Raimond: Quantitative Data Extraction from Twitter to Describe Events

Thibault Sellam[1] and Omar Alonso[2]

[1] CWI, Amsterdam, the Netherlands
thibault.sellam@cwi.nl
[2] Microsoft Corp, Mountain View, CA
omalonso@microsoft.com

Abstract. Social media play a decisive role in communicating and spreading information during global events. In particular, real-time microblogging platforms such as Twitter have become prevalent. Researchers have used microblogging for a number of tasks, including past events analysis, predictions, and information retrieval. Nevertheless, little attention has been given to quantitative data extraction. In this paper, we address two questions: can we develop a mechanism to extract quantitative data from a collection of tweets, and can we use the salient findings to describe an event? To answer the first question, we introduce Raimond, a virtual text curator, specialized in quantitative data extraction from Twitter. To address the second question, we use our system on three events and evaluate its output using a crowdsourcing strategy. We demonstrate the effectiveness of our approach with a number of real world examples.

Keywords: Microblogs, Information extraction, Events analysis

1 Introduction

Microblogging platforms constitute an incredible source of data about events, especially during time-critical matters like disasters. Consider for instance the series of earthquakes which shook Japan in March 2011. In the days which followed the first shocks, millions of posts were written and shared on Twitter. These tweets came from a wide range of sources, including individuals, official organizations, and news agencies from various places around the world. Many of them were produced in real-time. The combination of volume, diversity, brevity and instantaneous reaction makes Twitter a powerful medium to understand how the world was responding.

In this paper, we investigate how to extract *quantitative information* from microblogs. For example, in the case of Japan, how many earthquakes actually stroked the country? How many casualties were reported? How much funds were unlocked to help? The event has a number of objective *quantitative properties*, such as cardinalities and measures. These properties are often associated with numbers. Some of these properties change with time, e.g., the count of casualties.

Others remain constant, like the funds offered by a particular organization. Our aim is to develop a systematic mechanism to extract this information.

Once extracted, quantitative data is a powerful resource to describe events. Charles Minard's carte figurative of Napoleon's campaign in Russia is a famous example of how to convey an event with numbers [23]. The second question we investigate in this paper is the following: to what extent can an automatic system build a narrative from quantities? We will introduce methods to clean and organize quantitative information. But as Tufte suggests, "graphical excellence begins with telling the truth about the data" [23]; we cannot completely discard humans assessment from the edition process.

Researchers have studied how to extract information automatically from web pages since the early days of the Web. Ultimately, the objective is to produce structured data, such as tables, from natural text. This task is a challenge simply because computers cannot understand languages as well as humans do. When we target well-defined classes of information (e.g., the date of a cultural event), we can look for characteristic keywords or expressions. But seeking quantitative data, in general, is much harder. We must deal with an immense range of vocabulary, expressions, interpretations and topics.

In this paper, we present Raimond, a virtual text curator. Raimond's goal is to collect, clean, organize and recommend fragments of text which contain quantitative information. Our system is organized as a pipeline, where each stage solves a different sub-problem. First, Raimond identifies relevant tweets which contain quantitative data. Then, it groups those tweets into sub-topics, removes the low quality content, and display the results. Given the complexity of the problem, we designed Raimond as a hybrid system. On one hand, we automated the data intensive parts of the extraction process. On the other hand, we let humans interpret the text through a crowdsourcing platform. To summarize, we make the following contributions:

- We analyze how quantitative data is conveyed on Twitter
- We describe Raimond, a system to extract, filter and organize quantitative information to describe events.
- We study three real-world examples
- We evaluate the effectiveness of our approach with crowdsourcing

This paper is organized as follows. In the next section, we present the notion quantfrag. In Section 3, we detail how Raimond extract quantitative data. Section 4 showcases Raimond with real-word examples. An evaluation is presented in Section 5. A survey of related work is presented in Section 6. Finally, we present our conclusions and outline future work.

Table 1. Illustration of our terminology.

Tweet	Japan update: **five nuclar plants shut down in Japan**, tsunami waves continue to hit
Event	2011 Japan Earthquakes
Quantfrag	five nuclear plants shut down in Japan
Property	Nuclear plants shut down
Quantity	5
Is a Qweet?	Yes

2 Introducing the Quantfrag

Overview. The central concept behind Raimond is the **quantfrag**. A quantfrag is a snippet of text which contains a piece of quantitative information. Observe for instance the following tweets, recorded after the 2011 earthquakes in Japan[3]:

"Breaking News: **A 8.8 earthquake just hit #Japan.**"
"**At least 2,369 are missing after #quake.** I have no words."
"This is insane. **The Earth's rotation sped up by 1.6 microseconds.** #japan #planet"

Each post contains some quantitative information, surrounded by comments or details about the context. We call quantfrags the fragments of text which contain the quantities. We highlight these fragments in bold in the example. Ideally, a quantfrag should contain enough information to understand the quantity, but no more. It should be self-contained, but short. This leads to our first definition:

Definition 1 *A quantfrag is a complete, minimal piece of text which describes a fact based on a quantity.*

Not all tweets contain quantfrags. We use the term **qweets** for those which do: a qweet is a Twitter post which contains a quantity. We illustrate our terminology in Table 1. Raimond's aim is to detect qweets, extract quantfrags, and present the collection in a browsable form.

