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: Algorithms for Implicit Information Diffusion Networks (Revised)

Deliverable number: D3.2

Deliverable date: 31 December 2015

Partners responsible: MODUL Univeristy Vienna (MOD)

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*	11 May 2016

D3.2 Algorithms for Implicit Information Diffusion Networks Across Media

Svitlana Vakulenko, Arno Scharl (MODUL University Vienna)

Abstract.
FP7-ICT Collaborative Project ICT-2013-611233 PHEME
Deliverable D3.2 (WP3)

The deliverable describes the results of Task 3.2, proposing a novel approach for the analysis of information diffusion traces across different types of online media sources. Based on articles of independent news publishers as well as social media postings, the presented approach encompasses several algorithms to model and track information contagions propagated through the implicit diffusion network. We provide an experimental assessment and draw conclusions on how the presented approach will be used within the PHEME rumour detection framework.

Keywords. information diffusion, cross-media analysis, diffusion cascade, information contagion, user-generated content, Twitter.

Project	PHEME No. 611233
Delivery Date	(Revision) April 30, 2016; (Original) January 15, 2016
Contractual Date	December 30, 2015
Nature	Report
Reviewed By	Kalina Bontcheva
Web links	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

Nowadays, the news are communicated online not only through traditional publishers such as CNN, BBC and the New York Times, but also via social media platforms such as Twitter and Facebook. The dynamic interplay between such different types of communication channels, however, is yet not well studied - which poses significant challenges when investigating news propagation processes and and developing rumour detection algorithms. This deliverable describes methods for tracking information diffusion contagions across various media sources including major online news publishers as well as regular Twitter users.

In particular, this deliverable focuses on methods for modeling the information contagions extracted from the natural language text of news articles and social media posts. Specifically, we propose novel algorithms to assess the similarity between news articles and retrieve related social media posts. In order to address this problem, we represent the content of the news articles with a bag-of-relations model, which helps us to detect the information contagions.

The performance of the algorithms is assessed on a sample data set constructed using the media coverage contained within the initial prototype of the PHEME dashboard reported in D5.2.1. The experimental results show the advantages of the chosen approach when applied to real-world data. Revealing the latent structure of the implicit diffusion network (information sources, content similarity and network dynamics) will support the tracking of stories and rumours across different communication channels.

Contents

Chapter 1

Introduction

Information diffusion investigates the propagation of information through an actor network along a temporal axis. The actor network may be modeled as a graph describing all persons, objects, and semiotics that take part in the message passing, both implicitly or explicitly. Information diffusion is an established research field usually applied to explicit networks such as online social networks (OSNs), but less studied in real-world scenarios where the sources of information are often not provided explicitly.

1.1 Related Work

One way to link news articles is text reuse (plagiarism) detection to trace the origin of an information piece. This approach has been recently applied to study information exchange networks based on historical newspaper texts (Colavizza et al., 2014), the evolution of memes (Leskovec et al., 2009; Suen et al., 2013), and the unveiling of publisher biases via citation usage patterns (Niculae et al., 2015). In contrast to these projects, the diffusion work within PHEME does do not track stable phrases, but news items extracted from the natural language text and modeled as relations.

Yang and Leskovec (2010) model information diffusion in implicit networks and predict the future volume based on observing which nodes got infected in the past. They manage to derive and analyse influence functions for different topics and different types of media sources (blogs, Twitter users, traditional media web sites). Instead of trying to detect which node has infected which other nodes, they model the total number of infected nodes (volume of information diffusion) over time determined by the influence function of the nodes infected in the past. Moreover, they study the phenomena of novelty and imitation and their influence on the volume of information diffusion. They formulate the problem as an instance of non-negative least squares (NNLS) problem and use it to predict the future total volume by observing which nodes were infected in the past. Their approach differs from PHEME since it does not attempt to model the implicit network

explicitly by surfacing possible links between information sources.

A different approach with a focus on visualizing diffusion processes is *Cascade*[1], a tool that analyzes information diffusion cascades as they unfold through sharing activities via social media. In contrast to the PHEME project, it disregards data acquisition and preprocessing of the news articles - steps that are assumed to be used by the content providers. As a commercial solution, the details of the underlying algorithms and implementation are not provided.

