DELIVERABLE SUBMISSION SHEET

To: Susan Fraser *(Project Officer)*

EUROPEAN COMMISSION
Directorate-General Information Society and Media
EUFO 1165A
L-2920 Luxembourg

From:

Project acronym: PHEME Project number: 611233

Project manager: Kalina Bontcheva

Project coordinator The University of Sheffield (USFD)

The following deliverable:

Deliverable title: Evaluation Report – Interim Results (revision)

Deliverable number: D6.2.1

Deliverable date: 30 September 2015

Partners responsible: The University of Sheffield (USFD)

Status: ☒ Public ☐ Restricted ☐ Confidential

is now complete. ☒ It is available for your inspection.
☒ Relevant descriptive documents are attached.

The deliverable is:

☒ a document
☐ a Website (URL: .....)
☐ software (.....)
☐ an event
☐ other (.....)

Sent to Project Officer:	Sent to functional mail box:	On date:
Susan.Fraser@ec.europa.eu	*CNECT-ICT*-611233*@ec.europa.eu*	12 May 2016

D6.2.1 Evaluation report - Interim Results

Leon Derczynski; Michał Lukasik; P.K. Srijith; Kalina Bontcheva; Mark Hepple, University of Sheffield
Thierry Declerck; Piroska Lendvai, Univerity of Saarland
Tomás Pariente Lobo; Mateusz Radzimski, ATOS
Petya Osenova, Ontotext

Abstract.
FP7-ICT Collaborative Project ICT-2013-611233 PHEME
Deliverable D6.2.1 (WP6)

The deliverable describes the results of Task 6.4 in WP6 on preliminary evaluations of the PHEME algorithms and their integration. Following the description of work, the datasets created in Task 2.1, WP7, and WP8, are used for iterative development and parameter tuning of the PHEME content analytics methods from WP3 and WP4, as well as for testing their integration into a processing pipeline. The scalability of the integrated tools (Task 6.3) will be evaluated on the large-scale datasets collected in PHEME, as well as on historical data.
Keyword list: Evaluation, language processing, rumour detection, systems integration, performance evaluation

Project	PHEME No. 611233
Delivery Date	30 October 2015
Contractual Date	30 September 2015
Nature	Report
Reviewed By	Kalina Bontcheva
Web links	http://www.pheme.eu
Dissemination	PU

PHEME **Consortium**

This document is part of the PHEME research project (No. 611233), partially funded by the FP7-ICT Programme.

University of Sheffield
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Contact person: Kalina Bontcheva
E-mail: K.Bontcheva@dcs.shef.ac.uk

MODUL University Vienna GMBH
Am Kahlenberg 1
1190 Wien
Austria
Contact person: Arno Scharl
E-mail: scharl@modul.ac.at

ATOS Spain SA
Calle de Albarracin 25
28037 Madrid
Spain
Contact person: Tomás Pariente Lobo
E-mail: tomas.parientelobo@atos.net

iHub Ltd.
NGONG, Road Bishop Magua Building
4th floor
00200 Nairobi
Kenya
Contact person: Rob Baker
E-mail: robbaker@ushahidi.com

The University of Warwick
Kirby Corner Road
University House
CV4 8UW Coventry
United Kingdom
Contact person: Rob Procter
E-mail: Rob.Procter@warwick.ac.uk

Universitaet des Saarlandes
Campus
D-66041 Saarbrücken
Germany
Contact person: Thierry Declerck
E-mail: declerck@dfki.de

Ontotext AD
Polygraphia Office Center fl.4,
47A Tsarigradsko Shosse,
Sofia 1504, Bulgaria
Contact person: Georgi Georgiev
E-mail: georgiev@ontotext.com

King's College London
Strand
WC2R 2LS London
United Kingdom
Contact person: Robert Stewart
E-mail: robert.stewart@kcl.ac.uk

SwissInfo.ch
Giacomettistrasse 3
3000 Bern
Switzerland
Contact person: Peter Schibli
E-mail: Peter.Schibli@swissinfo.ch

Executive Summary

Social networks provide real time information on stories or events happening across the globe. However, often at least some online posts about a given event can turn out to be rumours, spread maliciously or unwittingly via the social networks. PHEME addresses this challenging problem of detecting and modeling rumour propagation in social networks. A major step towards PHEME's successful completion is the integration and complete mid-way evaluation of PHEME's technical components.

This deliverable describes evaluation results and integration details for the whole of the existing technical PHEME framework. Specifically, this includes the linguistic processing and linking; sub-story detection; detection of spatio-temporal context; and critically, preliminary identification of mis- and dis-information. A large part of the research included here has already been published at high-ranking academic venues; this deliverable demonstrates the consortium's successful application and continuation of this work, its application to the PHEME case studies, and its extension into living, distributable systems. Further, this evaluation deliverable reports on integration of all these individual tools into a connected system, demonstrated already at ICT 2015, and shortly after – again at EDF 2015.

The results of this work comprise:

- summaries of extensions to and evaluations of technical content so far, including language-specific pre-processing, cross-language linking, sub-story detection, and rumour classification;
- integration strategy and results;
- evaluation of the PHEME components against the two case-studies, medical information and journalism.

Contents

Chapter 1

Introduction

Recently people have started using social media not only to keep in touch with family and friends, but also increasingly as a news source. However, knowledge gathered from online sources and social media comes with a major caveat – it cannot always be trusted. Rumours, in particular, tend to spread rapidly through social networks, especially in circumstances where their veracity is hard to establish. For instance, during an earthquake in Chile rumours spread through Twitter that a volcano has become active and there was a tsunami warning in Valparaiso (Marcelo et al., 2010). This creates a large and real-time need for veracity assessments and feedback for social media data.

To build a research system for rumour detection and classification, we need accurate tools that can operate on very noisy text, in a variety of languages. These are vital to catching rumours as they emerge and providing the most possible information to stakeholders. Such tools often consist of multiple components and can be divided into discrete subparts. Each of these parts must be able to tolerate the variances of user-generated content (UGC) in the respective language. This places performance constraints on the system in terms of *quality*. The tools need to capture and provide enough information to enable accurate rumour recognition and classification, which is a novel demand which PHEME addresses.

Additionally, these tools also need to be able to inter-operate, and handle high volume streaming content in a timely fashion. Therefore, there are not only *quality* performance constraints on the system, but also computational performance constraints. Each service must be able to process information at an acceptable rate, handle bursts, and handle failure elegantly. Above this, common formats must be agreed by the consortium for exchanging data in a consistent and comprehensible way. To achieve such a format, we all need to know which information to add in order to supply complete information to other components in the system. Between the variety of languages, partners and subsystems in the consortium, this poses a challenging task.

This deliverable serves as a mid-way evaluation for the systems, methods and tools developed in PHEME to date. We report on quality performance of integrated components,

as well as describing the in-situ integration solution and the protocol agreed for this.

Chapter 2 examines all the content analysis methods built so far. This covers the linguistic pre-processing, spatio-temporal extraction, sub-story event detection, linking content across media and languages, detecting disputed information, and identification of mis-and dis-information. As specified in the description of work, systems are evaluated against the datasets gathered for the case studies in PHEME: medical information and journalism.

Chapter 3 puts forth the technical aspects of integration in PHEME. It describes the system used to share information across all consortium technical partners, and the choices behind this system (Kafka). This chapter also includes the shared data format we have arrived at.

1.1 Relevance to PHEME

The PHEME project aims to detect and study the emergence and propagation of rumours in social media, which manifest as dubious or false claims. In order to do this, there are many empirical and technical processes that need to have high quality performance and be inter-operable. This deliverable serves to measure progress towards both these goals. The output of this deliverable's content has also driven our live demonstrations, at the ICT '15 event and also the forthcoming EDF '15.

1.1.1 Relevance to project objectives

Producing integrated research on rumour detection is a key goal of PHEME, and so we require a prototype system for sharing results and demonstrating that our outputs work not only in theory but also in practice.

In particular, this deliverable reports on a number of quantitative evaluation experiments, and thus contributes directly to objective 5, defined in the PHEME description of work, as follows:

> Test and evaluate the newly developed methods through (i) quantitative experiments on gold-standard data, acquired both through traditional domain expert annotation and crowdsourcing; and (ii) qualitative assessments in the use cases on health and digital journalism, involving key stakeholders from two focus groups.

The focus of Task 6.4 and D6.2.1 is entirely quantitative, while qualitative assessment with stakeholders will be undertaken in the respective use cases (WP7 and WP8).

1.1.2 Relation to forthcoming and prior research in PHEME

This deliverable provides a single point of evaluation for much of the content analysis work in WP2 (Ontologies, Multilinguality, and Spatio-Temporal Grounding), WP3 (Contextual Interpretation) and WP4 (Detecting Rumours and Veracity). Specifically, the latest results of T2.1, T2.2, T2.3 and T2.4 are evaluated from WP2. From WP3, this deliverable evaluates work originating in T3.1 and T3.3. Finally, this deliverable reports preliminary results from T4.3 (underway until M30) and T4.4 (underway until M32). The results and findings from this document go forward to drive technical aspects of PHEME development and integration for the remainder of the project, with a second re-evaluation in D6.2.2 (M33).

1.1.3 Relation to other work packages

The datasets created in Task 2.1, WP7, and WP8 are used for iterative development and evaluation. This includes tuning of and reporting on content analytics methods from WP2, WP3, and WP4.

PHEME's internal evaluation process consists of two iterations. The first iteration started from month 14, in order to help improve the methods for M24 deliverables in WP3 and WP4, whereas the second iteration will start from month 26 to underpin the final WP3 and WP4 deliverables. This deliverable reports on the results of the first iteration. Evaluation on unseen data in the first iteration is based on use-case data (from D7.2 and D8.2).

Chapter 2

Evaluation of content analytics

2.1 Linguistic pre-processing

2.1.1 Bulgarian

In D2.2 (Declerck et al., 2014) the first steps in NLP processing of Bulgarian Tweet Corpus were reported. The Bulgarian Tweet Corpus has as its topic the Bank Crisis in Bulgaria in July 2014. The tweets were semi-automatically annotated with DBpedia URIs. This means that the named entities were annotated automatically by a specially designed rule-based module for Bulgarian, and then the disambiguation was performed manually. The corpus consists of 1150 tweets, containing 24721 tokens. There are 1410 named entities, annotated with DBpedia URIs.

At this stage we put our efforts in improving linguistic processing on the morpho-syntactic and syntactic levels. The state-of-the-art metrics are as follows.

The BulTreeBank NLP Pipeline has the following state-of-the-art accuracy on the BulTreeBank data:

- Morpho-syntactic Tagging: 97.98 %
- Dependency Parsing: 92.9 %

After some cleaning of the data, the NLP pipeline was run over the tweets. Then 100 sentences were manually checked with respect to the morphosyntactic features and the dependency relations. The estimated accuracy is as follows:

- Morpho-syntactic Tagging: 83.33 %
- Dependency Parsing: 82.52 %

As it can be seen, the parsing step relies very much on the tagging step. The main errors coming from the tagging are due to the consistently wrong choice of some frequent words - both open and close class ones, such as: said, will, somebody, everybody, etc. After the repairing, the result improved to around 87%.

The dependency parsing performs quite well in general. The main errors, as mentioned above, are due to the wrong morpho-syntactic tags. The other errors are the typical ones for a parser: mixing subject with object and vice versa, wrong root in verb-less sentences, wrong attachments. After the improvements on the previous level, the accuracy improves to around 86 %, which as a result is comparable to the current state-of-the-art results for constituency parsers over Bulgarian data.

The final version of this deliverable D6.2.2. will evaluate also the Bulgarian adaptation of the YODIE system for linked entities disambiguation. YODIE was developed originally in the TrendMiner[1] project and the algorithms are now being refined further by USFD as part of the DecarboNet[2] project.

2.1.2 English

Earlier, D2.2 (Declerck et al., 2014) reported details of linguistic pre-processing for English. These tools reached state-of-the-art accuracy on tweets, the toughest of the social media text types. Since then, we have rolled out the tools across a number of different text types, achieving strong results on other forms of social media text (e.g. forums, Reddit data). Additionally, various parts of the linguistic pre-processing pipeline have been wrapped and connected to the PHEME Kafka integrated framework, communicating and adding annotations in the consortium-agreed format.

Critical to this is our implementation of language identification, which is a key part of the D2.2 (Declerck et al., 2014) content and GATE application. Language identification allocates incoming content to one of multiple pipelines in Kafka, allowing language-specific systems to each receive a specific feed. The integration of the subsequent English pre-processing components, i.e. POS tagging, NER, and entity disambiguation, is ongoing and will be completed by the M24 D6.1.2 deliverable on the PHEME integrated veracity framework.

Here, we report benchmarks in terms of component throughput speed, and also performance in terms of items found in the case-study text.

Evaluation against USFD historical data from Twitter garden hose feed

For this process, we took a sample of 120 000 tweets from November 2014 from the USFD garden hose archive and replayed it through various components. The data pro-

[1] http://www.trendminer-project.eu
[2] http://www.decarbonet.eu

System	Overall	English	Dutch	French	German	Spanish
TextCat	89.5%	88.4%	90.2%	86.2%	94.6%	88.0%
langid	89.5%	92.5%	89.1%	89.4%	94.3%	83.0%
Cybozu	85.3%	92.0%	79.9%	85.8%	92.0%	77.4%
TextCat (twitter)	**97.4%**	**99.4%**	**97.6%**	**95.2%**	**98.6%**	**96.2%**
langid (twitter)	87.7%	88.7%	88.8%	88.0%	92.5%	81.6%

Table 2.1: Language classification accuracy on the ILPS dataset for systems before and after adaptation to the microblog genre.