Natural catastrophes are not the only events which yield quantitative data. The following quantfrags describe the 2014 World Cup Brazil-Germany game:

"BRA undefeated in 62 straight competitive home games since 1975"
"GER have now scored 221 goals in WorldCup history"

These quantfrags were produced during the 2014 Ukraine political crisis:

"EU to provide $15 billion help package to Ukraine"
"Crimea referendum: 97% voted to join Russia"

We will present these two topics in detail in Section 4.

[3] All the examples in this section are based on actual tweets. Nevertheless, we took the liberty to truncate the original posts to shorten the presentation.

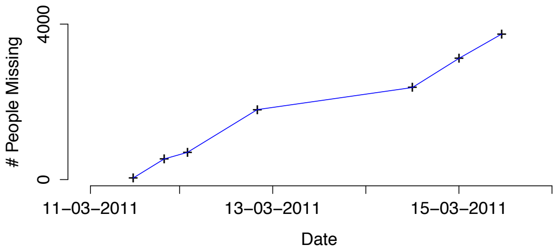


Fig. 1. Time series reconstituted from seven serial quantfrags.

Detection. We now discuss how to detect qweets and quantfrags algorithmically. Most qweets contain numbers, written with letters or digits. However, Twitter data also contains a plethora of counter-examples. A post can describe a quantity without using any number:

"the country's strongest earthquake on record"

Also, it is not difficult to find numbers without quantities:

"Japan I pray 4 U"
"Please text the words Text Red Cross to 90999"
"Barack Obama will give a special address at 1130"

To complicate the matter further, many fragments form valid quantfrags, but they teach us little about the event:

"A fire has broken out at Cosmo Oil's 220,000 b/d Chiba refinery after earthquake."
"I have a friend in japan. And he actually owes me ten bucks."

These examples show that reporting all tweets which contain numbers is a very naive solution. Raimond relies on the combination of several methods, which we will discuss thoroughly in the following section.

Single Quantfrags, Serial Quantfrags. During our experiments, we encountered two types of quantfrags. **Single quantfrags** state independent, self-contained facts. For instance, the following quantfrag is single:

"The Pacific Plate slid west by 79 feet"

Oppositely, **serial quantfrags** describe the same property of the event, but at different points in time. Therefore, they describe a time series. Here is an example of such fragments:

11 March 2011 - "530 people were reported missing after #earthquake in Japan"
12 March 2011 - "about 1800 missing in #japan as a result of #earthquake"
15 March 2011 - "at least 3,743 are missing #earthquake #tsunami"

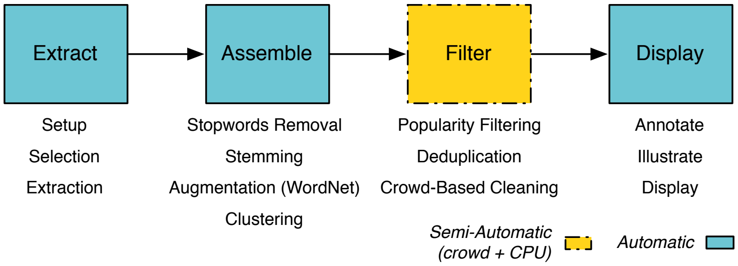


Fig. 2. Overview of the Raimond pipeline

These three quantfrags describe the number of people reported missing after the earthquakes, but at different points in time. They are particularly interesting because they let us reconstitute the original time series, as shown in Figure 1. One of Raimond's functions is to organize the quantfrags in subtopics, such that serial quantfrags are displayed together.

Validity. In general, qweets may contain approximations, omissions, exaggerations or time lags. Unfortunately, this noise is inherent to social data. For instance, thousands of tweets mentioned 88,000 missing people during the Japan earthquakes. We found no trace of the original report, and official sources hint that this number is largely overestimated[4]. Our aim is to depict microbloggers' views on events, regardless of their overlap with objective truth. Fact checking is, for now, beyond the scope of this study.

3 Methodology

Raimond's goal is to detect and organize quantfrags. To do so, it operates in four consecutive stages, pictured in Figure 2. First, Raimond detects the most promising tweets, and extracts the quantfrags. Then, it groups the fragments which cover the same topic. During the third phase, Raimond filters out the fragments which are irrelevant or not informative with a combination of coded rules and crowdsourcing. Finally, it labels and displays the clean groups.

3.1 Extracting Quantitative Data

During this first phase, Raimond detects the tweets associated to the event of interest, parses them and retrieves the quantfrags.

Setup. To seed the Raimond pipeline, we define an event configuration. The configuration specifies which authors to follow and which tweets to select. For our Japan example, we tracked the hashtag `#japan` during 5 days, and selected the posts with more than 25 retweets. Table 2 shows all the settings offered

[4] www.jst.go.jp/pr/pdf/great_east_japan_earthquake.pdf, page 13

Table 2. Seeding the Raimond pipeline.

Type	Input field
Content	Hashtags
	Keywords
	Language
Network	Twitter's `verified` flag
	Account's followers
	Message retweets

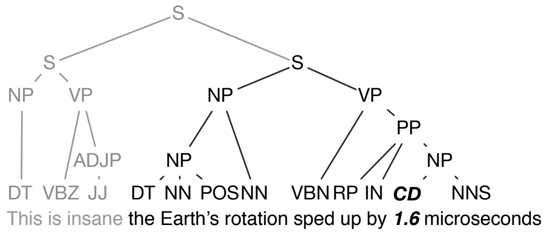

S
S
S
NP
VP
NP
VP
ADJP
NP
PP
NP
DT
VBZ
JJ
DT
NN
POS
NN
VBN
RP
IN
CD
NNS
This is insane the Earth's rotation sped up by 1.6 microseconds

Fig. 3. Extracting quantfrags from the parse tree. The nodes of the tree represent constituent tags, as defined by the Penn Treebank. Our aim is to extract the subtree which contains the quantfrag. The quantfrag is highlighted in bold.

by Raimond. The aim of content-related parameters is to spot relevant tweets. Network-related parameters measure trust and influence.