1.2 Relevance to PHEME

D3.2 introduces a novel approach to capture information flows propagating via traditional news media and their influence on social media. Since we attempt to analyze information diffusion across different communication channels (traditional news media vs. social media), PHEME does not only face the challenge of aligning various information sources (T3.1) and identifying stories across these sources (T3.3), but also modelling these stories (T3.4) and integrating them into a single holistic network model (T3.2).

The algorithms of D3.2 complement the cross-media and cross-language linking algorithm (CM-CL) of WP3 by focusing on information diffusion between the news articles. Both approaches help to model of the diffusion process. While D3.2 explores the news propagation flows from the authoritative sources such as traditional news media publishers to tweets, D3.1 covers the opposite direction - from tweets to related news articles.

In Year 3 of the project, once live data streams from the *Capture* data collection tools (T6.1) become available in the PHEME dashboard (T5.4) through the API integration of T6.3, these challenges will be jointly addressed - by applying the diffusion propagation module to the PHEME evaluation datasets from the WP7 and WP8 use case scenarios. The results will be reported in D6.2.2 Evaluation Report - Final Results.

1.3 Deliverable Structure

This deliverable describes methods for tracking the propagation of information diffusion contagions across the implicit network of online media sources. We discuss our approach to model information diffusion networks within PHEME in the following Chapter 2. Following an experimental assessment of our approach on a current news media sample dataset in Chapter 3, the deliverable concludes with implications and an outlook on future research in Chapter 4.

[1] `www.nytlabs.com/projects/cascade.html`

Chapter 2

Information Diffusion Model

This section presents a novel approach to reconstruct real-world information diffusion processes, taking place within the implicit network of the traditional (mainstream) news media sources as well as its further propagation in social media (micro-blogs).

Since we attempt to trace information propagation within an implicit cross-media network, where the underlying diffusion processes are hidden or unknown, we are facing the major challenge of inferring and surfacing these hidden relations to be able to construct a single network model. Our approach encompasses the following stages:

1. **Modeling information contagions** as relations extracted from natural language text that represent the information content of a post (e.g. a news article) and allow for the disambiguation of paraphrases, i.e. unifying semantically equivalent expressions by resolving synonyms.

2. **Modeling diffusion cascades** over the implicit network, which includes the computation of relation-based similarity to infer implicit links between news articles.

3. **Cross-Media Information Diffusion**, which involves tracing information propagation across different types of news media sources.

2.1 Modeling Information Contagions

We propose a bag-of-relations document representation model to capture the essential information contained in a textual document, e.g. a news article. The main idea behind the model is to represent each document as a set of relations, which are strings built from the sets of words similar to n-grams, but constructed following the grammatical dependency relations instead of the sequence order of words in the text. Each such relation we consider as a potential information contagion carrying certain semantic information within the diffusion network.

Relation extraction is a sub-task of information extraction aimed at identifying relations between entities expressed in natural language texts. It has already been successfully applied to analyze a collection of news articles (Del Corro and Gemulla, 2013). For each article we attempt to extract a set of relations following the procedure described in Algorithm 1. First, we employ a dependency parser to obtain the parse trees for each of the sentences and extract the relations by traversing these trees. The relations are modeled as triples of the form:

s (subject) – **p** (predicate) – **o** (object)

We start off with the task of finding all the predicates in the sentence (see Algorithm 1 line 4), which will play the role of triggers to finding the corresponding relations. We initialize the set of predicates with all the verbs used in various tenses including participles and gerund forms. Furthermore, we pre-process the predicates to bring them to the standardized form: '{*synsets (or lemmas)*} + {*flags*}', e.g. 'did not say' is transformed to 'state D N', by detecting for each verb of the compound predicate the corresponding WordNet synset (lemma, otherwise), tense, voice, negation and auxiliary verbs.

We define a set of words to be excluded from the predicate phrase, which could help to improve the results of our analysis. For example, there may be certain trivial relations, which are common for all news articles and which we would like to eliminate, e.g. the ones triggered by the predicates: 'print', 'post', 'update'. The helping words that do not carry the semantic information of the predicate, but used solely for grammatical purposes, e.g. 'have', 'to', 'will', 'be', are also good candidates to be included in the "noise filter".