Approach	Precision	Recall	F1
PTB Regexp	90%	72%	80%
PTB Regexp (twitter)	**98%**	**94%**	**96%**

Table 2.2: Tokeniser performance on sample microblog text

cessing rate (single core, i7; modern SSD I/O) is detailed below.

- **Language ID:** 15 460 documents/minute (per node)
- **Tokenisation, PoS tagging, normalisation, co-reference, NER:** 1 684 documents/minute (per node)

The quality performance was evaluated in D2.2 (Declerck et al., 2014), and included below for completeness.

Specifically, language identification results are shown in Table 2.1; tokenisation in Table 2.2; part-of-speech tagging in Table 2.3; and named entity recognition in Table 2.4.

In more recent research (Derczynski et al., 2015a), we discovered that using Brown clustering in a non-standard way actually increases performance in social media extraction above the state-of-the art reported in D2.2 (Declerck et al., 2014).

Brown clustering (Brown et al., 1992) is a hierarchical agglomerative clustering technique that relies on distributional information for its objective function. This means that it is both entirely unsupervised, and can also be used for reducing sparsity in language datasets, at the least for languages where word order is important. Our research found that, by ignoring default parameters and using a sufficiently large amount of USFD's Twitter archive data, we could in fact beat the state of the art without any feature engineering, using a relatively small training dataset. Setting $c = 1000$ and $T = 32M$, we

Tagger	Token accuracy	Sentence accuracy
Stanford	73%	2%
Ritter	85%	9%
(Derczynski et al., 2013)	89%	20%
PHEME	92%	26%

Table 2.3: Part-of-speech tagger accuracy (English)

	Per-entity F1				Overall		
System	**Location**	**Misc**	**Org**	**Person**	**P**	**R**	**F1**
ANNIE	40.23	0.00	16.00	24.81	36.14	16.29	22.46
DBpedia Spotlight	46.06	6.99	19.44	48.55	34.70	28.35	31.20
Lupedia	41.07	13.91	18.92	25.00	38.85	18.62	25.17
NERD-ML	**61.94**	23.73	32.73	**71.28**	52.31	**50.69**	51.49
Stanford	60.49	**25.24**	28.57	63.22	59.00	32.00	41.00
Stanford-Twitter	60.87	25.00	26.97	64.00	54.39	44.83	49.15
TextRazor	36.99	12.50	19.33	70.07	36.33	38.84	37.54
Zemanta	44.04	12.05	10.00	35.77	34.94	20.07	25.49
PHEME	43.93	20.09	**43.18**	65.13	**62.18**	49.00	**54.11**

Table 2.4: Named entity recognition performance over the evaluation partition of the Ritter dataset.

reached an F1 of 54.1 on this dataset.[3] This means that, given some annotations, we now have a cross-lingual high-performance tool for named entity recognition.

Entity recognition on medical use-case data

In a sample of 7149 tweets related to mephedrone, the following named entities were extracted:

- Locations: 1256
- Organizations: 968
- Person mentions: 1006

These are general observations without evaluation against a gold standard. The reference annotations come from D7.2.2, which includes named entity annotation over this corpus. This is due in M24, and will comprise the gold standard against which systems can be annotated. However, there are evaluation figures for the journalism use-case, detailed below.

Entity recognition on journalism use-case data

The experimental sample comprises 393 rumour-laden tweets from D8.2:

- Locations: 217
- Organizations: 146

[3]In this case, we use shearing to generate Brown cluster features at depths 4, 6, 8 and 10, and passive-aggressive CRF (Derczynski and Bontcheva, 2014).

Entity type	Precision	Recall F1	
Location	68.42	15.85	25.74
Organization	48.00	27.91	35.29
Person	20.28	40.28	26.98

Table 2.5: Named entity recognition results on human annotated tweets

- Person mentions: 123

SWI have provided human annotations for named entities in these data. Therefore, we are able to conduct an initial extrinsic evaluation over this dataset. We ran the English components of the linguistic pre-processing tool, and found results as given in Table 2.1.2. As with all results in this deliverable, these are strict results, requiring precise entity boundary recognition.

As can be seen from these results, a particular challenge is low recall, which is a well known challenge in all Twitter NER research (Derczynski et al., 2015b).

We are working on two threads of research for reducing this. Firstly, we are carrying out an investigation into the effect of studying solely newswire for statistical NER for many years. Our hypothesis is that systems and methods tend to overfit to newswire as a text type and are thus no longer able to generalise effectively. Initial experimental results have supported this, with more ongoing work currently focused on exhaustive method vs corpora investigation.

Secondly, we are aiming to address concept drift in entity recognition, both over time and over diverse media content. We will study the breadth of surface forms that entities may have, along with the multiple possible textual contexts in which named entities are embedded. This latter effect is present in newswire, though with sufficiently low magnitude and frequency so as not to be hugely evident; indeed, the field has often concentrated on older datasets for evaluation, sidestepping this drift entirely.

2.1.3 German

We report here on some work done in cooperation with colleagues at Saarland University (USAAR), with whom the PHEME USAAR team is sharing data and processing strategies for German text in social media, and in this case more specifically computer mediated communication (CMC). The following is summarizing work described in Beißwenger et al. (2015b) and further building on Horbach et al. (2014), work done in the context of the German national project "Schreibgebrauch" (http://www.coli.uni-saarland.de/projects/schreibgebrauch/en/page.php?id=index). Graphics are taken from the presentation by Michael Beisswanger (Uni Dortmund) and Stefan Thater (USAAR) at the NLP4CMC Workshop, a satellite event to GSCL 2015, in which PHEME was also presented (see https://sites.google.com/site/nlp4cmc2015/).

PoS annotation and tagset

The work describes experiences collected using also the Extended STTS tag-set ("STTS 2.0") with categories for CMC-specific items and for linguistic phenomena typical of spontaneous dialogic interaction. The original STTS tagset can be accessed at http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html and was described first in Schiller et al. (1999). The extensions to STTS, towards STTS 2:0 is described in Bartz et al. (2014) Zinsmeister et al. (2013). STTS 20.0 is downward-compatible with STTS (1999). Examples for STTS 2.0 are given in the table just below:

PoS tag	Category	Examples
I. Tags for phenomena which are specific for CMC / social media discourse:		
EMO ASC	ASCII emoticon	:-) :-(^^ O.O
EMO IMG	Graphic emoticon	
AKW	Interaction word	*lach*, freu, grübel, *lol*
H ST	Hash tag	Kreta war super! #urlaub
ADR	Addressing term	@lothar: Wie isset so?
URL	Uniform resource locator	http://www.tu-dortmund.de
EML	E-mail address	peterklein@web.de
II. Tags for phenomena which are typical for spontaneous spoken language in colloquial registers:		
VV PPER	Tags for types of colloquial contractions which are frequent in CMC (APPRART is already existing in STTS 1999)	schreibste, machste
APPR ART		vorm, überm, fürn
VM PPER		willste, darfste, musste
VA PPER		haste, biste, isses
KOUS PPER		wenns, weils, obse
PPER PPER		ichs, dus, ers
ADV ART		son, sone
PTK IFG	'Intensitätspartikeln', 'Fokuspartikeln', 'Gradpartikeln'	sehr schön, höchst eigenartig, nur sie, voll geil
PTK MA	Modal particles	Das ist ja / vielleicht doof. Ist das denn richtig so? Das war halt echt nicht einfach.
PTK MWL	Particle as part of a multi-word lexeme	keine mehr, noch mal, schon wieder
DM	Discourse markers	weil, obwohl, nur, also, ... *with V2 clauses*
ONO	Onomatopoeia	boing, miau, zisch

As described in Horbach et al. (2014), the strategy chosen for trying to improve the performance of off-the-shelf taggers for German, which were trained originally on news corpora, when applied to CMC text, was to annotate only a relatively small amount of the CMC data with the new STTS 2.0 tagset and to retrain the system on the aggregated corpora (news + CMC). Horbach et al. (2014) and Beißwenger et al. (2015a) can report on significant improvements delivered by the retrained taggers. The tables showing the

results, displayed just below, are reflecting the measured performances of the TreeTagger (http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/) Standard PoS taggers perform poorly on CMC data. Applied to a chat corpus, for example, TreeTagger reaches an accuracy of 71% (vs. *97% accuracy on Newspaper). The STTS 2:0 annotated CMC corpus was then aggregated to the existing TIGER gold standard[4] training data, and TreeTagger retrained on this aggregated corpus. A manual gold-standard annotation was provided for 12k tokens (4k for training and 8k for the test set). The resulting accuracy of TreeTagger when applied to the chat corpus is 83% (as a reminder: it was 71% before our experiment with the small manually annotated CMC data set). Errors could be reduced by 39%. On the basis of the added extended annotation, the tagger can now assign CMC-specific tags (emoticons, action words) - but the "non-standardness" of written CMC is still causing trouble in several respect. Beißwenger et al. (2015a) report the following figures:

- 25 out of 35 occurrences of Emoticons (EMO) tagged correctly (71%)
- 36 out of 59 occurrences of Interjections (ITJ) tagged correctly (61%)
- 22 out of 37 occurrences of Action words (AKW) tagged correctly (59%)
- 14 out of 15 occurrence of acronymic AKW tagged correctly (= 93%)
- 8 out of 17 tagged correctly simple verb-AKW tagged correctly (= 47%)
- 48 out of 72 NN and NE without capitalization tagged correcly

Particularly problematic are nominalisations without capitalization (das küssen/VVFIN, was verdauliches/ADJA, im zeugnis nur einsen/VVINF, leute zum anpacken/VVINF). Here only 8% of the data is correctly tagged. And for colloquial spellings, typos, character iterations we have 42 tokens out of 87 correctly tagged (= 48%).

Current work, to be reported in the next version of this deliverable, will consider the German data sets collected in the PHEME, but for which we did not have a gold standard at our disposal in the first year of the project.

2.2 Spatio-temporal extraction

In D2.3 (Derczynski and Bontcheva, 2015), the project developed tools for extracting information about time and space. These were specifically aimed at social media text. The temporal information was event mentions and mentions of times – this followed the ISO-TimeML standard. The spatial information was the document's expected origin location and mentions of locations from the document, following the ISO-Space standard.

[4] See http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.en.html

2.2.1 Annotation approach

PHEME takes a standardised approach to spatio-temporal annotation, relying on ISO standards. The standards we start from are ISO-TimeML (Pustejovsky et al., 2010) and ISO-Space (Pustejovsky et al., 2011), well recognised community standards for temporal and spatial annotation respectively.

Following recent research on customising annotation (Schneider, 2015), we reduce the many diverse types of entity supported by these ISO standards down to the set that is both applicable to social media text and also fits within the scope of our task. Annotating according the full standards is superfluous to needs; however, taking a strict subset of the standards is much cheaper than rebuilding a spatio-temporal annotation and encoding standard, especially given the large volume of guidelines, edge case guidance and other supplementary material that has accumulated for ISO-TimeML and ISO-SpatialML. This means that, for example, we will ignore all temporal relation information and data about spatial paths and other relations. While generally critical to the understanding of text, these relations are not immediately necessary to the spatial grounding of concepts and entities that is required in PHEME.

2.2.2 Datasets

Following D2.3(Derczynski and Bontcheva, 2015), we use four distinct datasets for evaluation. The goal is to develop a machine-learning based chunker for extracting events, and so training data is chosen from a variety of sources. Two pre-annotated datasets support this: the W-NUT/Ritter NE annotations (Ritter et al., 2011; Baldwin et al., 2015) for spatial, and the TempEval-2 data for temporal (Verhagen et al., 2010). Manually annotated evaluation data is drawn from the rumours gathered in deliverable D8.2 (Hoi, 2015), as detailed in D2.3 (Derczynski and Bontcheva, 2015).

In total, this evaluation dataset contains 605 events, 122 timexes, 139 spatial entities and 223 locations. Finally, we perform unsupervised feature extraction through Brown clustering using a sample of tweets from Twitter's 10% gardenhose feed. This is a fair sample (Kergl et al., 2014), drawn between 2009 and 2015 to induce resilience against entity drift (Masud et al., 2011; Fromreide et al., 2014). It is referred to here as the garden hose archive (GHA).

We evaluated two models: the first trained over the amassed prior annotated data, which was a blend of both social media and newswire text; the second using a split of the social media rumour data. The blended training data comprised about 69,000 tokens for temporal and about 10,600 for spatial entities. The second dataset is markedly smaller, at 5,400 tokens, and only for spatial entities, as there were no pre-existing temporal annotations.

Task	Precision	Recall	F1
Using blended data			
Event recognition	68.55	69.29	68.92
Timex recognition	59.57	52.83	56.00
Location recognition	81.25	64.36	71.82
Spatial entity recognition	48.15	18.06	26.26
Using only rumour data			
Location recognition	67.86	42.22	52.05
Spatial entity recognition	28.57	5.88	9.76

Table 2.6: Spatio-temporal entity recognition in tweets

2.2.3 Performance

We evaluated the performance of these tools over a newly-created spatio-temporally annotated corpus in D2.3 (Derczynski and Bontcheva, 2015). The construction of these tools is ongoing work, as no previous research has addressed either spatio-temporal information extraction from social media, or joint spatial and temporal information extraction, and we have seen encouraging results.

Table 2.2.2 describes the quality of spatial and temporal entity recognition in the system used for integration. The event and location recognition scores are approaching state-of-the-art for newswire, which is impressive, given that these are over one of the noisier twitter datasets. Regarding computational performance, these are labelled at 2053 documents / minute (3183 tokens / second).

