difference in performance of two methods is statistical significant, we use the unpaired bootstrap hypothesis test [16]. We consider the predictions with a target time τ of 5 days (120 hours) after publication of the article, and reference time r one to 24 hours after publication, with time intervals of one hour. For each considered reference time and popularity metric (direct views, Facebook views and Twitter views), we train and evaluate a separate regressor. We focus on the first 24 hours after publication, because in order to adapt the publishing strategy it is most important to get good predictions at an early stage after publication on the website. For example, 90% of the Facebook publications of the articles in our dataset appear within 13 hours after they were published on the website. We first describe our evaluation on the predictions of the direct views, after which we discuss the Facebook and Twitter predictions.

6.5.3.1 Direct views

We first consider the number of direct views five days after online publication as the popularity quantity to be predicted. The performance of the five methods we proposed in Section 6.5.2 is shown in Figure 6.8. The RMSLE is shown as a function of the reference time r. In other words, the RMSLE indicates for a particular reference time r how well the popularity at target time τ is predicted, given observations and features up to time r. The linear regression model trained on both the historical popularity and the curve features (LM history+curve) performs better than the linear model only trained on the historical popularity (LM history) before hour 20. However, the improvement is not statistically significant (bootstrap hypothesis test, $p > 0.2$). The error of the linear models (LM history and LM history+curve) decreases steadily up to 19 hours after publication, after which the

[5]https://sites.google.com/site/predictivechallenge2014/

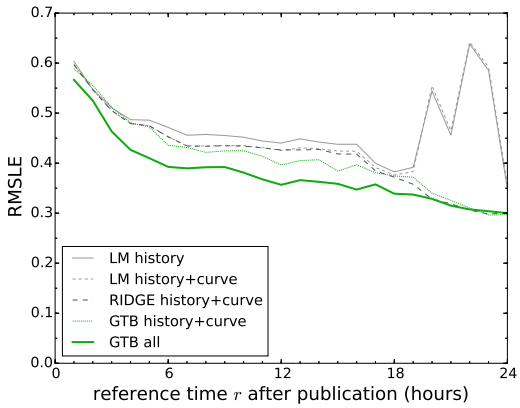


Figure 6.8: Performance of the four different versions of our proposed methodology, considering direct views.

error increases again and starts to fluctuate. This is mainly due to over-fitting of the linear model. When regularization is applied (RIDGE history+curve), we notice that the performance is similar for the first 19 hours after publication (bootstrap hypothesis test, $p > 0.2$). However, the error for the regularized model further decreases after hour 20, as over-fitting is avoided. The GTB regressors (GTB history+curve) are also robust to over-fitting, and lead to a slight improvement with respect to the ridge regressors (RIDGE history+curve) for reference hours 5 to 17 (bootstrap hypothesis test, $p > 0.2$). The last model, which applies GTB regression on all proposed features (GTB all), outperforms GTB history+curve for reference time r between hours 1 and 19. The improvement is statistically significant for $3 \leq r \leq 6$ and $10 \leq r \leq 14$ ($p < 0.05$). We conclude that adding content and meta-data related features on top of temporal features significantly improves the prediction effectiveness.

To investigate the contribution of each content and meta-data related feature type described in Table 6.2, a GTB regressor is trained using the history features and the features of the considered type. The performances of these models at reference time 10 can be found in Table 6.3. We observe that the author features leads to the best increase of performance (about 5% in RMSLE), closely followed by the manually annotated features (i.e. virality, target audience and emotion). The use of the publication or category features in addition to the history features also improves the performance (about 1% in RMSLE). On the other hand, the article source and title features hardly improve the GTB history model. Using all introduced content and meta-data features results in the best performance (increase of about 7% in RMSLE).

We now compare the baselines introduced in Section 6.5.1 with our best model

Table 6.3: Performance of the content and meta-data feature types at reference time 10, considering direct views.

model	**RMSLE**
GTB all	0.381
GTB history+author	0.405
GTB history+target	0.409
GTB history+virality	0.410
GTB history+emotion	0.415
GTB history+category	0.438
GTB history+publication	0.442
GTB history+source	0.451
GTB history+title	0.453
GTB history	0.453

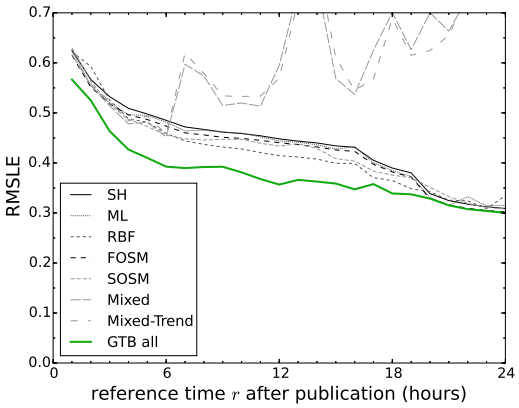


Figure 6.9: Performance of the baselines and our proposed methodology, considering direct views.

