Erasmus Mundus European Master in Language and Communication Technologies

University of Gronigen
Saarland University

Real time discussion retrieval from Twitter

Author:
Dmitrijs Miļajevs

Supervisors:
Gosse Bouma
Ivan Titov

August 2012

Abstract

This work studies discussion retrieval from a social network. Given an input stream of messages as an example of the discussion, the system extracts the most relevant words from it and queries the social network for more messages that contain these words. Then the system filters out the messages that do not belong to the discussion. The system was evaluated on a manually built dataset of tweets about Euro 2012 football championship. System precision and recall are 0.41 and 0.15. Suggestions on how the system performance can be improved are given.

Contents

Chapter 1

Introduction

Connected to the Internet mobile devices together with social networks have become a prominent human communication channel. People post tweets to Twitter, write status updates or comments in Facebook or any other social website to express their life experience. This activity results in an enormous amount of poorly structured natural language data, which reveals preferences of the community of social networks in general and personality of every individual in particular.

Companies are getting interest in joining social media as well. Obviously, they obtain yet another way to promote their product. Furthermore, social media reveals customer preferences giving an opportunity to reduce advertisement costs, to receive immediate product feedback, and most importantly to alter business strategy.

However, raw social data hardly meet expectations and fulfill needs of both humans and companies. This happens simply because of the amount of the information, which can not be handled manually in any reasonable amount of time, and the noise level of the data. This is why automatic analysis of massive unstructured text data receives a lot of attention from the scientific community and support from the industry.

Before being analyzed, the data have to be retrieved from a social network. Some services, for example Twitter, provide search functionality based on keywords, but it is not powerful enough if the goal of the retrieval is to get tweets about an event or a product.

Chapter 2 investigates whether people discuss events in social networks and gives insights on how the discussion is distributed in time as the study of three music festivals in the Netherlands and Belgium in 2011. In addition, keyword based document retrieval is evaluated. It is shown that discussions contain unexpected keywords and that several discussions may share the same hashtag (hashtags are special words that start with the character #, for example `#Groningen`).

Later, chapter 3 gives an overview how a two stage discussion retrieval system can be implemented taking into account limitations of the Twitter API. Given a stream of tweets that are highly associated with the discussion of interest, the first stage aims to retrieve more tweets with high recall by "subscribing" to trending keywords and hashtags found in the given stream. The second stage filters out tweets that are not relevant (maximizes precision).

The two stage stream retrieval process that explicitly decouples the high–recall retrieval step and the precision oriented filtering step together with the evidence that hashtags are not powerful enough for advanced tweet retrieval are the main contributions of the work.

Chapter 4 evaluates the proposed method on 5 data sets prepared manually about the UEFA football championship, 6 music festivals and the modern pentathlon event at the Olympics. Then suggestions how the method can be improved and adopted to a real system are given in chapter 5. Finally, chapter 6 concludes the work.

Chapter 2

Events and social media

A big entertainment event, such as a rock festival, is all about positive experience, which naturally is being shared between people before, during and after the event. The benefits that concert attendees and concert organizers may gain include, but not limited to safety, fun or commerce. This section describes three scenarios, which, we believe, reveal the properties of a useful way to make social media an important part of a concert.

Security Making a concert as secure as possible should be the main responsibility of the organizers and the main concern of visitors. An opportunity to notify and guide people in unexpected or risky situations may help to prevent injuries. Mobile push notification is a perfect way to have a one way communication from an the organizer to the event visitors.

To make a warning, a concert host has to detect a security threat. Moreover, it has to be done as soon as possible minimizing the delay between the beginning of a threat and the moment when visitors are notified. One way of the delay minimization can be social media monitoring for violent or worried messages. Once a sufficient amount of these messages is collected, security department is notified providing a possible nature of a threat, its location or any other relevant information. Then the security department will be able to make a decision on how to react.

Entertainment People don not just go to the concerts to feel safe, the main reason is to enjoy the time and to get positive memories. So, every visitor is concerned with being at the most interesting place according to her taste. How can she decide what stage to attend at a festival with multiple stages? The festival lineup is helpful for deciding in advance what to do, but not for making prompt decisions, for example, a lineup will not tell whether a band have already tuned their instruments and are ready to perform, or give a description of sound quality at the current moment.

In contrast, other people might have already stated their experience in social media, which has to be retrieved from a social media stream and showed to the user. It also may be the case, that a person just wants to read relevant messages about an event.

Commerce Once messages about the event are captured, an event organizer may be interested to know whether an event was a success or a failure, what has been done right or wrong, the difference in people perception comparing to previous similar events, the social structure of the event attendees and their social profile.

2.1 Case study: Music festivals in 2011

Two music festivals which took place in the Netherlands during 2011 and one in Belgium were chosen for examination to get the initial impression on how people share their festival experience in social networks.

The analysis is based on a tweet collection built by the University of Groningen. The collection consists of tweets which contain typical Dutch words, thus the tweets are mainly in the Dutch language and cover a broad variety of topics.

To filter tweets relevant to an event the following steps were carried out. The tweets which mention the name (both as a word or as a hashtag) or the short official hashtags of an event are selected. For Pukkelpop hashtags that reffer to the accident are added as well. Table 2.1 shows the hashtag distribution for the chosen events.

2.1.1 Pinkpop Festival

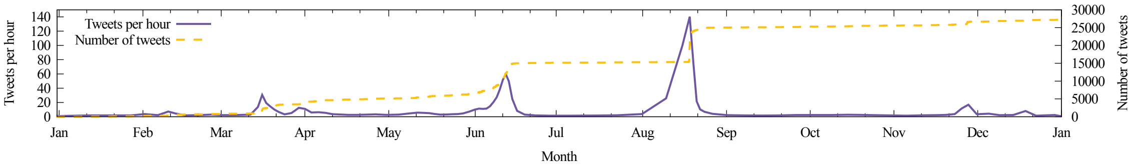

Figure 2.1: Pinkpop 2011

Pinkpop festival[1] takes place in Landgraaf every year. Pinkpop 2011 happened on 11–13 June. The headliners were Coldplay, Kings of Leon and Foo Fighters.

Figure 2.1 shows the distribution of Pinkpop related tweets in 2011. The tweet selection includes tweets containing either *pinkpop*, `#pinkpop` or `#pp11`.

27,251 tweets were selected, 6,974 (26%) of them are without any hashtag.

The Pinkpop Twitter community reacts on the news stories. Two peaks are related to the festival news: on March 16 the line up was announced[2], and on 31st of March three more bands were announced[3].

The peak in June is due to the festival itself. The peak in August caused by other event, Pukkelpop (see section 2.1.3 for a detailed description). Peaks in winter are about Pinkpop 2012.

2.1.2 A campingflight to Lowlands paradise

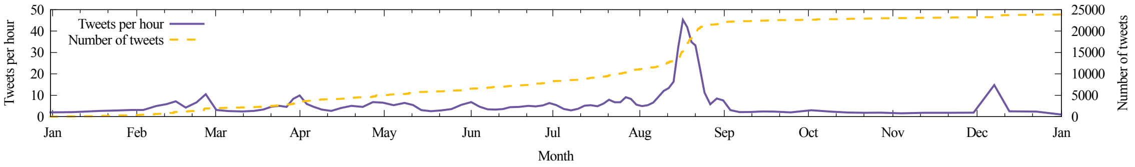

Figure 2.2: Lowlands 2011

Lowlands[4] is a Dutch music festival, which traditionally takes place in August. In 2011 the festival happened from 19th to 21th of August.

In order for a tweet to form the selection, it has to contain either *lowlands*, `#lowlands`, `#ll11` or `#ll`. 23,880 tweets were selected. 7,062 or 30% tweets are without hashtags.

Similarly to Pinkpop, the Twitter activity peaks for Lowlands (figure 2.2) in spring correspond to the line up announcements: in February among many artists The Wombads and Interpol were announced,[5] in April Arctic

[1] http://www.pinkpop.nl

[2] http://www.3fm.nl/nieuws/detail/350006/Pinkpop-2011-line-up

[3] http://spotlight.excite.nl/3-nieuwe-bands-op-pinkpop-2011-N7445.html

[4] http://lowlands.nl

[5] http://kickingthehabit.nl/2011-02-25/lowlands-2011-bevestigt-interpol-fleet-foxes-the-wombats-en-tren

Monkeys were announced[6] and in May The Offspring were announced.[7] Again, activity in winter is because of Lowlands 2012.

Table 2.3 shows the most important words of the Lowlands tweet collection. The words are ranked by the formula (2.1), where n_w is the frequency of a word in the event collection and N_w is the frequency of it in the whole collection. Other formulas were tried, but they did not yield satisfactory results. Especially, TF-IDF does not perform well, because a single tweet is unlikely to have repeated words, essentially making the score just the IDF.

$$score(w) = n_w(\frac{n_w}{N_w})^2 \tag{2.1}$$

The reason that the ratio of the word frequency in the collection to the global frequency is squared is to penalize words that are very frequent and not very unique to the event – their ratio will be much smaller than 1 and the square will make the value even less.

The majority of the words in the table are related to the event: they are mostly variations of the festival title, names of the artists and bands that performed at the festival or other entities relevant to the event. This suggests that tracking named entities should make search more accurate.

2.1.3 Pukkelpop

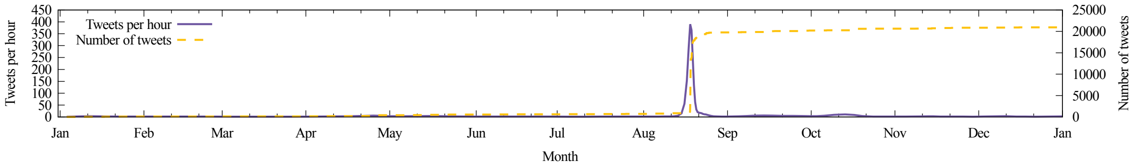

Figure 2.3: Pukkelpop 2011

Pukklepop[8] is a Belgium music festival. Pukkelpop 2011 was planned to happen from 18–20 August. The event was stopped[9] [10] due to a severe storm. Five people died. Twitter has been heavily used to organize help.

The query was the following: *pukkelpop*, `#pukkelpop`, `#pkp`, `#pkp11`, `#okpp`, `#hasselthelpt`, `#ppshelter`. 20,967 tweets were selected. 7,677 or 37% of selected tweets do not contain any hashtag.

@daHawkeyeCaller gives an overview[11] on how Twitter was used in the situation. According to him, the main challenge in using Twitter to help others was to unclatter the social stream from emotional and already known posts and access valuable information.

The conversation went beyond the official hashtag `#pkp11`. For example, `#hasselthelpt`, a special hashtag consisting of Hasselt, the name of the closest to the festival city and *help*, was used for asking or providing help. @WimLuyckx suggested to use `#hasselthelptmooi`[12] for expressing sentiments. `#ppok` was used to communicate that the author of a tweet is fine. `#pp` was also used, causing the clash with Pinkpop related tweets.

In September, the weather condition was recognized as a storm. This was reported in the media.[13] In October, Pinkpop 2012 was announced.[14]

Figure D.1 shows the hashtag distribution trough the evening of August 18 on a stacked graph [9].

[6] http://kickingthehabit.nl/2011-03-31/lowlands-2011-arctic-monkeys-warpaint-miles-kane-skunk-anansie

[7] http://kickingthehabit.nl/2011-04-28/lowlands-2011-aphex-twin-agnes-obel-lykke-li-the-naked-and-famou

[8] http://pukkelpop.be

[9] http://www.bbc.co.uk/news/world-europe-14582448

[10] http://www.inquisitr.com/135614/2011-pukkelpop-festival-stage-collapse-belgium-kills-five

[11] http://dahawkeyecaller.tumblr.com/post/9111077769

[12] Hasselt helps beautifully

[13] http://annemieturtelboom.be/2011/09/29/praktische-info-erkenning-storm-van-18-augustus-als-ramp

[14] http://www.guardian.co.uk/music/2011/oct/19/pukkelpop-2012

2.2 Case study: Global hashtag ambiguity

#ll and its usage The case study about major events in 2011 revealed that hashtags are ambiguous. However, due to the fact that only Dutch tweets were processed, the hashtag ambiguity issue was minimized.

The global usage of the hashtag `#ll` illustrates the problem. `#ll` is assumed to be a possible short hashtag for tweets about Lowlands. Alternatively, `#ll` means *love or lust*[15], *longitude lattitude*[16], *love life*, *Illinois* or *La Liga*[17].

#pp12 and music festivals `#pp12` is much less ambiguous than `#ll`, mostly because it is used by the Dutch Twitter users. Analysis of 150 tweets collected in 2012 shows that approximately 81% of tweets are about Pinkpop, 17% of them are about Pukkelpop (both are big music festivals), the rest was about Paaspop, Pauwenpop (other, much smaller festivals) or was about totally different matter.

2.3 Conclusion

The case study reveals the following:

Twitter activity correlates with media news The peaks in spring are triggered by news stories about the coming events. For all music festivals line up announcement triggered boost of activity in Twitter.

During an event twitter users are the most active The activity peak during the event is the highest for Lowlands and Pukkelpop. Due to the tag clash for Pinkpop and Pukkelpop (`#pp11` was used for both events), Pinkpop timeline also captured Pukkelpop.

Hashtags are ambiguous Depending on tweet context and time, a hashtag may refer to various entities. Again `#pp11` in June is most probably about Pinkpop, and in Augusts it is about Pukkelpop. June and August are the months the festivals take place.

Conversation goes beyond expected hashtags At Pukkelpop, where people were actually generating the content, instead of discussing it, new hashtags appeared. `#hasselthelpt`, `#ppok` and `#ppshelter` are among them.

Hashtags are not enough More than a quarter of captured tweets related to events did not contain any hashtag. Thus more elaborate linguistic analysis is required for an adequate social media analysis system.

Named entities may make the retrieval much more accurate Analysis of the Lowlands data revealed that words that correspond to the bands, places or other entities related to the event are ranked high. Identifying them should make tweet analysis more precise.

[15] It is also a Twilight Canon role play site's name. http://loveorlust.ning.com

[16] There is a service which exploits this tag. http://loctweet.appspot.com

[17] The top professional association football division of the Spanish football league system.

