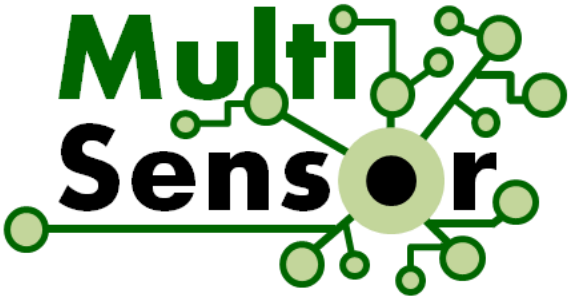

MULTISENSOR

Mining and Understanding of multilinguaL contenT for Intelligent Sentiment Enriched coNtext and Social Oriented inteRpretation

FP7-610411

D3.4

Modules for context representation, sentiment extraction and information propagation in social media

Dissemination level:	Public
Contractual date of delivery:	Month 32, 30/06/2016
Actual date of delivery:	Month 33, 11/07/2016
Workpackage:	WP3 User and Context-centric Content Analysis
Task:	T3.4 Information propagation and social interaction analysis
Type:	Prototype
Approval Status:	Final Draft
Version:	2.0
Number of pages:	39
Filename:	D3.4_ModulesContextSentimentSocialMedia_2016-07-11_v2.0.pdf

Abstract

This document presents the advanced modules for information propagation in social media (T3.4).To this end, the deliverable describes state-of-the-art algorithms fitted to the

requirements and solutions of the MULTISENSOR pilot use cases: journalism, commercial media monitoring and SME internationalization.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	15/05/2016	Initial draft	I. Arapakis (Eurecat)
0.2	12/06/2016	CERTH contribution	I. Gialampoukidis, D. Liparas, S. Vrochidis (CERTH)
0.3	22/06/2016	Second draft	F. Peleja (Eurecat)
0.4	28/06/2016	Third draft	D. García-Soriano (Eurecat)
0.6	04/07/2016	Internal review	I. Gialampoukidis (CERTH)
1.0, 2.0	11/07/2016	Final version	I. Arapakis (Eurecat)

Author list

Organization	Name	Contact Information
Eurecat	Ioannis Arapakis	ioannis.arapakis@eurecat.org
Eurecat	Filipa Peleja	filipa.peleja@eurecat.org
Eurecat	David García-Soriano	david.garcia@eurecat.org
CERTH	Ilias Gialampoukidis	heliasgj@iti.gr
CERTH	Dimitris Liparas	dliparas@iti.gr
CERTH	Stefanos Vrochidis	stefanos@iti.gr

Executive Summary

This document presents the advanced techniques and the modules developed on the information propagation and opinion mining. It describes algorithms for analysing the network of interactions among social media users, the detection of influential users, and the evolution of dynamic communities of information within the network.

Abbreviations and Acronyms

CEP	Context Extraction Pipeline
CNR	Central News Repository
GLM	Generalised Linear Model (GLM)
LDA	Latent Dirichlet Allocation
RMSE	Root Mean Squared Error
RRSE	Root Relative Squared Error
SMAP	Social Media Analysis pipeline
UC	Use Case

Table of Contents

1 INTRODUCTION

This document presents the advanced techniques and the modules developed on the information propagation and opinion mining. It describes algorithms for analysing the network of interactions among social media users, the detection of influential users, and the evolution of dynamic communities of information within the network.

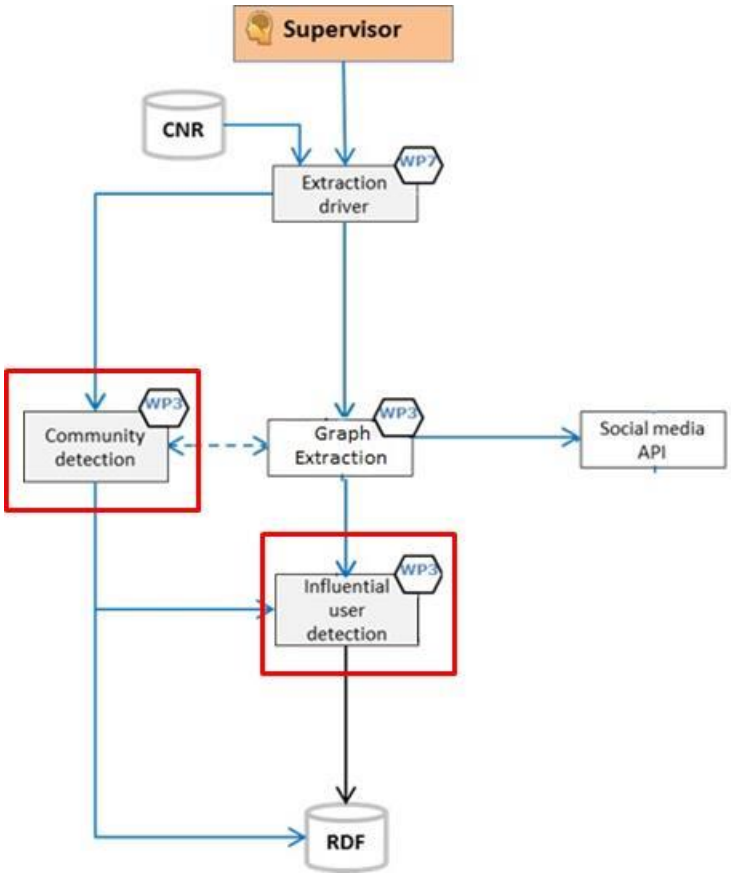

Figure 1: Social Media Analysis Pipeline architecture with the influential user detection and community detection modules highlighted

The "Information propagation and social interaction analysis" module is part of the Social Media Analysis Pipeline (SMAP), which consists of a set of processes related to analysis of social network data stored into the MULTISENSOR repositories (Figure 1). The SMAP pipeline performs social influence and interaction analysis on previously crawled Twitter data that includes hashtags, together with information about the profiles of the posters and the associations among them.

Given this data, the Graph Extraction service builds a topic-dependent network of contributors based on the mentions in the set of monitored tweets. It also computes retweet probabilities between users in this network, and finally the Influential User Detection module outputs two ranked lists of users, one by decreasing order of Pagerank and another one by decreasing order of influence in the Independent Cascade model.

The community detection partitions a network's nodes into subgroups of densely connected nodes, where there are only a few of links from one subgroup (community) to another, and there are many links within each community. In MULTISENSOR, the community detection

task involves the detection and visualisation of Twitter communities, given a list of desired keywords/hashtags. Contrary to the traditional modularity maximisation approaches for finding community structure, MULTISENSOR adopts the information-theoretic code length minimisation, known as the Infomap method.

The rest of this report is organized as follows. In Section 2, we cover the information propagation and social interaction analysis module and we provide some concluding remarks in Section 3.