(GTB all), as shown in Figure 6.9. First of all, and most importantly, we see that our method outperforms all seven considered baselines significantly between one hour and 16 hours after publication ($p < 0.05$). Furthermore, between reference hour 3 and 16, our method improves on all baselines with more than 10% in RMSLE. As it is most important to get good predictions in an early stage after publication, our proposed methodology has a high added value compared to the baselines. For instance, the RMSLE for the method GTB all at reference hour 7 (0.387) is only achieved at reference hour 17 for the best baselines. In other words, the prediction performance of the baselines 17 hours after publication is already achieved by our method after only 7 hours. Starting from reference hour 20, all methods (except for Mixed and Mixed-Trend) have similar performance (bootstrap hypothesis test, $p > 0.2$). This is because the popularity of articles typically becomes stable after having been published that many hours. As a result, for $r \geq 20$, the added value of more complex regression algorithms and additional features on top of the historical popularities is no longer significant.

When we compare the baseline methods, we see that one of the most complex methods (Mixed) introduced by [7] performs on average as the best baseline model between hour one and six. This is in line with the observation made by [7], testing their model with a reference time of one hour after publishing and target time of 48 hours after publishing. However, starting from 7 hours after publication, the RMSLE for the methods Mixed and Mixed-Trend increases and starts to fluctuate. This is due to over-fitting of their linear regression model trained on a large set of features. This could be resolved by using regularization, but from the description in [7], it was unclear if and which sort of regularization was used. Between 7 hours and 19 hours after publication, the RBF model introduced by [5] has the best baseline performance. However, the improvement above the other baselines (except for Mixed and Mixed-Trend) is not statistically significant ($p > 0.2$).

6.5.3.2 Facebook views

We now consider the number of Facebook views five days after publishing the article on Facebook as the popularity to be predicted. We only consider the training articles which effectively got published on Facebook (1940 out of 2614 articles). The performance of the baselines and our proposed methodology is shown in Figure 6.10. Our model (GTB all) is the best performing model between hour 1 and 16 after publication on Facebook. The improvement with respect to the baselines is significant between reference hour 5 and 8 ($p < 0.05$). In particular, for $2 \leq r \leq 11$, the RMSLE decreases with more than 8% when using our method instead of the baseline. As an example, the RMSLE for our method at reference hour 6 (0.223) is only achieved after 10 hours for the baselines. Starting from 12 hours after publication on Facebook, all considered methods (except Mixed and Mixed-Trend) display a similar performance ($p > 0.2$). We notice that the Mixed and

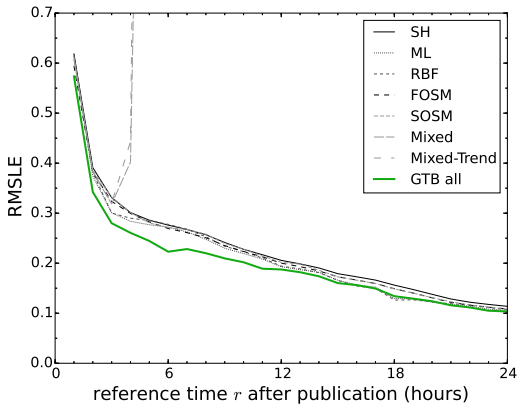


Figure 6.10: Performance of the baselines and our proposed methodology, considering Facebook views.

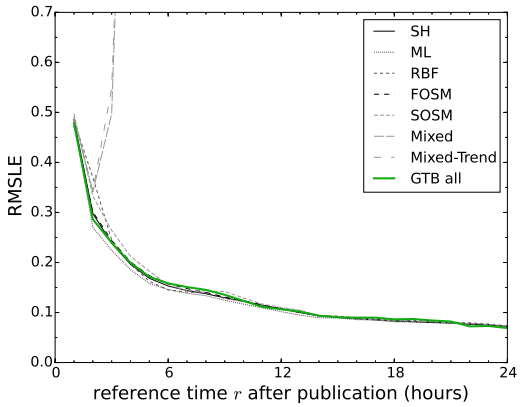


Figure 6.11: Performance of the baselines and our proposed methodology, considering Twitter views.