	Pinkpop			Lowlands			Pukkelpop		
Rank	Hashtag	Frequency	Ratio	Hashtag	Frequency	Ratio	Hashtag	Frequency	Ratio
1	pp11	16428	1.00	ll11	12798	1.00	pukkelpop	7517	1.00
2	pinkpop	3325	1.00	lowlands	1906	1.00	pp11	4014	0.24
3	pukkelpop	1857	0.25	ll12	1275	0.91	hasselthelpt	2806	1.00
4	hasselthelpt	944	0.34	zinin	540	0.03	ppok	1301	1.00
5	ppok	757	0.58	3fm	267	0.00	ppshelter	276	1.00
6	3fm	501	0.01	durftevragen	189	0.00	3fm	194	0.00
7	pp12	246	0.69	ll	187	1.00	brusselhelpt	161	0.92
8	durftevragen	203	0.00	3voor12	185	0.07	pp	143	0.21
9	3voor12	174	0.07	dtv	173	0.00	g	125	0.02
10	foofighters	163	0.39	llowlab	124	0.72	genthelpt	116	0.43
11	livexs	161	0.28	spannend	99	0.01	nos	114	0.01
12	countdown	151	0.34	pp11	96	0.01	okpp	114	1.00
13	chokri	142	0.83	lowleaks	90	0.81	ll11	97	0.01
14	ppshelter	130	0.47	pukkelpop	77	0.01	actualiteit	78	0.35
15	fb	129	0.01	loesje	69	0.00	hass	69	0.64
16	coldplay	123	0.01	foutjebedankt	62	0.32	fb	66	0.00
17	nos	111	0.01	fb	47	0.00	noodweer	65	0.02
18	pp	109	0.16	kaartje	40	0.16	pkp11	60	1.00
19	genthelpt	106	0.39	bavariacityrace	40	0.38	onweer	57	0.01
20	ll11	99	0.01	tekoop	40	0.01	helpaviking	57	0.85
21	ned3	95	0.02	lief	39	0.01	pukkelpopramp	55	0.98
22	actualiteit	78	0.35	ned3	38	0.01	p11	55	0.70
23	frs	77	0.55	hema	37	0.01	3voor12	54	0.02
24	dtv	73	0.00	pinkpop	37	0.01	terzake	53	0.10
25	jansmeets	71	1.00	rockon	36	0.48	rt	43	0.00
26	3onstage	68	0.30	randstad	35	0.02	b	42	0.00
27	hass	68	0.63	l	33	0.00	terzaketv	41	0.03
28	goudentip	68	0.28	ll2011	30	0.47	brus	36	0.88
29	app	61	0.02	waarishetfeestje	29	0.72	hasselt	34	0.08
30	stemmenmaar	60	0.67	ll11backstage	29	0.94	kvdb	31	0.00
31	android	53	0.01	dwdd	27	0.00	belgie	31	0.02
32	cultura24	52	0.32	a28	25	0.02	socialmedia	29	0.00
33	landgraaf	51	0.17	geruchten	24	0.02	blog	29	0.01
34	spoiler	51	0.41	onweer	24	0.01	pleaseretweet	29	0.03
35	ha	49	0.03	pvv	24	0.00	unexpected	28	0.85
36	kingsofleon	47	0.65	h	24	0.00	ravage	28	0.41
37	loesje	47	0.00	livexs	23	0.04	hamme	27	0.51
38	fail	46	0.00	n306	23	0.82	pu	25	0.08
39	weeatafrica	45	0.43	omarenomar	22	0.73	p	24	0.00
40	een	44	0.03	fiy	22	0.17	pkp	24	1.00
41	vrt	44	0.06	lowlandsweetje	22	1.00	pp10	19	0.39
42	okpp	43	0.38	wistjedat	21	0.00	lowlands	19	0.01
43	repudo	43	0.09	arcticmonkeys	21	0.23	twitterhelpt	18	0.30
44	helpaviking	43	0.64	festival	21	0.02	be	17	0.01
45	twitterhelpt	42	0.69	elbow	20	0.06	rip	17	0.00
46	hasselhelpt	39	0.67	in	20	0.00	drama	17	0.01
47	rt	39	0.00	nosop3	19	0.01	demorgen	16	0.01
48	p	38	0.00	effeekdom	19	0.02	storm	16	0.01
49	theboss	38	0.58	rookuitserver	19	1.00	vrt	16	0.02
50	nosop3	36	0.01	nieuwesite	19	0.28	durftevragen	16	0.00

Table 2.1: The top 50 most frequent hashtags. Ratio of a tag is the ratio of tag's frequency in the event collection to the tag's frequency in the whole collection.

	Pinkpop			Lowlands			Pukkelpop		
Rank	Hashtag	Frequency	Ratio	Hashtag	Frequency	Ratio	Hashtag	Frequency	Ratio
1	pp11	16428	1.00	ll11	12798	1.00	pukkelpop	7517	1.00
2	pinkpop	3325	1.00	lowlands	1906	1.00	hasselthelpt	2806	1.00
3	jansmeets	71	1.00	ll	187	1.00	ppok	1301	1.00
4	chokri	142	0.83	lowlandsweetje	22	1.00	ppshelter	276	1.00
5	twitterhelpt	42	0.69	rookuitserver	19	1.00	okpp	114	1.00
6	pp12	246	0.69	ll11backstage	29	0.94	pkp11	60	1.00
7	hasselhelpt	39	0.67	ll12	1275	0.91	pkp	24	1.00
8	stemmenmaar	60	0.67	n306	23	0.82	pukkelpopramp	55	0.98
9	kingsofleon	47	0.65	lowleaks	90	0.81	brusselhelpt	161	0.92
10	helpaviking	43	0.64	omarenomar	22	0.73	brus	36	0.88
11	hass	68	0.63	waarishetfeestje	29	0.72	helpaviking	57	0.85
12	ppok	757	0.58	llowlab	124	0.72	unexpected	28	0.85
13	theboss	38	0.58	rockon	36	0.48	p11	55	0.70
14	frs	77	0.55	ll2011	30	0.47	hass	69	0.64
15	ppshelter	130	0.47	bavariacityrace	40	0.38	hamme	27	0.51
16	weeatafrica	45	0.43	foutjebedankt	62	0.32	genthelpt	116	0.43
17	spoiler	51	0.41	nieuwesite	19	0.28	ravage	28	0.41
18	foofighters	163	0.39	arcticmonkeys	21	0.23	pp10	19	0.39
19	genthelpt	106	0.39	fiy	22	0.17	actualiteit	78	0.35
20	okpp	43	0.38	kaartje	40	0.16	twitterhelpt	18	0.30
21	actualiteit	78	0.35	3voor12	185	0.07	pp11	4014	0.24
22	countdown	151	0.34	elbow	20	0.06	pp	143	0.21
23	hasselthelpt	944	0.34	livexs	23	0.04	terzake	53	0.10
24	cultura24	52	0.32	zinin	540	0.03	pu	25	0.08
25	3onstage	68	0.30	geruchten	24	0.02	hasselt	34	0.08
26	goudentip	68	0.28	randstad	35	0.02	pleaseretweet	29	0.03
27	livexs	161	0.28	a28	25	0.02	terzaketv	41	0.03
28	pukkelpop	1857	0.25	effeekdom	19	0.02	vrt	16	0.02
29	landgraaf	51	0.17	festival	21	0.02	g	125	0.02
30	pp	109	0.16	lief	39	0.01	3voor12	54	0.02
31	repudo	43	0.09	spannend	99	0.01	belgie	31	0.02
32	3voor12	174	0.07	hema	37	0.01	noodweer	65	0.02
33	vrt	44	0.06	pinkpop	37	0.01	demorgen	16	0.01
34	ha	49	0.03	pukkelpop	77	0.01	onweer	57	0.01
35	een	44	0.03	tekoop	40	0.01	drama	17	0.01
36	app	61	0.02	ned3	38	0.01	nos	114	0.01
37	ned3	95	0.02	onweer	24	0.01	storm	16	0.01
38	android	53	0.01	nosop3	19	0.01	lowlands	19	0.01
39	nosop3	36	0.01	pp11	96	0.01	be	17	0.01
40	nos	111	0.01	3fm	267	0.00	blog	29	0.01
41	3fm	501	0.01	h	24	0.00	ll11	97	0.01
42	ll11	99	0.01	l	33	0.00	b	42	0.00
43	coldplay	123	0.01	dtv	173	0.00	kvdb	31	0.00
44	fb	129	0.01	fb	47	0.00	3fm	194	0.00
45	p	38	0.00	loesje	69	0.00	p	24	0.00
46	fail	46	0.00	durftevragen	189	0.00	rip	17	0.00
47	durftevragen	203	0.00	dwdd	27	0.00	fb	66	0.00
48	loesje	47	0.00	in	20	0.00	socialmedia	29	0.00
49	dtv	73	0.00	pvv	24	0.00	rt	43	0.00
50	rt	39	0.00	wistjedat	21	0.00	durftevragen	16	0.00

Table 2.2: The top of hashtags by ratio. The ratio equal to $\frac{n_w}{N_w}$, where n_w is the frequency of a word w in the collection of tweets relevant to an event, and N_w is the frequency of the word in the whole collection.

Rank	Word	Frequency	Ratio	Score	Description
1	lowlands	18013	1.00	18013	Festival's name.
2	nachtjes	4113	0.39	635	A special account posted messages like: *Nog 144 nachtjes slapen tot Lowlands*
3	lowlandsnieuws	375	0.94	328	
4	soulyman	253	1.00	253	Artist Omar Souleyman
5	lijstj	252	0.90	203	
6	lowlanders	253	0.84	178	
7	junip	212	0.88	164	Band Junip
8	wombats	393	0.59	139	Band The Wombads
9	warpaint	228	0.76	131	Band Warpaint
10	interpol	339	0.62	130	Band Interpol
11	fleet	422	0.55	126	Band Fleet Foxes
12	foxes	399	0.56	124	Band Fleet Foxes
13	beady	298	0.57	97	Band Beady Eye
14	zwarthandel	133	0.84	93	*Black trade* in Dutch
15	horeeeeeeh	83	1.00	83	
16	jeeeuuuh	83	0.97	77	
17	castles	158	0.70	77	Band Crystal Castles
18	namencircus	77	1.00	77	
19	aphex	189	0.61	71	Band Apex Twin
20	eerdenburg	145	0.69	68	Eric van Eerdenburg is the director of Lowlands
21	offspring	225	0.51	59	Band The Offspring
22	marketingtrucje	59	0.98	57	*Marketing trick* in Dutch
23	lowmap	50	1.00	50	A mobile application about the festival
24	residentie	264	0.41	45	Band Residentie Orkest
25	ticketfraude	43	1.00	43	*Ticket fraud* in Dutch
26	bleeps	46	0.94	41	
27	lowlandsterrein	44	0.96	40	*Lowlands area* in Dutch
28	natiom	39	1.00	39	
29	biddinghuizen	281	0.37	38	The closest town to Lowlands
30	lykke	156	0.48	36	Artist Lykke Li
31	jannekegelul	36	1.00	36	
32	campingflight	36	1.00	36	Part of the festival name
33	festivalsshows	35	1.00	35	
34	lowlands2011	43	0.88	33	
35	vaccines	188	0.41	32	Band The Vaccines
36	festivalgras	30	1.00	30	
37	windgevoelig	33	0.94	29	
38	lowlandsnaam	37	0.88	29	*Lowlands' name* in Dutch
39	arctic	414	0.26	28	Band Arctic Monkeys
40	evabol	30	0.97	28	
41	afhaalindonees	27	1.00	27	
42	biddinghuize	36	0.86	26	The closest town to Lowlands
43	groentetoren	24	1.00	24	
44	leeggekomen	24	1.00	24	
45	monkeys	418	0.24	24	Band Arctic Monkeys
46	trentemoller	31	0.86	23	Band Trentemoller
47	gaslamp	74	0.56	23	Band The Gaslamp Killer
48	lowlandsflasht	24	0.96	22	
49	llow	79	0.52	21	
50	chromeo	48	0.66	21	Band Chromeo

Table 2.3: The top of words by score, where score of a word w is $n_w(\frac{n_w}{N_w})^2$.

Chapter 3

Discussion retrieval

This chapter explains how the discussion about an event is retrieved from social media. Informally, a discussion retrieval system should be able to get messages relevant to the discussion from a global stream of messages.

Chapter 2 revealed that the events in the real world give rise to discussions in Twitter, and gave some use cases how the analysis of the discussion can be used.

Discussion retrieval is a challenging task. The tweets about Pukkelpop (section 2.1.3) show that it is almost impossible to predict what keywords and phrases will be used the most in the discussion of an event. Even so, a part of the discussion can be retrieved easily. This part is made up of messages that contain obviously related words or are written by concrete users. For an event discussion it could be the name of the event, or the names of event organizers. We refer to these messages as the core stream of a discussion.

A core stream, being a part of a discussion, is similar to it. This similarity can be exploited in order to retrieve the other part of the discussion. The other part will share the same topics that the discussion is about, and will share the same words.

It is assumed that everything in the core stream belongs to the discussion. In other words, the user has to be very careful in choosing the keywords that form the core stream, and avoid ambiguous keywords. However, we believe that it is always possible to narrow down the core stream so it contains mostly the relevant messages, for example, in case of a music festival, by tracking only the timelines of the official festival representatives.

3.1 Related work

Nowadays, Twitter is a popular social media service where users publish short 140 character long messages. The ability to post messages, called tweets, via a mobile phone made the service extremely popular. Also, linguists use data collected from Twitter in their research.

User experience In Twitter users generate a huge amount of data. Issuing a query of interest, a user receives so many tweets that are impossible to digest.

[6] proposes a topic–based browsing of social streams. Instead of showing a stream of tweets, the topic streams are shown. Topics are rendered as a timeline (in a similar way to [9]) or as a tag cloud.

To assign tweets to topics, instead of applying a clustering algorithm, the authors used search engine as a distributed knowledge base. The main idea behind this is that tweets are very similar to search engine queries, and the answer to the query based on a tweet will contain required information to identify tweet's topics.

A tweet is labeled with topics in 3 steps. In the first step, the noun phrases are used to form a search query. Then a search engine is queried. Sometimes, the query is too specific and little or no results are returned. In this case iterative backoff is used to remove terms from the query that have the fewest occurrences in the web. Finally, the algorithm identifies the popular words in the search results and treats them as topics.

The user study reveled that event though the precision of the best algorithm was only 40%, the topic organization was reported meaningful and accurate.

[23] is another study of a compact representation of a stream of tweets as a timeline. They assume that events are tightly associated with peaks in the user activity, so it makes sense to detect peaks and then to label them with few descriptive words. Once a peak is detected, the terms of the tweets in the peak are ranked using TF-IDF, the top ranked terms label the peak. Evaluation on a collection of tweets shows that the system captures the major events, but did not recognized yellow cards.

Tweet annotation Tweets apart from the text of the message contain metadata such as username, creation time, geographical location and so on. Though, the message it self contain references to people, places.

[20] describe a semi–supervised method of named entity recognition. They split the task to two sub tasks: boundary detection and type classification. A boundary define the beginning and the end of a named entity. Typical types are person, location, organization or product. [33] propose a similar technique but in addition exploit Freebase[1] dictionaries as a source of distant supervision.

[25] develop a semantic linking of phrases in tweets to Wikipedia articles based on a high–recall concept ranking and high–precision concept selection.

Event detection A number of studies was performed to retrieve and detect events in a social media stream.

[36] maps tweets to late breaking news in online fashion, so the system is able to identify news stories almost as soon as they happen. [5] maps events found in a local city guide to tweets using a graphical model. [38] compared Twitter and traditional media focusing on topics that different media covers.

[31] and [30] extended the event detection task to find the first message about event of interest.

Analysis [37] observed that twitter is used in political deliberation on studying the tweets about the federal election of the national Parliament in Germany in 2009 and predicted the election results.

[18] performed target-dependant sentiment classification of tweets.

Trend detection Trends are the core notion for many Twitter based applications. [4] define a trend as:

Definition 1 (Trend) A word or phrase that is experiencing an increase in usage, both in relation to its long–term usage and in relation to the usage of other words. □

[34] use Hodrick-Prescott Trend Filtering. They estimate trend components of topics and then figure out whether a topic is emerging by introducing a margin based loss function which penalizes static or decaying topics.

[13] are looking for trending words that co–occur with a product of interest. For them an interesting phrase *should both be mentioned frequently and should be relatively unique to the product with which it is associated.* The significance score of phrase is the ratio of phrase co-occurrence with product to phrase occurrence raised to the power 0.96. The score requires a “global” resource, which in case of Twitter is difficult to acquire.

[4] describe a selection criterion based on several measures: term frequency, term frequency–inverse document frequency and entropy. The entropy formula requires term co–occurrence monitoring, which has a long tail of infrequently used items, see section 5.2.1 for an example.