Selection. Once the event configuration is defined, Raimond fetches the tweets from our archive, and it applies a filter to discard tweets with no quantities. At this point, we include every tweet which could potentially be interesting, regardless of it its quality - we value recall much more than precision. The filter relies on two tests, assembled in a disjunction. The first test uses a quantity classifier. The classifier is based on statistical learning, and it was trained internally for production purposes. For the second test, we wrote a set of regular expressions. These regular expressions detect cardinal and ordinal numbers, expressed with letters or numbers. At the end of this phase, we obtain a set of potential qweets, which typically contains lots of false positives.

Extraction. During this phase, Raimond extracts the quantfrags. Previously, we defined quantfrags as complete, minimal pieces of texts which convey a quantity. Unfortunately, evaluating whether a quantfrag is complete and minimal depends a lot on the user and the use case. Our definition is not practical. We propose to operationalize the notion as follows:

Definition 2 (operational) *A quantfrag is a grammatical clause which contains a quantity.*

To detect clauses with numbers, we use a grammatical parser. The parser takes a tweet as input, and returns a tree, as pictured in Figure 3. In this tree each node represents a grammatical constituent. We check if the tree contains a quantity, tagged `CD` (Cardinal number) in the example. If it does, we extract the smallest clause which contains this quantity (`S` in our example). If we detect several numbers, we extract one clause for each. We used an internal parser trained specifically for tweets, but several open source NLP suites can handle this type of task (e.g., Stanford NLP).

3.2 Assembling Quantfrags

In this phase, Raimond aggregates the quantfrags which describe the same topic, or, in some cases, the same variable (cf. serial quantfrags in Section 2). To achieve this, Raimond uses cluster analysis. As the quantfrags are short and noisy, preprocessing is crucial.

Preprocessing and Augmentation. To clean the quantfrags, we apply classic preprocessing operations: we replace smileys by keywords, we remove punctuation symbols and stop words, and we stem every term. Typically, the quantfrags we obtain are very short. This is problematic for clustering, because they are not likely to share terms. Consider for instance the following two quantfrags:

```
"Troops of 500+ to provide help"
"More than 500 militaries sent for assistance"
```

Both phrases have exactly the same meaning, yet they do not have any word in common. We use a lexical database, WordNet [13], to tackle this problem. For a given term, WordNet gives us hypernyms. Intuitively, a hypernym is a semantic superclass of a term. For instance, `army unit` is a hypernym of `troop`. Thanks to hypernyms, we can *augment* our quantfrags. We query the WordNet database for each noun and append the results to the fragment. This increases the chance that similar tweets share words. For instance, if we augment the first noun in each of our example tweets, we obtain:

"Troops *army unit* ***military force*** of 500+ to provide help"

"More than 500 militaries ***military force*** *organization* sent for assistance"

WordNet entries are organized in a hierarchy: hypernyms themselves have hypernyms. Therefore, we can expand our terms with several levels of generality. We used two levels of recursion in the example, we use three in our system.

In many cases, nouns have several competing WordNet entries. Each entry is represented by a set of synonyms, such as `assistance - aid - help`, or `assistance - financial aid - economic aid`. To resolve the ambiguity, we check how many of the synonyms are contained in the corpus, and keep the entry with the highest count. If the procedure finds no match, we take the most frequent sense. We refer the reader to the work of Hotho et al. for an empirical validation of this method [8].

Clustering. We represent the quantfrags with bags of words, and cluster them with agglomerative clustering [21]. We chose this approach because

Table 3. Parameters for the cluster analysis.

Parameter	Range	Default
Distance	Cosine, Euclidean, p-Minkowski	Cosine
Linkage	Single, Complete, Average	Average
Maximum distance	0 - 1.0	0.9

it is simple enough to be tuned by non-technical users. Recall that agglomerative clustering operates bottom-up. To initialize the algorithm, we assign each quantfrag to its own cluster. Then, at each iteration, we detect which two clusters are the closest, and merge them. As the algorithm runs, the clusters get larger. We stop when we reached a threshold. The algorithm requires three parameters, summarized in Table 3. We must chose a distance function for quantfrags. For instance, the cosine distance is a well-established choice. We must also define how to compute the distance between clusters. Consider two clusters C_1 and C_2, and let d describe the distance measure we use for quantfrags. There are different ways to define how close these clusters are. We can use the the distance between their two closest points (single-link). In this case, we set $D(C_1, C_2) = \min\{d(x, y) : a \in C_1, y \in C_2\}$. We can use the distance between their two closes furthest points (complete-link). Then, $D(C_1, C_2) = \max\{d(x, y) : a \in C_1, y \in C_2\}$. This usually results in tighter clusters.

3.3 Filtering Irrelevant Quantfrags

Table 4. Sequence of filters used to remove false positives.

Precision Level	Filter	Computation
Cluster	Size	Machine
Quantfrag	Near-duplicates	Machine
Cluster	Relevance	Machine + Crowd
Quantfrag	Relevance	Machine + Crowd

During the two first phases, Raimond typically accumulates lots of false positives. Some quantfrags do not contain any quantity (`"Japan, I pray 4 u"`), are not related to the topic (`"Japan, thank you for Playstation 4!"`), are not informative (`"3 reasons why we must help Japan"`) or simply redundant. To make things worse, the clusters we detect are rarely perfect, as they may combine unrelated but lexically similar topics. To address this problem, we developed a cascade of filters, based on automatic rules and crowdsourcing. We summarize the filters in Table 4, and detail them below.