Mixed-Trend baselines start to over-fit at hour 4, which could again be avoided using regularization. The other baselines show similar prediction performances ($p > 0.2$).

6.5.3.3 Twitter views

We now evaluate the performance of the models in their ability to predict the number of Twitter views. We only consider the 1724 training articles effectively published on Twitter, and predict the number of Twitter views five days after publishing the article on Twitter. The performance of all models is shown in Figure 6.11.

We see that their performance (except for Mixed en Mixed-Trend) is very similar ($p > 0.2$). The main reason is that the average number of Twitter views received for articles published on Twitter is very low (around 80 views), and becomes constant soon after publication. It is thus not obvious to improve the prediction performance in terms of RMSLE by using more advanced features and algorithms. Note that this behavior is not representative for any news data, but in Belgium Twitter is not as widely adopted as in other countries [19].

We can conclude that our method outperforms all baselines during the first hours after publication, when direct views or Facebook views are considered. As mentioned before, the prediction of the direct views and Facebook views at these early hours is most relevant to optimize the publishing strategy of online articles. The significant improvement with respect to previously published methods therefore has a high added value for popularity predictions in practice.

6.6 Conclusion

In order to improve the online publishing strategy of news content, methods to model and predict the popularity of online news articles are required, which forms the main topic of this chapter. We first identified the distributions which underlie the view patterns of online news articles. These consist of several distinct components. The first component becomes visible as soon as the article is published on the news publisher's website. The corresponding views are referred to as the direct views and originate from e.g. search and browsing. When the article is published on social media, clear additional components in the view patterns start to appear. In this chapter, we focused on the views originating from Facebook and Twitter. We then introduced a model that allows to accurately model these view pattern components. This model captures the popularity behavior, is simple to fit to observed views, and has parameters that are intuitively interpretable. Based on real-world data from a young Belgian publisher that actively targets the distribution of its content over social media, we demonstrated that this model outperforms previously proposed log-normal fits. In addition, we took the influence of the day versus night on the view patterns into account to further increase the accuracy, without leading to a more complex model. By transforming each actual time interval into an equivalent time interval with an effective duration equal to the normalized total number of views for that time interval, the influence of the average hourly variations in number of views is largely canceled out, which allowed for a better fit of the view pattern to smooth basic functions.

As a second contribution, we proposed a methodology to predict the final popularity for each component adding to the total popularity of an article (i.e., direct views, Facebook views, and Twitter views). We focused on articles which are at

most one day old, as the predictions of those articles are most useful. Our primary model was based on existing methods, with linear regression algorithms and features based on the historical popularity of the articles. We then proposed models with improved prediction effectiveness, based on the following three ideas. First, we used the parameters of our proposed popularity model as additional input features, leading to a small overall improvement during the first hours after publication, although not significant. Second, we showed that the use of a more advanced regression technique, i.e., Gradient Tree Boosting, gives more accurate predictions. Third, the prediction performance was significantly improved by considering features based on the content and meta-data of the articles. Our best model outperformed all discussed baselines during the first hours after publication, at least for the direct views or Facebook views. In particular, we considered seven baseline methods, with features mainly capturing the historical popularity of the considered articles. The performance of the Twitter view predictions appeared similar to the baseline predictions. However, the average number of Twitter views per article appeared very low in our experimental setup, which prevented further improvements by using a more complex method. As the prediction of the direct views and Facebook views at the early hours are most relevant in order to optimize the publishing strategy of online articles, the significant improvement with respect to previously published methods has a high added value for popularity predictions in practice.

In this chapter, we proposed a method which predicts the final popularity of news articles. In future work, these predictions will be improved by considering additional features such as the positions of the articles on the home page, the popularity of articles with similar content, and the relationship between the articles and the hot news topics at the moment they are published. Additionally, we focused on articles with at most one Facebook and Twitter push. In future work, we will extend our monitoring and prediction approach to handle more than one push per social media platform. In addition to predicting the final popularity, it is often useful to also predict the popularity dynamics between the current time stamp and its final popularity. This information can for instance be used to better decide which would be the most suited moment to publish and promote an article over social media. Therefore, in future work, we will propose a methodology that predicts the entire future popularity pattern. This will be achieved by combining the knowledge of the proposed popularity model, including the day-night behavior of the different components, with the prediction method with content and meta-data related features in a single time series prediction setup.