[28] describe a very simple method of trend detection based on burst detection. The method is based on 3 values. f is the observed frequency of a term in a window of length n. μ is the mean of the frequencies of the term over the windows of length n. σ is the standard deviation of the term frequencies over windows of length n. Given these three values the trending score of the terms is:

$$\frac{f-\mu}{\sigma} \tag{3.1}$$

[1] https://www.freebase.com is a structured collection of 23 million entities.

Search [24, 3] propose a query expansion in searching based on a language modeling framework.

It is usually assumed that the relevance of a document is correlated with the likelihood of a query. So $P(D|Q)$, where D is a document and Q is a query, is considered to be the score of a document in a search result.

The conditional probability can be rewritten applying Baye's rule (formula (3.2), and the probability of the query $P(Q)$ may be ignored for ranking purposes, because it stays constant for all documents (formula (3.3)).

$$\begin{aligned} P(D|Q) &= \frac{P(Q|D)P(D)}{P(Q)} \qquad (3.2) \\ &\propto P(Q|D)P(D) \qquad (3.3) \end{aligned}$$

Assuming that the query terms are independant from each other, and to prevent numerical underflows performing the computations in the log domain, the score becomes:

$$\begin{aligned} P(Q|D)P(D) &\propto P(D) \prod_{t \in Q} P(t|D)^{n(t,Q)} \qquad (3.4) \\ \log P(Q|D)P(D) &\propto \log(P(D)) + \sum_{t \in Q} n(t,Q) \log P(t|D) \qquad (3.5) \end{aligned}$$

where ($n(t,Q)$ is the number of time term t is presented in the query Q).

[3] generalize $n(t,Q)$ so it takes also real values, it can be interpreted as the weight of a term in the query. $n(t|Q)$ is replaced with a query model $P(t|\theta_Q)$. $P(t|D)$ is generalized as a document model $P(t|\theta_D)$.

$$\log P(Q|D)P(D) \propto \log(P(D)) + \sum_{t \in Q} P(t|\theta_Q) \log P(t|\theta_D) \qquad (3.6)$$

If an uniform prior in formula (3.6) is assumed, then the Kullback-Leibler divergence between the query model and the document model provides document ranking:

$$D(\theta_Q||\theta_D) = -\sum_{t} P(t|\theta_Q) \log P(t|\theta_D) + \mathrm{const}(Q) \qquad (3.7)$$

where const, document–independant entropy of the query model, can be ignored. Minimizing formula (3.7) is the same as maximizing formula (3.6).

To overcome data sparseness of $P(t|\theta_D)$ when a term t that does not present in a document make the whole probability in formula (3.4) equal to zero, the document model is built from the empirical estimate of $P(t|D)$ and the estimate of the term, using the coefficient λ:

$$P(t|\theta_D) = (1-\lambda)P(t|D) + \lambda P(t) \qquad (3.8)$$

[24] elaborates the model for Twitter expecting low language reuse in single tweets by applying set semantics to the terms in tweets:

$$\begin{aligned} P(t|D) &= \frac{\hat{n}(t,D)}{\sum_{t' \in D} \hat{n}(t',D)} \qquad (3.9) \\ P(t) &= \frac{\sum_D \hat{n}(t,D)}{N} \qquad (3.10) \\ \hat{n}(t,D) &= \begin{cases} 1 & \text{if } n(t,D) > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (3.11) \end{aligned}$$

where N is the total number of tweets in the collection, $n(t, D)$ is the term frequency of t in a document D.

In addition to [3], [24] adds quality indicators that estimate the global prior probability $P(D)$ (see formula (3.6)). the indicators are bases on:

- the number of times a tweet has been reposted
- the number of followers the author of a tweet has
- the recency of a post, the time a post was created with respect to the query time.

Here is an assumption that the search is performed over a static collection of tweets, thus the recency indicator makes sense, since recent tweets are expected to be shown first. The number of reposts also makes sense, since by the time a query is issued, some of the tweets will be reposted. However, when a query operates on a stream of data, the receny and repost indicators are not useful — the tweets are ranked as soon as they are posted to a social network: all of them are recent, and there is no chance for anyone to make reposts in such a small amount of time.

A search query usually contains few terms, though sometimes users are able to provide few relevant documents. This additional information may be exploited to expand the query terms in Q, coming up with query expansion $\hat{Q}$. So, the query model $P(t|\theta_Q)$ is a linear combination:

$$P(t|\theta_Q) = (1-\mu)P(t|\hat{Q}) + \mu P(t|Q) \tag{3.12}$$

[24] ranks terms for query expansion by the score:

$$\text{score}(t, Q) = \log\left(\frac{|N_c|}{|\{d : t \in D, d \in N_c\}|}\right) \sum_{d \in \{N_c : q \in Q \wedge t, q \in D\}} e^{-\beta(c - c_d)} \tag{3.13}$$

where c is query submission time, c_d is post's creation time, N_c is the set of posts that are posted before c. β controls the contribution of each post based on its creation time.

3.2 Twitter and the Streaming API

The Twitter Streaming API offers two ways of data collection.

The **GET statuses/sample** API resource,[2] on the moment of writing, provides approximately 1% of all public tweets. This resource might be useful for exploring global trends, though is not promising for analysis of smaller local events, which most probably are filtered out.

The **POST statuses/filter** entry point[3] returns public statuses that match one or more filtering predicates. There are 3 types of filtering predicates:

- **follow**[4] A list of user IDs. The resulted stream will include the tweets created by the listed users.
- **track**[5] A list of phrases. Tweets containing mentioned phrases are included to the stream.
- **locations**[6] A list of coordinates. The stream will include the tweets from the defined locations.

The **POST statuses/filter** entry point is useful for discussion retrieval, since it can be queried for tweets with desired properties.

[2] https://dev.twitter.com/docs/api/1/get/statuses/sample
[3] https://dev.twitter.com/docs/api/1/post/statuses/filter
[4] https://dev.twitter.com/docs/streaming-apis/parameters#follow
[5] https://dev.twitter.com/docs/streaming-apis/parameters#track
[6] https://dev.twitter.com/docs/streaming-apis/parameters#locations

3.3 Methodology

Given the Twitter Streaming API[7] **POST statuses/filter** method the system has to explicitly query the keywords of interest.

In the beginning the system is given the core stream of a discussion. It can be formed by tweets that contain certain phrases, hashtags or user names. For example, to collect the tweets about Lowlands, the core stream could contain the tweets that contain the word *lowlands* or tweets written by the user `@Lowlands_12`. The included words should be chosen carefully. In the example, `#ll` should not be included, because then the core stream will contain tweets unrelated to Lowlands, section 2.2 describes hashtag ambiguity.

The only way to retrieve the rest of the discussion given a core stream is to query Twitter for more keywords. However, it is not guaranteed that incoming tweets will belong to the discussion mainly due to ambiguities. Thus the are the following streams of tweets in the system:

- **The core stream** is a stream of tweets that is given to the system and is guaranteed to be a part of the discussion.
- **The noisy stream** contains the tweets that are requested by the system. This stream contains relevant and irrelevant to the discussion tweets.
- **The discussion stream** is made of the tweets in the core stream and the relevant tweets from the noisy stream.

Definition 2 (Filtered stream of tweets) Let $P = \{p_1, \dots, p_n\}$ be a set of n distinct filtering predicates. Then $S(P) = t_1, \dots, t_m$ is a filtered stream of m tweets, such that every tweet in the stream satisfies at least one filtering predicate in P. □

In the system, the noisy stream is a filtered stream.

Definition 3 (Discussion stream) Given a core stream of tweets S and a filtered stream $S(P)$, the sequence of tweets $S(S, S(P))$ is the discussion stream of S with $S(P)$ which consists of tweets that are relevant to the discussion that is described by S. □

In case of collecting tweets about Lowlands, the set up could be the following: S would be the core stream and it would contain the tweets with the word *lowlands* and the tweets that are written by the user `@Lowlands_12`. $P = \{\texttt{wombats}, \texttt{\#ll}\}$. The discussion stream would consist of the tweets in S and some tweets in $S(\{\texttt{wombats}, \texttt{\#ll}\})$.

For successful discussion retrieval, the system has to be able to a) decide whether a tweet from the noisy stream is relevant to the discussion and should be included to the discussion stream. In addition, since P is empty in the beginning the system has to b) update P with words that tweets relevant to the discussion are likely to contain.

3.4 Discovering potential keywords

Suppose we are given a core stream S and we need to provide a list of keywords P that tweets relevant to the discussions are likely to contain.

The discussion is going to evolve trough the time: new topics will appear, other topics will not be discussed anymore. The core stream will mirror this development by the change of word frequency usage. Emergence of a new topic will give rise to the usage of certain words.

The words that experience an increase of usage over time are called trending words. There exist several methods on how trends are detected [13, 34, 4] which apply different mathematical methods.

The method proposed in [28] is very effective to implement because the mean and the standard deviation (formula (3.1)) can be accumulated without storing the frequencies of a term for every window. This contrasts to formula (3.13) where iteration over documents is needed to compute a score of a term.

[7] Even though the method is based on the Twitter Streaming API limitations, it may be applied in a situation when the access to a resource is not limited. Since the method queries only a part of the global stream, the amount of obviously irrelevant data is reduced, and less computational resources are required to process the data.

For the mean, the accumulated sum of frequencies and the number of observations has to be stored. For the standard deviation, the sum of frequency squares has to be stored in addition to the information about the mean.

The system evaluated in chapter 4 implements Burst detection as described in [28].

3.5 Including a tweet in a discussion

The task is to distinguish relevant tweets from irrelevant tweets regarding a discussion. The system needs to decide whether a message in the noisy stream is part of the discussion.

A relevant candidate tweet discusses one of the topics that are in the core stream, but an irrelevant tweet's topic won't be discussed in the core stream. Thus the decision should be based on the content of a tweet, on it's topics. Topic models in general [7] and LDA in particular are highly applied methods for topic induction.

A model is trained on the core stream. For every tweet in the noisy stream, its topic distribution is predicted. So, the model returns a probability distribution $p_1, \ldots, p_n$ where n is the number of topics and for $0 < i \leq n$: p_i is the probability of ith topic. Then the entropy of prediction is:

$$-\sum_{i=1}^{n} p_i \log p_i$$

For a uniform prediction, when the model can not associate an incoming tweet with a small amount of topics, the entropy value will be high. For a skewed distribution, when the model is confident and is willing to assign a small amount of topics to the incoming tweet, the entropy is low. The system in chapter 4 includes a tweet to the discussion if its entropy is lower than a predefined threshold.

Chapter 4

Evaluation

This chapter evaluates the proposed discussion retrieval method. The system has to retrieve relevant tweets from the noisy stream given the core stream.

4.1 The data set

The system is tested on a collection of tweets about the UEFA European Football Championship 2012[1]. During the tournament tweets that contain the hashtag `#euro2012` or are created by the user `@uefacom` were gathered. Manual inspection of the collected tweets showed that the hasthag, at least during the championship, was unambiguous: most of the collected tweets are related to football.

The data set consists of tweets collected during the games in the tournament. For every game the data collection started half an hour before the match and finished half an hour later. Appendix A shows in details the collected tweet distribution in time during the championship in general and every game in particular.

For the simulation the data set has to provide two streams: a stream of tweets about the event of interest, this stream is referred as the core stream, and another stream of tweets which are either related to the discussion or not, this stream is related as the noisy stream. The two streams are disjoint – there is no message which belongs to both streams. Figure 4.1 graphically shows how the streams were formed from the collected tweets.

The core stream consists of 2,097 tweets randomly sampled from the collection of tweets about the game Germany versus Italy.

The nosy stream includes the rest 2,171 tweets about the game Germany versus Italy – denoted on the diagram as the unseen stream – and 100,265 tweets about other games.

The creation time of the tweets was replaced with the time relative to the beginning of data collection. For example, the creation time of a tweet posted at 16:33 UTC on 8th June 2012 is set to 01:03, because the data collection of the game Poland versus Greece started at 15:30 UTC. Such alignment of the tweets merges the tweets of the games, which are scattered in time, to one short, but intensive stream.

The game Germany versus Italy The semi-finals game Germany vs. Italy took place at the National Stadium Warsaw in Warsaw on 28th of June 2012 at 20.45 CET[2].

`@UEFAcom` posted all key match events during the game. Table 4.1 shows the tweets by `@UEFAcom` announcing the key events.

While tweets produced by `@UEFAcom` are structured and are easy to be parsed, tweets generated by users are much more diverse. Table 4.2 shows tweets written by non official users are much less structured and more informal.

[1] http://www.uefa.com/uefaeuro/season=2012

[2] http://www.uefa.com/uefaeuro/season=2012/matches/round=15174/match=2003379/postmatch/index.html

ID	Time	@UEFAcom tweet
1	00:32	GER 0-0 ITA Kick-Off #GERITA #EURO2012 http://uefa.to/NSfroA
2	00:52	GER 0-1 ITA Goal: #Balotelli (20) #GERITA #EURO2012 http://uefa.to/NSfroA
3	01:08	GER 0-2 ITA Goal: #Balotelli (36) #GERITA #EURO2012 http://uefa.to/NSfroA
4	01:09	GER 0-2 ITA Yellow card: #Balotelli (37) #GERITA #EURO2012 http://uefa.to/NSfroA
5	01:18	GER 0-2 ITA End 1st half #GERITA #EURO2012 http://uefa.to/NSfroA
6	01:33	GER 0-2 ITA Substitution: #Klose In #Gomez Out (46) #GERITA #EURO2012 http://uefa.to/NSfroA
7	01:34	GER 0-2 ITA Substitution: #Reus In #Podolski Out (46) #GERITA #EURO2012 http://uefa.to/NSfroA
8	01:34	GER 0-2 ITA Start 2nd half #GERITA #EURO2012 http://uefa.to/NSfroA
9	01:46	GER 0-2 ITA Substitution: #Diamanti In #Cassano Out (58) #GERITA #EURO2012 http://uefa.to/NSfroA
A	01:49	GER 0-2 ITA Yellow card: #Bonucci (61) #GERITA #EURO2012 http://uefa.to/NSfroA
B	01:52	GER 0-2 ITA Substitution: #Motta In #Montolivo Out (64) #GERITA #EURO2012 http://uefa.to/NSfroA
C	01:58	GER 0-2 ITA Substitution: #DiNatale In #Balotelli Out (70) #GERITA #EURO2012 http://uefa.to/NSfroA
D	02:00	GER 0-2 ITA Substitution: #Müller In #Boateng Out (71) #GERITA #EURO2012 http://uefa.to/NSfroA
E	02:12	GER 0-2 ITA Yellow card: #DeRossi (84) #GERITA #EURO2012 http://uefa.to/NSfroA
F	02:18	GER 0-2 ITA Yellow card: #Motta (89) #GERITA #EURO2012 http://uefa.to/NSfroA
G	02:21	GER 1-2 ITA Goal: #Özil (92, p) #GERITA #EURO2012 http://uefa.to/NSfroA
H	02:22	GER 1-2 ITA Yellow card: #Hummels (94) #GERITA #EURO2012 http://uefa.to/NSfroA
I	02:23	GER 1-2 ITA End 2nd half #GERITA #EURO2012 http://uefa.to/NSfroA

Table 4.1: Germany versus Italy match events and the corresponding tweets. Time is the relative time of the tweet since the data collection start.

4.2 Evaluation metrics

Precision and recall are widely adopted measures for document retrieval tasks.