2 INFORMATION PROPAGATION AND SOCIAL INTERACTION ANALYSIS

2.1 Motivation and user requirements

The spread of information in a society does not happen all at once, but it is rather a complex process that, by exploiting channels of communication among individuals, starts from few adopters and propagates gradually, reaching eventually the mass market. The phenomenon of information propagation is strictly tied with the structure of the underlying network.

We analyse data from interactions in Twitter and information propagation cascades to detect relevant communities, interesting patterns in the propagation flow, and influential users. We are capable not only of finding popular opinion leaders (i.e., those with high visibility) but also some "hidden" opinion leaders that are relevant due their ability to push new (innovative) content into the network.

Popular social network analysis tools rely on just some basic (and many times superficial) metrics and tools, such as number of mentions, re-tweets, or keyword-based sentiment analysis. However, those metrics are generic and often provide misleading information. Our service goes deeply into the particularities of a particular domain, being able to design personalized analysis for our clients, generating a rich set of context-aware metrics that would help them a better understanding of their performance in social media.

Service capabilities

Social network extraction: build a topic-dependent network of contributors based on mentions and/or retweets and specific keywords or hashtags. The network should be updated dynamically and our system supports early detection of new relevant nodes (people, accounts). We have software for Twitter, but can develop solutions to acquire data from other platforms.

*Community detectio*n: analyse social relationships and information cascade logs to identify relevant communities (either social or topical) and the role (authority/susceptible) of each user within these communities. The goal is to analyse social interaction data and information propagation cascades to detect relevant communities, interesting patterns in the propagation flow and influential users.

Identification of influential users: interaction logs can be analysed to detect influential users and the strength of their influence on peers. We compute scores of local influence (measuring the influence of a user in a given-topic within their immediate neighbourhoods) and global influence (measuring the typical reach of users, i.e., how many other users end up re-posting content created by him/her, and/or how likely a user's profile is to be visited by someone following). We also can perform categorisation of users into interest categories, and display a ranked list of most influential users on a given topic.

Spheres of influence and consistency: we compute a Consistency of Sphere of Influence (CSI) metric that quantifies the consistency of information propagation cascades in a social graph for a given user. Put more simply, the CSI metric measures the variability of the set of users influenced by the targeted user on different instances. For example, a user with a high CSI score is expected to influence approximately the same set of users and, by targeting it him

(e.g., advertising) we have a higher degree of confidence over how many and which users will be influenced. On the other hand, a user with a low CSI score means that the influence process starting from him/her is highly unreliable and unpredictable. The CSI metric considers all possible cascades that a user can produce, given a probabilistic graph, and finds what we call the "typical cascade" which is the set, which is closest to all these cascades. The CSI score will be the expected distance between the typical cascade and a random cascade from this user. The CSI is the result of recent novel research, which addresses the problem of information propagation inconsistency in social networks. Therefore, nodes with higher CSI are preferable: they are more reliable influencers.

Strategic Influence Maximization. For viral marketing it is very important to detect the optimal set of users to target in order to maximize influence in a given context, such as targeted advertising. We can define which users to target/contact to maximise the number of users that our advertising campaign will consistently reach. We have developed a system to solve the influence maximisation problem based on spheres of influence and set cover. This is the first method for influence maximisation outperforming the theoretically optimal greedy algorithm for influence maximisation.

Our new contributions since the last deliverable are in the last three points (identification of influential users, spheres of influence and consistency, and strategic influence maximization). Below we outline how they fit into the uses cases and then we delve into details regarding how to formalize our problems and our solutions. The results of this research have been published at SIGMOD (Mehmood et al., 2016).

2.1.1 Use cases

Some use cases to which this service may be applied to, are the following:

Use Case 1: Journalism

- Find the most central/influential contributors about a given topic (e.g., users who post about European energy policy).
- Collect information about individual contributors of information and data.

Use Case 2: Commercial Media Monitoring

- Article selection: find the most central/influential contributors about a given topic.
- Assess topics and key influencers for a client’s business.
- Display network-graphs that show how different pieces of information and different contributors correlate.

Use Case 3: SME Internationalization Display sector information.

- Evaluate an author’s impact in the network and provide background information on him.

2.2 Consistency of Sphere of Influences (CSI): overview

2.2.1 Introduction

The phenomenon of influence-driven cascades in social networks has received tremendous attention in the last years thanks to its applications, among which one of the most appealing is viral marketing. The idea of viral marketing is to exploit a pre-existing social network in order to increase brand awareness or to achieve other marketing objectives (such as product sales) through self-replicating viral processes, analogous to the spread of viruses. More concretely, the idea is to target a few “influentials”, in the hope that, through word-of-mouth mechanism, they will be able to spread the marketing message to a large portion of the network, as it was a viral contagion.

This notion was formalized by (Kempe et al., 2003) as the Influence Maximization problem, i.e., the problem of finding the set of k influential nodes (usually named “seed set") such that activating them maximizes the expected number of nodes that eventually get activated in a social network where the contagion is governed by a stochastic propagation model. This problem has received a great deal of attention by the research community in the last decade.

However, (Watts et al., 2007) challenges what he calls “The Influentials Hypothesis", i.e., the assumption that a small set of super-star users can act as sparks to start a large forest fire. Watts states that influence processes are highly unpredictable and unreliable, and relying on a small seed set simply aggravates the unpredictability. Therefore, in order to implement viral marketing in the real world, Watts suggests we should target a large seed set of ordinary individuals who might trigger their small, but more reliable, sphere of influence. Even if each seed manages to activate a handful of other users, the large size of the seed set makes possible to reach a critical mass that can make the campaign go viral.

Inspired by this vision, we study how to compute the sphere of influence of each node s in the network, together with a measure of stability of such sphere of influence, representing how predictable the cascades generated from s are. We then devise an approach to influence maximization based on the spheres of influence and maximum coverage, which is shown to outperform in quality the theoretically optimal method for influence maximization when the number of seeds grows.

In order to better explain our contributions, we first need to provide some preliminary background on the influence maximization problem.

(Kempe et al., 2003) modelled viral marketing as a discrete optimisation problem, named influence maximization, and based on the concept of propagation model: i.e., a stochastic model that governs how users influence each other and thus how contagion happens. Given a propagation model and a set of nodes $S \subseteq V$, the expected number of nodes “infected” in the viral cascade started with S is called the expected spread of S, denoted by $\sigma(S)$. For a given $k \in \mathbb{N}$ the influence maximization problem asks for a set $S \subseteq V$, $|S| = k$, such that $\sigma(S)$ is maximum.

The most studied propagation model is the so-called Independent Cascade (IC) model. We are given a directed probabilistic graph $G = (V,E,p)$ where each arc $(u,v) \in E$ is labelled with a contagion (or influence) probability $p(u,v) \in (0, 1]$, representing the strength of the influence

of u over v. At a given time step, each node is either active (an adopter of product) or inactive. At time 0, a set S of seeds is activated. Time unfolds deterministically in discrete steps. When a node u first becomes active, say at time t, it has one chance to influence each inactive neighbour v with probability p(u,v), independently of the history thus far. If the attempt succeeds, v becomes active at time t + 1.