Acknowledgment

We thank Ke Zhou for useful suggestions on drafts of the manuscript. Steven Van Canneyt is funded by a Ph.D. grant of the Agency for Innovation by Science and Technology in Flanders (IWT). Part of the presented research was performed within the MIX-ICON project PROVIDENCE, facilitated by iMinds-Media and funded by the IWT.

References

[1] A. Kaltenbrunner, V. Gómez, and V. López. *Description and prediction of Slashdot activity*. In Proceedings of the Latin American Web Conference, pages 57–66, 2007.

[2] M. Tsagkias, W. Weerkamp, and M. De Rijke. *Predicting the volume of comments on online news stories*. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, pages 1765–1768, 2009.

[3] A. Tatar, P. Antoniadis, M. D. de Amorim, and S. Fdida. *From popularity prediction to ranking online news*. Social Network Analysis and Mining, 4(1):174–186, 2014.

[4] G. Szabo and B. Huberman. *Predicting the popularity of online content*. Communications of the ACM, 53:80–88, 2008.

[5] H. Pinto, J. M. Almeida, and M. A. Gonçalves. *Using early view patterns to predict the popularity of YouTube videos*. In Proceedings of the 6h ACM International Conference on Web Search and Data Mining, pages 365–374, 2013.

[6] C. Castillo, M. El-Haddad, M. Stempeck, A. Jazeera, and J. Pfeffer. *Characterizing the life cycle of online news stories using social media reactions*. In Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing, pages 211–213, 2014.

[7] F. Figueiredo, M. Gonçalves, and J. M. Almeida. *Improving the effectiveness of content popularity prediction methods using time series trends*. In ECML/PKDD Discovery Challenge on Predictive Analytics, pages 1–6, 2014.

[8] R. Bandari, S. Asur, and B. a. Huberman. *The pulse of news in social media: Forecasting popularity*. In Proceedings of the 6th International Conference on Weblogs and Social Media, pages 26–33, 2012.

[9] I. Arapakis, B. B. Cambazoglu, and M. Lalmas. *On the feasibility of predicting news popularity at cold start*. In Proceedings of the 6th Internation Conference on Social Informatics, pages 290–299, 2014.

[10] S.-D. Kim, S.-H. Kim, and H.-G. Cho. *Predicting the virtual temperature of web-blog articles as a measurement tool for online popularity*. In Proceedings of the 11th International Conference on Computer and Information Technology, pages 449–454, 2011.

[11] M. Tsagkias, W. Weerkamp, and M. De Rijke. *News comments: Exploring, modeling, and online prediction*. In Proceedings of the 32nd European Conference on Advances in Information Retrieval, pages 191–203, 2010.

[12] S. Kong. *Predicting future retweet counts in a microblog*. Journal of Computational Information Systems, 4(10):1393–1404, 2014.

[13] A. Oghina, M. Breuss, M. Tsagkias, and M. De Rijke. *Predicting IMDB movie ratings using social media*. In Proceedings of the 34th European Conference on Advances in Information Retrieval, pages 503–507, 2012.

[14] M. H. DeGroot and M. J. Schervish. *Probability and statistics*. 2010.

[15] B. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[16] T. Sakai. *Evaluating evaluation metrics based on the bootstrap*. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 525–532, 2006.

[17] J. Deleu and A. D. Moor. *Named entity recognition on Flemish audio-visual and news-paper archives*. In Proceedings of the 12th Dutch-Belgian Information Retrieval Workshop, pages 38–41, 2012.

[18] J. Berger and K. L. Milkman. *What makes online content viral?* Journal of Marketing Research, 49(2):192–205, 2012.

[19] A. Cheng, M. Evans, and H. Singh. *Inside Twitter: An in-depth look inside the Twitter world*. Technical report, 2014.

7 Conclusion

In this dissertation, we have addressed several research challenges with regard to data science in the context of social media. These challenges can be subdivided into three main parts: (1) extracting useful content in a structured form from a large corpus of noisy and diverse social media data; (2) constructing frameworks to collect and analyze large amounts of data in real-time; and (3) predicting the popularity of social media content. This chapter highlights the most important contributions of this work and summarizes the perspectives for future research.