Definition 4 (Precision) Let A be the set of the relevant documents and B be the set of the retrieved documents, then precision is $p = \frac{|A| \cap |B|}{|B|}$. □

Definition 5 (Recall) Let A be the set of the relevant documents and B be the set of the retrieved documents, then recall is $r = \frac{|A| \cap |B|}{|A|}$. □

Since the system includes two stages: filtering and classification, it worth to measure performance of each stage. Filtering aims to retrieve as many tweets as possible maximizing recall. The second stage, classification, filters out unrelated tweets and maximizes precision. Thus precision and recall were measured after filtering and classification. These two numbers represent intermediate and final performance of the system. Also, the performance of the classifier is measured in isolation (see table 4.3).

In the example on the figure 4.1, there are 6 documents in the unseen stream, filtering retrieved 4 tweets, 2 of them are relevant, classifier returned 3 tweets, one tweet is relevant.

User	Tweet
@AlexSnowww	We watch today a football ! #EURO2012 #GERMANY #ITALIA
@joeeeschiavone	C'mon #Italy let's get to the final, tough German side. #Euro2012
@ABCiE	#Euro2012. I want to see Germany in the finals. Come on Deutschland
@NorthEastCorner	Italy appear to be managed by Bob Mortimer! #euro2012
@Snowietiger	#euro2012 can u germans win faster so we can see überhuman Deutsch precision vs Spanish ninjavoodoo in the final?
@abanvini	Buffon has really been kept busy by the German machine #Euro2012
@paulmwatson	That's a pretty good goal that by the Italians. #Euro2012
@drewmcdevitt	Wow, unexpected #supermario #Euro2012
@KoptimusPrime	Great goal.Fantastic cross #Euro2012
@dunn_sam9	Italy to play Spain in the final #Euro2012 Mario Balotelli #legend

Table 4.2: Selection of tweets during the game Germany versus Italy.

Precision and recall of filtering (p_f, r_f), classification (p_c, r_c) and the whole system (p, r) are:

$$p_f = \frac{2}{4} = 0.5 \qquad p_c = \frac{1}{3} = 0.3 \qquad p = \frac{1}{3} = 0.3$$

$$r_f = \frac{2}{6} = 0.3 \qquad r_c = \frac{1}{2} = 0.5 \qquad r = \frac{1}{6} = 0.17$$

4.3 Experiments

4.3.1 The baseline

Setup The trend detection component captured word frequencies over 15 15-minute long windows, so words' trending scores were computed every minute. The mean and the standard deviation values of the word frequencies were updated every 15 minutes. A word had to have a trending score of more than 20 to be considered as trending.

The LDA model was trained on the tweets from the core stream. Every minute the model was retrained on the already collected and newly arrived tweets. The number of topics was set to 50. The decision whether a tweet from the noisy stream belonged to the discussion was based on the distribution of topics the model predicted. The entropy of the estimated topic distribution had to be less than 1.2.

Hashtags were not distinguished from words, for example, `#euro2012` was seen by the system as *euro2012*.

[32] LDA implementation was used in the experiment.

Results The system retrieved 321 relevant and 468 irrelevant tweets from the unseen stream and from the other game streams respectively. Precision was 0.41, recall was 0.15. Table 4.3 sums up the experiment results. Figure 4.2 shows precision and recall scores for various entropy threshold values.

Figure B.1 shows the component reaction to the game events. Trending words appeared after almost all major events such as goals and game interrupts. The second goal by Balotelli was not noticed, because it was his second goal and his name had been already tracked. Not all minor events, such as substitutions and yellow cards, were recognized. The results are similar to the results reported in [23], their system also failed to capture minor events.

Interestingly, many trending words are hashtags that represent player surnames. The series of the substitutions in the beginning of the second half made `#Klose`, `#Gomez`, `#Reus` and `#Podolski` trending. Notice that `@UEFAcom` referred to players using hashtags, and since `@UEFAcom` has many followers and its tweets are often retweeted that could be the reason these hashtags got trending. Majority of the trending words were used by `@UEFAcom`.

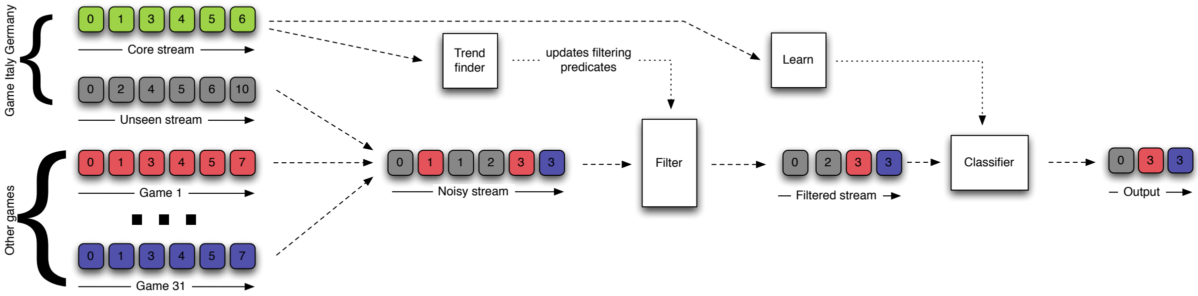

Figure 4.1: Experimental setup Rounded rectangles represent tweets, numbers represent their creation time. Green are the tweets in the core stream. Grey are the tweets in the unseen stream. Other colors represent other games. Rectangles are components of the system. Dashed arrows are tweet flows, so the core stream is consumed by the trend finder and the model learning component. Dotted arrows represent other information exchange.

4.3.2 Hashtags are different from words

Hashtags are labels that tag a tweets with topics, or categories the tweet belongs to. The main point of their existence is to facilitate searching in Twitter. Also, users may monitor discussions of interest, which are supposed to be have dedicated hashtags.

Tweet categorization is accomplished in various ways. For instance, to indicate that a tweet is related to the weather, one can write a tweet like this: *It's exceptionally sunny in Groningen.* `#weather` But, other person could make accent that the tweet is related to Groningen: *It's exceptionally sunny in* `#Groningen`.

While in the first case, the hashtag only categorizes the tweet, in the second example, `#Groningen` is also a part of the message, without it, the sentence will be incomplete. So hashtags are not used exclusively for classification.

[19] evaluated characteristics of Twitter hashtags by the following properties:

- **Frequency** The number of tweets labeled with a give hashtag and the number of users that used it.
- **Specificity** The measure that shows how the meaning of a hashtag is different from the meaning of a *non-tag*, the corresponding word without a hashtag.
- **Consistency in usage** Since one of the goals of the study was to link hashtags and strong identifiers in the Semantic Web, the authors measured the variety of usage contexts of a hashtag.
- **Stability over time** This measure shows what hashtags have recently appeared.

They showed that not all hashtags are used in the same manner and have the same quality. Clearly, user provided classification can help topic inference. But it would be helpful to take apart different hashtag categories and relay only on hashtags that only mark a topic. However, since not all hashtags are of the same quality, only high quality, non ambiguous and specific hashtags should be used.

Setup The experiment is based on the baseline settings, but hashtags were considered to be different from words.

Results The system retrieved 298 documents from the unseen stream and 328 from the stream of other games. Filtering retrieved 802 documents from the unseen stream and 19,686 documents from the stream of other games.

Less words were considered trending in comparison to the baseline. Only hashtags used by `@UEFAcom` became trending showing that the user was using them consistently. Filtering recall lowered.

Classification performed slightly better then in the baseline experiment, though the whole system performance stayed at the same level. Note that 298 relevant tweets were retrieved versus 321 in the baseline experiment.

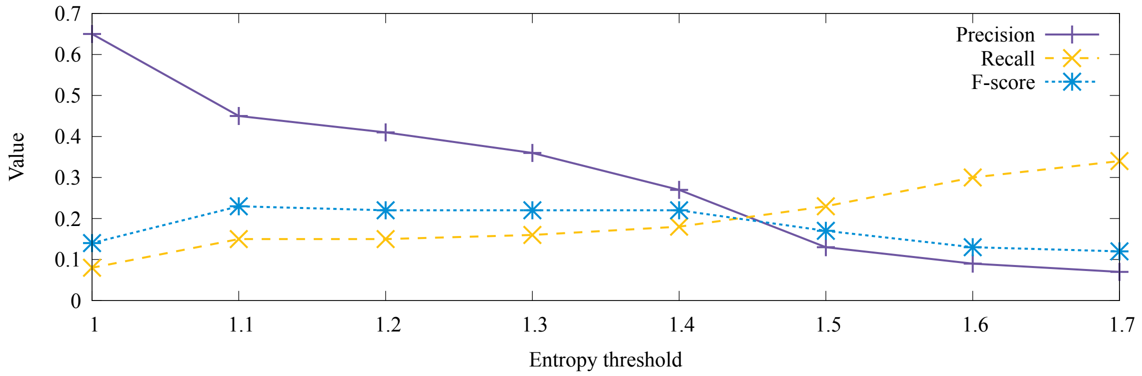

Figure 4.2: Performance for different entropy values. F–score is the harmonic man of precision and recall: $F = \frac{2\cdot\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}}$

	Filtering		Classifier		System	
Experiment	Precision	Recall	Precision	Recall	Precision	Recall
Baseline	0.047	0.65	0.41	0.23	0.41	0.15
Hashtags are different from words	0.039	0.37	0.48	0.37	0.48	0.14
Without retweets	0.058	0.16	0.1	0.046	0.1	0.0073
Primavera Sound	0.26	0.86	0.83	0.15	0.83	0.13
Rock am Ring	0.49	0.87	0.91	0.11	0.91	0.094
Les Eurockeennes de Belfort	0.27	0.75	0.94	0.2	0.94	0.15
Rockwerchter	0.49	0.66	0.98	0.17	0.98	0.11
Pukkelpop	0.24	0.67	0.73	0.099	0.73	0.066
Lowlands	0.64	0.8	0.97	0.071	0.97	0.071
Pentathlon	0.0074	0.84	0.17	0.14	0.17	0.12

Table 4.3: Experiment results

4.3.3 Without retweets

The main functionality of a retweet is to spread a tweet being retweeted among followers. From this perspective, retweets are not useful for discussion retrieval, because the original tweet most probably will be retrieved as well.

On the other side, a reason a person wants to share a particular tweet is that the tweet is considered to be important and worth being shared. Then retweets indicate “importance” of the original tweet. [8] show that retweets sometimes contain new information, often to begin a conversation.

However, in the current setting retweets of a tweet form a cluster of many almost identical items, leading the model away from inferencing the actual discussion topics, but concentrating on retweets.

Taking this into account, retweets should be treated differently than plain tweets, however, there is no way to process retweets differently in the implemented system, thus they are ignored in this experiment.

Setup The setup is based on the baseline experiment but the retweets were totally ignored in the dataset. There were 1,738 tweets in the core stream, 1,780 tweets in the unseen stream and 84,606 tweets about other games.

Results The system retrieved 13 documents from the unseen stream and 115 from the stream of other games. Filtering retrieved 285 documents from the unseen stream and 4,654 documents from the stream of other games.

Filtering queried for much less tweets. Classification performance drastically dropped indicating that the model could not reliably detect topics.

Reasons of performance decrease are lower amount of data: 88,124 versus 104,533 tweets; and change of the content: retweets introduce many almost identical items that are much easier to cluster.

It is useful to include tweets if the further analysis is focused on quantitative results, for example, how many users took part in the discussion, what are the most influential users and what are the most important messages.

Retweets will not help much if the goal of the later analysis is to provide content insights of the data, for example, topics that were discussed.

4.3.4 Primavera Sound and Rock am Ring

The data set based on the tweets about Euro 2012 (section 4.1) is rather challenging because the corpus shares the same topic — football. Classification whether a tweet belongs to the discussion has to be well tuned. In the previous experiments, classification precision is less than 0.5.

To evaluate the system in different setting more evaluation datasets were built. In contrast to the UEFA experiments, the tweets are not merged together to produce the noise stream. Instead, the noisy stream consists of tweets about some similar event that happened at the same time as the event of the discussion of interest.

Setup Tweets about two rock festivals were collected: Rock am Ring in Germany, and Primavera Sound in Spain. Both events took place in the first weekend of June 2012. 27,495 and 16,351 tweets were collected from May, 28 to June 3 about Rock am Ring and Primavera Sound respectively.

To be a part of the *Primavera Sound* collection, a tweet has to contain the phrase *primavera sound* or has to be written by the user `@Primavera_sound`. To belong to the *Rock am Ring* stream, a tweet has to contain the phrase *rock am ring* or be written by `@rockamringblog`.

The task was given half of the tweets about the events, find the other half, tweets about the other events formed the noise. Thus two experiments were run: in the first run the system had to retrieve the hidden tweets about Rock am Ring, in the second run the system had to get tweets about Primavera Sound.

The classification model was updated every half an hour. Other system parameters were not altered and stayed the same as in the previous experiments.

Results

Primavera Sound The system retrieved 1,098 documents from the unseen stream (in total there were 8,239 tweets in the unseen stream) and 219 from the stream of Rock am Ring (27,495 tweets in total). Filtering retrieved 7,125 documents from the unseen stream and 20,353 documents from the stream of Rock am Ring.

Rock am Ring The system retrieved 1,292 documents from the unseen stream (in total there were 13,679 tweets) and 131 from the stream of Primavera Sound (16,351 tweets in total). Filtering retrieved 11,956 documents from the unseen stream and 11,394 documents from the stream of Primavera Sound.

Both filtering precision and recall are higher than the baseline experiment. The reason for this improvement is that the noisy streams is comparable in size to the discussion: tweets, which was not the case in the baseline experiment, where the number of noisy tweets (100,265) was almost 20 times more that the discussion size (4,268 tweets).

Classification also increased in comparison to the baseline. Even though the discussion and the noise streams are about music festivals, the events took place in different countries, no artist performed at both concerts at the same time. All this made the streams more different from each other.

The only measures that decreased are classification (and consequently classification) recall. Since the discussion is broader, it includes many bands. The entropy threshold 1.2 is too conservative for such a discussion type.

The results for Rock am Ring are better in comparison to Primavera Sound because there are more Rock am Ring tweets than tweets about Primavera Sound.

4.3.5 Rockwerchter and Les Eurockeennes de Belfort

Setup This is another experiment based on music festivals. Rockwerchter is a Belgian festival, Les Eurockeennes de Belfort is a French festival.

The *Rockwerchter* stream consist of tweets contained of the word *rockwerchter* or created by the user `@RockWerchter`. There are 3,318 tweets in this stream.

The *Les Eurockeennes de Belfort* stream's tweets contain either the word *eurockeennes* or *eurockéennes*, or are written by `@eurockeennes` or `@EurockeennesUK`. There are 3318 tweets in this stream.

Results

Rockwerchter The system retrieved 308 documents from the unseen stream (in total there were 3,318 tweets) and 8 from the stream of Le Eurockeennes de Belfort (3,348 tweets in total). Filtering retrieved 2,226 documents from the unseen stream and 2,289 documents from the stream of Le Eurockeennes de Belfort.

Les Eurockeennes de Belfort The system retrieved 259 documents from the unseen stream (in total there were 1,687 tweets) and 18 from the stream of Rockwerchter (6,620 tweets in total). Filtering retrieved 1,269 documents from the unseen stream and 3,406 documents from the stream of Rockwerchter.

The evaluation measures are similar to the result of the experiment in section 4.3.4.

4.3.6 Pukkelpop and Lowlands

Setup The experiment is based on the data about Lowlands and Pukkelpop in 2012, and is similar to the experiment in section 4.3.4.

16,351 tweets made up the *Pukkelpop stream*. A tweet of the stream has to contain the word *pukkelpop*.

31,584 tweets about Lowlands were collected. The tweets contained either the word *lowlands*, or are written by `@Lowlands_12` or `@Rapid_Razor_Bob`.