Flags are used to preserve the grammatical information expressed by the helping words, which are our conventional codes indicating the tense of the main verb, passive voice, negation, etc. We use 'D' to indicate the past tense, 'F' – future tense, 'N' – negation, 'A' - auxiliary verb ('would'). Thus, the standard form, which consists only of lemmas without any additional flags, indicates that the predicate is used in present tense and is not negated. Since there are multiple ways to represent negation, past tense, etc. this approach helps us to disambiguate and unify semantically-equivalent relations.

Next, for each of the predicates we are looking for the adjacent branches which represent clauses used to express its subject or objects. Currently we are using a simple heuristic assigning the node to the subject-element, if it precedes the predicate in the sentence, and to the object otherwise (if it follows the predicate), which will undoubtedly fail to succeed in many cases. However, we favor this solution over a set of more complex manually constructed or learned rules, which may also not be easily generalizable to other languages. Nevertheless, further improvements beyond this heuristic may be considered for future work.

We allow for a relation to have multiple partially overlapping instances, which enables us to represent the same relation in several different ways. We achieve this by constructing a separate relation for each object-element related to the predicate including an empty object if the relation has a non-empty subject: $\forall r \in R \mid r.subject \neq \emptyset$: $r.objects.add(\emptyset)$

(see Algorithm 1 line 8).

This simple heuristic allows us to create several fine-grained relations with different levels of detail. For example, a sentence "The plane landed in Panama on Tuesday" will be decomposed into: 'plane - land D', 'plane - land D - in Panama', 'plane - land D - on Tuesday'. This way we are able to group articles, which may report on the same event but provide complementing or contradicting details.

Algorithm 1 Relation extractor

Require: $article$

```
1:  relations ← {}
2:  for sentence in article do
3:      graph ← generate_dependency_graph(sentence)
4:      predicates ← extract_predicates(graph)
5:      for predicate in predicates do
            // predicate may be related to a subject
6:          relation.subject ← extract_subject(graph, predicate)
            // predicate may have multiple relations to different objects
7:          objects ← extract_objects(graph, predicate)
            // if relation has a non-empty subject...
8:          if relation.subject≠ ∅ then
                // ...we generate one instance of it with an empy object, i.e. "s - p"
9:              objects.add(∅)
10:         end if
11:         for object in objects do
12:             relation.object ← object
13:             relations.add(relation)
14:         end for
15:     end for
16: end for
17: return relations
```

2.2 Modeling Diffusion Cascades

We propose to model the diffusion process as a network graph, with two types of edges:

- edges linking posts explicitly referenced through url of the source (edge direction is from the source to the post containing the url);

- edges linking similar posts that share information contagions (edge direction from the older to the newer posts).

Since news articles often do not cite their information sources explicitly, we focus on developing an approach generating edges between the news articles that propagate the same information contagion. Once we retrieved the set of articles, which contain the information contagion of interest (modeled as a relation), we construct the diffusion graph with edges generated using the pairwise similarity values computed on the relation bags that were extracted for each of the articles beforehand (see Chapter 2.1 for details on our relation extraction method).

The similarity between the two news articles can be calculated by considering the intersection of their relation bags, or by first hashing the relation bags and then computing the similarity for these relation hashes. We employ both methods to compute similarity for any two relation bags complementary to each other to evaluate which of them performs better in practice.

While the first method is straight forward returning an integer for the number of shared relations, it has the obvious limitation of considering only the exact matches for the relations. The second method, which involves additional relation hashing routine, allows for continuous relation similarity values and can indicate more and less similar relations even if they do not have an exact overlap. The second method is potentially more powerful, but it requires rigorous fine-tuning and evaluation for generating the relation hashes and their comparison.

Currently we use Nilsimsa hashing and Hamming distance, but we may also consider alternative algorithms in future work. Nilsimsa (Capitani et al., 2004) is one of the most popular algorithms for locality-sensitive hashing and is traditionally employed for spam detection in emails. Hamming distance (Hamming, 1950) measures the proportion of positions in the hash at which the corresponding symbols are different.

Our approach can be briefly summarized as follows: (1) we use previously extracted relations to find similar articles; (2) we apply relation hashing to be able to compare the relations that are not exact matches; (3) we identify the original information source for each of the articles via pairwise similarity comparison using a distance measure applicable to hash values, e.g. Hamming distance (see Algorithm 2).