Filtering on Popularity. Typically, the size of the clusters obey approximately a power-law distribution. We observe a few large clusters, and a long tail of micro-topics. Raimond gives the the option to select the large clusters (the head of the distribution) and discard the smaller groups. The rationale is that large clusters describe popular topics, while smaller clusters may contain noise, such as personal reaction or irrelevant facts.

Near-Duplicates Removal. So far, we have kept (near) duplicates to assess the popularity of the topics. We now eliminate the redundancy. In fact, this task is close to the clustering phase, described in 3.2. We detect near-duplicates with the exact same method, but we operate at a thinner granularity. We reuse the dendrogram structure produced at the end of the clustering phase, and we cut it at a low level of dissimilarity (by default, 0.1). We obtain lots of micro-clusters, we represent each of them by a representative quantfrag (by default, the most frequent one).

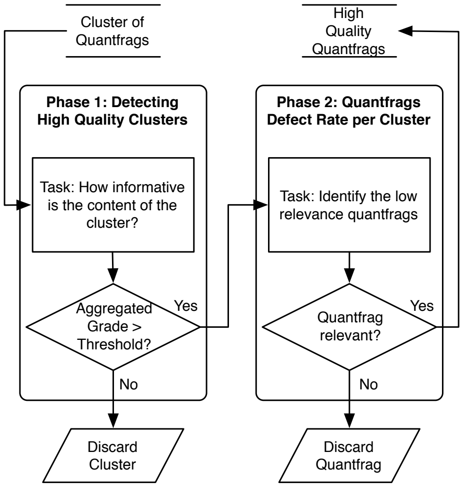
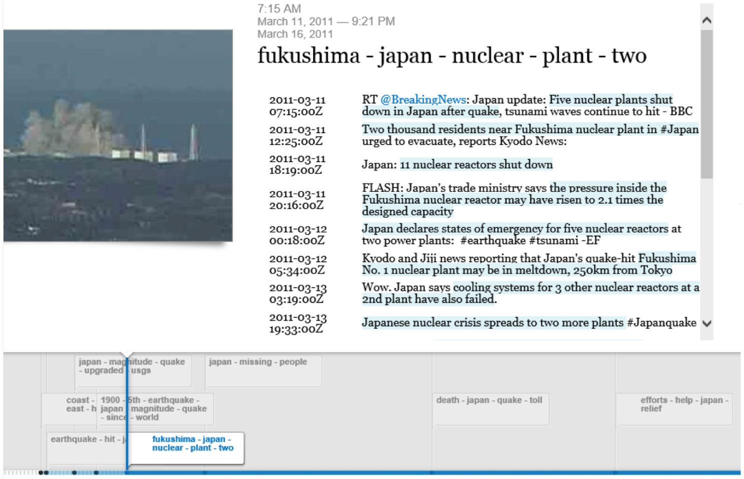

Cluster of Quantfrags
High Quality Quantfrags
Phase 1: Detecting High Quality Clusters
Task: How informative is the content of the cluster?
Aggregated Grade > Threshold?
Yes
No
Discard Cluster
Phase 2: Quantfrags Defect Rate per Cluster
Task: Identify the low relevance quantfrags
Quantfrag relevant?
Yes
No
Discard Quantfrag

Fig. 4. Flow-Chart illustrating our crowdsourcing strategy to select high quality quantfrags.

Crowd-Based Cleaning. At this stage, the collection of quantfrags still contains false positives, with numbers but no quantities. It also contains uninformative quantfrags, i.e., quantfrags which are technically valid but provide no useful information about the event. We discard those with human computation.

Our crowdsourcing strategy is based on two consecutive tasks. During the first task, workers evaluate the overall quality of the clusters. They assign a grade to each cluster, based on a relevance. We aggregate the scores, and check if the value is above a certain threshold. If not, we discard the cluster. We then run another task, in which the goal is to identify low quality quantfrags within the clusters. Figure 4 describes the overall process. We can think of this approach as a two-step quality control: the first phase checks if the cluster is relevant to the event. The second phase provides a defect rate per cluster. The final output



7:15 AM
March 11, 2011 — 9:21 PM
March 16, 2011

fukushima - japan - nuclear - plant - two

2011-03-11 07:15:00Z RT @BreakingNews: Japan update: Five nuclear plants shut down in Japan after quake, tsunami waves continue to hit - BBC

2011-03-11 12:25:00Z Two thousand residents near Fukushima nuclear plant in #Japan urged to evacuate, reports Kyodo News:

2011-03-11 18:19:00Z Japan: 11 nuclear reactors shut down

2011-03-11 20:16:00Z FLASH: Japan's trade ministry says the pressure inside the Fukushima nuclear reactor may have risen to 2.1 times the designed capacity

2011-03-12 00:18:00Z Japan declares states of emergency for five nuclear reactors at two power plants: #earthquake #tsunami -EF

2011-03-12 05:34:00Z Kyodo and Jiji news reporting that Japan's quake-hit Fukushima No. 1 nuclear plant may be in meltdown, 250km from Tokyo

2011-03-13 03:19:00Z Wow. Japan says cooling systems for 3 other nuclear reactors at a 2nd plant have also failed.