Results

Pukkelpop The system retrieved 454 documents from the unseen stream (in total there were 6,891 tweets) and 170 tweets from the stream of Lowlands (31,584 tweets in total). Filtering retrieved 4,588 documents from the unseen stream and 14,660 documents from the stream of Lowlands.

Lowlands The system retrieved 890 documents from the unseen stream (in total there were 15,818 tweets) and 29 tweets from the stream of Pukkelpop (13,618 tweets in total). Filtering retrieved 12,586 documents from the unseen stream and 7,171 documents from the stream of Lowlands.

4.3.7 Pentathlon

The discussion in this experiment is about the modern pentathlon competitions at the Olympic games on August 11–12.

Setup The stream about modern pentathlon was build by tweets that contain words *pentathlon*, *modpen* or written by `@UIPM_HQ`. 30,853 tweets were included to this stream.

The noise was formed by tweets with words *olympic* (*olympics* is also matched by the Twitter API in this case), *london2012* (and also the hashtag `#london2012`), however the tweets that were also in the modern pentathlon stream were excluded. 1,996,585 tweets formed the noise.

Results The system retrieved 1,817 documents from the unseen stream (in total there were 15,345 tweets) and 9,142 tweets from the stream of Olympics. Filtering retrieved 12,846 documents from the unseen stream and 1,727,377 documents from the stream of Olympics.

Low performance is due to the difference in stream size: about 30 thousand in the pentathlon stream and 2 million in the Olympics stream.

In general, performance of the system highly depends on the dataset. Some of them a too simple or difficult to retrieve the discussion. No human evaluation was performed, it might be the case that similarly to [6] the output of our system is meaningful and accurate.

Chapter 5

Future work

5.1 Improving the classification

Experiments in section 4.3 showed that the classification is very conservative and inaccurate: classification precision in the baseline experiment was 0.41, and classification recall was 0.23. There are several potential improvements.

Tweet lexical normalization Tweets are very different from other linguistic resources in a sense that they contain a lot of typos, abbreviations and emotions. The task of lexical normalization is close to spell checking.

[16] perform lexical normalization in three steps

1. **Confusion set generation**. During this step, candidate words are generated given an input word.
2. **Ill–formed word identification**. A word is classified as ill-formed and has to be replaced with a candidate from the corresponding confusion set.
3. **Candidate selection**. The words that are predicted to be informed are replaced with the most appropriate candidate.

[14] investigated the writing conventions among different user groups and proposed a context aware noisy text cleaner. They studied the most common lexical transformations performed by users of various applications: desktop Twitter clients, mobile clients or web based.

The main motivation of sophisticated analysis for lexical normalization is to be able to detect lexical substitutions that are located in the long tail, that is, are performed extremely rare. A small amount of simple rules, however, is sufficient to normalize the most frequent mistakes.

Named entity recognition Named entity recognition is another challenging task for Twitter data. In the discussion retrieval system, named entity recognition may be useful because discussion is usually built around some person, place or event.

Recognized named entities also might help to normalize the content of tweets by substituting different forms of the same named entity with some canonical form.

Adopting the model The core of the classification process, the LDA model, was trained on rather challenging data: the tweets were short and there were only 2,097 of them to train the model.

Probabilistic topic models are heavily used in discovery of topics and classification of large collections of documents [7]. Latent Dirichlet Allocation is one of many such models [15]. It associates words and documents, which both are observed, with topics, which are hidden.

One way of adopting the model to Twitter data is to assume that a tweet, being a very short entity, is about one topic [38]. In addition, their model distinguishes between background and topic words. Background words are the words that are common and are used in many topics. Once named entities are identified, another model can treat them explicitly as it done in [29].

Every extension of the LDA model requires a custom inference procedure, that has to be described before the model can be used. [27] and later [17] proposed a method that allows one to associate various temporal, geographical and social features with the input documents without designing the inference procedure.

Making the classification more robust The classification is based on entropy of the predicted topic distribution for a tweet in the noisy stream. This is rather crude because not all the induced topics by the model are equally significant.

[2] propose a way to rank topics to find meaningful or important topics. The basic idea is to measure the distance between a topic distribution and a "junk distribution" which gives high probability to stop words. Then if a tweet is predicted to belong to a junk topic, it is not considered to be a part of the discussion.

5.2 Big data crunching challenges

Apart from complex theoretical foundations, a production discussion retrieval system should be effectively implemented.

Appendix C shows the volume of tweets retrieved for selected events in 2012. The amount of tweets received per hour varies from hundreds (most of the music festivals) to thousands (Queens day, a Dutch national festival). These are the possible sizes of the core stream that the system has to deal with.

The proposed method does not need to store all the incoming tweets. Rather it stores word aggregates, but even storing and accessing simple term count on a big scale is a challenging task.

5.2.1 Word and co-occurrence frequency

The majority of the computational approaches to text analysis is based on word frequencies and their co-occurrence frequencies.

Vector space models see documents as vectors where every component corresponds to a certain element [35]. In case of Twitter analysis, an individual tweet or a collection of tweets may be seen as a document. The words of the tweets may be elements. For example, given the dictionary of 4 words *Mary*, *John*, *loves* and *knows*, the tweet *John loves Mary* is represented as the vector $(1, 1, 1, 0)$. Here the first vector component corresponds to *Mary*, the second corresponds to *John*, the third corresponds to *loves* and the last component corresponds to *knows*. The components in the example define the number of times a word occurred in the tweet.

The described approach assumes that a dictionary of the elements is built before documents are converted to vectors. Usually, the dictionary is limited in size. To limit the dictionary, it excludes stop words and words that appear less times than a predefined threshold. In addition, the elements in the dictionary are explicitly mapped to vector components, which are ultimately the integer indexes.

The first challenge in analyzing a stream of documents is that the dictionary is not known beforehand. Moreover, a stream may experience topic shifts, when usage of elements (words) decreases or increases over time. Thus, a dictionary has to be occasionally updated, depending on the element frequencies.

One way of keeping the dictionary up to date is to keep the precise counts of all the elements. However, because the distribution of words in natural languages is Zipfian (figure 5.1 illustrates word frequency distribution for a collection of tweets about Koninginnedag[1]), there will be a long tail of "outliners", the elements that are seen for just a few times and won't be included in the dictionary. The size of the counter will grow at least proportionally to the number of distinct elements in the stream and may be too expensive to keep it in memory [1, 10] and impractical, because most of the elements will not be included to the dictionary.

[1] Queen's Day

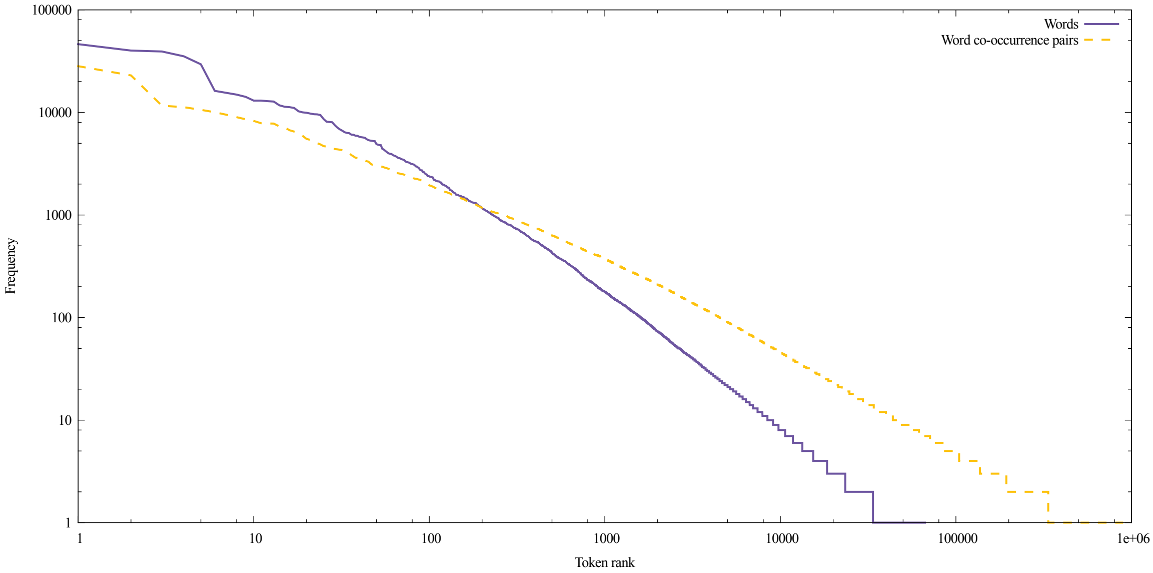

Figure 5.1: Word and word co-occurrence distribution in the collection of tweets about Koninginnedag, a Dutch national holiday. 214769 tweets were collected from February 9 to May 11. There are 1839067 words, 66672 are distinct. There are 1588177 word co-occurrence pairs, 7636752 are distinct.

An alternative solution is to estimate the frequencies.

Definition 6 (Frequency of an element) Given a stream S of n elements $t_1, \dots, t_n$ the frequency of an element i is $f_i = |\{j|t_j = i\}|$. □

To build the dictionary, we need to query the stream to get the *top-k elements* (the k elements with the highest frequencies) [26]. We might be interested in representing a stream of elements as a vector, for this we need the frequencies of the top-k elements as well.

There are 2 approaches to the problem of frequency estimation in the literature [10]. The counter based approach keeps individual counters for elements of the interest. The sketch based method does not monitor a subset of the elements, but tracks all of them, providing frequency estimates of all elements with fewer guarantees.

5.2.2 Count-min sketch

[11] describes a sketch based method for estimating element frequencies.

The main idea is to store frequencies in a $w \times d$ array. w corresponds to the number of columns in the array and is called width. d corresponds to the number of rows and is called depth. The algorithm keeps d different hash functions, each of them is assigned to a row in the array.

To update the sketch, a cell in each row incremented. Each hash function defines which cell to modify in the corresponding row. To avoid the overflow, the hash value is divided by the sketch width, so the remainder points to the column that has to be updated.

To retrieve an element frequency, the minimum count is retrieved from the cells assigned by the hash functions.

[12] extends count-min sketch retrieval procedure, so it considers the “noise” introduced by other counters. This enhancement makes the sketch behave better on the less skewed distributions.

Adoption to the discussion collection task The core of the algorithm are the hash functions. In the setup described in chapter 2 the elements of which frequencies are estimated are strings. It is not possible to have a perfect hash function over strings. However, indexing of the seen word solves the problem. Then the hash of a word is its index.

The size of the index will grow proportionally to the number of distinct elements in the stream (or streams, if the index is a global resource and shared among many stream counters). However, the size of the counter (or counters) can be set in advance depending on the precision the counter has to guarantee.

To count pairs of words (for example, word co-occurrence frequencies) the hash value of a pair of words $h(a, b)$ can be built by concatenation of the hashes of each word in the pair. For example, let indexes be 3 bit unsigned integers. $h(a) = 000$ and $h(b) = 101$, then the hash of the pair (a, b) could be $h(a, b) = 0001101$, the hash of the pair (b, a) could be $h(b, a) = 1011000$. Note that the additional 1 in the middle is required to distinguish the pair (a, b) and just the word b.

This solution limits the number of distinct items that the system is able to track. However, the count-min hash function includes the parameter p, a prime number which is higher than any number being hashed [11]. Setting this value high enough should be sufficient to deal with real world problems.

5.2.3 Counter–based algorithms

In contrast to the sketch algorithms, counter–based algorithm keep in memory a limited number of counters which are associated with some input items [10, 22, 21, 26].

For a newly arrived input item, it is being checked whether there is a counter associated with it. If there is such a counter, it is updated. If it does not exist it is either ignored, or some algorithm-dependent action is performed [26].

Lossy Counting [22] divides the input stream S to buckets of width $w = \lceil \frac{1}{\epsilon} \rceil$, where ϵ is a parameter that controls estimate error. The buckets are assigned bucket ID's, starting from 1. The data structure $\mathcal{D}$ is a set of tuples (e, f, Δ), where e is an element of the stream, f is the estimated frequency, and Δ is the maximum possible error of the estimate f.

In the beginning $\mathcal{D}$ is empty. When a new element arrives, it is checked whether there is a frequency estimation for it in $\mathcal{D}$. If there is a corresponding estimation, it is incremented by one, otherwise, a new entry $(e, 1, b_{curreint} - 1)$ is added to $\mathcal{D}$, $b_{current}$ is the ID of the current bucket.

At bucket boundaries $\mathcal{D}$ is pruned by deleting some of its entries. An entry (e, f, Δ) is deleted if $f + \Delta \leq b_{current}$.

Counter–based algorithms in general take less space and require less computation on item insertion and have provable error bounds. Sketched-based approaches are not affected by the item order in the input stream.

These are suggestions how a discussion retrieval system can be effectively implemented, this techniques were not implemented in 4.

Chapter 6

Conclusion

At the beginning of the XXX Olympic Games in London in 2012 spectators are believed to disturb television coverage of a cycling road race[1]. People tweeted so much about their experience that mobile network could not handle the load. This is just one of many examples on how social media connect the people. Clearly, in the vast amount of produced tweets, there is a certain number of discussions that mirror the event and reveal spectators attitude to the event which is very fruitful to analyze. However, before analyzing the data it has to be collected.

This work proposed a discussion collection method from a social network. The task is split into two subtasks: trending word identification with a goal to be able to query the Twitter streaming API for tweets that contain the trending words; and incoming message classification stage determines whether incoming tweets belong to the discussion or not.

The method was discribed and evaluated. In addition, further improvements and suggestions were given.