We assume that all the articles that share the same information contagion are related to each other, i.e there is a path for every pair of articles within the diffusion graph. We included this assumption into our model by enforcing the connectivity requirement over our diffusion graph: for each node, except the root node, we generate an incoming edge that links the post to its source (see Algorithm 2 line 11). Here, we also use the single source assumption: for all nodes, except the root node, there is exactly one incoming edge linking the post to its source, by selecting the closest neighbour as the information source. We could, as well, store the whole pairwise comparison matrix instead and/or select a threshold for similarity value. We introduced this assumption in order to simplify

Algorithm 2 Similarity comparison for the information source detection

Require: $articles$ (stack of articles sorted in ascending order by their publication time)

// Initialize a new stack for processed articles
$published \leftarrow \{\}$
// remove the first published article from the stack
article $\leftarrow articles$.pop()
// and assign it to be the root of the information diffusion cascade
article.source $\leftarrow$ "ROOT"
$published$.push(article)

// process the rest of the articles in the stack
while $articles \neq \{\}$ **do**
  article $\leftarrow articles$.pop()
  // compute pairwise similarity to all the articles that were already published
  $distances \leftarrow \{\}$
  for previous in $published$ **do**
    $distances \leftarrow hamming\_distance$(article.relation_hash, previous.relation_hash)
  end for
  // assign the most similar article to be the source for the current article
  article.source $\leftarrow$ min($distances$)
  $published$.push(article)
end while
// output all articles with the corresponding sources
return $published$

the model and to avoid making assumptions about the similarity threshold value, i.e. how similar the articles should be to be linked in the diffusion model (we considered this problem later in the experimental evaluation, see Section 3.3).

2.3 Cross-Media Information Diffusion

Traditional and social media are not two isolated worlds. On the contrary, they constantly share and exchange information propagating the information contagions across the networks. Links to news articles are distributed via social media and news articles cite individual tweets or even tell whole stories via a thread of related tweets.

Twitter is likely to be the most popular social media channel for news propagation since the whole concept of this service is built around the goal to facilitate massive crowd-driven information (news) sharing. From this requirement also stems the enforced limitation on the character count for the post on Twitter, i.e. the message has to be concise and

to the point, carrying only the essential information (contagion).

Due to the short length of the posts users often adopt special writing style including slang, abbreviations and hashtags. Twitter is a frequent example of such a language subculture (lingo).[1]

The assumption we made, when developing our relation-based approach, was that the input text follows the standard grammar rules, which is usually reasonable, when dealing with traditional narratives, such as news articles, but is often no longer the case for the social media posts. Therefore, this approach may not be efficient for processing tweets or other social media posts. Also, computing content similarity between a news article and a tweet may be difficult due to differences in length and language style. Therefore, we propose to exploit the direct (explicit) links between news articles and tweets, when possible, such as the URL of a news article available from the metadata of a tweet or a tweet ID extracted from the tweet snippet embedded within a news article.

Finally, it is not feasible to retrieve all the tweets due to the enormous velocity of the Twitter data stream and the restrictions imposed by the Twitter API. Hence, we propose a set of heuristics, which help us to build specific queries to the Twitter API retrieving the tweets relevant for our diffusion network. Our approach involves double propagation of initial information contagions extracted from the set of news articles to obtain a seed set of tweets, which we use to extract additional information contagions and retrieve more of the relevant articles and tweets, thereby enriching our diffusion model (see Algorithm 3 for details).

[1] http://mashable.com/2013/07/19/twitter-lingo-guide

Algorithm 3 Cross-media information retrieval

Require: $articles$

// Initial sets of tweets and hashtags may or may not be empty
$tweets \leftarrow \{\}$ $hashtags \leftarrow \{\}$
$new\_articles \leftarrow articles$
$new\_hashtags \leftarrow \{\}$
while $new\_articles \neq \{\}$ **or** $new\_hashtags \neq \{\}$ **do**
  // I. Retrieve new tweets
  $new\_tweets \leftarrow \{\}$
  for article in $articles$ **do**
    new_tweets.add($request\_TwitterAPI$(q=article.url))
  end for
  for hashtag in $new\_hashtags$ **do**
    new_tweets.add($request\_TwitterAPI$(q=article.hashtag))
  end for
  $new\_tweets \leftarrow new\_tweets$.diff($tweets$)
  $tweets$.add(new_tweets)