Influence maximization is generally NP-hard. (Kempe et al., 2003), however, show that the objective function σ(S) is monotone and submodular. When equipped with such properties, the simple greedy algorithm that at each iteration greedily extends the current set of seeds S with the node w providing the largest marginal gain gives a (1 - 1⁄e)-approximation to the optimum.

Another source of complexity is the fact that the computation of the expected spread which is itself #P-hard. Therefore in the work of (Kempe et al., 2003), Monte Carlo simulations are run sufficiently many times to obtain an accurate estimate of the expected spread. In particular they show that for any ϕ>0, there is a k = poly(n⁄ϕ) such that by using k samples, we can obtain a (1 - 1⁄e - ϕ)-approximate solution for influence maximization.

Finally, it is important to note that the (1 - 1⁄e) approximation ratio for influence maximization cannot be further improved, at least under the IC propagation model. This is due to the fact that influence maximization under the IC model encodes max-k-cover as a special case, which is known to be not approximable within ratio (1 - 1⁄e + ϵ) unless P= NP. For this reason, while a very literature has been produced on the efficiency and scalability of influence maximization, understandably very little attention has been devoted to improving the quality (at least in practice, given that in theory it is not possible).

2.2.2 Problem studied

The problem studied here is, abstractly, to compute the set of nodes that, under a probabilistic contagion model, would get infected if a given node s get infected. This can be seen as a novel type of reachability query in uncertain or probabilistic graphs. More in details, our data is a probabilistic directed graph G = (V,E,p), where p : E → (0, 1] is the contagion probability, i.e., the probability that the arc will exist, or participate, in a contagion cascade. Our query is a source node s ∈ V , and the result is a set of nodes C ⊆ V , which we call the sphere of influence of s, i.e., the set of nodes that would get infected if the node s get infected.

This type of query can find application in many contexts besides viral marketing: from epidemics (given an Ebola case, which other individuals should we quarantine?), to corporate workflows, computer and financial networks (given a node failure, which is the typical cascade we can expect?).

As the contagion is a stochastic process, we need to define a way to identify a unique set of nodes C. In fact, each possible subset of V can be a possible cascade from s, each one with its own probability of materializing.

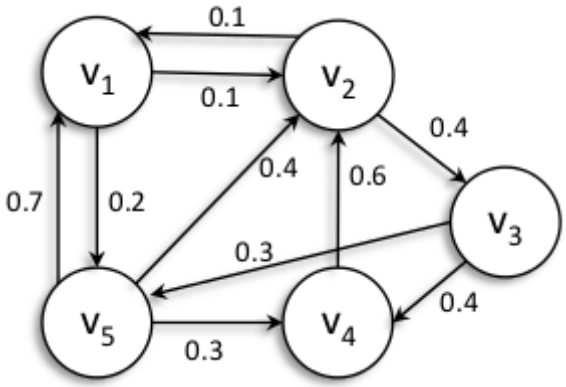

Figure 2: Example 1 - Probabilistic Graph

Consider the probabilistic graph in Figure 2, and suppose v5 is our query node. The probabilities associated to the arcs define a probability distribution over the possible subsets of {v1, v2, v3, v4}. For instance the set {v1} is the cascade of v5 with probability 0.7 · (1 - 0.4) · (1 - 0.3) · (1 - 0.1) = 0.2646 (i.e., the arc (v5, v1) succeeds transmitting the contagion, while (v5, v2), (v5, v4), and (v1, v2) all fail). Similarly the set {v2, v4} is the cascade of v5 with probability (1-0.7)·0.3·(1-((1-0.4)·(1-0.6))·(1-0.1)·(1-0.4) = 0.036936 (i.e., the arc (v5, v1) fails transmitting the contagion, (v5, v4) succeeds and at least one among (v5, v2) and (v4, v2) succeeds, finally (v2, v1) and (v2, v3) both fail).

As a final example, the set {v1, v3, v4} has null probability of being the cascade of v5 as v3 can only be infected by v2.

Thus, how to identify a unique set of nodes *C* from this probability distribution? One could think to select the most probable cascade, but this would not be a good choice as explained next. If we have |V | = n nodes there are 2n possible cascades and n is usually large. This means that we have a probability distribution over a very large discrete domain, with all the probabilities that are very small. Therefore, the most probable cascade still has a tiny probability, not much larger than many other cascades. Finally, the most probable cascade might be very different from many other equally probable cascades.

Instead here we study the problem of computing set of nodes, which is the closest (in expectation) to all the possible cascades of s. For our purpose we need a set-similarity measure: the Jaccard similarity is the most natural choice and it has the benefit of being a metric.

2.2.3 Contributions

Our main contributions are summarised as follows:

We formalize the Typical Cascade problem which requires, for a given source node s, to find the sets of nodes minimizing the expected Jaccard distance to all the possible cascades from s. Such expected cost also represents, for a given node s, a measure of the stability of its sphere of influence, i.e., how much a random cascades from a source node s deviate from the “typical” cascade. In this sense source nodes with lower expected cost are preferable: e.g., in the context of viral marketing they can be considered more reliable influencers.

We show that for a given source node s, computing the expected cost for a candidate set of nodes is #P-hard. We then devise a solution based on sampling possible worlds and then

computing the Jaccard median of the obtained cascades (Chierichetti et al., 2010). The next question we face is how many samples are needed in order to get a “good" approximation. We answer this question by providing theoretical bounds showing that, quite surprisingly, we can obtain a multiplicative approximation with a constant number samples, i.e., not dependent on the size of the network.

Backed by our theoretical results, we turn our attention to the practical deployment of our algorithm and we devise an index that allows to efficiently computing the sphere of influence for any node in the network.

Finally, we apply our framework to the influence maximization problem and propose a max-cover based solution over the spheres of influence. Through exhaustive evaluation using real-world networks and different methods of assigning the influence probability to each arc, we show that our approach outperforms in quality the theoretically optimal greedy algorithm. Our method based on spheres of influence has several interesting features that can explain its quality.

The first observation is that with our method we intuitively steer the attention of the greedy algorithm from the average size of cascades (i.e., the expected spread), to the size of the “average cascade”. This gives us a more reliable approach to the influence maximization problem. In fact, as suggested by intuition, the typical cascade of a node gets larger when all the possible cascades from that node have a large common portion, or in other terms, are similar. Therefore, by picking nodes with large typical cascades, not only do we pick nodes that are influential, but also implicitly favour influentials that are reliable. The connection between the size of the typical cascade and its cost is confirmed empirically in Section 6.3.