2011-03-13 19:33:00Z Japanese nuclear crisis spreads to two more plants #Japanquake

japan - magnitude - quake - upgraded - usgs
japan - missing - people
coast - east - h
1900 - japan - since
5th - earthquake - magnitude - quake - world
death - japan - quake - toll
efforts - help - japan - relief
earthquake - hit - j
fukushima - japan - nuclear - plant - two

Fig. 5. Quantfrags presentation.

is a set of high quality clusters, with useful quantfrags. In our surveys, we avoid spammers with purposely trivial questions and redundancies.

3.4 Annotation and Visualization

The aim of the last step is to annotate and display the clusters of quantfrags. The operations described in this section do not add content, but they enhance the presentation of the quantfrags.

Title. Raimond summarizes each cluster with a title. To produce the title, it creates documents by concatenating the quantfrags of each cluster. Then, it computes a tf-idf matrix, and reports the top k terms for each cluster/document (we set $k = 5$ for the rest of this paper).

Illustration. We observed that many qweets contain links to images. Our idea is to exploit these links to illustrate the clusters. Raimond parses the tweets for image URLs with a set of regular expressions. If it encounters such URLs, it tries to download the documents. It then presents the images side-by-side with the quantfrags in the interface. If a cluster links to several images, Raimond presents them sorted by decreasing order of popularity (using the number of retweets).

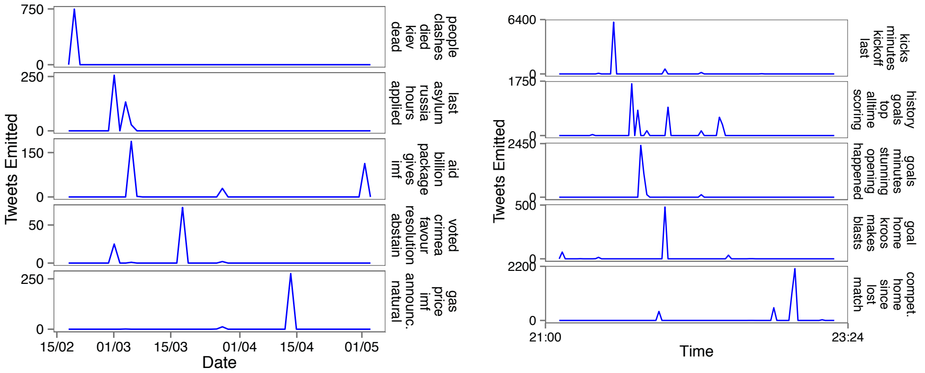
Display. Raimond's last task is to display the quantfrags. We provide a screenshot of the interface in Figure 5. The bottom part of the display presents the titles of the clusters on a timeline. To anchor the labels, we calculate *peak dates*. The peak date of a cluster is the timestamp at which it is the most popular. To calculate it, we retrieve the dates at which the quantfrags of the cluster are mentioned, estimate a density function with Gaussian density estimation and compute the mode of this distribution. We will present some examples in Section 4. Users can focus on a cluster by clicking on its title. Then, Raimond

Table 5. Data collection methodology and event configuration.

	Ukraine	BRAvGER	Japan

Hashtags	`#ukraine`	`#bra #ger #bravger`
		`#brazil #germany`
			`#japan`

Start date	1 Jan 2014	08 Jul 2014	10 Mar 2011
End date	15 May 2014	08 Jul 2014	15 Mar 2011
Author checks	Min. 200,000 followers Verified	Min. 200,000 followers Verified	Min. 25 retweets
#tweets	7,362,838	16,481,551	3,049,463



(a) Ukraine Data Set.

(b) Brazil-Germany dataset.

Fig. 6. Popularity of the Clusters with Time.

displays the quantfrags with their timestamps and the qweets from which they were extracted.

4 Use Cases

In this section, we present our experiments with three datasets. The first dataset is based the 2011 Japan earthquakes, discussed throughout the paper. The second dataset describes the political crisis in Ukraine, still ongoing at the time of writing. To obtain it, we tracked the hashtag `#ukraine` during 134 days. The third dataset contains tweets about the Brazil-Germany football game of the 2014 World Cup. Using five hashtags, we gathered approximately 16 millions of Tweets in less than 24 hours. We detail our data collection methodology and event configurations in Table 5.

In terms of implementation, Raimond runs partly on a cluster, and partly on a local machine. The cluster gives a huge throughput, but a low latency. The local machine operates the other around. Therefore, we implemented the operations which require no user intervention on the cluster (in particular the extraction). We run the Clustering step and parts of the Filtering step on the

Table 6. Hints about resource consumptions.

Phase	Computation	Runtime	Resources
Extraction	Machine	10-120 min	<500 nodes
Selection			
Extractiom			
Preprocessing	Machine	30-90 min	1 node
Augmentation			
Clustering		1-5 min	
Popularity	Machine	<2 min	1 node
Deduplication			
Cleaning	Human	1-5 hours	>100 workers
Annotation	Machine	<1 min	1 node
Illustration		5-10 min	
Display		<1 min	

local machine, because these tasks require several rounds of trial and error. We provide hints about the execution times and resource consumptions in Table 6 (as Raimond runs on a shared production cluster, its exact runtime depends on the on ressources available).

Table 7. Filtering and extraction of quantfrags. The sets are sorted by inclusion - each set is refinement of the previous one. The Japan set was filtered and deduplicated before our experiments.