  // II. Compute frequent hashtags
  $new\_hashtags \leftarrow frequent\_hashtags$($tweets$).diff($hashtags$)
  $hashtags$.add($new\_hashtags$)

  // III. Retrieve new articles
  $new\_articles \leftarrow \{\}$
  for tweet in $tweets$ **do**
    $new\_articles$.add($retrive\_articles$(tweet.id))
  end for
  for hashtag in $hashtags$ **do**
    $new\_articles$.add($retrive\_articles$(hashtag))
  end for
  $new\_articles \leftarrow new\_articles$.diff($articles$)
  $articles$.add($new\_articles$)
end while

Chapter 3

Experiment

In this chapter, we report on a small-scale experiment, which we performed to evaluate our approach as a proof-of-concept using the recent dump of the media coverage and a sample use case. Thereby, we aim to: (1) demonstrate how our approach can be applied in practice with an illustrative example, (2) qualitatively assess the performance of our approach in a real-world scenario, and (3) discover the limitations and the directions for future work.

3.1 Settings

In order to compile a sample dataset to use in our experiment we took a recent snapshot of the media coverage contained within the PHEME dashboard reported in D5.2.1 (Scharl et al., 2016), to be formally assessed as part of the evaluation report of D6.2.2. We retrieved $\sim$ 71,000 news articles published within one week (from 27th of November until 3rd of December 2015).

After relation extraction procedure the most frequent relations were manually inspected. For the purpose of demonstration of the proposed approach we selected the following relation as an information contagion to be tracked through online media:

s: president barack obama – **p**: state D – **o**:,

which captures all expressions with a predicate that belongs to the WordNet synset 'state' and is used in the past tense (flag 'D'),such as: 'president barack obama stated', 'president barack obama said', 'president barack obama has said', 'president barack obama told' etc., thereby indicating statements that were made by President Obama. This use case has a practical application, e.g. the resource `www.dailymail.co.uk/obama` groups articles covering news about Barack Obama.

Then, for each of the 12 unique news articles containing this relation from our data set, we retrieved the tweets using Twitter Search API that contain a URL identifier pointing to this article, which resulted in 150 tweets (127 and 23 for two of the articles).

We constructed the diffusion graph following our approach with edges generated using the direct links for the tweets and the pairwise similarity between the contents for the articles. We used networkx[1] and matplotlib[2] Python libraries to visualize the resulting diffusion graph (see Figure 3.1) and manually inspected the results. We analyzed the content of the original articles and assessed whether the model constructed by our algorithm makes sense in terms of content coherence and similarity between the news articles.

3.2 Results

Figure 3.1 shows the graph of the propagation of the information contagion (in this case 'president barack obama state D' relation) through the diffusion network. The graph exhibits cascading behaviour, which reflects the dynamics of the information diffusion within the implicit network of the news sources. The diffusion process starts from the single root node and propagates further to cover the whole network of the infected sources.