[1] http://www.guardian.co.uk/media/2012/jul/29/olympics-2012-twitter-bbc-cycling

Appendix A

UEFA Euro 2012

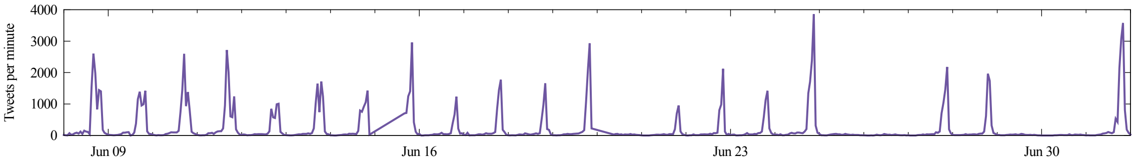

Figure A.1

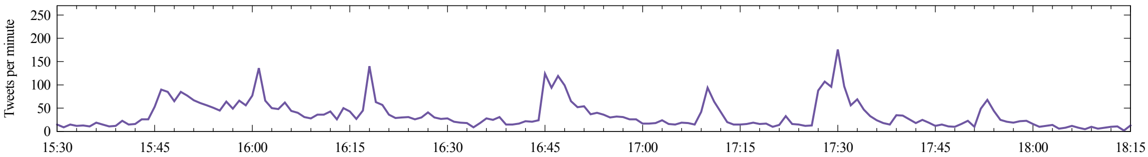

Figure A.2: Poland vs. Greece

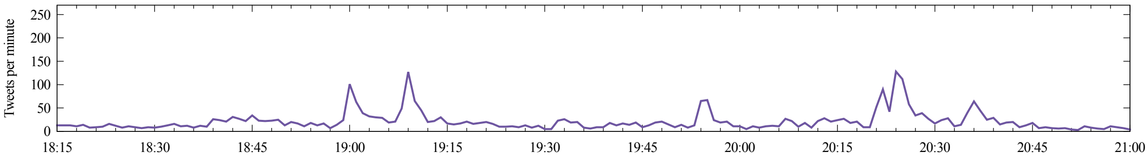

Figure A.3: Russia vs. Czech Republic

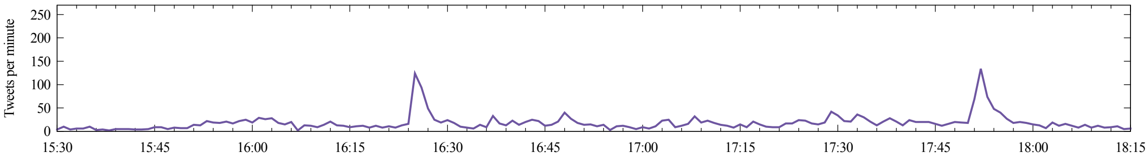

Figure A.4: Netherlands vs. Denmark

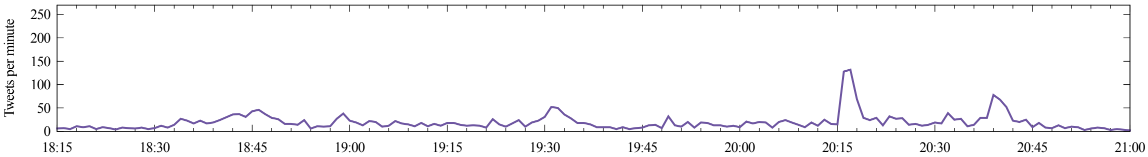

Figure A.5: Germany vs. Portugal

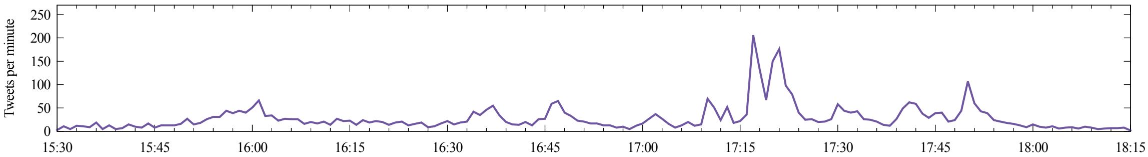

Figure A.6: Spain vs. Italy

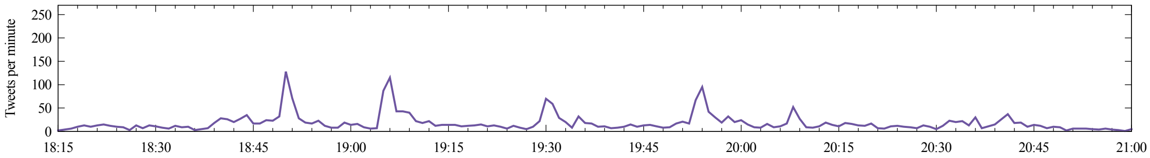

Figure A.7: Republic of Ireland vs. Croatia

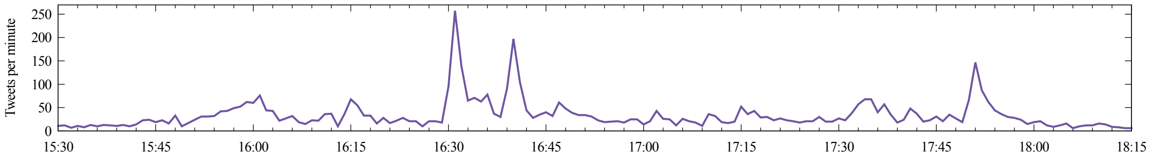

Figure A.8: France vs. England

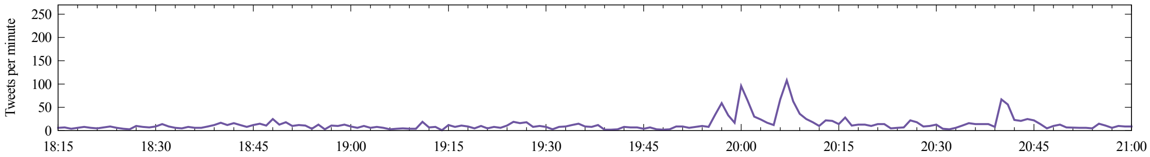

Figure A.9: Ukraine vs. Sweden

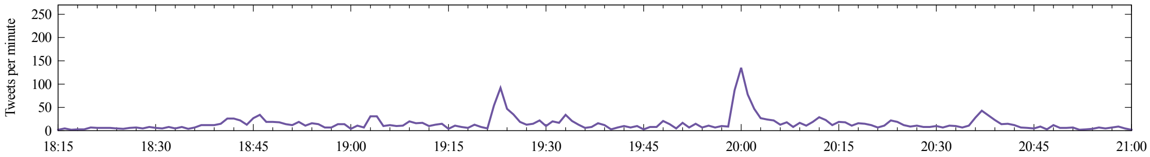

Figure A.10: Poland vs. Russia

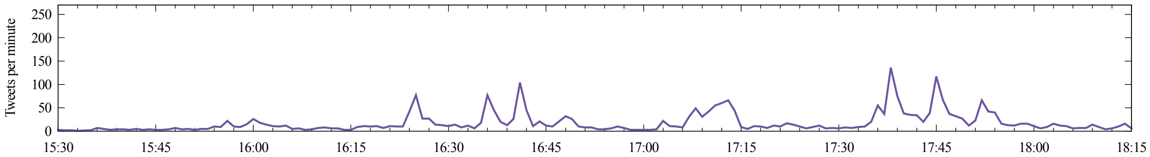

Figure A.11: Denmark vs. Portugal

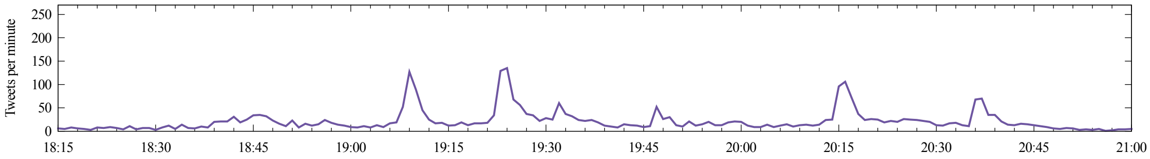

Figure A.12: Netherlands vs. Germany

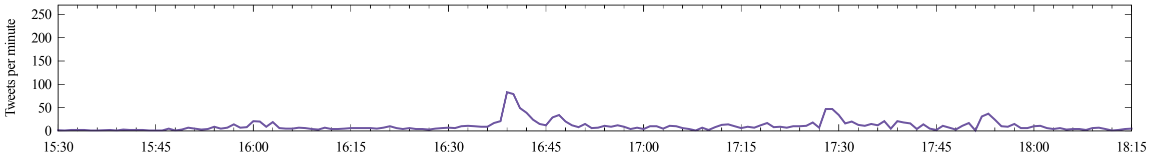

Figure A.13: Italy vs. Croatia

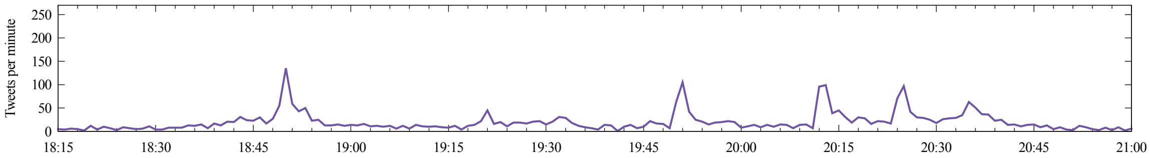

Figure A.14: Spain vs. Republic of Ireland

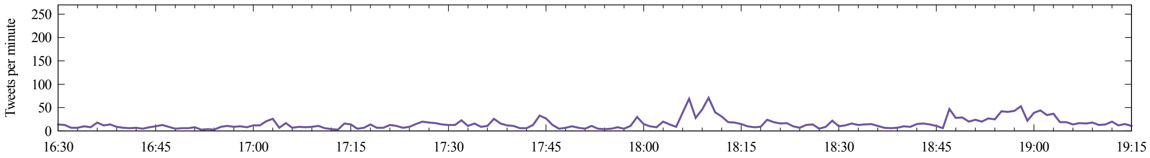

Figure A.15: Ukraine vs. France

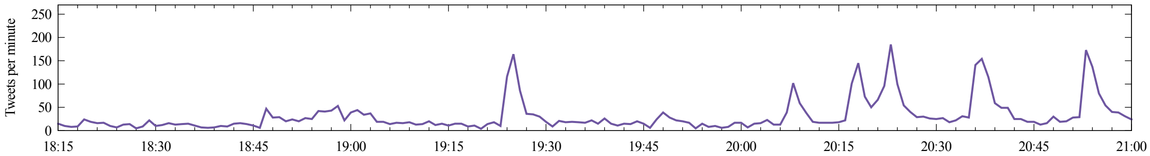

Figure A.16: Sweden vs. England

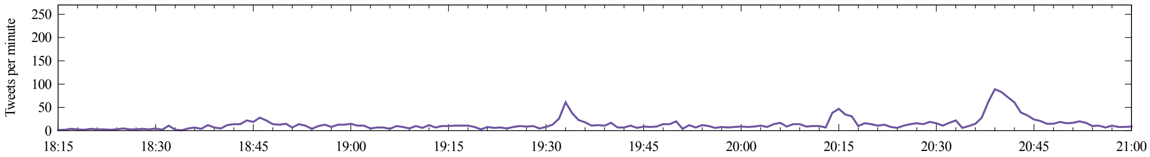

Figure A.17: Germany vs. Russia, Czech republic vs. Poland

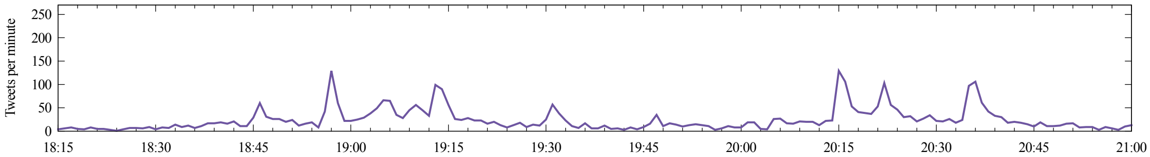

Figure A.18: Portugal vs. Netherlands, Denmark vs. Germany

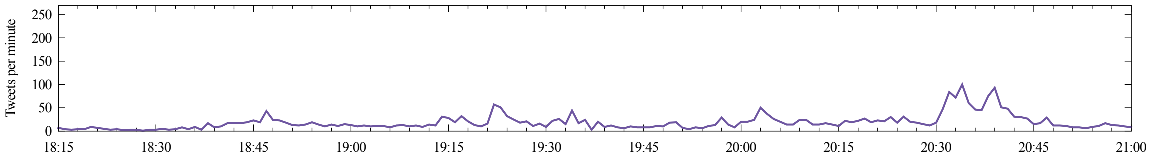

Figure A.19: Croatia vs. Spain, Italy vs. Republic of Ireland

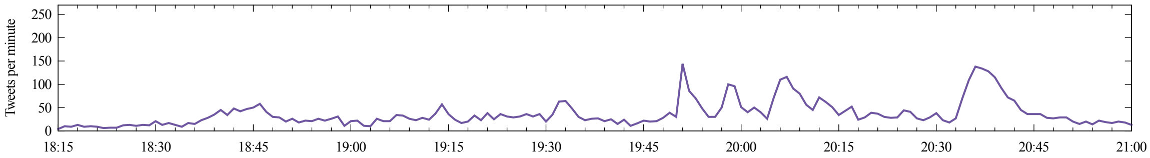

Figure A.20: Sweden vs. France, England vs. Ukraine

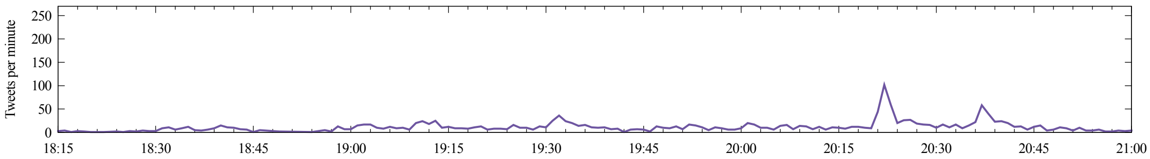

Figure A.21: Czech republic vs. Portugal

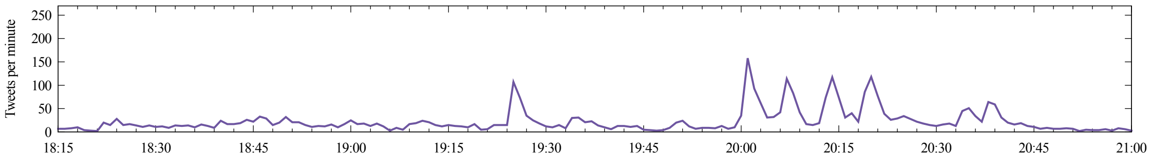

Figure A.22: Germany vs. Greece

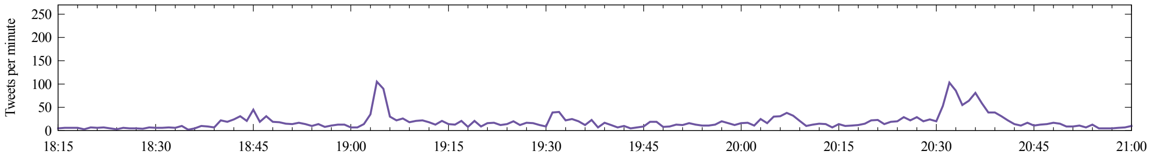

Figure A.23: Spain vs. France

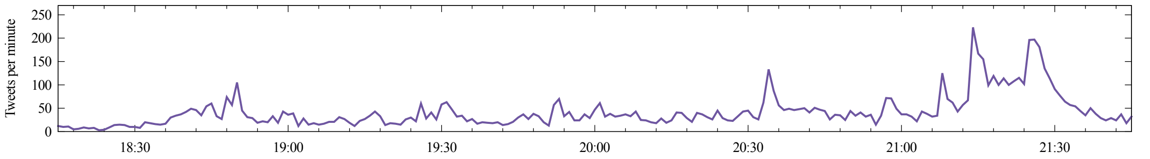

Figure A.24: England vs. Italy

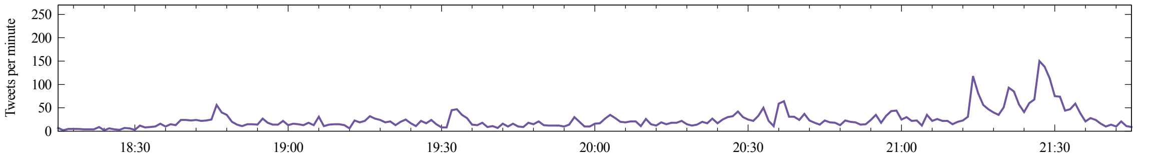

Figure A.25: Portugal vs. Spain

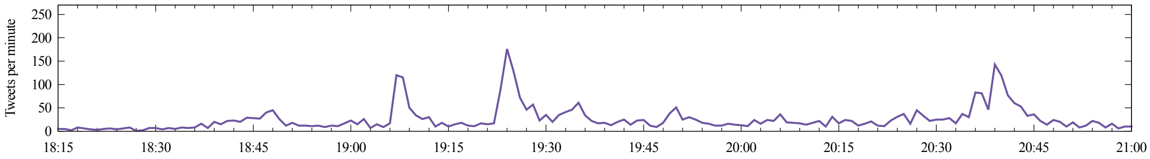

Figure A.26: Germany vs. Italy

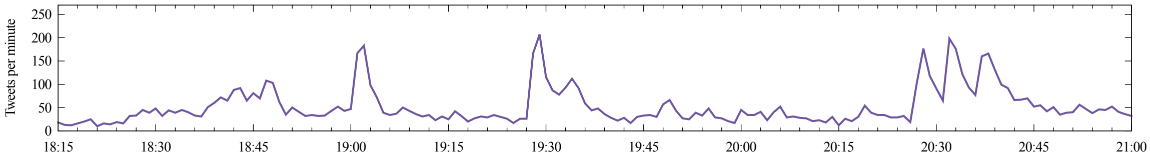

Figure A.27: Spain vs. Italy

Appendix B

System performance

Curves represent the total number of tweets in each stream. Continuous lines correspond to the relevant tweets (higher half of the plot), dashed lines correspond to irrelevant tweets (lower half of the plot). Violet are all the tweets, Yellow: the tweets that were returned by the filtering component. Blue: the tweets that were classified as being part of the conversation. The values were normalized. In the unseen stream there were 2,171 tweets, 100,265 tweets were in the streams of other games.