2.2.4 Future work

Future technical work includes upgrading the representations to include more Brown clusters, and replacing the token references with continuous embeddings; using a higher-order model to account for the non-linear structure of language and especially dysfluency-rich language on social media. Additionally, the scope of the work will be broadened, to account for not only the types of temporal and spatial expressions, but also their normalisations. We include information on spatio-temporal bisemy as described in D2.3 (Derczynski and Bontcheva, 2015). Finally, the methods will be adapted to account for the class skew present in social media, exacerbated by the multiclass nature of e.g. EVENT and LOCATION type.

2.3 Sub-Story detection

Task 3.3 focused specifically on developing algorithms for clustering posts about a given event (e.g. the Ferguson unrest) into topically related sub-stories, the veracity of which

can then be examined by the PHEME veracity intelligence methods. Two algorithms (LSH and SC) and their evaluation on the London riots and selected PHEME rumour datasets from D8.2 (Hoi, 2015), were reported in deliverable D3.3.1 (Srijith et al., 2015). In the current deliverable, evaluation results are extended with an additional algorithm (HDP). Next the three algorithms are discussed briefly, followed by their comparative evaluation.

2.3.1 Method Overview

This deliverable investigates sub-story detection based on Hierarchical Dirichlet Processes (HDP) (Teh et al., 2006). HDP is a non-parametric Bayesian model, which can effectively model the sub-story detection task and automatically identify the number of sub-stories. The rest of this section compares this approach to the two algorithms investigated in D3.3.1: firstly, Spectral Custering (SC) using pointwise mutual information (Preotiuc-Pietro et al., 2013), and secondly, a streaming model based on Locality Sensitive Hashing (LSH) (Rajaraman and Ullman, 2011).

HDP has the advantage of being a non-parametric approach. It can detect automatically the number of topics based on the distribution of words in the data set. We investigate using HDP for sub-story detection in particular, since it can model the hierarchical structure underlying the topic distribution. For sub-story detection, we need to find subtopics associated with a main topic, such as the Ferguson protests, and HDP is developed specifically to handle this kind of problem. HDP achieves this by extending the Dirichlet Process Mixture Model (DPMM) (Murphy, 2012) to a hierarchical setting.

In particular, for sub-story detection in Twitter, all tweets relate to the same main topic, with individual tweets addressing various sub-topics of the main topic. HDP can thus be used to identify shared topics (the shared main topic) and tweet specific topics, where each topic is characterized by a set of words. These identified topics are used to cluster tweets based on maximal overlapping of words in tweets with the words associated to the topics.

2.3.2 Comparative evaluation using precision and recall

The three methods are evaluated using precision and recall, as detailed in D3.3.1 (Srijith et al., 2015). Tweet text is pre-processed, including stop word removal, stemming, etc. (see Section 3.3. in (Srijith et al., 2015) for details).

The HDP experiments are conducted using HCA, a topic modelling toolkit based on Dirichlet processes (Buntine and Mishra, 2014) [5]. It learns the concentration parameter of HDP from data (initialized to default value of 1). It requires the user to provide an upper bound on the number of topics (k). The spectral clustering approach depends mainly on the parameter k, which determines number of clusters in the dataset. The approach is run

[5]The software can be downloaded from `http://mloss.org/software/view/527/`

Table 2.7: Results of HDP, SC, and LSH on London riots for different parameter settings. Best results are indicated in bold.

	London Riots		
Method	P_{micro}	R_{micro}	F_{micro}
HDP (k50)	0.4188	**0.2759**	**0.3326**
HDP (k100)	0.4194	0.2013	0.2720
SC (k50)	0.1833	0.2666	0.2172
SC (k100)	0.4522	0.2539	0.3252
LSH (k12h56b10)	**0.5948**	0.2258	0.3273
LSH (k13h71b10)	0.4976	0.2323	0.3167

Table 2.8: Results of HDP, SC, and LSH on Ferguson and Ottawa data sets for different parameter settings. Best results are indicated in bold letters.

	Ferguson			Ottawa		
Method	P_{micro}	R_{micro}	F_{micro}	P_{micro}	R_{micro}	F_{micro}
HDP (k200)	0.0536	0.0889	0.0668	0.1799	0.1431	**0.1594**
HDP (k300)	0.1366	0.1057	**0.1191**	0.2182	0.1249	0.1588
SC (k400)	0.0131	**0.1622**	0.0242	0.0519	**0.1821**	0.0807
SC (k2000)	0.0422	0.0861	0.0566	0.0873	0.1244	0.1025
LSH (k12h56b10)	**0.3441**	0.0301	0.0554	**0.4797**	0.0314	0.0589
LSH (k13h71b10)	0.3430	0.0407	0.0728	0.3768	0.0285	0.0529

by filtering out words with a threshold of 0.1 for the NPMI score and with threshold of 10 for word frequency. We perform experiments with different values of k for HDP and SC. The LSH approach depends on the parameters k (number of bits), h (number of hash tables) , and b (bucket size). The experiments are conducted with different values of these parameters. We present only the results obtained with best two parameter settings of the approaches.

The experimental results obtained on the London riots data set are presented in Table 2.7. The HDP and SC approaches are run in this data set by partitioning the data set into 50 partitions with approximately 50,000 tweets in each partition. The table provides number of clusters per partition for HDP and SC approaches.

As can be seen, LSH has very high precision but low recall. HDP and SC provide better recall than LSH. The precision obtained with HDP is higher than that obtained for SC. Overall, HDP has a higher F-score than all other approaches.

In particular, the LSH algorithm produces a high number of clusters which contain almost only tweets and their retweets. This results in higer precision but low recall.

The spectral clustering algorithm, on the other hand, produces results with high recall but low precision. It is found to cluster tweets from related sub-stories into the same

Table 2.9: Results of HDP, SC, and LSH on Ferguson and Ottawa data sets (considering conversational structure) for different parameter settings. Best results are in bold.

	Ferguson			Ottawa		
Method	P_{micro}	R_{micro}	F_{micro}	P_{micro}	R_{micro}	F_{micro}
HDP (k200)	0.273	0.3674	0.3132	0.4749	0.4612	0.4679
HDP (k300)	0.3822	**0.4199**	**0.4001**	0.4398	**0.5691**	**0.4968**
SC (k400)	0.0722	0.4091	0.1227	0.1786	0.3581	0.2383
SC (k2000)	0.2149	0.3034	0.2515	0.1588	0.3143	**0.2109**
LSH (k12 h56 b10)	**0.5589**	0.3087	0.3977	0.5428	0.3038	0.3895
LSH (k13 h71 b10)	0.5079	0.3106	0.3854	**0.7777**	0.2877	0.4200

cluster, resulting in few very big clusters.

HDP provides a more balanced result with comparatively higher precision and recall. It is a more fine grained approach which can distinguish subtle differences in various sub-stories, due to the hierarchical modeling of the topics with some shared vocabulary.

The experimental results obtained on the Ferguson and Ottawa datasets are provided in Table 2.8. We observe that the evaluation scores calculated for these data sets are lower compared to those on the London riots data. This is due to the presence of conversational tweets in these data sets. The conversational tweets which are not topically similar get assigned to a completely different cluster. Even in this case, we observe that the F-score obtained with HDP is much better than that obtained by the other approaches.

In order to handle conversational tweets, we follow a strategy where we cluster only source tweets and the conversational tweets are assigned to the cluster of its corresponding source tweet. This is possible in these data sets as they maintain the source-reply structure.

We show the performance of the approaches upon considering the conversational structure in Table 2.9. We can see that the results have improved considerably compared to those in Table 2.8. There is an order of magnitude improvement in recall and F-score. In this case, we observe that LSH provides better precision while HDP provides better recall. The F-score obtained with HDP is higher than SC and LSH.

We also consider a baseline method which clusters the tweets together using only their conversational structure. By construction, this approach will have a precision of 1. We want to investigate if the story detection approaches can get a better recall than this baseline. We found that the baseline has a recall of 0.2545 and 0.1696 for Ferguson and Ottawa respectively. As can be observed from Table 2.9, the story detection approaches perform better than the baseline in terms of recall.

2.3.3 Comparative evaluation using adjusted mutual information

The main drawback of evaluating based on precision, recall and F-measure is that they do not penalize methods producing large number of small clusters (high precision and low recall). Such large numbers of small clusters, however, are not useful in practical applications, since users struggle to navigate effectively large number of clusters. Therefore, a second set of comparative evaluation experiments was performed, using adjusted mutual information (AMI) (Vinh et al., 2009), which takes cluster numbers and size into account.

Information theoretic scores such as mutual information have been used to compare clustering approaches (Banerjee et al., 2005; Meilă, 2005). They are theoretically grounded and capture better cluster quality. The mutual information (MI) between two clusters $\mathbf{U} = \{U_1, \ldots, U_R\}$ (true clustering of tweets) and $\mathbf{V} = \{V_1, \ldots, V_C\}$ (generated clustering of tweets) quantifies the information shared among them and provides the reduction in uncertainty on $\mathbf{U}$ upon observing $\mathbf{V}$. The MI score between $\mathbf{U}$ and $\mathbf{V}$, $I(\mathbf{U}, \mathbf{V})$ can be computed as

$$MI(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{R} \sum_{j=1}^{C} p(i,j) \log \frac{p(i,j)}{p(i)p'(j)}. \tag{2.1}$$

Here, $p(i)$ provides the probability that tweets belong to cluster U_i, $p'(j)$ provides the probability that tweets belong to cluster V_j, and $p(i,j)$ provides the probability that tweets belong both to clusters U_i and V_j. When the clusterings are identical, MI score takes a higher value upper bounded by $min\{H(\mathbf{U}), H(\mathbf{V})\}$, where $H(\mathbf{U}) = \sum_{i=1}^{R} p(i) \log(p(i))$ is the entropy of the clustering $\mathbf{U}$. If the clusterings are not identical but independent, MI score will be close to zero. One uses a normalized MI (NMI), score which normalizes the MI score to be between zero and one.

A drawback of the MI and NMI scores is that they are not corrected for chance, i.e. they do not have a constant baseline value which is the average obtained for random clustering of the data (Vinh et al., 2009). These scores tend to be higher for a clustering with larger number of clusters, or when the ratio of the total number of data points to number of clusters is small. Note that the MI score can be high for a clustering approach which categorizes each point into a separate cluster.

Therefore here, we consider the adjusted mutual information (AMI) score (Vinh et al., 2009), corrected for chance by subtracting the expected mutual information score from both the numerator and denominator of the normalized mutual information score. Table 2.10 provides the AMI scores obtained by the HDP, SC and LSH approaches on different data sets.

We observe from Table 2.10 that HDP again achieves the best performance. In this case, we also note that SC has improved performance, which is often better than LSH. The AMI score penalizes the LSH algorithm which produces a very large number of small clusters using the expected mutual information score which grows with the increase in number of clusters.

Table 2.10: Adjusted mutual information scores obtained for HDP, SC and LSH on various data sets. We report the best AMI score obtained for different parameter setting of the approaches.

Method	Ferguson	Ottawa	Method	LondonRiot
HDP (k100)	0.46	0.59	HDP (k25)	**0.32**
HDP (k200)	0.46	0.55	HDP (k50)	0.31
HDP (k300)	**0.47**	**0.60**	HDP (k100)	0.29
SC (k400)	0.40	0.39	SC (k50)	0.31
SC (k400)	0.38	0.43	SC (k100)	0.31
SC (k2000)	0.39	0.42	SC (k200)	0.28
LSH (k12 h56 b10)	0.40	0.46	LSH (k12 h56 b10)	0.29
LSH (k13 h71 b10)	0.40	0.47	LSH (k13 h71 b10)	0.30

Table 2.11: Running times of the approaches on FSD, London riots, Ferguson and Ottawa data sets.

Method	LondonRiot	Ferguson	Ottawa
HDP	2 hours	196 seconds	55 seconds
SC	1.5 hours	183 seconds	52 seconds
LSH	4 hours	151 seconds	35 seconds

2.3.4 Runtime Comparison

Table 2.11 provide runtime comparisons of HDP, LSH and SC on different data sets. The algorithms are run on a Linux computer with a 4 core Intel CPU 3.40 GHz, with 16 GB RAM.

In terms of run time, all approaches are comparable on Ferguson and Ottawa data. In the case of London riots, LSH is significantly slower.

2.4 Cross-media and cross-language linking

In Task 3.1 ("Cross-Media and Cross-Language Linking") of PHEME we propose a cross-media (CM) and cross-lingual (CL) linking algorithm to connect User-Generated Content (UGC) to topically relevant information in complementary media. The basis data set we use as the starting point is a subset of the collection of tweets that have been selected by the Digital Journalism use case partners (WP8). More precisely we are working on the datasets that have been collected in the context of the events of the Ottawa Shooting and

the Gurlitt art collection.[6]

Threads of tweets have been in PHEME manually annotated with a "story" or what we also call "event". E.g. the tweet 'RT @SWRinfo: Das Kunstmuseum Bern nimmt das Erbe des Kunstsammlers Cornelius #gurlitt an.' is assigned the event 'The Bern Museum will accept the Gurlitt collection', while 'NORAD increases number of planes on higher alert status ready to respond if necessary, official says. http://t.co/qsAnGNqBEw #OttawaShooting' is assigned the event 'NORAD on high-alert posture', etc. The story or the event is assigned to a different number of tweets, depending on how long their containing thread is. We can call such pairs "story/event" - thread a manually created cluster of tweets. While the main aim in task 3.1 is to generate links to documents outside of the UGC media, we are also interested in seeing if we can automatize the assignment of a "story" or "event" to a collection of tweets. We are reporting on those two aspects here, and present first results.