We also show empirically that, as the seed set size grows, at a certain point the standard influence maximization approach reaches a saturation point where it is no longer able to distinguish well among nodes to be added to the solution. Essentially, by focusing on the marginal gain w.r.t. the expected spread, the standard method finds itself choosing among many practically equivalent nodes. Instead, our method, by focusing on the sets themselves, is still able to distinguish the next good candidate when the standard influence maximization has reached its saturation point. From this moment on (that is to say, for large seed sets) our method starts outperforming the theoretically optimal algorithm w.r.t. the expected spread objective function.

Our empirical findings are consistently confirmed by a thorough experimentation over several influence networks, which are the typical benchmarks used in the influence maximization literature, and by using different ways of learning/assigning the influence probability to each link.

To the best of our knowledge, our work is the first to show consistent improvement in terms of quality over the standard greedy algorithm for influence maximization.

2.3 Spheres of influence: problem formulation and algorithms

Let G = (V,E,p) be a probabilistic directed graph, where p : E → (0, 1] is a function that assigns a probability of existence to each edge. Following the literature, we consider the edge probabilities independent. In this setting, the possible-world semantics is a principled way of defining the meaning of a query over uncertain data. Specifically, the possible-world

semantics interprets G as a probability distribution over subgraphs of (V,E) defined by choosing every edge e ∈ E independently at random with probability p(e).

Let q(G) be a function that when applied to a deterministic graph G returns a value in $\mathbb{R}$. Following the possible-world semantics querying q over the probabilistic graph G is typically done by asking for its expected value.

When q(G) is a binary predicate then the expectation corresponds to the probability that the predicate is satisfied. For example, this is the case for instance of reachability query r(u,v), which returns true if v is reachable trough a directed path from u. In the context of probabilistic graphs, the corresponding reliability query would ask for the probability of v being reachable from u.

In this work, we are interested in a type of query, which returns neither a scalar nor a binary, but a set of nodes. Given a directed probabilistic graph G = (V,E,p) and a node s ∈ V we are interested in the cascade originated from s, i.e., the set of nodes that would get infected if s get infected. In the case of a deterministic graph, that would be the set of nodes reachable from s trough directed paths. But how is it possible to determine the typical cascade in a probabilistic graph?

2.3.1 Problem formulation

In a sense we want to define a notion of “expected” or “typical” cascade: this is a cascade which is the closest to the set of possible cascades of s. Towards formalizing this intuition, we need a metric to compute the distance among two possible cascades. As previously stated, a cascade simply corresponds to a set of nodes, and therefore we use Jaccard distance. Given two sets of nodes A,B ⊆ V , their Jaccard distance dJ(A, B) is defined as the size of their symmetric difference divided by the size of their union. This is known to be a metric.

Given a deterministic graph G ⊑G and a node s ∈ V , we denote by Rs(G) the set of nodes reachable from s in G. Given G, s ∈ V and a set of nodes C ⊆ V , we define the expected cost, ρG,s(C), of C as the expected Jaccard distance between C and a random cascade generated from s. (We omit the dependence on G or s when appropriate.)

This represents a measure of *stability* or *consistency*, i.e., how much random cascades from s deviate from C. It is therefore desirable to find the set C*⊆ V that minimizes the expected cost ρ s(C*). This set represents the typical cascade of the node s, or what we abstractly call its sphere of influence. The cost of this set is the *Consistency of the sphere of influence of s.*

Problem 1 (Typical cascade). Given a probabilistic graph G = (V,E,p) and a source node s ∈ V find the set C*⊆ V that minimizes the expected cost.

The first source of complexity for the Typical Cascade problem is that for a given source, computing the expected cost of a set of nodes is #P-hard.

Theorem 1. Given a probabilistic graph G = (V,E,p), a source node s ∈ V , and a set of nodes C ⊆ V , computing ρG,s(C) is #P-hard.

The proof (Mehmood et al., 2016) employs a reduction from s-t reliability on graphs: given a directed probabilistic graph G and two nodes s,t ∈ V (G), compute the probability that there is a path from s to t, denoted rel(G,s,t). This problem was shown #P-hard by (Valiant, 1979).

A natural approach to deal with some #P-hard problems is by means of Monte-Carlo sampling: this means to sample a large enough number ℓ of independent cascades S = {S1,…,Sℓ} from s, and use them to compute an estimate as the empirical mean of dJ(C,Si) over the cascades sampled. This is an unbiased estimator of the actual cost ρs(C) as defined above, so one may hope to use our estimation as a proxy for the actual cost ρs(C) and attempt to solve the following related optimization problem.

Problem 2 (Jaccard median). Given a finite set V and a collection S of ℓ sets S1,…, Sℓ ⊆ V , find a set C*⊆ V that minimises the average Jaccard distance of C* from the elements of S.

(Chierichetti et al., 2010) show that Problem 2 is NP-hard, and present a polynomial-time approximation scheme.

The difference between Problems 1 and 2 is that in the latter we are given a list of sets, while the first one defines implicitly a distribution over exponentially many sets. Neither one seems easily reducible to the other, though: on the one hand, enumerating all subgraphs to apply Jaccard median to our problem would require exponential time; on the other hand, a solution to the typical cascade problem may not extend to a general solution to Jaccard median, since the set of possible cascades from a vertex in a graph has certain special properties (for example, closure under unions).

2.3.2 Sampling and Jaccard Median

Before we hinted at a possible approach to tackle the Typical Cascade problem:

1. Sample ℓ random cascades from source nodes s;
2. Compute their Jaccard median as the typical cascade.

An important question is how many deterministic graphs we need to sample in order to obtain a good estimate of the median quality, and to avoid overfitting in the scheme above. To fix notation, let C denote a distribution over non-empty subsets of [n] (for example, C could be the reachability sets from a given vertex in an n-vertex uncertain graph). Let ρ(X) denote the cost of a candidate solution X, and M denote an optimal median with cost $\epsilon^* = \rho(M^*)$.

Recall that we cannot evaluate ρ(X) efficiently, so we resort to sampling ℓ independent elements of C and using the empirical mean $\hat{\rho}(X)$ as an estimator for ρ(X). Then we derive a median M that approximately minimises ^ρ s on the sample, with the hope that its actual cost ρ(M) will be close to optimal.

While one can easily show that the cost of any particular set X is approximately preserved (with additive error) in the sample, this may not hold simultaneously for all sets. The situation is analogous with the problem of overfitting in learning theory: while the error of any given classifier can be accurately estimated from the training set, if the learner's hypothesis class is large enough it may happen that we find a hypothesis that does exceptionally well on the training set, but performs badly on the test set. (In our setting, the set 2[n] of all "candidate medians" play the role of the hypothesis class.) In fact, as there are

2n candidate medians, a naive estimate via the union bound would suggest that Θ(n) samples are needed, which is too large a sampling size to be practical (recall that n is the number of nodes of the graph in our application). Fortunately, this is far from tight: our next result shows that these bounds can be substantially improved, as long as the cost of the optimal median is small: a constant number of samples suffice to get good multiplicative approximations.