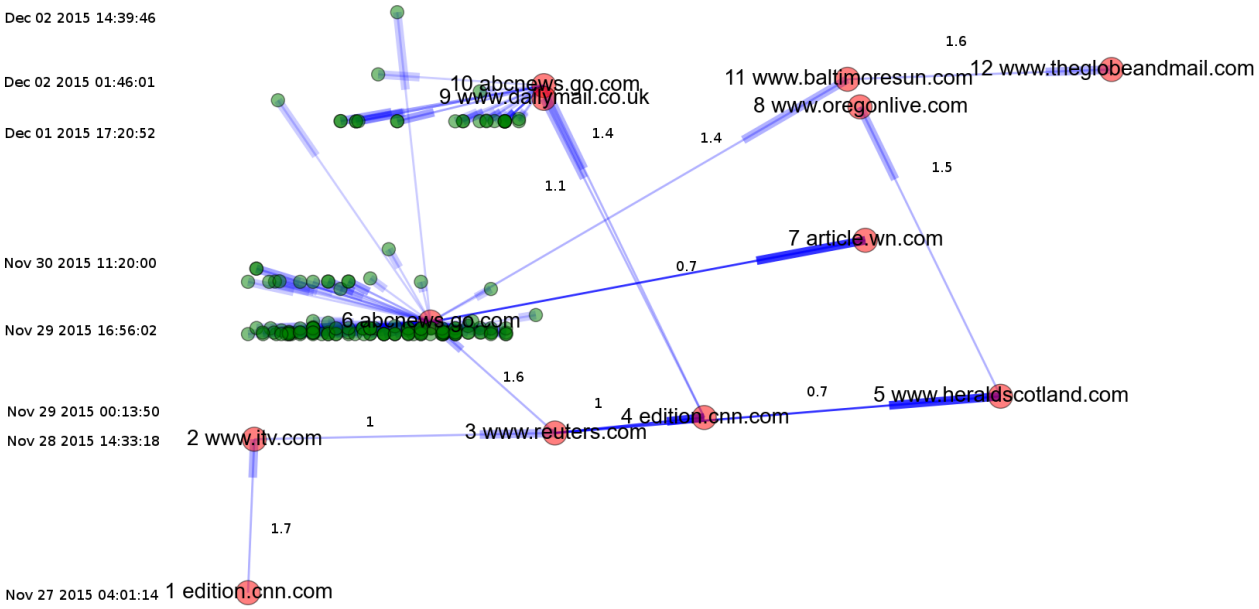

Figure 3.1: Information diffusion graph - initial results

The nodes of the graph represent the individual posts (red – news articles, green – tweets) published at discrete time intervals by the specified news source (an independent news publisher or a Twitter user). The news sources get infected in the sequence order

[1] networkx.github.io
[2] matplotlib.org

aligned with the Y time axis and explicitly indicated on their label. The same source may appear more than once within the same network if it published several articles in the given time period, which contain the same information contagion, e.g ABC News or CNN on Figure 3.1. More details about the individual articles (including their titles, URLs and the sets of extracted relations) and the tweets that constitute the graph are provided in the diffusion timeline, which lists all the posts in the order of publication.

The edges of the graph represent direct links in case of the tweets (using the quoted article's URL or the re-tweet meta-relation) or the content similarity measure in case of the articles. The content similarity is indicated as weights over the corresponding edges. It is computed as the hamming distance for the hash values of the extracted relations. Therefore, the values closer to 0 indicate the more similar articles and the larger values - the less similar articles.

There are two types of edges between the articles: (1) *light edges* indicate that the adjacent articles share a single information contagion, which is the relation common for all the articles in the graph that we initially used to select these articles, i.e. 'president barack obama state D' in our use case; (2) *solid edges* indicate that the articles have more than one information contagion in common. For example, in our use case we observe additional information contagions appearing and propagating from Reuters news source to the next two articles that follow in time:

1. 3: Reuters $\rightarrow$ 4: CNN (Relation-hash similarity: 1)

   The posts have an additional shared relation: 'make easy accessibility weapon war street people', which was extracted from the same citation of President Obama's statement. Both articles use the same citation just frame it differently, when quoting it in direct speech:

   2015-11-28 15:57:48
   Reuters: "We have *to do something about the easy accessibility of weapons of war on our streets to people* who have no business wielding them. Period," Obama said in a statement.

   2015-11-28 19:26:33
   CNN: President Barack Obama told the American people on Saturday that "we have *to do something about the easy accessibility of weapons of war on our streets to people* who have no business wielding them."

2. 3: Reuters $\rightarrow$ 5: Herald (Relation-hash similarity: 0.7)

   Later on, Herald re-uses the same citation as in the previous two sources:

   2015-11-29 00:13:50
   Herald: "We have *to do something about the easy accessibility of weapons of war on our streets to people* who have no business wielding them. Period," Obama said in a statement.

Additionally, our algorithm detects another relation shared between Herald and Reuters publications: “united state necessitate”, which points us to the paraphrased sentences that contain snippets of indirect speech:

2015-11-28 15:57:48
Reuters: President Barack Obama said on Saturday *the United States needs* to “do something” to make it harder for criminals to get guns after a shooting in Colorado killed three people and injured nine.

2015-11-29 00:13:50
Herald: President Barack Obama said yesterday *the United States needs* to “do something” to make it harder for criminals to get guns after a shooting in Colorado Springs on Friday left three dead and nine injured.