2.4.1 Cross-Linking UGC to other media: Augmenting the number of search terms for cross-media linking

Our first approach for establishing linking between UGC (e.g Twitter texts) and complementary media is based on the straightforward use of URLs that are used in the tweets. Beyond this we explore the possibility to link back the URLs to tweets that are in the same thread, sharing the manual story/event annotation, but lacking an URL. For each URL-containing tweet within each story/event thread, a tweet-to-document similarity calculation cycle is run between tweets that link an external web document, and the linked web document. Similarity is evaluated in terms of the Longest Common Subsequence (LCS) metric. LCS returns a similarity value between 0 (lowest) and 1 (highest) based on the longest shared n-gram for each text pair, without the need for predefined n-gram length and contiguity of tokens (cf. (Lin, 2004)). It also returns the string(s).

2.4.2 LCS terms extraction

We use LCS to collect the top-5 scored longest common token subsequences identified for a linked document, based on a series of LCS computations producing LCSs between one or more tweets linking this document and each sentence of the document. No linguistic knowledge is used, except for stopword filtering by the NLTK toolkit (see (Bird et al., 2009) or http://www.nltk.org/). Then the LCS cycle is applied to the same document set but paired with tweets that did not link external documents, based on the hand-labeled events. We are able to extract more, and lexically different phrases due to the link transfer. For example, for the web document with the headlines "Swiss museum accepts part of Nazi art trove with 'sorrow'— World news — The Guardian" the extracted top terms

[6]See respectively https://en.wikipedia.org/wiki/2014_shootings_at_Parliament_Hill,_Ottawa and https://en.wikipedia.org/wiki/2012_Munich_artworks_discovery for more details on the background news.

based on tweets linking to this document are: 'swiss museum accepts part nazi art trove', 'nazi art', 'swiss museum', 'part nazi', 'nazi', whereas the extracted top terms based on tweets not linking any document but being annotated with the same event as the tweets referring to this document, are'kunstmuseum bern cornelius gurlitt', 'fine accept collection', 'museum art', 'kunstmuseum bern cornelius gurlitt', 'kunstmuseum bern gurlitt', exemplifying that the Gurlitt dataset holds multilingual data, since we obtain terms not only in English, but in German as well.

2.4.3 Term extraction evaluation

Transfer to URL-less tweets

We are able to grow the set of extracted unique terms significantly if we perform the web link transfer step, when compared to not performing this step: from 110 to 186 in Gurlitt, and from 171 to 320 in Ottawa. The obtained term sets are highly complementary: about 70-90% of the phrases extracted from URL-less tweets are unseen in the phrase set extracted from URL-ed tweets.

Transfer based on automatically grouped tweets We have also compared the results of our LCS approach to experimental results where instead of using tweet clusters based on manual event annotations, we create tweet clusters by computing tweet similarity between each tweet and a centroid tweet for each event (designated by the phrase used in the manual event annotation), via a LCS similarity threshold. Inspired by Bosma and Callison-Burch (2007), who use an entailment threshold value of 0.75 for detecting paraphrases, we obtained our LCS similarity threshold t empirically by averaging the third quartile of LCS value distributions relating to an event over all events in a dataset (t ¿ 0.22). Figure 2.4.3 illustrates tweet similarity distribution in terms of LCS values for two events from the Gurlitt dataset. We computed LCS values both in an intra-tweet way (i.e., LCS for all pairs of tweets within a tweet event cluster, the size of which is indicated in the upper right corner), and in the centroid-tweet way (i.e., LCS for all centroid-tweet pairs within the event cluster). Since Gurlitt is a multilingual set, the LCS scores often have a very wide distribution, also indicated by the large number of outliers in the plot.

The approach on the current toy datasets achieves an event-based-mean precision of 1.0 for Gurlitt and 0.32 for Ottawa, while a event based- mean recall of 0.67 for Gurlitt and 0.78 for Ottawa. With this approach, we get much less URL-less tweets as with the manually annotated set (Gurlitt: 16 vs 43, Ottawa:117 vs 182), but this seems to have an impact only on the Gurlitt multilingual dataset on the amount of extracted unique phrases from URL-less tweets (Gurlitt: 64 vs 93, Ottawa: 178 vs 197).

Importantly, the quality and semantics of the extracted phrases for both datasets remain in line with those based on link transfer via hand-labeled events.

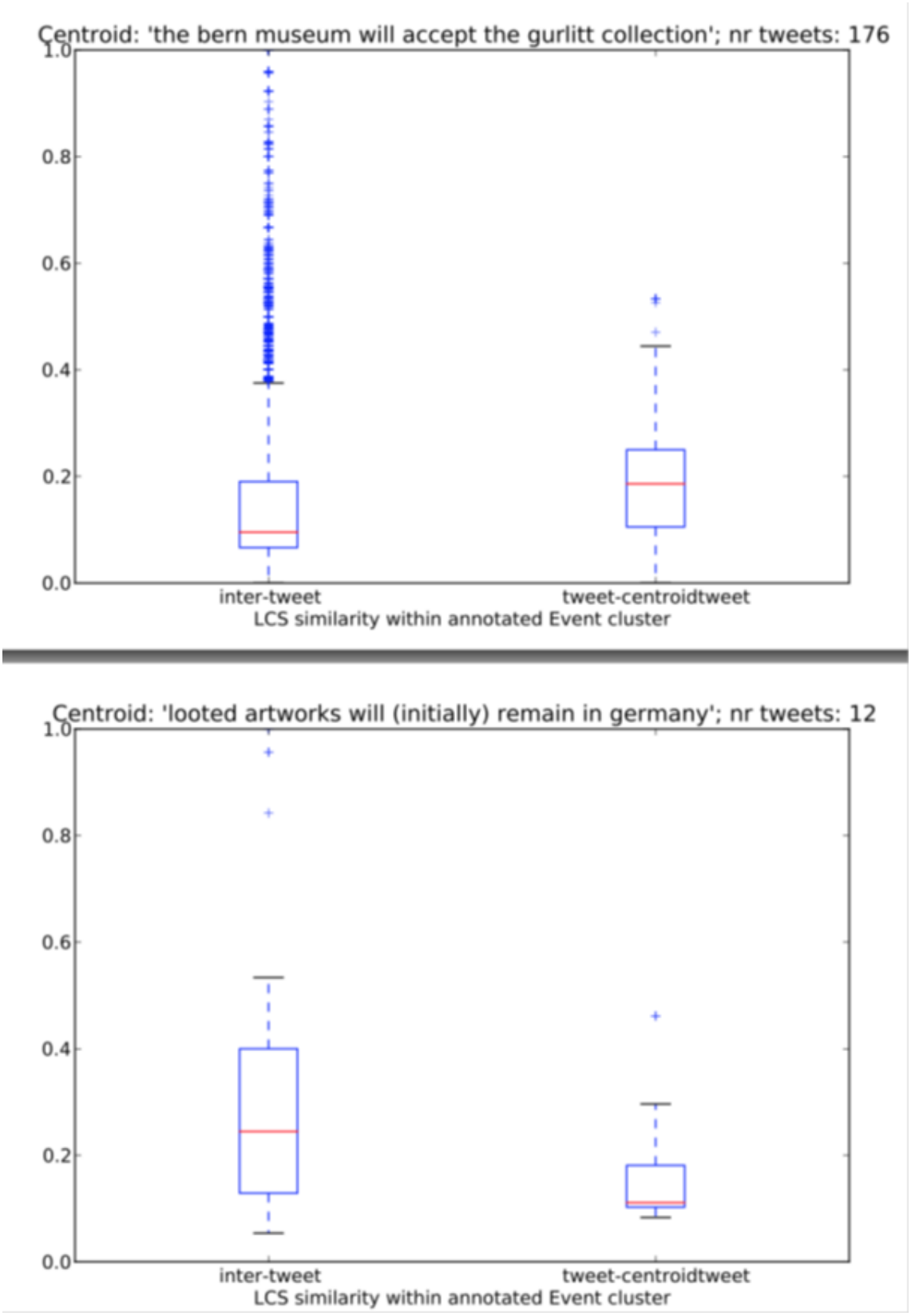

Figure 2.1: Tweet similarity distribution in terms of LCS values for two events from the Gurlitt dataset: tweet-tweet similarities within an event cluster, as well as centroid tweet - tweet similarities are plotted.

2.4.4 Comparing to Frequency-based term extraction

We extracted a document-based term set from all tokens in the fetched documents that were automatically classified as nouns; part-of-speech information was obtained from the NLTK platform. These sets seem semantically more general than the terms obtained by the LCS approach (e.g. 'ausstellung','sammlung', 'suisse', i.e., 'exhibition', 'collection', 'switzerland') and are also smaller in size: 75 unique terms from all documents linked from the Gurlitt set, obtained in a top-5-per-document cycle to simulate the LCS procedure, and 116 for Ottawa. The obtained term set consists of single tokens only, while the average phrase length using the LCS approach is 3.65 for Gurlitt and 3.13 for Ottawa.

2.4.5 Results and Conclusion

Our approach for linking UGC to complementary media, based on longest common subsequence computation, uses human input (use of URLs by the author) for extracting semantically meaningful terms of flexible length. We link tweets to complementary web documents, and create lexical descriptors extracted from tweets aligned with documents. The method is language-independent and unsupervised. The extracted phrases are expected to have indexing potential and could be used in their multi-word form or could be tokenized further. Scaling up from our current pilot setup, we are going to report on further qualitative and quantitative results on cross-media, cross-lingual text linking in forthcoming deliverables.

We will also investigate how our study relates to the recently created Shared Task in the Natural Language Processing community (Xu et al., 2015), and which deals with the creation of systems for Semantic Textual Similarity judgments on Twitter data. Given two sentences, the participating systems needed to determine a numerical score between 0 (no relation) and 1 (semantic equivalence) to indicate semantic similarity on the hand-annotated Twitter Paraphrase Corpus. The sentences were linguistically preprocessed by tokenisation, part-of-speech and named entity tagging. The system outputs are compared by Pearson correlation with human scores: the best systems reach above 0.80 Pearson correlation scores on well-formed texts. The organizers stress that "while the best performed systems are supervised, the best unsupervised system still outperforms some supervised systems and the state-of-the-art unsupervised baseline."

2.5 Detection of disputed information

We report here on on-going work on the initial T4.2 prototype for algorithms for detecting disputed information. This work is dedicated to the description on investigation of and experiments made with the open-source entailment platform that was resulting from the

European Excitement project.[7]

2.5.1 The Excitement Open Platform (EOP)

EOP is a generic architecture for textual inference. It implements a modular approach that allows various configurations to be deployed and adapted. State-of-the-art linguistic pre-processing has been included as well as some lexical-semantic resources, but EOP observes a clear separation between linguistic analysis pipelines and entailment components, aiming at easy language extensions. EOP comes with a Java API and with source code.[8]

Excitement implements textual inference as a matching between different text segments in order to decide if a text segment has the same meaning as the other or if meaning from one text segment can be implied from the other text segment. The text segment used to be compared with is normally just named "text" (or "premise") and the segment that is compared with the "text" is called the "hypothesis". In the case of the meaning of the hypothesis text being implied from the first text, the developers of EOP speak of a directional textual entailment ($T \rightarrow H$). In the case of the two text segments bearing the same meaning, the developers of the system speak of a bi-directional paraphrasing entailment relation ($T \rightarrow H \& H \rightarrow T$).

Among the 6 configurations implemented in EOP for processing (German) text, it has been reported that one of the most successful configurations is the alignment-based algorithm, and this is one of the algorithms we first tested by applying to EOP examples from our annotated German corpus in PHEME, doing this in a first phase using the online User Interface of EOP.[9] The intuition between the alignment-based algorithm is that: The more material in the hypothesis can be "explained" or "covered" by the premise, the more likely entailment is. As we can see, this approach implies a certain degree of lexical commonalities between the premise and the hypothesis, but EOP makes also use of lexical semantic networks (i.e. WordNet) and of computed distributional similarity in order to gain some independence from a pure lexical form similarity and so has a better coverage by identifying links between words or phrases across the premise and the hypothesis texts. EOP also makes use of paraphrase resources.

The entailment is then computed with the help of features relevant for the alignment. Features used for the time being are "Word Coverage", "Content Word Coverage", "Verb Coverage" and "Proper Noun Coverage". As an example, the premise text "Peter was Susan's Husband" and the hypothesis text "Peter was married to Susan" have the following features coverage: "Word Coverage = 4/5 and 100% (the four words of the premise are all linked to at least one word of the hypothesis, whereas "husband" is linked to "married" and "to"). We have two named entities in both text segments and they can be linked, so

[7] http://excitement-project.eu/

[8] http://hltfbk.github.io/Excitement-Open-Platform/

[9] http://hlt-services4.fbk.eu/eop/

that we have also 100% for the "Proper Noun Coverage", etc.

The textual entailment (TE) task is then defined as a classification task: on the basis of certain values associated to the features defined for the aligned text segments, the hypothesis can be (or not) classified a being entailed in the premise.

In order to be able to make concrete statements on the possible use of Textual Entailment technologies for PHEME, we need first to have a relevant corpus, both for training and testing the EOP platform.