Theorem 2 (informally stated). For any α >ϵ*, a sample of size ℓ = log(1⁄α)⁄α^2 suffices to obtain an (1 + O(α))-approximate median.

This is remarkable, because the number of samples is independent of n and moreover, it does not suffice in general to estimate the cost of a candidate solution with small multiplicative error (this would require ℓ = Ω(1⁄ϵ*)). This means that the empirical and true costs may differ significantly, yet the cost of the solution found by solving the empirical problem is very close to the true optimal. The result also yields a sublinear-time randomized approximation algorithm for standard Jaccard median (Problem 2) when the number of input sets is large: sample O(log(1⁄α)⁄α^2) of the input sets and work on the smaller instance.

Roughly speaking, the proof of Theorem 2 proceeds as follows. We show that a) any nearly optimal median X gives rise to an easily manageable approximate cost function fX with certain properties implying that no median can do much better than X; and b) these properties of fX are approximately preserved after sampling. This trick allows us to convert the statement "for all candidate medians, their sample cost is not much smaller than ρ(M*)" into a statement regarding the single function fX, which can be proved directly. The details may be found in (Mehmood et al., 2016).

2.3.3 Practical Algorithms

Next, we put together the pieces from the theoretical insights introduced in the previous section, and discuss practical efficiency considerations. First, we describe an indexing scheme enabling efficient simulations of the cascades needed. Then we present the main algorithm to compute a typical cascade for every node of G.

In order to obtain the typical cascade for a given vertex v, we first need to produce a certain number ℓ of cascades from v. As we saw before, taking ℓ = O(log(1⁄α)) samples is enough to obtain a (1 + α)-approximation provided that the cost is Ω(α); if we wish this guarantee to hold simultaneously for all vertices, we may take ℓ = O(log(n⁄α)⁄α^2). Rather than sampling separately for each vertex, we sample ℓ possible worlds G1,…,Gℓ from G, each of which implicitly defines a sample cascade from each vertex v ∈ V (G), which may be obtained by performing a DFS traversal of G rooted at v.

A key observation that we exploit to speed up this process is that all the vertices in the same strongly connected component (SCC) have the same reachability set: since any two vertices u, v in the same SCC are reachable from each other, any vertex reachable by u is also reachable by v, and vice versa. Therefore, we can represent each sampled possible world G_i by its SCC structure. Representing G_i in terms of its SCCs yields savings in both space usage and computational runtime, because of the compactness of representation and because a single DFS is sufficient to identify the reachability set of all vertices in the same component.

Based on these observations we build an index that contains for all the sampled possible worlds $G_1,\ldots,G_\ell$:

1. The condensation C_i of G_i, that is, the directed acyclic graph of links between SCCs, obtained by contracting each component of G_i to a single vertex;

2. For each vertex v and index i, the identifier of the connected component of v in G_i (see Figure 3)

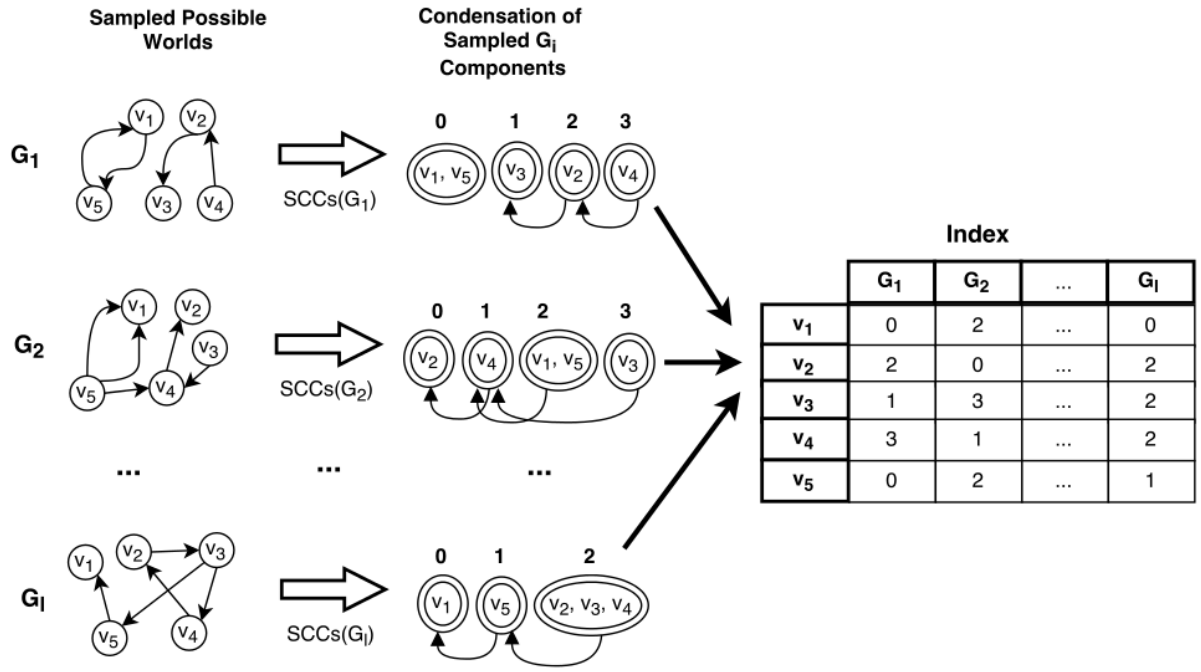

Figure 3: Indexing scheme for all sampled possible worlds

As depicted in Figure 3, for each sampled possible world G_i it is stored the structure made of the condensation of the SCCs and a matrix indicating for each vertex v and each possible world G_i, the index of the component to which v belongs in G_i.

Computing the SCCs and performing their contraction can be performed in time linear in the total number of vertices and edges of the graphs sampled.

To further reduce the space consumption, we perform the transitive reduction of the condensation of Ci, i.e., find the unique graph Ti (not necessarily a subgraph of Ci) with vertex set V (Ci) that preserves the reachability/non-reachability between every pairs of vertices of Ci and has the smallest number of edges. While the worst-case computational complexity of this task is theoretically equivalent to that of Boolean matrix multiplication, for which the best algorithms known run in time $O(n^{2.373})$, in the practical instances arising in our experiments the algorithm from (Aho et al., 1972) proved adequate.

The procedure to construct the index is summarized in Algorithm 1 (Figure 4).