3.3 Discussion

Our initial assumption was that all the articles that contain the same information contagion are related, i.e there is a path between every pair of articles within the diffusion graph. We included this assumption into our model by enforcing the connectivity requirement over our diffusion graph: for each node, except the root node, we generate exactly one incoming edge that links the post to its source (see Algorithm 2).

In practice, the degree of similarity already gives us an idea that some groups of articles can be more similar between each other and at the same time more distinct from all other documents. However, at this point we do not know which threshold for the similarity value is reasonable to choose. Therefore, we manually identified two prevalent topics within this set of articles (see Figure 3.2):

- statements made by President Obama related to the **Colorado clinic shooting** (orange nodes);
- statements made by President Obama at the **Climate Change Conference in Paris** (purple nodes)

Using Figure 3.2 we make an observation that in order to separate our diffusion graph into several non-overlapping subgraphs that correspond to the topics listed above we need to make the following cut (an instance of the minimum cut problem): $C = [(1,2),(3,6),(5,8)]$. All of the edges in this cut have comparatively high weights: $\forall e_i \in C : w(e_i) \geq 1.5$, i.e. they indicate less similar articles.

To achieve the sub-topic separation, which we manually detected before, we can set the threshold value for the similarity measure accordingly and delete all the edges with the weights above the threshold (highlighted yellow on Figure 3.2): $\forall e_i \in E | w(e_i) \geq 1.5 : del(e_i)$.

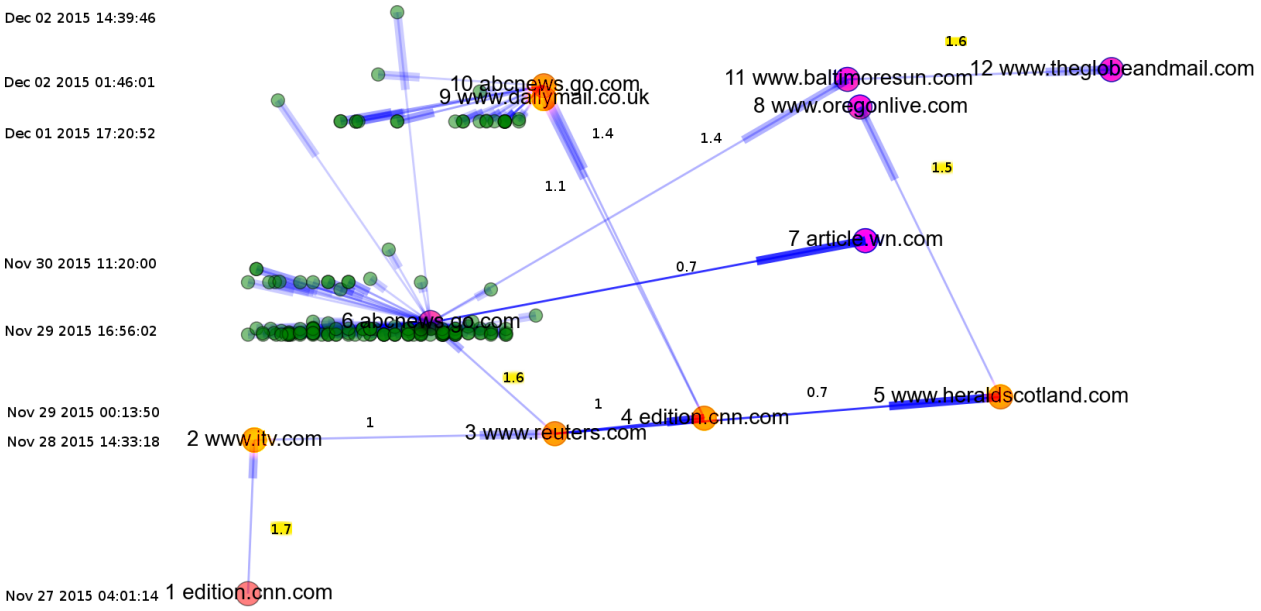

Figure 3.2: Information diffusion graph - subgraph analysis

This sub-topic separation via threshold values introduces the following two errors into our diffusion model: Nodes 8 (Oregon Live)[3] and 12 (The Globe and Mail)[4] become completely disconnected from the rest of the nodes and are not able to join the nodes assigned to the same topic (the purple sub-graph).