2.5.2 Methods for the Generation of gold-standard new Data

The goal is to acquire a large set of text pairs from social media and to label them with their correct entailment judgement. We consider here the current PHEME datasets "Ferguson", "Ottawa Shooting" and "Gurlitt".[10] Three methods are envisaged for the generation task: 1. Source Tweet : Story pairs; 2. Source Tweet : Replying Tweets pairs, and 3. Cross-media linking.

The first method is based on manual annotations by PHEME partners, by which a Source Tweet is always hand-labelled with a Story, examples of which are:

> **Source:** 2 of the 4 police departments rampaging through #Ferguson were trained by Israel in methods of domination and control http://t.co/ztZUZpzHJb
>
> **Story:** Two of the four police departments were trained in Israel

> **Source:** MORE: #Ferguson police chief identifies Darren Wilson as cop who shot Michael Brown http://t.co/Qojlgp8mlc
>
> **Story:** Ferguson police to release name of police officer who shot M. Brown today (August 15)

We make use of the fact that such source tweet : story pairs are in all cases true positives for entailment when submitted to EOP.

The second method for acquiring text pairs is based on the PHEME annotated source/reaction structure, and proceeds automatically. Each Reaction Tweet binds to a Source Tweet (and thus indirectly to the hand-labelled Story). We notice that such pairs do not always stand in a positive entailment relationship when applied to EOP. As an extension to the default classification scheme of TE, Entailment vs Non-Entailment, Non-Entailment pairs can further be sub-classified into Contradiction or Unknown judgments,

[10] Cf. the annotation scheme and the current data sets' description in D2.1 Qualitative Analysis of Rumors, Sources, and Diffusers across Media and Languages, D7.2.1 Annotated Corpus - Initial Version, and in D8.2 Annotated Corpus of Newsworthy Rumors.

in accordance with the RTE-3 task of multi-way entailment judgment assignment.[11] Hierarchical entailment labels have also been envisioned within EOP, but this feature was unavailable to us in the EOP Platform.

Examples of Contradiction as well as Unknown relationships are given in Figure 2.5.2. Pairs of texts are generated from the source tweet and each of its replying tweets. The text pair that has Reply tweet nr. 1 will be labelled as an Unknown relationship, the same happens to the pair that holds Reply tweet nr 2. The text pairs holding Reply tweets nr 3-6 will get the Contradiction label, because the proposition in the Source tweet is argued against in each of these replying tweets.

The third method of entailment data generation consists in establishing a cross-media linking between the tweet and the online article's headlines, based on the URL that is contained in the tweet. An example of text pairs automatically generated this way between a tweet and headlines is given in Figure 2.5.2:

2.5.3 Additional Features for performing the TE Task on the PHEME Data

We investigate if and how additional features can support the TE task when applied to PHEME data, taking for example into account the information on the usage of hashtags and URLs in the tweets. We checked for this again the above mentioned datasets "Ferguson", "Ottawa Shooting" and "Gurlitt". Figure 2.5.3 below displays a chart quantifying the use of Hashtags and URLs in the Source Tweets of three data sets.

The chart in Figure 2.5.3 shows:

- how many tweets we have in each dataset, split between rumourous and non-rumourous (marked as "rum-nonrum") tweets (back line, in blue)
- how many hashtags these tweets contain (middle line, purple)
- how many URL references these tweets contain (front line, orange)

Figure 2.5.3 shows similar statistics, but including also the number of included media material (mainly images) in the tweets, and taking into accounts also the replying tweets.

We have to analyse the relevance of those features for applying TE to the PHEME data sets, mainly by acquiring more data to generate more robust statistical insights.

[11] On differentiating the judgment 'Unknown' from 'False/Contradicts', see nlp.stanford.edu/RTE3-pilot

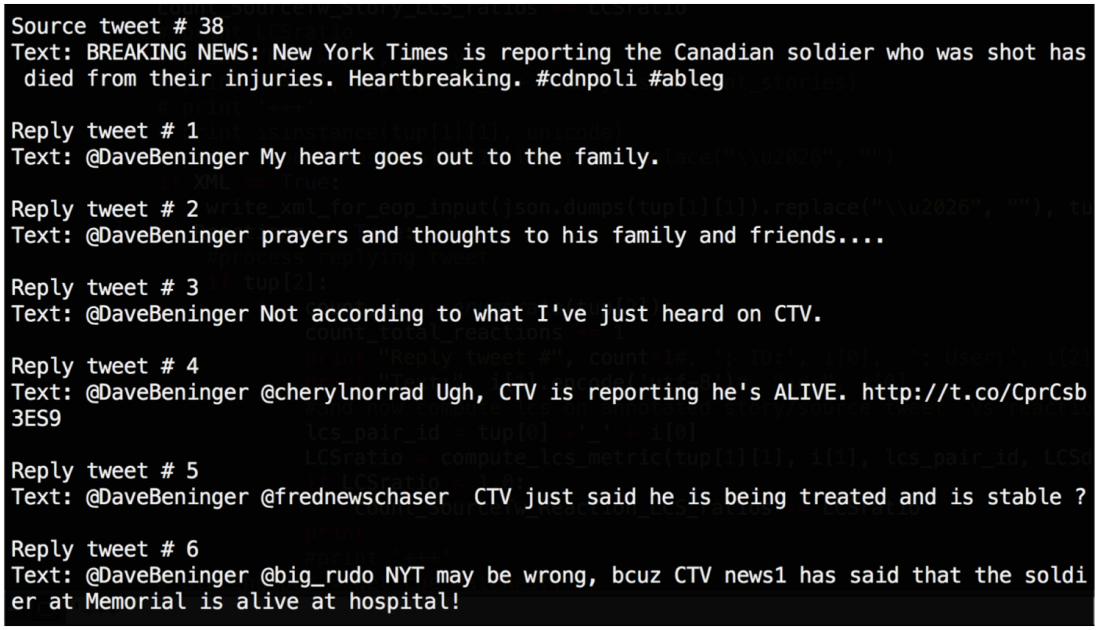

Figure 2.2: Examples of Contradiction/Unrelated relationships between a source and reaction tweets

```
Ferguson police to release name of police officer who
shot M. Brown today
```

```
Police name cop who shot Michael Brown
Police Have Named the Cop Who Shot Michael Brown as
Ferguson's Anger Turns to Peace
Ferguson Chief Names Darren Wilson as Cop Who Shot
Michael Brown - NBC News
Ferguson Police Officer Darren Wilson Revealed as
Missouri Fatal Shooting Cop - ABC News
```

Figure 2.3: News headlines extracted from URL contained in a tweet (first line)

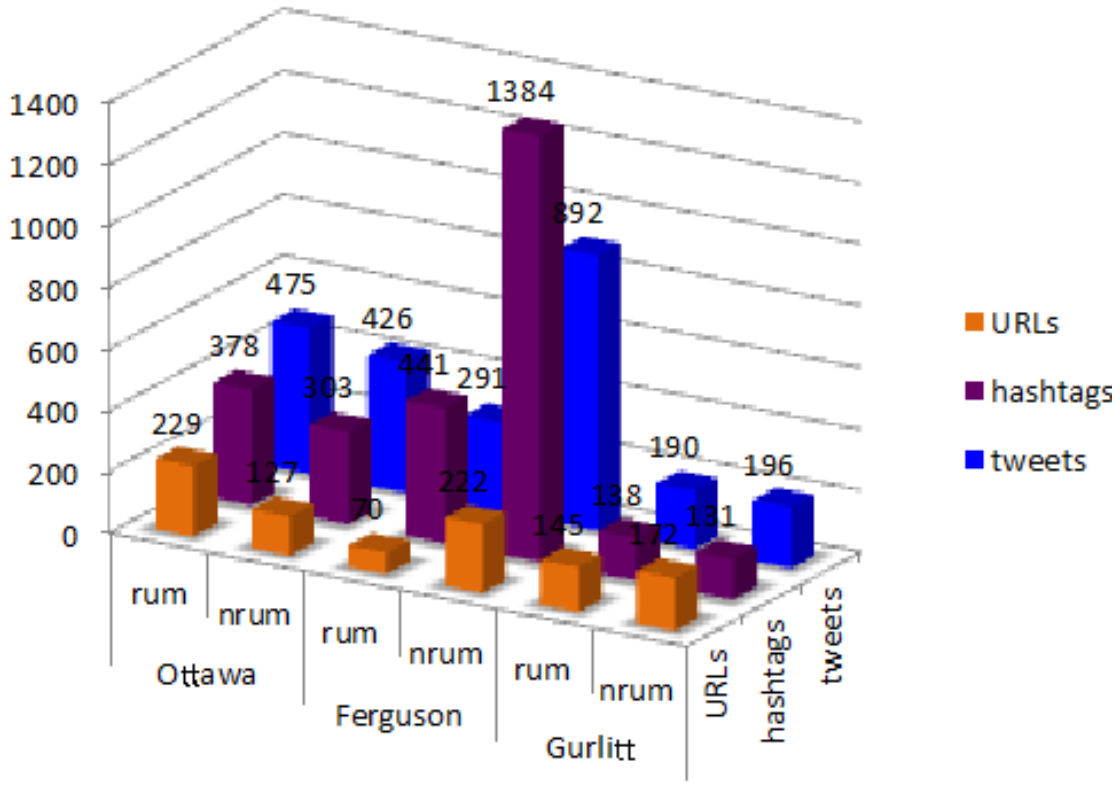

Figure 2.4: Chart showing the numbers of Hashtags and URLs used in the three considered source tweet data sets.

	Ottawa		Ferguson		Gurlitt	
	rum	nrum	rum	nrum	rum	nrum
URLs	810	1439	505	1821	159	181
Hashtags	1810	476	2386	7889	157	136
Media	503	449	305	2246	18	30
Source + Reply Tweets	6496	5893	6625	18547	243	243

Figure 2.5: Number of Hashtags and URLs across the data sets, including the replying tweets

```
Story # 1 Text: Two of the four police departments in Ferguson were trained by Israel

Source tweet # 1 ; ID: 500217516611108864.json ; User: RaniaKhalek
Text: 2 of the 4 police departments rampaging through #Ferguson were trained by Israel
 in methods of domination and control http://t.co/ztZUZpzHJb
shared: ["of", "the", "police", "departments", "were", "trained", "by", "israel"]
LCS ratio: 0.666666666667
```

Figure 2.6: A text pair labelled with the LCS ratio

2.5.4 Establishing a baseline Method for detecting Textual Entailment for the PHEME Data Sets

A potential baseline for our work on TE for the PHEME datasets is described in [15]. This would consist in computing the Longest Common Subsequence (LCS) between text pairs and labelling these with a corresponding LCS ratio, as shown in Figure 2.5.4.

We performed initial experiments with the LCS ratio on the PHEME data sets (considering for the time being only the relations between hand-labelled Stories and Source Tweets), and compared it with two EOP algorithms. The results, illustrated by the chart in Figure 2.5.4, may suggest that LCS ratio can be regarded as an initial baseline method for entailment detection: the scores of the EOP algorithm that computes edit distance correlate with the LCS ratio. Figure 2.5.4 displays the comparative results of LCS and two EOP algorithms applied to the story : source tweet pairs of our data sets. As performance measure we depict only recall, since precision in these experiments is always 100%.

The chart in Figure 2.5.4 shows several pieces of information about entailment in our 3 datasets.

- The numbers in brackets below each dataset name show how many annotated stories each dataset holds.
- The yellow first row indicates the following:
  - Gurlitt has a very low amount of Stories. This already can cause that Gurlitt has the smallest mean LCS ratio between a tweet text and its annotated story

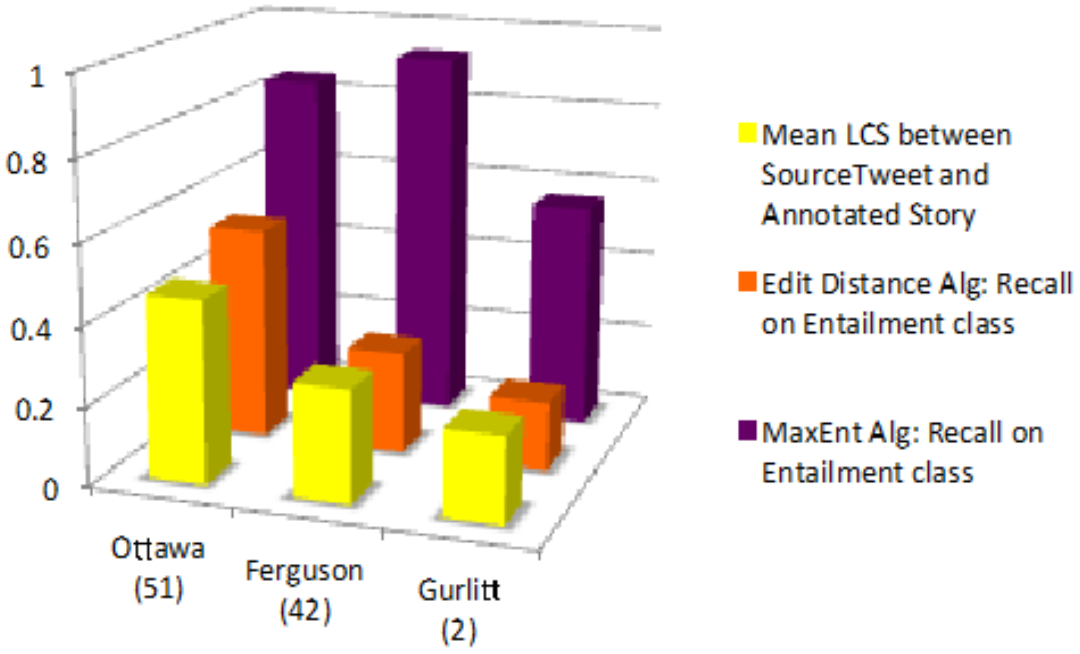

Figure 2.7: Comparing the recall of the LCS approach and two EOP algorithms, when applied to {story-source} pairs in the PHEME data sets.

pair - i.e., lexical variation can be large when a tweet is compared to the annotated story. But additionally, Gurlitt is a multilingual set (Source Tweets are in EN, DE, NL, FR), while Stories are in English, so simple token overlap between such pairs will obviously be relatively low.