Algorithm 1: Index construction

Input : Input graph $\mathcal{G}$ and number of samples ℓ.
Output: Index $\mathcal{I}$, Component Pointers $\mathcal{P}$

$\mathcal{I} \leftarrow [|V| \times \ell]$
$\mathcal{P} \leftarrow [1 \times \ell]$
for $i \leftarrow 1$ ***to*** k **do**
  Sample G_i from $\mathcal{G}$
  $SCCs \leftarrow StronglyConnectedComponents(G_i)$
  $\mathcal{P}[i] \leftarrow transitiveReduction(SCCs)$
  foreach $v \in V$ **do**
    $\mathcal{I}[v, i] \leftarrow nodeComponentIndex(v,\ SCCs)$
  end
end
return $(\mathcal{I}, \mathcal{P})$

Figure 4: Algorithm 1: Index construction

Given a node v and i ∈ [ℓ], the cascade of v in G_i can be obtained as follows: look at the identifier of the SCC of v in G_i; recursively follow the links from the associated condensed vertex in Ci to find all the reachable components; and output the union of the elements in the reachable components. The time to perform this computation is linear in the number of nodes of the output and the number of edges of the condensation Ci, which is typically much smaller than the number of edges of G_i.

For the computation of the typical cascade Cv of a node v, we need to compute an approximate Jaccard median of the collection of ℓ cascades S1,…,Sℓ from v. To this end, we use the work of (Chierichetti et al., 2010). Their PTAS to achieve arbitrarily good approximations is mostly of theoretical interest, so we use the following algorithm (see Figure 5), which achieves an 1 + O(ϵ) factor approximation (where ϵ is the cost of the optimal median of the instance) and runs in linear time.

Algorithm 2: All Typical Cascades

Input : Input graph $\mathcal{G}$, number of samples ℓ.
Output: The typical cascades for each $v \in G$.

$(\mathcal{I}, \mathcal{P}) \leftarrow \text{Index}(\mathcal{G}, \ell)$
for $vinG$ **do**
  $S \leftarrow [1 \times \ell]$ (list of cascade sets)
  for $i \leftarrow 1$ ***to*** ℓ **do**
    $c \leftarrow \mathcal{I}[v, i]$
    $cG \leftarrow reachable\_components(\mathcal{P}[i], c)$
    $S[i] \leftarrow \bigcup\{nodes(c) \mid c \in cG\}$
    $C_v \leftarrow JaccardMedian(S)$ (by Chierichetti et al. [11])
  end
end
return $\{(v, C_v) \mid v \in V(\mathcal{G})\}$

Figure 5: Algorithm 2: All Typical Cascades

2.3.4 An application to Influence Maximization

In this Section, we present, as an application of the typical cascade computation, a novel approach to influence maximization. While our approach is heuristic in nature, it is motivated by the observations below.

1. Given a seed set S, we can define its stability as the expected cost of its typical cascade C*. If this cost is small we say that the seed set is reliable.

2. If S is a highly reliable seed set, the size of its typical cascade C* is very close to the mean size of cascades from S. In other words, by optimizing for size of the typical cascade we are also indirectly optimizing for expected spread, unless the optimal solution is unreliable (which is ruled out by the next item).

3. It is an empirically observable phenomenon (see the stability analysis in Sec. 6) that the expected cost of the typical cascade of S tends to decrease as S grows. Intuitively, this means that the cascading process becomes more and more deterministic (or predictable) as the size of the seed set increases. We want to leverage this fact by acting as if the cascade from S were effectively C*. However, for sake of efficiency, in our method we will not use the typical cascade of S, but instead we will use the union of the typical cascades of all the seed nodes in S. This is justified by the next point.

4. It can be shown that some nearly optimal typical cascade from seed set S is a superset of the typical cascades for the cascades induced by the individual elements of S. This is because if the typical cascade has cost ϵ, then simply selecting all elements that are present with probability at least 1⁄2 can be proved to be a solution with cost at most $\epsilon + O(\epsilon^{3/2})$. But note that the probability that a given vertex is reachable from S is monotonically increasing with S, so the set of elements reachable with probability at least 1⁄2 is monotonically increasing as well. Consequently the nearly optimal typical cascade for S can be assumed to contain the typical cascades for all its elements. (It may contain further elements, which would increase the solution size and decrease its cost, so we are being conservative by ignoring them.)

These observations motivate us to approach the influence maximization problem as max-cover problem over the typical cascades of the singleton nodes. Let $S \subseteq V$ denote a set of nodes and let Cv be the median cascade of each $v \in V$. Write $\Phi(S) = \bigcup_{v \in S} Cv$ for the elements covered by the typical cascades of all the nodes in S, Now, given an finite integer $k \leq |V|$, our goal is to find a set S* such that the coverage $\Phi(S^*)$ is maximized for $|S^*| = k$.

This is an instance of the maximum coverage problem, which can be approximated by the standard greedy algorithm that runs for k iterations and at each iteration it selects a node v whose addition increases the value of Φ the most. This approach, whose pseudocode is shown in Algorithm 3 (see Figure 6), is named InfMax_TC: Influence Maximization using Typical Cascades.

Algorithm 3: InfMax_TC

Input : Typical cascades for all $v \in V$: $\{C_1, C_2, \cdots, C_{|V|}\}$
Output: Seed set of k nodes: $\mathbb{S}^*$

$\mathbb{S}^* \leftarrow \emptyset$
for $i \leftarrow 1$ ***to*** k **do**
  $u = \arg\max_{u \in V \setminus \mathbb{S}^*} \Phi(\mathbb{S}^* \cup u)$
  $\mathbb{S}^* \leftarrow \mathbb{S}^* \cup u$
end
return $\mathbb{S}^*$

Figure 6: Algorithm 3: InfMax_TC

Next, we compare this method for influence maximization with the standard method, with respect to the objective function of the influence maximization problem: the expected spread.

2.4 Performance evaluation

In our evaluation, we first report basic statistics about the spheres of influence (or typical cascades) and their computation, such as their size, cost and the running time of our procedure. We then focus on the main goal of our experimental assessment, which is to show the performance, in terms of quality, of our method for influence maximization.

The majority of the literature on influence maximization uses a set of benchmark social graphs, where the influence probability for each edge is artificially assigned according to some certain standard methods. Perhaps more appropriately, some authors (Goyal et al., 2010; Saito et al., 2008) have started to use influence probabilities learnt from a log of past user activity. In our experiments, for sake of exhaustiveness, we follow both approaches. We use six datasets, which are often used as benchmarks in the literature for influence maximization: in three of these, the edge probabilities are learnt, whereas for the other three the probabilities are assigned artificially as described later. Moreover, we use two different methods for learning the contagion probabilities and two different methods for assigning them. This gives us a total of 12 datasets to work with.

Dataset description. The three datasets that come with a log of users activity and can thus be used for learning the influence probabilities are Digg, Flixster and Twitter.

Digg is a news portal that allows users to submit news stories, as well as rate the posted stories by means of voting. The ratings are then used to promote stories on the front page of Digg portal. The data snapshot we use is related to the voting history of all the stories promoted during June 2009. It has 3M votes for 3.5K stories. The data also provides a fan network from which a directed social graph is induced.

Flixster is an online social networking service (https://flixster.com/) enabling its users to rate and review movies. Here, we have user ratings from November 2005 to November 2009. This amounts to 8.2M ratings for 49K items/movies.