[3] www.oregonlive.com
[4] www.theglobeandmail.com

Chapter 4

Conclusion and Future Work

Modeling information diffusion networks helps us to understand how information spreads within an online community, and to identify key actors within the network who play a significant role in the propagation process. Once latent relations between publications are uncovered and structured in terms of content similarity and time of publication, it is possible to construct an implicit network model where the contagion is propagated. The ability to model pieces of information (contagions) travelling through the network, as well as their subsequent transformations, is an important element of rumour detection research.

We assessed the performance of the presented contagion modeling approach on a crawled news media data set, in order to track the propagation of contagions within an implicit network. The results of this initial experiment indicate that this relation-based similarity approach is promising and merits further research. More specifically, the assessment results suggest the following directions for future work:

- Improve the relation extraction method to identify those relations that are most useful to compare articles;
- revise the relation hashing method to preserve the semantics of underlying relations;
- consider other available distance measures applicable to hash values with the goal of maximising precision; and
- further divide the diffusion graph into smaller subgraphs based on document similarity metrics.

Overall, the experimental evaluation shows that the resulting information diffusion model reflects a meaningful subset of the implicit network where the information contagion of interest is propagated. In regards to the scope of the actual diffusion network (as compared to the part of the network that we capture), we are unable to make specific claims as the recall has not been determined.

One reason for this is that the underlying diffusion process is hidden or unknown, which represents a general obstacle when analysing implicit networks. Nevertheless, it is possible to perform a quantitative evaluation of the similarity measure by using a gold-standard data set such as the WikiTimes (Tran and Alrifai, 2014) classification. Another limitation of the evaluation is the focus on one particular use case, generating a single diffusion cascade. Therefore, the conclusions can not be generalized to all other use cases and shall be dealt with caution. More detailed evaluation results will be reported in the evaluation report of D6.2.2 using live data streams from the *Capture* data collection tools (WP6).

Bibliography

Capitani, D. D., Damiani, E., De, S., Vimercati, C., Paraboschi, S., and Samarati, P. (2004). An open digest-based technique for spam detection. In *International Workshop on Security in Parallel and Distributed Systems*, pages 15–17.

Colavizza, G., Infelise, M., and Kaplan, F. (2014). Mapping the Early Modern News Flow: An Enquiry by Robust Text Reuse Detection. In *Social Informatics - SocInfo 2014 International Workshops, Barcelona, Spain, November 11, 2014, Revised Selected Papers*, pages 244–253.

Del Corro, L. and Gemulla, R. (2013). Clausie: Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 355–366, Geneva, Switzerland.

Hamming, R. (1950). Error detecting and error correcting codes. *Bell System Technical Journal, The*, 29(2):147–160.

Leskovec, J., Backstrom, L., and Kleinberg, J. (2009). Meme-tracking and the Dynamics of the News Cycle. In *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 497–506, New York, NY, USA. ACM.

Niculae, V., Suen, C., Zhang, J., Danescu-Niculescu-Mizil, C., and Leskovec, J. (2015). QUOTUS: the structure of political media coverage as revealed by quoting patterns. In *24th International Conference on World Wide Web, Florence, Italy, May 18-22, 2015*, pages 798–808.

Scharl, A., Weichselbraun, A., Goebel, M., Rafelsberger, W., and Kamolov, R. (2016). Scalable knowledge extraction and visualization for web intelligence. In *49th Hawaii International Conference on System Sciences (HICSS-2016)*, pages 3749–3757. IEEE Press.

Suen, C., Huang, S., Eksombatchai, C., Sosic, R., and Leskovec, J. (2013). NIFTY: A System for Large Scale Information Flow Tracking and Clustering. In *22nd International Conference on World Wide Web*, WWW '13, pages 1237–1248, Geneva, Switzerland.

Tran, G. B. and Alrifai, M. (2014). Indexing and analyzing Wikipedias Current Events Portal, the Daily News Summaries by the Crowd. *Proceedings of World Wide Web 2014, Web Science Track.*

Yang, J. and Leskovec, J. (2010). Modeling information diffusion in implicit networks. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on Data Mining*, pages 599–608. IEEE.