  - Ferguson and Ottawa have a large number of Stories, so tweets associated to these stories are lexically less varied, making the LCS ratio higher.

- The purple back row shows scores by an entailment detection algorithm that is built by a different principle: it uses grammatical information (in the current setup: POS tags) and the Maximum Entropy Modelling framework. This supervised algorithm is pre-trained on RTE benchmark data, and obtains much higher scores on our datasets than the baseline technique or the edit distance algorithm.

2.5.5 Ongoing Work

Current work related to evaluation studies for Task 4.2 is summarized by the following bullet points:

- Quantify how EOP‘s confidence scores correlate with

  - Lexical-syntactic phenomena across tweet pairs
  - Twitter structure and chronology
  - Argumentation cues
  - Reference to real-world entities
- Retrain the entailment algorithms using these features
- Make entailment judgments
  - On unlabelled tweets
  - based on Entailment with a rumourous vs non-rumourous pre-classified tweet
  - On unseen data
- Test in big data setting.

We hypothesize that the extraction of core argumentation elements from a tweet, such as a statement, will benefit the contradiction detection task.

2.5.6 Conclusion

Although we are in the starting phase of our investigation on if and how Textual Entailment can support the tasks at hand in PHEME, we could already recognize steps to be taken to optimize the available EOP for supporting the detection of contradictions and controversies. We do think that algorithms implemented in EOP can be a very useful basis, but we need to define an approach that is abstracting much more over the lexical and phrasal alignments implemented in EOP. Lexical semantics is not enough for our objectives, we need to access real world knowledge and to bridge statements formulated in unstructured (and also noisy) text, with statements formulated in knowledge data sets and trusted sources.

2.6 Detection of mis- and dis-information

As detailed in Lukasik et al. (2015), we carry out tweet-level judgement classification automatically, in order to assist in (near) real-time rumour monitoring by journalists and authorities (Procter et al., 2013). In addition, we plan on using information about tweet-level judgements to assist forthcoming veracity estimation and early rumour detection (Zhao et al., 2015).

In this deliverable we evaluate tweet-level judgement classification on unseen rumours, based on a training set of other already annotated rumours.

Table 2.13: Counts of tweets with supporting, denying or questioning labels in each rumour collection from the England riots dataset.

text	position
Birmingham Children's hospital has been attacked. F***ing morons. #UKRiots	support
Girlfriend has just called her ward in Birmingham Children's Hospital & there's no sign of any trouble #Birminghamriots	deny
Birmingham children's hospital guarded by police? Really? Who would target a childrens hospital #disgusting #Birminghamriots	question

Table 2.12: Tweets on a rumour about hospital being attacked during 2011 England Riots.

Rumour	Supporting	Denying	Questioning
army bank	62	42	73
hospital	796	487	132
London Eye	177	295	160
McDonald's	177	0	13
Miss Selfridge's	3150	0	7
police beat girl	783	4	95
zoo	616	129	99

2.6.1 Datasets

We evaluate our work on two datasets, which we describe next.

The first consists of tweets from the England riots in 2011, which we used for initial evaluation in deliverable D4.3.1 (Lukasik et al., 2015). A summary of that dataset appears in Table 2.13 for reference. As can be seen from the dataset overview in Table 2.13, different rumours exhibit varying proportions of supporting, denying and questioning tweets, which was also observed in other studies of rumours (Marcelo et al., 2010; Qazvinian et al., 2011). These variations in majority classes across rumours underscores the modeling challenge in tweet-level classification of rumour attitudes.

Secondly, we make use of the 8 PHEME rumour datasets introduced in Zubiaga et al. (2015); Hoi (2015). As can be seen from the summary Table 2.14, some datasets contain relatively few tweets. In order to simulate light supervision by introducing from 10 to 50 tweets for training, we exclude small datasets from our experiments (as it would bias results unduly). We limit our attention to datasets with a ✓sign in the last column of Table 2.14.

The statistics show that there is a substantial number of comments among the tweets. Nevertheless, for consistency and comparability with our experimental results obtained on the London riots dataset, we continue with a 3-way classification into supporting, denying, questioning. We leave consideration of a 4 way classification for future work.

Table 2.14: Counts of tweets with supporting, denying or questioning labels in each event collection from the PHEME rumours.

Dataset	Rumours	Supporting	Denying	Questioning	Commenting	Large
Ottawa shooting	58	161	76	64	481	✓
Ferguson riots	46	192	83	94	685	✓
Prince in Toronto	12	19	7	11	59	×
Charlie Hebdo	74	236	56	51	710	✓
Ebola Essien	2	6	6	1	21	×
Germanwings crash	68	177	12	28	169	✓
Putin missing	9	17	7	5	33	×
Sydney siege	71	89	4	99	713	✓

2.6.2 Gaussian Processes for Classification

We apply Gaussian Processes and multi-task learning methods, following the problem formulation introduced in Lukasik et al. (2015).

2.6.3 Features

We conducted a series of preprocessing steps in order to address data sparsity. All words were lowercased; stopwords removed; all emoticons were replaced with words[12]; and stemming was performed. In addition, multiple occurrences of a character were replaced with a double occurrence (Agarwal et al., 2011), to correct for misspellings and lengthenings, e.g., *looool*. All punctuation was also removed, except for ., *!* and *?*, which we hypothesize to be important for expressing emotion. Lastly, usernames were removed as they tend to be rumour-specific, i.e., very few users comment on more than one rumour.

After preprocessing the text data, we use either the resulting bag of words (BOW) feature representation or replace all words with their Brown cluster ids (Brown), using 1000 clusters acquired from a large scale Twitter corpus (Owoputi et al., 2013). In all cases, simple re-tweets are removed from the training set to prevent bias (Llewellyn et al., 2014).

Apart from using the above described text features, we consider additional features. We use counts of punctuation marks *?*, *!* and . treated separately. These punctuation marks seem to convey important information about the sentence. Moreover, we employ emoticon counts (using the same emoticon dictionary as described above), as they convey important information about sentiment. We use a count of each hashtag as a feature. Lastly, we employ count of URLs in a tweet and binary indicator if a tweet is a complex re-tweet[13].

[12]We used the dictionary from: `http://bit.ly/1rX1Hdk` and extended it with: :o, :|, =/, :s, :S, :p.

[13]A complex re-tweet is a re-tweet which is not simple. A simple re-tweet is a re-retweets not modifying content in any other way than just adding information about user being re-tweeted at the front of the message (information added automatically by Twitter).

method	acc
Pooled Majority	0.68
GPPooled Brown	0.72
GPPooled BOW	0.69

Table 2.15: Accuracy taken across all rumours in the LOO setting.

2.6.4 Experiments and Discussion

Evaluation results on the London riots

Table 2.15 shows the mean accuracy in the LOO scenario (Leave One Rumour Out) following the GPPooled method, which pools all reference rumours together ignoring their task identities. ICM can not use correlations to target rumour in this case and so can not be used. The majority baseline simply assigns the most frequent class from the training set.

We can observe that methods perform on a level similar to majority vote, outperforming it only slightly. This indicates how difficult the LOO task is, when no annotated target rumour tweets are available.

Figure 2.8 shows accuracy for a range of methods as the number of tweets about the target rumour used for training increases. Most notably, performance increases from 70% to around 75%, after only 10 annotated tweets from the target rumour become available, as compared to the results on unseen rumours from Table 2.15. However, as the amount of target rumour increases, performance does not increase further, which suggests that even only 10 human-annotated tweets are enough to achieve significant performance benefits. Note also how the use of reference rumours is very important, as methods using only the target rumour obtain accuracy similar to the Majority vote classifier (GP Brown and GP BOW).

The top performing methods are GPCIM and GPPooled, where use of Brown clusters consistently improves results for both methods over BOW, irrespective of the number of tweets about the target rumour annotated for training. Moreover, GPICM is better than GPPooled both with Brown and BOW features and GPCIM with Brown is ultimately the best performing of all.

Notice that the methods GP Brown and GP BOW exhibit big improvement when even 10 tweets are used for training, comparing to when no target rumour data is available for training. However, when more annotation is available, the improvement is not so drastic anymore. This might be due to the fact, that initial tweets cover a broad spectrum of possible stances (support, deny, questioning) and further tweets mostly replicate these stances.

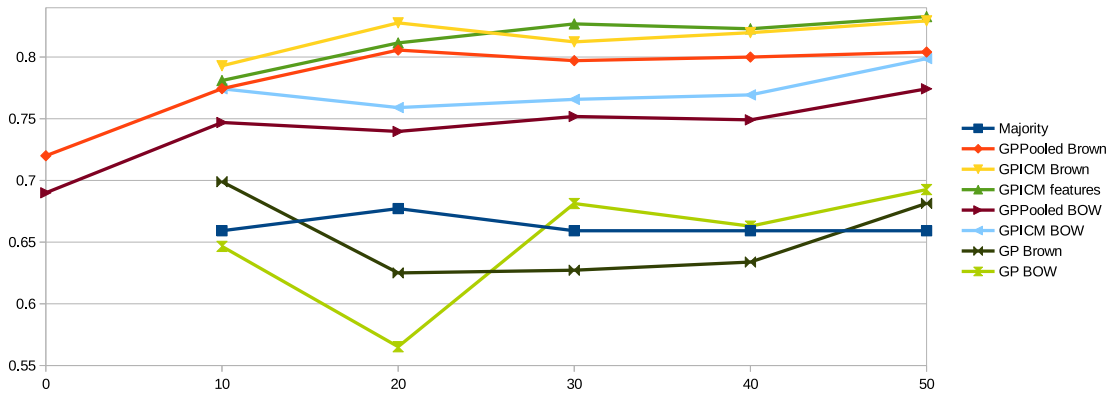

Figure 2.8: Accuracy measures for different methods versus the size of the target rumour used for training in the LPO setting. The test set is fixed to all but the first 50 tweets of the target rumour.

supporting	denying	questioning
? 10001101	fake 11111000001	? 10001101
! 10001100	not 001000	! 10001100
not 001000	? 10001101	hope 01000111110
fake 11111000001	! 10001100	true 111110010110
true 111110010110	bullshit 11110101011111	searching 01111000010

Table 2.16: Top 5 Brown clusters, each shown with a representative word. For further details please see the cluster definitions at the appendix.

train size	Majority	GPPooled BROWN	GPICM BROWN	GPICM features
10	0.66	0.78	80.42	78.92
20	0.72	0.82	83.82	82.39
30	0.66	0.81	82.68	83.58
40	0.66	0.81	83.22	83.39
50	0.66	0.81	84.07	84.13

Table 2.17: Performance on accuracy for class **support** of selected methods on London riots rumours.

train size	Majority	GPPooled BROWN	GPICM BROWN	GPICM features
10	0.81	0.87	88.42	86.71
20	0.81	0.89	91.42	89.19
30	0.81	0.88	89.33	88.73
40	0.81	0.89	89.86	90.29
50	0.81	0.89	91.43	90.84

Table 2.18: Performance on accuracy for class **deny** of selected methods on London riots rumours.

However, we can observe, that additional features start helping comparing to using Brown cluster features only when at least 30 tweets are observed from the target rumour. This indicates, that some more supervision is useful to levarage the more advanced features.

In order to analyse the importance of Brown clusters, Automatic Relevance Determination (ARD) is used (Rasmussen and Williams, 2005) for the best performing GPICM Brown in the LPO scenario. Only the case where the first 10 tweets are used for training is considered, since it already performs very well. Using ARD, we learn a separate length-scale for each feature, thus establishing their importance. The weights learnt for different clusters are averaged over the 7 rumours and the top 5 Brown clusters for each label are shown in Table 2.16. We can see that clusters around the words *fake* and *bullshit* turn out to be important for the denying class, and *true* for both supporting and questioning classes. This reinforces our hypothesis that common linguistic cues can be found across multiple rumours. Note how punctuation proves important as well, since clusters *?* and *!* are also very prominent.

Evaluation on the PHEME datasets

In this subsection we aim at validating the results, that multi-task learning improves performance. We also intend to see whether features can improve over BROWN cluster features. Due to high number of rumours, we merge rumours from different datasets into separate tasks. We leave exploration of setting when each rumour is a separate task for

train size	Majority	GPPooled BROWN	GPICM BROWN	GPICM features
10	0.84	0.90	89.86	90.56
20	0.82	0.91	90.29	90.71
30	0.84	0.90	90.59	93.06
40	0.84	0.90	90.50	90.90
50	0.84	0.90	90.60	91.58

Table 2.19: Performance on accuracy for class **question** of selected methods on London riots rumours.