The final dataset in this category is a snapshot of Twitter, obtained by crawling its public timeline. The items in Twitter represent the URLs propagating across the network. Unlike the

previous two datasets, in Twitter the user activity corresponds to sharing/re-sharing of the URLs instead of rating items. The data contains 6K items and 383K user activities.

The other three datasets are from the SNAP dataset collection. These include NetHEPT, Epinions, and Slashdot. The first is a network of citations, whereas the other two are social networks. These datasets are widely used in the study of social networks and influence maximization. Table 1, reports basic statistics on the datasets used. When a graph is undirected, we just consider the edges existing in both directions.

Datasets	V	ϵ	Type	Probabilities
Digg	68K	875K	directed	learnt
Flixster	137K	1.2M	undirected	learnt
Twitter	23K	650K	undirected	learnt
NetHEPT	15K	31K	undirected	assigned
Epinions	76K	509K	directed	assigned
Slashdot	77K	905K	directed	assigned

Table 1: Dataset Characteristics

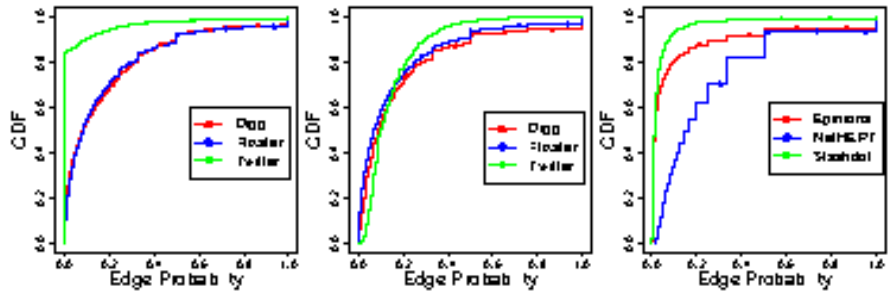

Figure 7: CDF of edge probability learnt from (Saito et al., 2008) (left), (Goyal et al., 2010) (centre) and WC model (right). We don't report the distribution for the fixed probability method, as this is not meaningful

Edge probabilities. Below we describe the two different ways of learning the influence probabilities, and the two different ways of artificially assigning the influence probabilities, that we use in our experiments.

Learning from real-world cascades. The datasets in the first category (Digg, Flixster, and Twitter) provide us two key elements: (i) a social network (ii) log of user activities (for different items) with the corresponding timestamps. Both methods we use exploit these two pieces of input to learn the edge probabilities. The first method is (Saito et al., 2008), which model the learning of the influence probabilities as a likelihood maximization problem and devise an EM algorithm to solve it.

The second method (Goyal et al., 2010) follows a frequentist approach. Among the various models they propose we use the simplest one: the probability assigned to an edge (u,v) is simply the number of times in the propagation log in which v performs an action after u, divided by the number of actions performed by u.

In the following we will use a suffix -S and a suffix -G to denote the datasets with the probabilities learnt by following (Saito et al., 2008) and (Goyal et al., 2010) respectively.

Artificial assignments. For the second group of datasets (NetHEPT, Epinions, and Slashdot), we use two different methods for artificially assigning probability to each edge. The first the methods is the weighted cascade (WC) model, which sets the probability pu,v over an edge (u,v) as: pu,v = inD1eg(v). Here, inDeg(v) is the in-degree of node v. In the second method, we assign a fixed probability pu,v = 0.1 to each edge (u,v).

In the following we will use a suffix -W and a suffix -F to denote the datasets with the probabilities assigned by weighted cascade method and fixed respectively.

Using the methods of learning/assigning edge probabilities explained above, we have 12 datasets in total for our experiments detailed in this Section. Table 2, demonstrates the CDFs of the edge probabilities in all datasets.

Datasets	avg($\overline{C_\cdot}$)	sd($\overline{C_\cdot}$)	max($\overline{C_\cdot}$)
Digg-S	9.0	22.2	263
Flixster-S	4.3	12.0	439
Twitter-S	17.0	86.4	1459
Digg-G	7.3	17.0	130
Flixster-G	999.5	822.7	2589
Twitter-G	24.9	58.4	1727
NetHEPT-W	3.0	1.2	13
Epinions-W	3.6	8.6	684
Slashdot-W	4.8	19.4	420
NetHEPT-F	1067.5	915.5	4138
Epinions-F	4774.5	1574.5	6345
Slashdot-F	1337.0	841.5	5574

Table 2: $\overline{C_\cdot}$ denotes the size of the approximated typical cascade computed and we report its average, standard deviation and maximum over all models in the graph

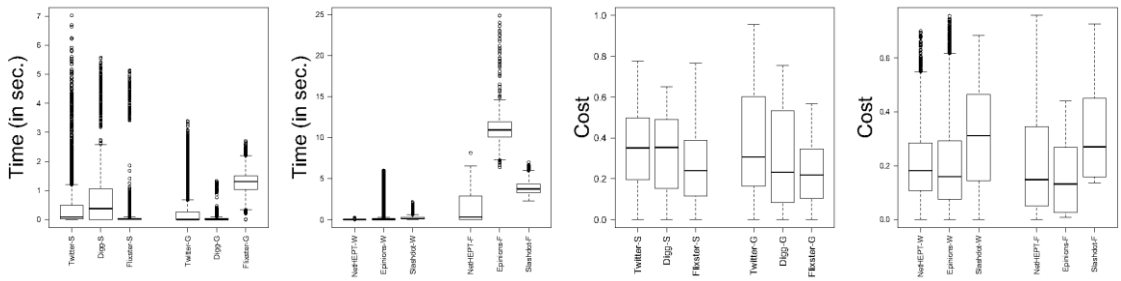

Figure 8: Time taken to compute the typical cascade ^C* (two left most plots) and its expected cost ρG,s(^C*) (two right most plots)

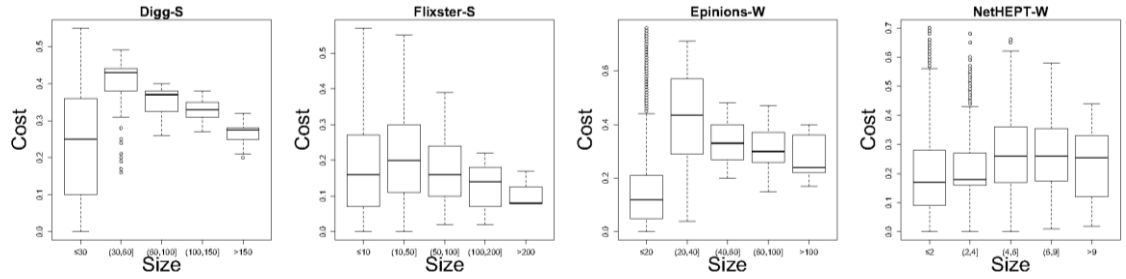

Figure 9: Distribution of the expected cost of the typical cascade ρG,s(^C*) with regards to its size

Computing the typical cascades. Table 2 reports basics statistics on the size of the sampled cascades (Si) and the typical cascade computed from the samples (C^*). Given that the edge probabilities learnt using the method of (Goyal et al., 2010) are larger than the probabilities learnt by means of (Saito et al., 2008) method (see Figure 7), not surprisingly also the average size of typical cascades is larger for the former than for the latter.