Dataset	Ukraine	BRAvGER	Japan
Tweets	7,326,838	16,481,551	3,049,463
. Trusted	441,151	992,980	NA
.. Unique	10,508	6,438	6,210
... Contain quantities	1,093	1,207	1,729
.... Quantfrags	718	762	1,354

Table 7 shows the size of the data as Raimond processes the tweets. We start with several million tweets. We tuned the pipeline to extract only those that come from official sources and news accounts (cf. Table 5). We obtain less than a million tweets (about 5% of the initial volume). This number includes the tweets written by official sources, but relayed by non-trusted individuals. After removing the retweets and the duplicates, we obtain less than 10,000 posts. This decrease is spectacular, but not surprising: by definition, popular accounts are massively retweeted. For instance, in the BRAvGER dataset, posts about spectacular actions and goals are retweeted by thousands of supporters. At the end of the pipeline, after filtering, cleaning and aggressive deduplication, we obtain a few hundred quantfrags.

Table 8 displays the labels of a few clusters generated by Raimond for the Japan dataset. As in our interface, we ordered the clusters by peak date. We

Table 8. Clusters from the Japan Dataset

Keywords	Peak	Size
quake, magnitude, upgraded, usgs, felt	11/03	4,029
nuclear, fukushima, plant, two, explosion	11/03	2,464
axis, moved, shifted, feet, earths	12/03	5,771
people, missing, tsunami, dead, quake	12/03	10,761
toll, death, quake, missing, tsunami	13/03	5,007
effort, help, donate, relief, redcross	13/03	7,414
plant, radiation, nuclear, fukushima, says	15/03	3,062

Table 9. Examples of clusters for the Ukraine dataset.

Keywords	Peak Date	Size	Qweet
people, clashes, died, kiev, dead	18/02	1,128	`"#Ukraine police say four officers have died in today's riots, 39 have sustained gunshot wounds and more than 100 others have been injured"`
last, asylum, russia, hours, applied	01/03	451	`"#UKRAINE: 143,000 Ukrainians have asked for asylum in #Russia for last two weeks"`
aid, billion, package, gives, imf	05/03	330	`"BREAKING: Top official says EU to provide #Ukraine $15 billion aid package in loans and grants"`
voted, crimea, favour, resolution, abstained	14/03	765	`"#Crimea parliament declares independence from #Ukraine after referendum. Final tally shows 97% voted to join #Russia"`
gas, price, imf, announces, natural	13/04	406	`"As the IMF announces aid package of $14-18bn for #Ukraine, the Ukrainian PM warns the price paid to Russia for gas will rise 79% from 1 Apr"`
imposes, officials, sanctions, entry, russia	28/04	587	`"BREAKING NEWS: #EU imposes sanctions on 21 officials from #Russia and #Ukraine over Crimea. More soon..."`
donetsk, ballots, region, results, selfdefense	11/05	578	`"Preliminary results show 89.7% support of self-rule in #Donetsk region, #referendum election commission says"`

observe that the topics are semantically intelligible. The first cluster describes physical properties of the earthquake. The second one mentions the nuclear plant explosion which followed. Twitter users discuss the impact of the disaster on people and on the environment. Then, then they give more details about casualties, and encourage donations.

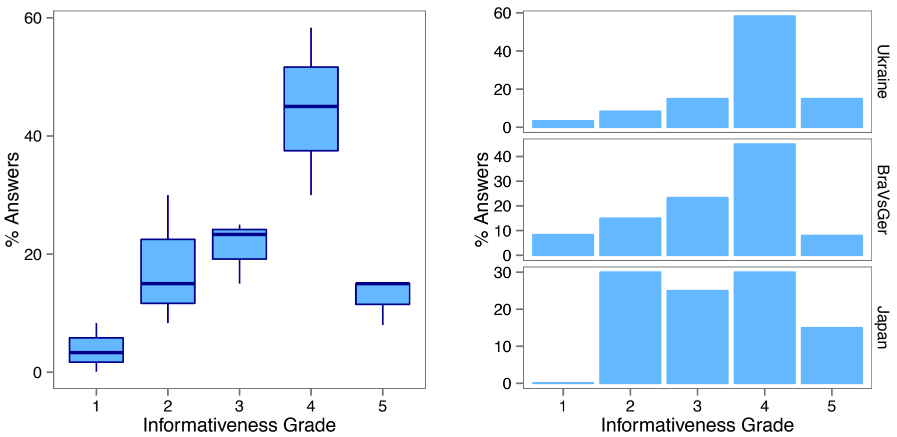
We show a few clusters created from the Ukraine dataset in Table 9. The quantfrags spread across a variety of small topics, such as casualties ("people, clashes, died"), international help ("aid, billion, package"), gas markets ("gas, price, imf") or sanctions ("imposes, officials, sanctions"). We describe the dynamics of the five first clusters in Figure 6a. To obtain these charts, we tracked

Table 10. Examples of clusters for the BRAvGER dataset.