The grey vertical lines correspond to the game events, see table 4.1. The black vertical lines correspond to the filtering predicate update.

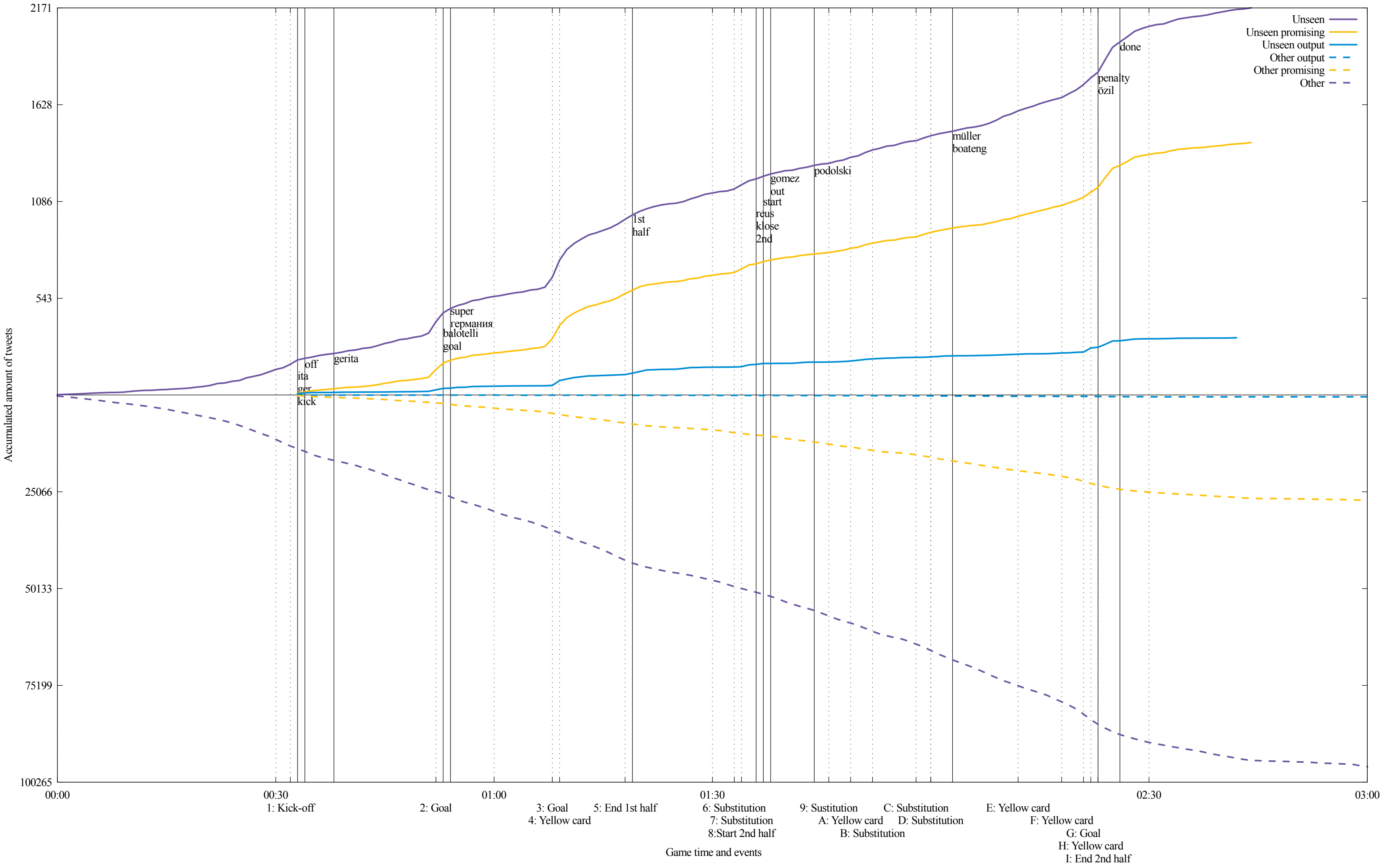

Figure B.1: The baseline (section 4.3.1)

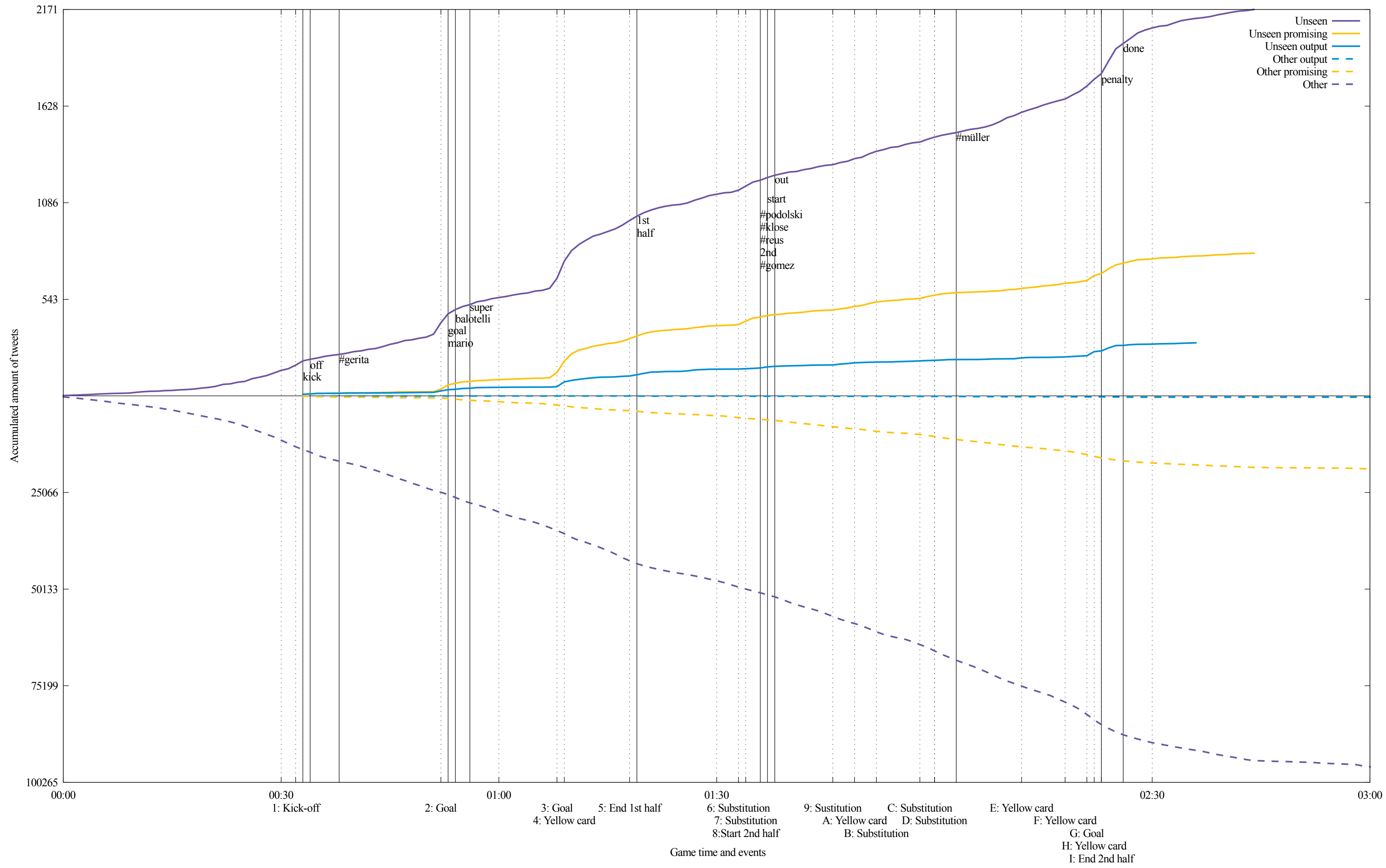

Figure B.2: Hashtags are different from words (section 4.3.2).

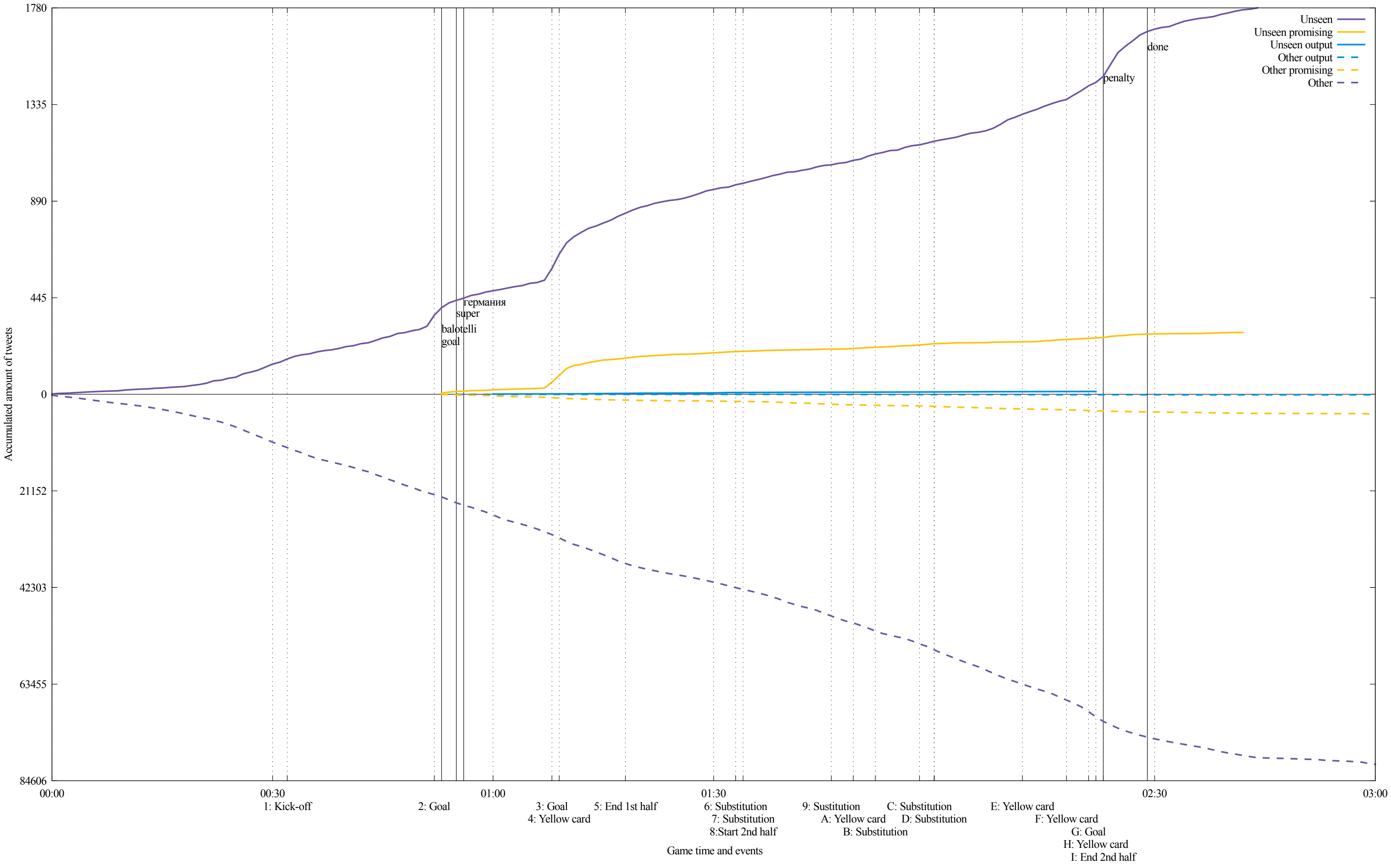

Figure B.3: Without retweets (section 4.3.3)