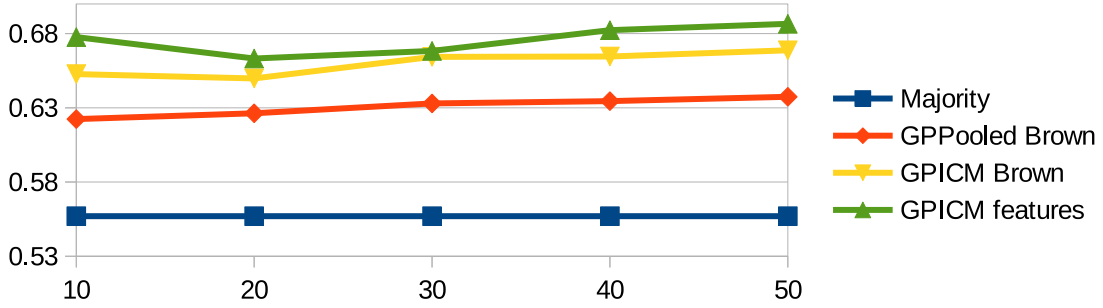

Figure 2.9: Accuracy measures for different methods versus the size of the target rumour used for training in the LPO setting on the selected PHEME datasets. The test set is fixed to all but the first 50 tweets of the target rumour.

future work (in such settings a problem of high number of hyperparameters arises, which yields a computationally hard problem).

We show the results in Table 2.20 and Figure 2.9. The results are worse than in the previous deliverable, presumably due to the fact that data is more heterogenous than in case of London riots dataset. We make similar observations, that multi-task learning makes for the best method. Moreover, additional features consistently improve results.

Time speed evaluation

Here we report experiments on the PHEME datasets, aim of which is to estimate time of the tweet classification process. We only estimate time for the actual prediction time, therefore assuming training has been already done.

The specifications of the used workstation are as follows: 4 cores of Intel(R) Core(TM) i7-3687U CPU @ 2.10GHz, 7.7 GB RAM.

We test the pipeline, where as input serve raw Twitter jsons and as output another json is acquired with added field denoting the class field (see Table 2.24).

Method	train tweets	Ottawa	Ferguson	Charlie H	German w.	Sydney	mean
Majority	10	52.59	50.16	66.89	53.45	55.40	55.70
Majority	20	52.59	50.16	66.89	53.45	55.40	55.70
Majority	30	52.59	50.16	66.89	53.45	55.40	55.70
Majority	40	52.59	50.16	66.89	53.45	55.40	55.70
Majority	50	52.59	50.16	66.89	53.45	55.40	55.70
GPPooled Brown	10	62.55	55.49	70.65	63.79	58.73	62.24
GPPooled Brown	20	64.14	55.17	71.33	63.79	58.73	62.63
GPPooled Brown	30	64.54	55.80	71.67	63.79	60.66	63.29
GPPooled Brown	40	64.94	56.11	72.01	63.79	60.39	63.45
GPPooled Brown	50	64.14	56.43	72.01	63.79	62.33	63.74
GPICM BROWN	10	65.34	59.25	74.4	67.24	60.11	65.27
GPICM BROWN	20	61.35	57.37	75.09	70.69	60.39	64.98
GPICM BROWN	30	64.94	57.99	74.74	72.41	62.05	66.43
GPICM BROWN	40	65.74	58.31	75.77	70.69	61.77	66.46
GPICM BROWN	50	66.14	58.62	75.77	70.69	63.16	66.88
GPICM features	10	62.95	59.56	76.79	74.14	65.37	67.76
GPICM features	20	63.75	55.49	76.45	74.14	61.77	66.32
GPICM features	30	62.95	57.68	76.79	74.14	62.6	66.83
GPICM features	40	68.92	59.87	78.5	70.69	63.16	68.23
GPICM features	50	69.72	58.31	77.13	74.14	63.99	68.66

Table 2.20: Accuracy for each of the PHEME datasets in the LPO setting according to different methods.

train size	Majority	GPPooled BROWN	GPICM BROWN	GPICM features
10	55.70	69.16	71.06	74.01
20	55.70	69.31	71.10	72.91
30	55.70	69.91	71.29	72.55
40	55.70	69.90	71.65	74.02
50	55.70	70.43	71.97	74.28

Table 2.21: Performance on accuracy for class **support** of selected methods on PHEME rumours.

train size	Majority	GPPooled BROWN	GPICM BROWN	GPICM features
10	78.86	74.74	76.84	78.20
20	78.86	74.87	76.47	75.98
30	78.86	75.21	79.01	77.85
40	78.86	75.43	78.79	79.24
50	78.86	75.73	79.44	79.40

Table 2.22: Performance on accuracy for class **deny** of selected methods on PHEME rumours.

train size	Majority	GPPooled BROWN	GPICM BROWN	GPICM features
10	76.84	80.58	82.64	83.32
20	76.84	81.09	82.38	83.75
30	76.84	81.47	82.56	83.27
40	76.84	81.57	82.46	83.20
50	76.84	81.32	82.34	83.63

Table 2.23: Performance on accuracy for class **question** of selected methods on PHEME rumours.

number of tweets	time
1	0m 4.019s
10	0m 3.797s
100	0m 4.416s
1000	0m 9.474s
10000	0m 50.626s
100000	8m 28.644s

Table 2.24: Speed tests on different dataset sizes. We duplicated an example tweet from the PHEME datasets appropriate number of times to measure the processing speed.

Chapter 3

Integration Evaluation

3.1 Approach

PHEME's initial integration approach was explained in deliverable D6.1.1. Many of the PHEME components already covered in this document need to be pipelined in order to accomplish certain tasks, such as language detection, text pre-processing, processing in several languages. etc. These components are heterogeneous, developed by different partners often using different programming languages (Java and Python mainly) and sometimes even hosted remotely. These facts poses requirements to the integration approach followed in the project.

From the integration perspective, the main goal is to ensure that the whole system and all its components are able to fulfill the project requirements of processing social network data in a streaming fashion for real-time rumour classification and detection, cross-language and cross-media and providing timely results. Some of the components will perform other tasks by themselves, such as Machine Learning training. In these cases integration with other components is not required. Where needed, the integration approach should also allow easy and loosely coupled integration and communication for batch processing.

The PHEME integrated framework is meant to scale. In order to ensure the scalability of the solution the project is on the one hand improving the performance and scalability of the different individual components as reported in the previous sections. On the other hand, from the integration perspective, the project is following a global integration strategy to ensure the performance and scalability of the overall system. This global integration strategy presents project-wide approaches orthogonal to the individual scaling plans and common for most technical components. The focus is on integration aspects to provide an infrastructure to integrate components while enabling big data scaling techniques, such as scaling up, scaling out, parallelization, etc. This global integration strategy takes into account limits of individual components to align them into a common plan.

The integration in PHEME is following an incremental approach. The first year witnessed several integration tests (reported in D6.1.1.) aimed at defining the approach to follow. Based on those experiments, the project decided to go for a message oriented architecture in order to ease the integration process, especially, but not only, for the real-time processing pipelines.

This real-time processing integration follows the concept of pipelines. Pipelines allow the addition of multiple components in a process. From the integration and scalability perspectives, pipelines should be able to increase the throughput and decrease the latency as much as possible. In order to do that, enabling parallelisation means processing of several inputs coming from components in a pipeline with other identical components that work in parallel. In an optimal scenario, it is simply adding more processing units for the same components that work slower compared to other components in a pipeline. More processing units can be provided to components by scaling horizontally or vertically. Scaling horizontally is achieved by adding more nodes (computers) to a system, while vertical scaling can be achieved by running the whole pipeline on a faster machine.

For programming language independence, messaging systems support multiple platforms and programming languages and are a clear solution for the integration problem (Nannoni, 2015). There are many popular messaging systems, such as ZeroMQ,[1] RabbitMQ,[2] Apache Flume[3] and Apache Kafka,[4] among others. However, as explained in D6.1.1, the Capture module provides the infrastructure for messaging and several other integration points that PHEME may take advantage of. Capture is built on top of a big data-enable infrastructure that provides batch and streaming processing engines. In particular Capture uses Apache Kafka pub-sub mechanism for message exchange. This ability has been exploited during the second year of the project as baseline for data and component integration for the veracity framework.

Figure 3.1 shows a detailed view of the Capture IT layer that has been used for the integration experiments done in the second year of PHEME:

The previous Figure shows that Capture is built on top of several Open Source frameworks from the Apache Foundation that provides processing and messaging capabilities, namely Apache Hadoop,[5] Apache Flink,[6] Apache Storm[7] for processing, and Apache Kafka for messaging. Apache Kafka is a distributed publish-subscribe messaging system from the Apache Big Data environment. It is a messaging system that is mainly used for various data pipeline and messaging uses. Kafka is designed to allow a single cluster to serve as the central data backbone for a large organization. It can be elastically and transparently expanded without downtime. Data streams are partitioned and spread over a

[1] http://zeromq.org/
[2] https://www.rabbitmq.com/
[3] https://flume.apache.org/
[4] http://Kafka.apache.org/
[5] https://hadoop.apache.org/
[6] https://flink.apache.org/
[7] https://storm.apache.org/

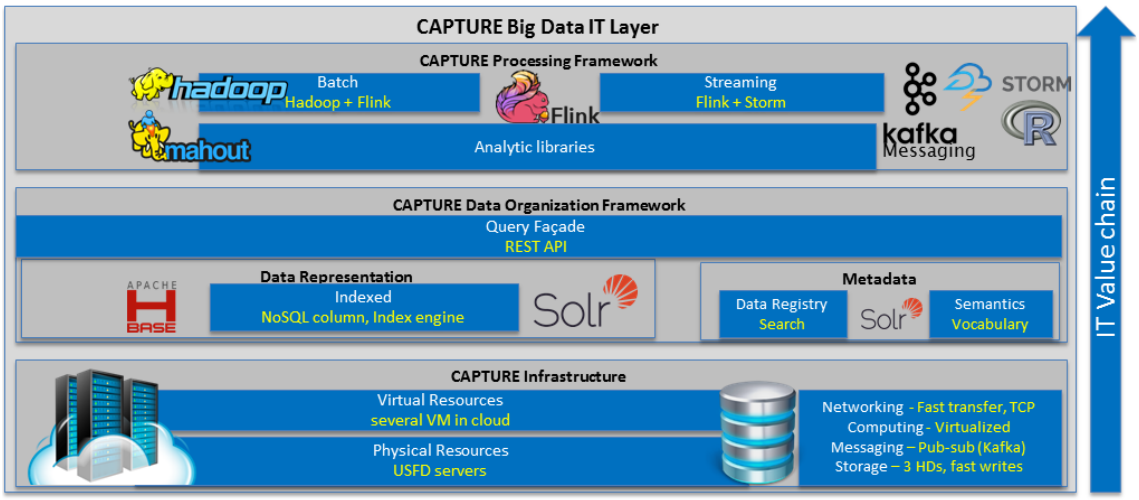

Figure 3.1: Capture integration points

cluster of machines to allow data streams larger than the capability of any single machine and to allow clusters of coordinated consumers.

Apache Kafka differs from traditional messaging systems in:

- It is designed as a distributed system which is very easy to scale out.
- It offers high throughput for both publishing and subscribing.
- It supports multi-subscribers and automatically balances the consumers during failure.
- It persist messages on disk and thus can be used for batched consumption such as ETL, in addition to real time applications

Kafka is a general purpose publish-subscribe model messaging system, which offers strong durability, scalability and fault-tolerance support. For the pipelining approach of integration of PHEME components, Apache Kafka is used to pass streams (e.g. Twitter streams) from one component to the next. PHEME components and algorithms publish and subscribe to data streams to decouple the different processes, thus creating pipeline of loosely-coupled components.

It is worth mentioning that the approach followed in PHEME consist of the incremental addition of annotations to the original Kafka message. The first component in the pipeline (Capture) publishes a stream of tweets in a Kafka queue (called Kafka topic) and the next component adds new annotations to those tweets (i.e. language of the tweet) and publishes them in a new queue. This incremental approach enables a very easy integration, as components just need to publish a new stream with the addition of the new annotation made. New components simply need to subscribe to those queues, in order to start consuming the streams.

3.2 Descriptions of systems integrated

The integration experiments done in year two consists of the integration of some components of the real-time PHEME rumour classification pipeline. This pipeline is depicted below.