Keywords	Peak Time	Size	Qweet
reach, semifinals, first, country, consecutive	17:40:22	2,671	"GER is the first team ever to reach four straight #WorldCup semifinals."
kicks, minutes, every, kickoff, less	18:01:22	7,095	"Still more than two hours to go until kick-off... #Copacabana #Brazil "
history, goals, top, alltime, scoring	21:37:26	4,683	"#GER have now scored 221 goals in #WorldCup history, more than any other side and one ahead of #BRA."
goals, minutes, stun, opening, happened	21:47:02	3,535	" That. Just. Happened. Germany stun Brazil with 5 goals in the opening 29 minutes."
goal, home, kroos, makes, blasts	21:51:09	921	"#GER 5 goals in the first 29 minutes!" ‘‘6-0... Germany got once and GOAL... #Amazing’’ ‘‘GOAL!!!! ’79 Schurrle blasts home a pitch-perfect pass from Mueller to make it 7-0.’’
klose, record, now, miroslav, goals	22:57:58	5,331	"#GER’s Mirsolav Klose has a chance to break his record of 15 #WorldCup goals against Brazil."
competitive, home, since, lost, match	23:00:02	4,275	"Entering this match, Brazil had not lost a competitive game on home soil in 14,161 days. Until today.... #BRAvsGER"

the number of quantfrags produced for each cluster. We observe bursts, which last several hours, sometimes days. These bursts actually reflect real events. The first cluster describes the clashes which took place on February 18th and 20th. According to the quantfrags, this was the worse day of violence that Ukraine had known in 70 years. During the followed two weeks, several hundred thousands Ukrainians asked for asylum to Russia and a $15 billion Dollars help package was approved by the European Union. The fourth cluster describes the outcome of the Crimean status referendum, which happened on March 16th. Finally, the last cluster discusses a raise in consumer gas tariffs, requested by the IMF in exchange for a rescue loan.

Table 10 presents our Brazil-Germany dataset. As opposed to our previous example, the clusters are semantically close to each other - they are all somehow related to scoring goals. We highlight serial quantfrags in the fifth cluster ("goal, home, kroos"): the count of German goals is regulary incremented, finally reaching seven goals. We detail the dynamics of the clusters in Figure 6b. We see that they appear in short, intense bursts of several minutes. The game starts at 21.00, the first cluster discusses the kick-off. Within the first 30 minutes, the German team scores five goals. This triggers two consecutive clusters, explaining with quantities why the event is "historical" and "stunning". For instance, Germany is the first country to score 221 goals in a World Cup. With two goals



(a) Distribution of the grades for every datasets combined.

(b) Grades for each dataset.

Fig. 7. Crowdsourcing experiment results.

in two minutes, the main attacker, Tony Kroos, has a cluster on his own. The last cluster shows that Brazil had not been defeated at home since 1975.

5 Crowdsourcing Experiments

In this Section, we evaluate the effectiveness of Raimond's output. We process the three datasets introduced in Section 4, and present the clusters to a set of crowdworkers. We ask them if the quantfrags contain quantitive information, and how *informative* this information is, with a grade between 1 (not informative) and 5 (very informative). As we only have a limited pool of workers, we decided to remove the crowd-based filtering step from the pipeline - to avoid having workers check their own work. Thus, our evaluation is conservative. We evaluated 70 clusters (20 for Ukraine and Brazil-Germany, 30 for Japan), containing between 2 and 75 quantfrags. Each cluster is reviewed by at least two workers.

Figure 7a represents the overall distribution of the grades. The neat dominance of the the value 4 indicates that most clusters are informative. Nevertheless, Raimond also returns some noise: about a fifth of the clusters have a grade lower than 2.

Figure 7b shows the grades for each dataset. The Ukraine and Brazil-Germany clusters have good scores. In the Ukraine case, more than 90% of the clusters have at least a grade of 3. Most of the noise comes from the Japan dataset. There are many informative clusters, but there are about as many irrelevant clusters. Further inspection revealed lots of calls for donations, such as:

```
"Txt ASIA to 30333 to donate $5."
"100% donations go to Canadian Red Cross"
"text REDCROSS to 90999 to donate $10 from your phone"
```

Also, some personalities are so popular that any quantfrag involving them will be retweeted thousands of times:

```
"Justin Bieber donated $1,000,000 to Japan."
"Lady Gaga donated 16 million to Japan"
"Disney made a $2.5 million donation to the Red Cross"
```

Such fragments are difficult to filter programmatically, because they form valid quantfrags and they are extremely popular. To conclude, Raimond does generate useful clusters. Nevertheless, with popular events such Japan earthquake, the diversity of the data justifies our choice for human computation.

6 Related Work

Studying events on social media has gained considerable interest in the last five years. In particular, catastrophes and emergency situations have attracted lots of attention [9]. The resulting works can be classified in four categories: event detection, event summarization, information extraction and visualization (note that these areas overlap). We describe these works below. There is to our knowledge no previous work on quantitative data extraction from Twitter.

Sayyadi et al. have published one the first study on event detection with social media, based on lexical community detection [20]. Sakaki et al. use microblogging to detect earthquakes and track their location [19]. Popescu and Pennachiotti focus on controversial events, which they recognize with supervised learning [16]. Petrović et al. focus on computational efficiency. They present a scalable algorithm based on Locality-Sensitive Hashing [14].

Authors have investigated how to extract key sentences to summarize a text for decades [11]. In 2001, Allan and Khandelwal proposed a method to summarize news coverage. They decompose a main event in sub-events with language models, and describe each sub-topic with a piece of news [2]. Several studies have extended this method to social data with more advanced statistical models. For instance, Chakrabarti et al. use a custom version of Hidden Markov Models to segment the events [6].

Extracting structured information about events from social media involves complex NLP methods. One of first the research effort on the topic was presented by Popsecu et al., who use entity extraction to recognize actors [17]. Benson et al. go one step further, as they infer structured records about entertainment events from Twitter [5]. Imran et al. combine several classifiers and a sequence labelling algorithm to extract structured information about disasters [10]. These approaches are generalized by Ritter et all, who introduce a method to analyze events in open domains. They present a pipeline, somehow similar to Raimond, which extracts names entities, event phrases, calendar dates and event type. Their pipeline combines cutom NLP tools and unsupervised learning [18].