Appendix C

Events in 2012

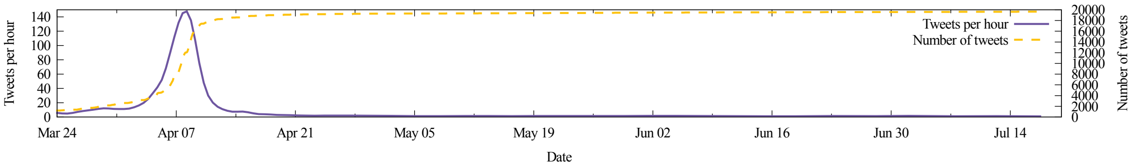

Figure C.1: Paaspop

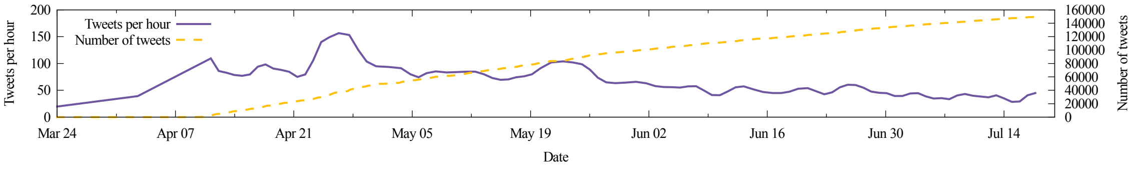

Figure C.2: Le Printemps de Bourges

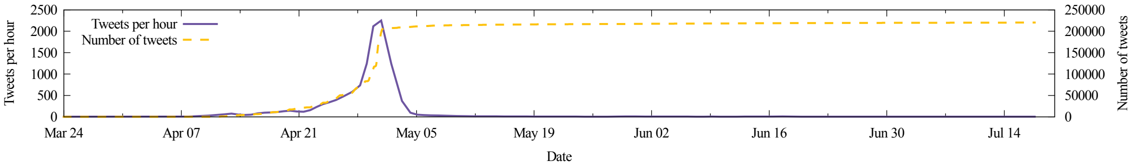

Figure C.3: Koninginnedag

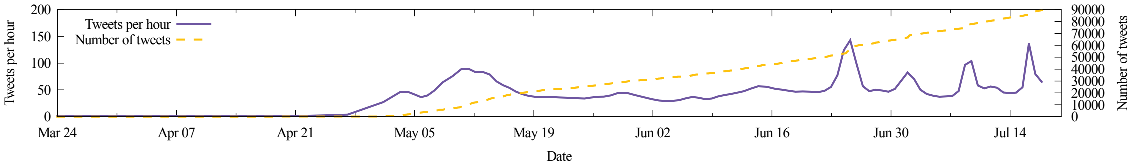

Figure C.4: The Great Escape

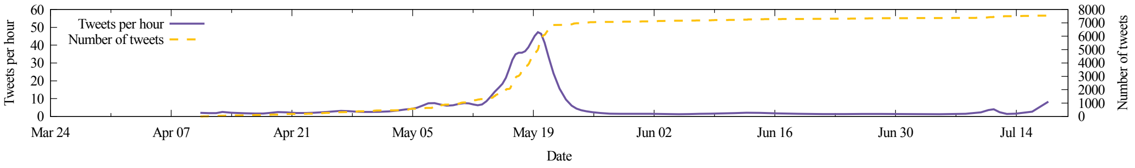

Figure C.5: Nuits Sonores

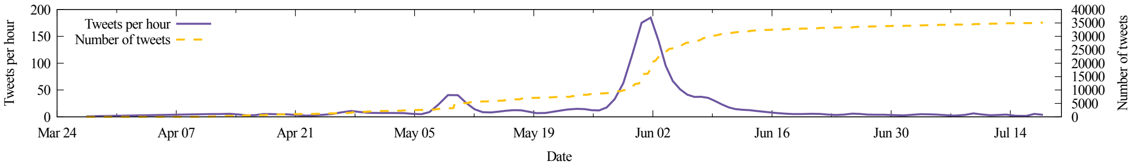

Figure C.6: Primavera Sound

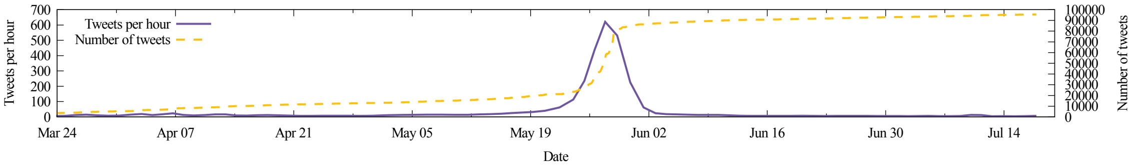

Figure C.7: Pinkpop

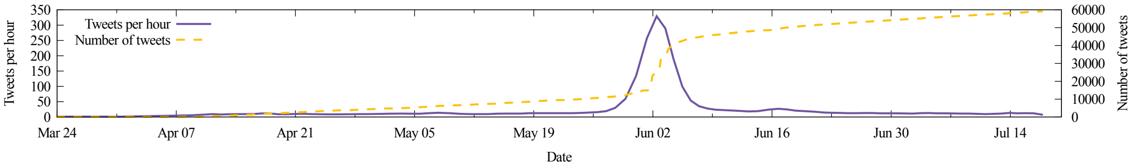

Figure C.8: Rock am Ring

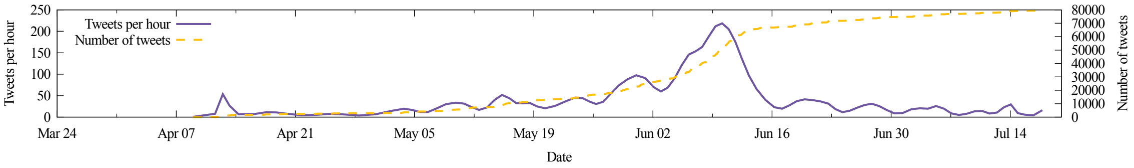

Figure C.9: Downloadfest

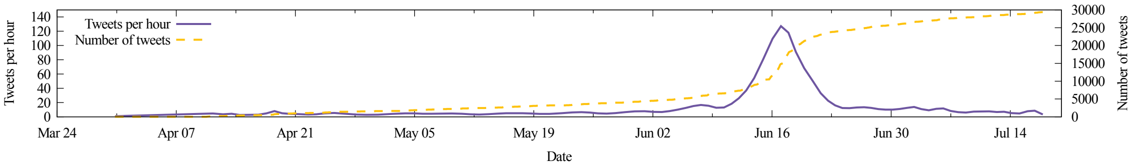

Figure C.10: Hellfest

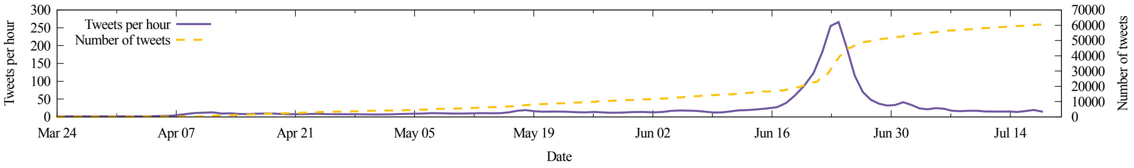

Figure C.11: Defqon

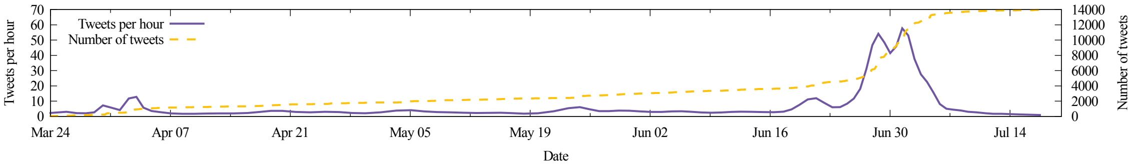

Figure C.12: Rockwerchter

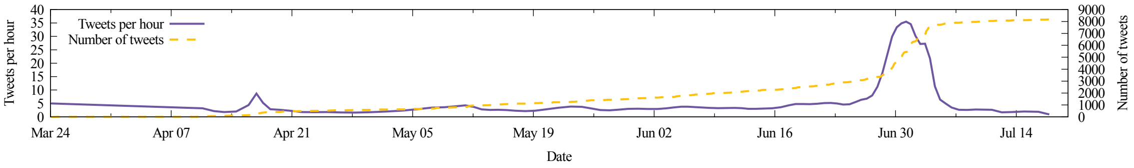

Figure C.13: Le Eurockeennes de Belfort

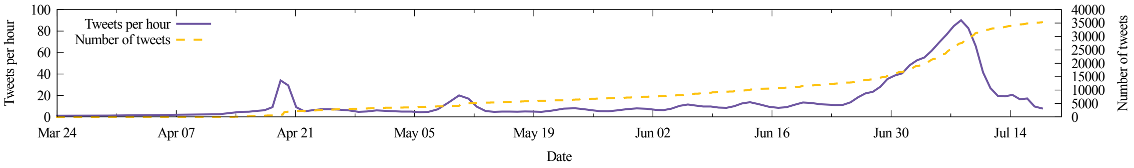

Figure C.14: Roskilde

Appendix D

Pukkelpop 2011

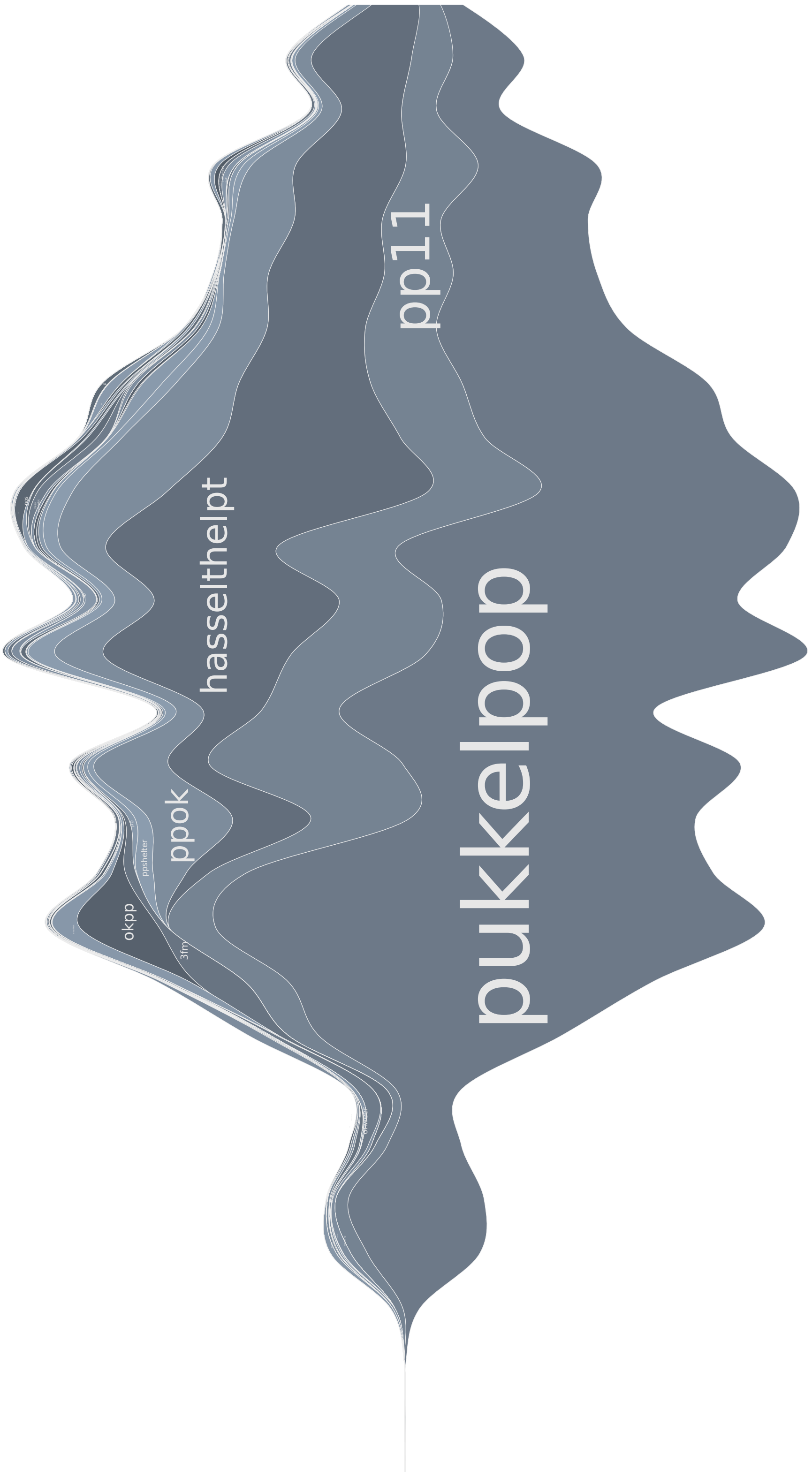

Figure D.1: Pinkpop 2011 streamgraph of hashtags on August 18 from 16 to 24 UTC.

Glossary

#hasselthelpt Hasselt helps 4, 5

#hasselthelptmooi Hasselt helps beautifully 4

#ll Lowlands short hashtag 3, 5

#ll11 Lowlands 2011 short hashtag 3

#lowlands Hashtag of the Lowlands festival. `http://www.lowlands.nl/` 3

#okpp Was used at Pukkelpop 2011 to communicate that the person is fine. 4

#pinkpop Hashtag of the Pinkpop festival. `http://www.pinkpop.nl/` 3

#pkp Short hashtag of the Pukkelpop festival. 4

#pkp11 Short hashtag of the Pukkelpop 2011 festival. 4

#pp Pinkpop short hashtag 4

#pp11 Pinkpop 2011 short hashtag 3, 5

#ppok Was used at Pukkelpop 2011 to communicate that the person is fine. 4, 5

#ppshelter 4, 5

#pukkelpop Hashtag of the Pukkelpop festival. 4

Bibliography

[1] N. Alon, P. Gibbons, Y. Matias, and M. Szegedy. Tracking join and self-join sizes in limited storage. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 10–20. ACM, 1999.

[2] L. AlSumait, D. Barbara, J. Gentle, and C. Domeniconi. Topic significance ranking of lda generative models. *Machine Learning and Knowledge Discovery in Databases*, pages 67–82, 2009.

[3] K. Balog, W. Weerkamp, and M. de Rijke. A few examples go a long way: constructing query models from elaborate query formulations. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 371–378. ACM, 2008.

[4] J. Benhardus. Streaming trend detection in twitter. *National Science Foundation REU for Artificial Intelligence, NLP and IR*, 2010.

[5] E. Benson, A. Haghighi, and R. Barzilay. Event discovery in social media feeds. In *The 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, USA. To appear*, 2011.

[6] M. Bernstein, B. Suh, L. Hong, J. Chen, S. Kairam, and E. Chi. Eddi: interactive topic-based browsing of social status streams. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 303–312. ACM, 2010.

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

[8] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10. IEEE, 2010.

[9] L. Byron and M. Wattenberg. Stacked graphs–geometry & aesthetics. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1245–1252, 2008.

[10] G. Cormode and M. Hadjieleftheriou. Finding the frequent items in streams of data. *Communications of the ACM*, 52(10):97–105, 2009.

[11] G. Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. *LATIN 2004: Theoretical Informatics*, pages 29–38, 2004.

[12] F. Deng and D. Rafiei. New estimation algorithms for streaming data: Count-min can do more.

[13] S. Goorha and L. Ungar. Discovery of significant emerging trends. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–64. ACM, 2010.

[14] S. Gouws, D. Metzler, C. Cai, and E. Hovy. Contextual Bearing on Linguistic Variation in Social Media. In *Proceedings of the ACL-11 Workshop on Language in Social Media*. Association for Computational Linguistics, 2011.

[15] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228, 2004.

[16] B. Han and T. Baldwin. Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics*, 2011.

[17] P. Hennig, D. Stern, R. Herbrich, and T. Graepel. Kernel topic models. *Arxiv preprint arXiv:1110.4713*, 2011.

[18] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 151–160, 2011.

[19] D. Laniado and P. Mika. Making sense of twitter. *The Semantic Web–ISWC 2010*, pages 470–485, 2010.

[20] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011), Portland, Oregon*, 2011.

[21] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani. Counter braids: a novel counter architecture for per-flow measurement. In *ACM SIGMETRICS Performance Evaluation Review*, volume 36, pages 121–132. ACM, 2008.

[22] G. Manku and R. Motwani. Approximate frequency counts over data streams. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 346–357. VLDB Endowment, 2002.

[23] A. Marcus, M. Bernstein, O. Badar, D. Karger, S. Madden, and R. Miller. Twitinfo: Aggregating and visualizing microblogs for event exploration. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 227–236. ACM, 2011.

[24] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. *Advances in Information Retrieval*, pages 362–367, 2011.

[25] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 563–572. ACM, 2012.

[26] A. Metwally, D. Agrawal, and A. El Abbadi. Efficient computation of frequent and top-k elements in data streams. *Database Theory-ICDT 2005*, pages 398–412, 2005.

[27] D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *Uncertainty in Artificial Intelligence*, pages 411–418, 2008.

[28] M. Naaman, H. Becker, and L. Gravano. Hip and trendy: Characterizing emerging trends on twitter. *Journal of the American Society for Information Science and Technology*, 62(5):902–918, 2011.

[29] D. Newman, C. Chemudugunta, and P. Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686. ACM, 2006.

[30] S. Petrovic, M. Osborne, and V. Lavrenko. Using paraphrases for improving first story detection in news and twitter.

[31] S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189. Association for Computational Linguistics, 2010.

[32] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[33] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[34] A. Saha and V. Sindhwani. Learning evolving and emerging topics in social media: A dynamic nmf approach with temporal regularization. In *Proceedings of the 5th International Conference on Web Search and Data Mining (WSDM)*, 2012.

[35] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[36] J. Sankaranarayanan, H. Samet, B. Teitler, M. 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. ACM, 2009.

[37] A. Tumasjan, T. Sprenger, P. Sandner, and I. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 178–185, 2010.

[38] X. Zhao and J. Jiang. An empirical comparison of topics in twitter and traditional media. *Singapore Management University School of Information Systems Technical paper series. Retrieved November*, 10:2011, 2011.