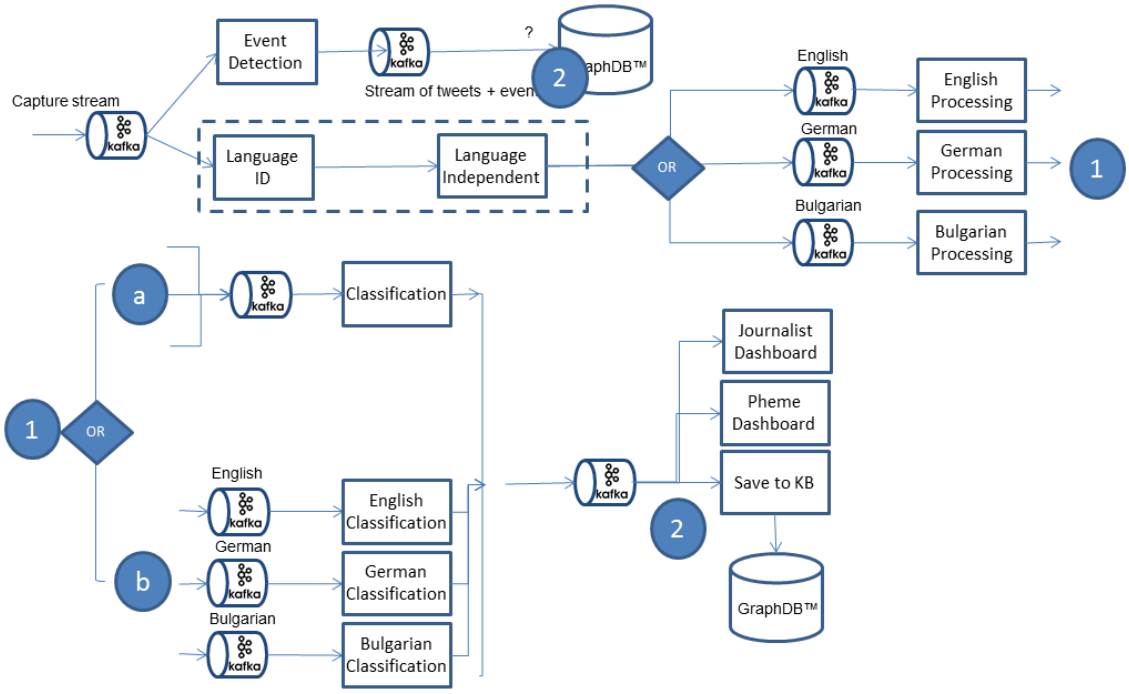

Figure 3.2: PHEME pipeline

It is worth mentioning that the pipeline depicted is a high-level working representation of the components to integrate, but it does not display all the components of PHEME. Some of the elements (i.e. as the English processing) will be broken into different components that could be tightly or loosely integrated themselves.[8]

The pipeline above is therefore a sample real-time process used to create a first integrated prototype, including language dependent and independent components. It starts with a stream provided by **Capture**. The process forks on the one hand to the **Event Detection** component, depicted in parallel to the main flow. It detects candidate stories in the stream of tweets. The Event Detection ends by publishing a Kafka topic with the

[8]Some atomic components may need a tight integration with others for simplicity, optimization or performance reasons (for instance the Language Independent component is in reality a set of components). From the integration perspective they are seen as a single block that reads from Kafka and eventually produces output to Kafka.

stories detected. A new component (a Kafka consumer) will be developed to subscribe to that topic and persist data of events in the Knowledge Base (GraphDB).

The language ID is the first element to add language annotation to the stream of tweets. This component produces different Kafka streams for each of the 3 languages treated so far in PHEME (English, German and Bulgarian). The language-independent components are depicted after the language ID in the diagram. Those components have not been integrated so far, but the idea is that those components will perform typical text processing tasks independent of the language, publishing finally the 3 Kafka topics corresponding to the language pipelines.

The different language dependent pipelines (for English, Bulgarian and German) are treated as separated pipelines in the diagram. This involves the creation of 3 similar (the same format) Kafka topics (one per each language). This eases the process as, for instance, English tweets will be listened by the English Processing components using a Kafka consumer that is subscribed only to that particular topic. An initial version of the Rumor classification is meant to be added to the process, adding annotations to the tweet about the rumor.

The final components in the pipeline produce a final Kafka topic with all the previous annotations. This final Kafka topic can be then used for several purposes, such as persistence in the Knowledge Repository or visualization of the real-time data.

Messages are passed in Kafka as well-formed JSON

- As Twitter, our main source, and other minor sources all follow the convention of placing the document text in a top-level key called “text”, we will follow this
- The social media message identified should be a top-level “id_str” object, following Twitter convention
- When producing, components re-publish the entire original JSON object, and add additional keys
- If there is a chance of colliding with a social media source’s top level key, or a key name is non-descriptive, prefix it with “pheme_”
- We anticipate the following top-level keys:
  - event extraction: “event_cluster”, long unsigned int
  - language: “lang_id”, tuple of (string, float), where string is a two-letter lowercase country code and float the confidence in [0..1]
  - tokens: “tokens”, list of [tuples of (int, int)], giving token start/end offsets
  - named entities, spatio-temporal entities: “pheme_entities”, a list of [tuples of (int, int, string)], corresponding to string start offset, end offset, and entity type; e.g. “person”, “timex”

  - support/deny/query: ”pheme_sdq”, tuple of (string, float) where string is support—query—deny—comment and the float the confidence in [0..1]
  - spatial grounding: ”pheme_location”, a dict of {string: *}, where the source key is “latlong”, “dbpedia”, “nuts” or “geonames”, and the value is either a string reference or a tuple (float, float) for latitude and longitude. All keys are optional.

- New top-level keys will be defined upon agreement, when a need arises.

3.3 Scalability evaluation

In pipelines the problem is the slowest component (bottlenecks). The goal of the coming months will be on finding bottlenecks and applying techniques to overcome them (both at component and at integration levels). Techniques such as parallelisation of the pipeline will be tried, based on the average latency of each component and the processing power which is available. Based on the results, slow components will have more instances that work in parallel. Other techniques, such as splitting the work in parallel for some of the components (for instance creating several consumers from the same Kafka topic), or distributing the pipeline in several instances on more machines (horizontal scalability), will be studied to maximize throughput, while minimizing latency. The advantage is that Kafka allows this type of distribution and has been tested in the current evaluation by having nodes in Sheffield and in Ontotext’s premises. Current tests shows that Kafka consumers from Ontotext are able to subscribe and retrieve messages from the Kafka producers located in Sheffield.

Any number of producers may fill any number of queues, from which any number of consumers can simultaneously or not consume the data. It makes horizontal scaling very easy: the more HTTP requests arrive, the more public facing back-end servers can be set up, sending data to the broker. The same can be done with the consumers, to increase the consuming rate of messages. The broker can also be clustered, to perform both load balancing and replication. However, the instances with which the producers and consumers would have to exchange data (e.g. databases, file storage systems) may get overloaded too. As such, they should be able to scale as well, to really give an architecture implementing a messaging middleware the power to scale up entirely.

Chapter 4

Conclusion and Future Work

The current version of the experiments done to integrate components using Kafka and the underlying Capture infrastructure shows good results. Several components have been integrated so far and the performance and potential scalability prospects are promising.

The goal of the integration is to enable the delivery of the PHEME Integrated Veracity Framework. The steps towards integration done so far show that the selected infrastructure and technical guidelines work well, but further experiments and moreover the identification of potential problems and bottlenecks need to be addressed.

For next integration evaluation cycles, the performance of the integration framework as a whole will be measured and evaluated further. Bottlenecks have to be found and overcome from an overall scaling strategy perspective. Deliverable D6.2.2. will report on the results of the evaluation of the final PHEME Veracity Framework.

In order to achieve the concrete goals of the integration, the project will follow both component-level and global scaling strategies. Frequent integration meetings and hackathons are planned for the rest of the project, in order to ensure efficient and optimal development towards an Integrated Veracity Framework that scales.

Bibliography

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38.

Baldwin, T., Han, B., de Marneffe, M. M. C., Kim, Y.-B., Ritter, A., and Xu, W. (2015). Findings of the 2015 Workshop on Noisy User-generated Text. In *Proceedings of the Workshop on Noisy User-generated Text (WNUT 2015)*. Association for Computational Linguistics.

Banerjee, A., Dhillon, I. S., Ghosh, J., and Sra, S. (2005). Clustering on the unit hypersphere using von mises-fisher distributions. *J. Mach. Learn. Res.*, 6:1345–1382.

Bartz, T., Beißwenger, M., and Storrer, A. (2014). Optimierung des Stuttgart-Tübingen-Tagset für die linguistische Annotation von Korpora zur internetbasierten Kommunikation: Phänomene, Herausforderungen, Erweiterungsvorschläge. *Zeitschrift für germanistische Linguistik*, 28(1):157–198.

Beißwenger, M., Bartz, T., Storrer, A., and Westpfahl, S. (2015a). Tagset und Richtlinie für das PoS-Tagging von Sprachdaten aus Genres internetbasierter Kommunikation.

Beißwenger, M., Ehrhardt, E., Horbach, A., Lüngen, H., Steffen, D., and Storrer, A. (2015b). Adding Value to CMC Corpora: CLARINification and Part-of Speech Annotation of the Dortmund Chat Corpus. In *Proceedings of the 2nd Workshop on Natural Language Processing for Computer-Mediated Communication / Social Media*.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media, Inc.

Bosma, W. and Callison-Burch, C. (2007). Paraphrase substitution for recognizing textual entailment. In *Evaluation of Multilingual and Multi-modal Information Retrieval*, pages 502–509. Springer.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Buntine, W. and Mishra, S. (2014). Experiments with non-parametric topic models. In *20th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*.

Declerck, T., Osenova, P., and Derczynski, L. (2014). D2.2 Linguistic Pre-processing Tools and Ontological Models of Rumours and Phemes. Technical report, PHEME project deliverable.

Derczynski, L. and Bontcheva, K. (2014). Passive-aggressive sequence labeling with discriminative post-editing for recognising person entities in tweets. *EACL 2014*, page 69.

Derczynski, L. and Bontcheva, K. (2015). D2.3 Spatio-Temporal Algorithms. Technical report, PHEME project deliverable.

Derczynski, L., Chester, S., and Bøgh, K. S. (2015a). Tune Your Brown Clustering, Please. In *Proceedings of the conference on Recent Advances in Natural Lang Processing (RANLP)*.

Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., Petrak, J., and Bontcheva, K. (2015b). Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49.

Derczynski, L., Ritter, A., Clark, S., and Bontcheva, K. (2013). Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*, pages 198–206.

Fromreide, H., Hovy, D., and Søgaard, A. (2014). Crowdsourcing and annotating NER for Twitter #drift. *European language resources distribution agency*.

Hoi, G. W. S. (2015). D8.2 Annotated Corpus of Newsworthy Rumours. Technical report, PHEME project deliverable.

Horbach, A., Steffen, D., Thater, S., and Pinkal, M. (2014). Improving the performance of standard part-of-speech taggers for computer-mediated communication. In *Proceedings of KONVENS*. Universitätsbibliothek Hildesheim.

Kergl, D., Roedler, R., and Seeber, S. (2014). On the endogenesis of Twitter's Spritzer and Gardenhose sample streams. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 357–364. IEEE.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8.

Llewellyn, C., Grover, C., Oberlander, J., and Klein, E. (2014). Re-using an argument corpus to aid in the curation of social media collections. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC'14, pages 462–468.

Lukasik, M., Bontcheva, K., Cohn, T., Tolosi, L., and Georgiev, G. (2015). D4.3.1 Algorithms for Detecting Misinformation and Disinformation: Initial Prototype. Technical report, University of Sheffield.

Marcelo, M., Barbara, P., and Carlos, C. (2010). Twitter under crisis: Can we trust what we RT? In *1st Workshop on Social Media Analytics (SOMA)*.

Masud, M. M., Gao, J., Khan, L., Han, J., and Thuraisingham, B. (2011). Classification and novel class detection in concept-drifting data streams under time constraints. *Knowledge and Data Engineering, IEEE Transactions on*, 23(6):859–874.

Meilă, M. (2005). Comparing clusterings: An axiomatic view. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 577–584, New York, NY, USA. ACM.

Murphy, K. P. (2012). *Machine learning: A Probabilistic Perspective*. The MIT Press.

Nannoni, N. (2015). *Message-oriented Middleware for Scalable Data Analytics Architectures*. PhD thesis, Kth Royal Institute of Technology, School of Information and Communication Technology.

Owoputi, O., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL*, pages 380–390.

Preotiuc-Pietro, D., Samangooei, S., Lampos, V., Cohn, T., Gibbins, N., and Niranjan, M. (2013). Clustering models for discovery of regional and demographic variation. Technical report, Public Deliverable for Trendminer Project.

Procter, R., Crump, J., Karstedt, S., Voss, A., and Cantijoch, M. (2013). Reading the riots: What were the police doing on twitter? *Policing and society*, 23(4):413–436.

Pustejovsky, J., Lee, K., Bunt, H., and Romary, L. (2010). ISO-TimeML: An international standard for semantic annotation. In *LREC*.

Pustejovsky, J., Moszkowicz, J. L., and Verhagen, M. (2011). ISO-Space: The annotation of spatial information in language. In *Proceedings of the Sixth Joint ISO-ACL SIGSEM Workshop on Interoperable Semantic Annotation*, pages 1–9.

Qazvinian, V., Rosengren, E., Radev, D. R., and Mei, Q. (2011). Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1589–1599.

Rajaraman, A. and Ullman, J. D. (2011). *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named entity recognition in tweets: An experimental study. In *Proc. of Empirical Methods for Natural Language Processing (EMNLP)*.

Schiller, A., Teufel, S., and Thielen, C. (1999). Guidelines für das Tagging deutscher Textcorpora mit STTS.

Schneider, N. (2015). What i've learned about annotating informal text (and why you shouldn't take my word for it). In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 152–157. ACL.

Srijith, P., Hepple, M., and Bontcheva, K. (2015). D3.3.1 Longitudinal models of users, networks, and trust: Initial Prototype. Technical report, PHEME project deliverable.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581.

Verhagen, M., Sauri, R., Caselli, T., and Pustejovsky, J. (2010). Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 57–62. Association for Computational Linguistics.

Vinh, N. X., Epps, J., and Bailey, J. (2009). Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1073–1080, New York, NY, USA. ACM.

Xu, W., Callison-Burch, C., and Dolan, W. B. (2015). Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*.

Zhao, Z., Resnick, P., and Mei, Q. (2015). Early detection of rumors in social media from enquiry posts. In *International World Wide Web Conference Committee (IW3C2)*.

Zinsmeister, H., Heid, U., and Beck, K. B. (2013). Das STTS-Tagset für Wortartentagging - Stand und Perspektiven. *Journal for Language Technology and Computational Linguistics*, 28(1).

Zubiaga, A., Tolmie, P., Liakata, M., and Procter, R. (2015). D2.4 Qualitative Analysis of Rumours, Sources, and Diffusers across Media and Languages. Technical report, PHEME project deliverable.