Finally, several authors have studied how to create visual dashboards from Twitter to describe events. Diakopoulos et al. combine raw data, automatically-generated statistics (such as sentiment or relevance) and timelines to help journalists [7]. Marcus et al. propose a similar system, with geographical information and peak detection [12]. Alonso and Shiells introduce a display based on multiple timelines, and illustrate their method with sports events [4].

A number of studies resemble ours by their methods, but target other problems. Alonso et al. study to what extent crowdsourcing can be used to assess the interestingness of tweets [3]. For instance, NIFTY by Suen et al. is also an information extraction pipeline based on Twitter and unsupervised learning. However, it focuses meme-tracking [22]. More generally, news processing is an active related domain of research [1, 15].

7 Conclusions and Future Work

Short posts on social networks provide lots of opportunities to communicate quantitative information. We described Raimond, a pipeline to extract this content from Twitter. We introduced quantfrags, and illustrated the concept with a number of examples. We presented how to extract quantfrags with the help of NLP techniques, how to organize them with clustering, and how to clean them with a hybrid automatic/crowdsourcing approach. Finally, we showcased quantfrags about a three real events. We described their semantics, their dynamics and evaluated their content.

We believe that many exciting developments can come from our work. We will generalize our pipeline to more general topics (not just events) and other data sources. We will also adapt it to real-time, incremental settings. Finally, we will investigate how to exploit our crowdsourced labels for machine learning.

More generally, the road for further automation lays wide open. Reconstituting time series from text without human intervention is still an open problem. This task implies many challenges: how can we normalize the quantfrags? How can we check the facts? How do we resolve inconsistencies? The technologies to be developed go far beyond the strict realm of social networks.

Acknowledgements. We thank Aitao Chen, from Microsoft Research for his NLP suite, his time and his insights. We thank Martin Kersten and Stefan Manegold for their support.

References

1. Ahmed, A., Ho, Q., Eisenstein, J., Xing, E., Smola, A.J., Teo, C.H.: Unified analysis of streaming news. In: Proc. WWW. pp. 267–276 (2011)
2. Allan, J., Gupta, R., Khandelwal, V.: Temporal summaries of new topics. In: Proc. SIGIR. pp. 10–18. ACM (2001)
3. Alonso, O., Marshall, C.C., Najork, M.: Are some tweets more interesting than others? #hardquestion. In: HCIR. p. 2. ACM (2013)

4. Alonso, O., Shiells, K.: Timelines as summaries of popular scheduled events. In: Proc. WWW. pp. 1037–1044 (2013)
5. Benson, E., Haghighi, A., Barzilay, R.: Event discovery in social media feeds. In: Proc. ACL. pp. 389–398. Association for Computational Linguistics (2011)
6. Chakrabarti, D., Punera, K.: Event summarization using tweets. In: Proc. ICWSM. pp. 66–73. AAAI Press (2011)
7. Diakopoulos, N.: Diamonds in the rough: Social media visual analytics for journalistic inquiry. In: Proc. VAST. pp. 115–122. IEEE (2010)
8. Hotho, A., Staab, S., Stumme, G.: Ontologies improve text document clustering. In: Proc. ICDM. pp. 541–544. IEEE (2003)
9. Imran, M., Castillo, C., Diaz, F., Vieweg, S.: Processing social media messages in mass emergency: A survey. In: CoRR. arXiv preprint: 1407.7071 (2014)
10. Imran, M., Elbassuoni, S., Castillo, C.: Practical extraction of disaster-relevant information from social media. In: Proc. WWW. pp. 1021–1024 (2013)
11. Luhn, H.: The automatic creation of literature abstracts. In: IBM Journal of research and development. pp. 159–165 (1958)
12. Marcus, A., Bernstein, M., Badar, O.: Twitinfo: aggregating and visualizing microblogs for event exploration. In: Proc. CHI. pp. 227–236. ACM (2011)
13. Miller, G.A.: Wordnet: a lexical database for english. In: CACM. vol. 38, pp. 39–41. ACM (1995)
14. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: NAACL. pp. 181–189. Association for Computational Linguistics (2010)
15. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proc. WWW. pp. 91–100 (2008)
16. Popescu, A.M., Pennacchiotti, M.: Detecting controversial events from twitter. In: Proc. CIKM. p. 1873. ACM (2010)
17. Popescu, A.M., Pennacchiotti, M., Paranjpe, D.: Extracting events and event descriptions from Twitter. In: Proc. WWW. p. 105 (2011)
18. Ritter, A., Etzioni, O., Clark, S.: Open domain event extraction from twitter. In: KDD. p. 1104. ACM (2012)
19. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proc. WWW. pp. 851–860 (2010)
20. Sayyadi, H., Hurst, M., Maykov, A.: Event Detection and Tracking in Social Streams. In: Proc. ICWSM. pp. 311–314. AAAI Press (2009)
21. Sokal, R.R.: A statistical method for evaluating systematic relationships. In: U. Kansas Scientific Bulletin. vol. 38, pp. 1409–1438 (1958)
22. Suen, C., Huang, S., Eksombatchai, C., Sosic, R., Leskovec, J.: Nifty: a system for large scale information flow tracking and clustering. In: Proc. WWW. pp. 1237–1248 (2013)
23. Tufte, E.: The visual display of quantitative information. Graphics press Cheshire, CT (1983)