The research described in Chapters 2, Chapter 3 and Chapter 4 focused on the challenge of structured information extraction from social media. In Chapter 2, we demonstrated how social media can be used to improve existing databases of places. We first used mean shift clustering on the locations of a set of Flickr photos to obtain the locations which potentially correspond to places of interest (POIs) in a given city. We then associated with each candidate POI a feature vector based on the tags of the Flickr photos that are associated with locations nearby. Afterwards, we associated a query with each place type t based on the descriptions of known places of that type. The obtained query was used to rank the candidate POIs based on the likelihood that they belong to type t. In the optimization phase of our proposed methodology, we analyzed the behavior of different feature selection techniques. We concluded that for the Flickr data, correlation coefficient feature selection performs significantly better than χ^2. Based on a large-scale evaluation on 88 different cities, we concluded that Flickr tags are more informative for finding places of a given type than Twitter posts. However, it appeared that

using tweets in addition to Flickr photos contributes to obtaining better results yet. We further examined the results for London in more detail to analyze to what extent our approach can discover new places of a particular type. Based on this evaluation, we could conclude that our method is able to detect places which were not yet included in LinkedGeoData, Geonames, Google Places, and Foursquare. Additionally, we explained how our methodology can be used to identify errors in existing databases of places.

Chapter 3 focused on discovering the semantic type of events, using the relationship between an event type and other event characteristics. These characteristics were estimated using the metadata of the Flickr photos associated with the events. We first used a dataset collected from Upcoming to examine the performance of each considered characteristic. We considered high-level event types such as 'sport', 'music', and 'conferences'. When using our methodology instead of the baseline which only uses the text of the Flickr photos related to an event to estimate its semantic type, the classification accuracy increased significantly. We observed that considering the type of events visited previously by event participants leads to a substantial improvement over the baseline approach. The classification performance was further improved by also including the types of known events organized nearby, the textual content of the photos taken nearby, and the time and date of the event. Second, a dataset from Last.fm was used to demonstrate that the proposed methodology also works for more fine-grained event types (in this case, subtypes of music events). Finally, we showed how our methodology can be used to discover events not yet mentioned in existing event datasets.

In Chapter 4, we proposed a methodology to automatically mine Twitter streams in order to provide journalists with a set of headlines and complementary information that summarizes the most important topics for a number of time intervals of interest. We started with a dataset of tweets related to the problems of Syria, Ukraine, terror, and bitcoin, mentioned on Twitter during February 2014. As we were only interested in newsworthy topics, we used tweets of users classified as 'news publishers'. These tweets were then grouped into topics using the DBSCAN clustering algorithm, where the similarity between the tweets was determined using the cosine similarity on their boosted *tf-idf* representations. Thereafter, a classifier was trained to estimate which of the detected topics was newsworthy. Finally, for each obtained newsworthy topic, a descriptive headline, together with relevant tweets, keywords, and pictures were determined. The obtained topics and information were evaluated by the organizers of the SNOW2014 Data Challenge who indicated the effectiveness of our proposed methodology.

The challenges about collecting and analyzing big data in the context of online news content were covered in Chapter 5. To help optimizing publishing strategies of news agencies, we proposed a framework that can be used to monitor and analyze the consumption and social sharing behavior of users on news websites. To

test the potential of this framework, we evaluated it thoroughly on two major news websites. We concluded that it is able to monitor the popularity of online news articles in real-time, as it has been running in a stable way between April 2015 and January 2016, having registered more than 40,000 articles, 15 million visits and 4 million social shares. The framework was constructed in a scalable way, such that it can handle many more articles, visits, shares, and websites in the future. While anallyzing the obtained data, we observed that the optimal publishing strategy depends on the considered popularity metric, and differs for both of the considered news websites. For example, there is a clear difference in sharing behavior via Twitter versus Facebook for newsmonkey articles. In particular, light-weighted newsmonkey articles with emotional content, funny pictures, or lists perform best on Facebook. On the other hand, newsmonkey articles containing breaking news and a thorough analysis of politics or economy perform much better on Twitter than on Facebook. The difference in behavior between Facebook and Twitter was less clear for the deredactie.be articles. Studying new article features led to better understanding of the sharing behavior. For instance, the feature indicating the emotion that captures why users want to share articles, led to the insight that taboo may be a good indicator to read an article, but not to share it with friends. These observations confirmed the need for a data-driven framework to support online news publishing strategies.

In Chapter 6, the last challenge of this dissertation was described. We presented a novel methodology to model and predict the popularity of online news. We first identified the distributions which underlie the view patterns. These consist of several distinct components. The first component becomes visible as soon as an article is published on the news publisher's website. The corresponding views are referred to as the direct views and originate from, e.g., search and browsing. When the article is published on social media, clear additional components in the view patterns start to appear. In this paper, we focused on the views originating from Facebook and Twitter. We then introduced a model that allows to accurately model these view pattern components. This model captures the popularity behavior, is simple to fit to observed views, and has parameters that are intuitively interpretable. Based on real-world data, again from newsmonkey, we demonstrated that this model outperforms previously proposed log-normal fits. In addition, we took the influence of the day versus night on the view patterns into account to further increase the accuracy, without leading to a more complex model. As a second contribution, we proposed a methodology to predict the popularity for each component adding to the total popularity of an article (i.e., direct views, Facebook views, and Twitter views). We focused on articles of at most one day old, as the predictions of those articles are most useful. Our primary model was based on existing methods, with linear regression algorithms and features based on the historical popularity of the articles. We then proposed models with improved prediction effectiveness,

based on the following three ideas. First, we used the parameters of our proposed popularity model as additional input features, leading to a small overall improvement during the first hours after publication, although not significant. Second, we showed that the use of a more advanced regression technique, i.e., Gradient Tree Boosting, gives more accurate predictions. Third, the prediction performance was significantly improved by considering features based on the content and meta-data of the articles. Our best model outperformed all discussed baselines during the first hours after publication, at least for the direct views or Facebook views. In particular, we considered seven baseline methods, with features mainly capturing the historical popularity of the considered articles. The performance of the Twitter view predictions appeared similar to the baseline predictions. However, the average number of Twitter views per article appeared very low in our experimental setup, which prevented further improvements by using a more complex method. As the prediction of the direct views and Facebook views at the early hours are most relevant in order to optimize the publishing strategy of online articles, the significant improvement with respect to previously published methods has a high added value for popularity predictions in practice.

We see a number of opportunities for future work. We proposed methodologies in Chapter 2 and Chapter 3 to discover places and events from social media, and to estimate their semantic type. This can be extended in future work by discovering additional features of the places and events using social media. For instance, by encoding the final score for football matches or to detect the best items on the menu of a restaurant. This additional structured information can be used to perform more complex queries, e.g. 'In which countries did U2 perform during 2016?'.

In recent years, the importance of neural networks to solve big data problems has increased a lot. For instance, in March 2016, there was a breakthrough for artificial intelligence as the Go-playing AI AlphaGo, developed using neural networks, has beaten world-class player Lee Se-dol.[1] Additionally, Microsoft built a neural network which is able to identify objects in a photograph or video with an accuracy that meets and sometimes exceeds human-level performance.[2] Due to the large amount of data in social media and the new developments in neural networks, in future work, the potential of neural networks should be investigated to handle the challenges discussed in this dissertation. For instance, neural networks can be developed to classify events and places into semantic types. This may improve the performance of the feature engineering approaches introduced in Chapters 2 and 3. In addition, it should be investigated if neural networks can improve the performance of popularity prediction of news articles as handled in Chapter 6.

[1] https://deepmind.com/alpha-go

[2] http://blogs.microsoft.com/next/2015/12/10/microsoft-researchers-win-imagenet-computer-vision-challenge/

In Chapters 5 and Chapter 6, we proposed a framework which monitors, analyzes and predicts the consumption behavior of online news articles in real-time. This framework has been deployed at newsmonkey and deredactie.be, and will be used to optimize their publishing strategy. The framework will also be made available for other news websites, to monitor and analyze their data and to optimize their strategy. In future work, the knowledge into consumption behavior of online news and their predicted popularity should be used to construct methodologies which actively suggest how the publishing strategy can be optimized. For instance, by recommending in real-time which articles should be published at what time on which social media platform to reach the highest possible audience.

Data scientists constantly need to extend their skills and knowledge to improve state-of-the-art methods and discover and solve new challenges. Therefore, I want to end my PhD dissertation with the same words with which Steve Jobs concluded his famous commencement speech at Stanford University in 2005:

Stewart and his team put out several issues of The Whole Earth Catalog, and then when it had run its course, they put out a final issue. On the back cover of their final issue was a photograph of an early morning country road, the kind you might find yourself hitchhiking on if you were so adventurous. Beneath it were the words: 'Stay Hungry. Stay Foolish.' It was their farewell message as they signed off. Stay Hungry. Stay Foolish. And I have always wished that for myself. And now, as you graduate to begin anew, I wish that for you.

Stay Hungry. Stay Foolish.

Thank you all very much.