The pattern is even more evident in Flixster, which showcases the fact that different strategies used to assign probabilities may greatly impact the size of the samples, and thereby, the size of the corresponding typical cascade. We also note that when artificially assigned, the probabilities set fixed to 0.1 also result in larger sampled cascades, and thus larger typical cascade, than produced when assigning probabilities by means of the WC model.

In Figure 8, we report the time taken to compute the typical cascade ^C* and its expected cost ρG,s(^C*), excluding the index construction. That is, this is the time to extract the cascades from the index and run the Jaccard median approximation on this instance. (Recall that we are using 1000 samples so the number of elements to process per vertex is often in the hundreds of thousands.) The times reported use a Python implementation, on a Intel Xeon 2.2 Ghz with 6 cores and 16 GB memory. As depicted, the time remains almost always well under 1 second except for a small number of nodes. As regards the expected costs they rarely exceed 0.4, and in most of the dataset the average is around 0.2.

Figure 9 reports the distributions of the expected cost concerning the size of the typical cascade, in order to assess whether the quality (or reliability) of the solution also depends on its size. In every plot, if we disregard the bucket of very small cascades, which is in any case not very interesting for applications, we can observe that the larger is the typical cascade, the more reliable it is (smaller cost). This becomes even more evident when observing the maximum cost observed: it is practically impossible to find a large typical cascade with large cost.

Evaluation of Influence maximisation

We next present our main practical result: the fact that our method for influence maximization based on spheres of influence outperforms the standard influence maximisation method for what concerns quality, i.e., the expected spread achieved.

Quality of influence maximisation. In the following, the standard greedy (theoretically optimal) algorithm for influence maximization (Kempe et al., 2003) is denoted InfMax_std, and our greedy algorithm for maximum coverage using the sphere of influence (typical

cascade) of each node is denoted InfMax_TC. In all the experiments we use k = 200 for the seed set size and we use the same number of sampled cascades (1000) for both methods: to estimate the expected spread for InfMax_std, and for computing the typical cascades for InfMax_TC. The expected spread σ(S) is reported in each iteration of the two greedy algorithms from |S| = 1 to 200.

The results for all combinations of datasets and ways of assigning edge probabilities are reported in Figure 10: on the X-axis we report the size of the seed set |S|, on the Y -axis we report the expected spread σ(S). We can observe the same pattern emerging in all settings: InfMax_std outperforms InfMax_TC in the selection of the first several seeds, but at a certain point, as the seed set size grows, the two curves cross and InfMax_TC starts outperforming the standard method.

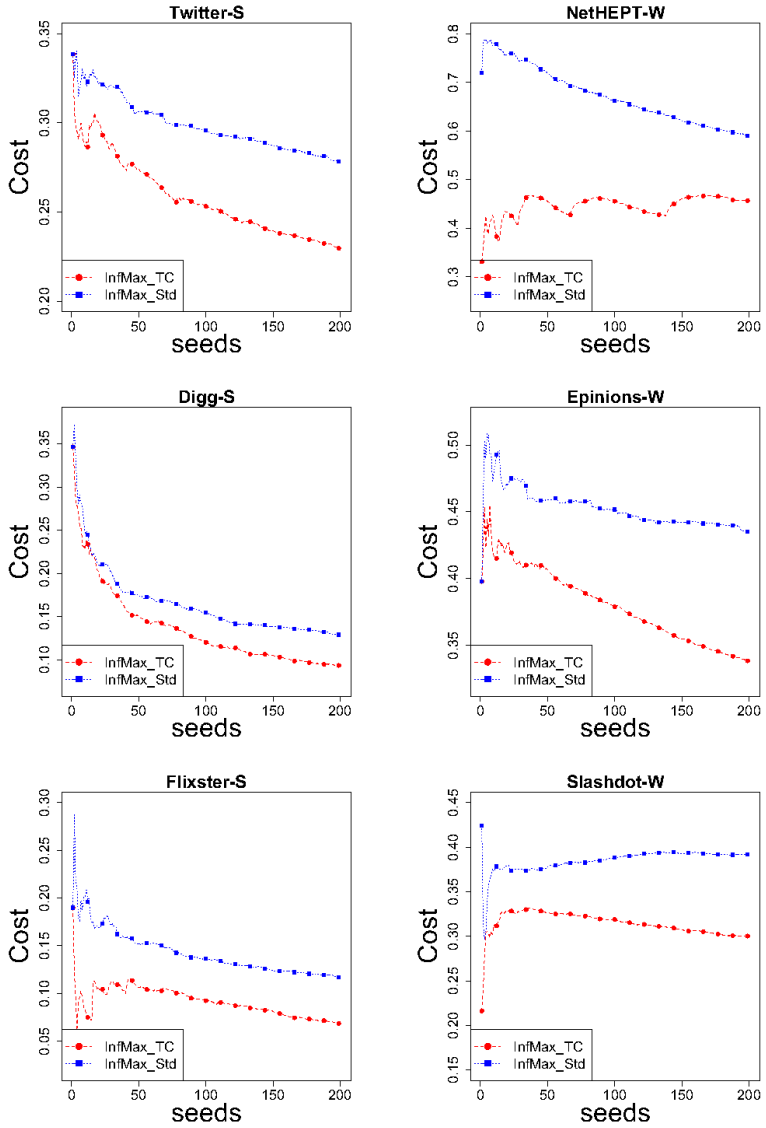

Figure 10: Stability Analysis: expected cost of the seed sets extracted by InfMax_std and InfMax_TC over six datasets. By expected cost we mean the expected Jaccard distance between the typical cascade generated by the seed set and 1000 random cascades generated by the same seed set. The smaller this value the more reliable the behaviour of the seed set

2.5 Spheres of influence: summary

We studied the problem of computing the sphere of influence for each node in a social influence network. We formalize this as the Typical Cascade problem over a probabilistic directed graph where each directed edge (u, v) has associated a probability representing the contagion probability, or the strength of influence of u over v. We devise a method based on sampling and computing the Jaccard median of the samples. Then we propose a novel approach to influence maximization based on max-cover applied to the sphere of influence of all nodes in the network.

Our main theoretical result is a bound showing that we can obtain a multiplicative approximation to our problem with a constant number samples, i.e., not dependent on the size of the network (Theorem 2).

Our main practical contribution is the first method for influence maximization outperforming the theoretically optimal greedy algorithm for influence maximization, for large seed sets.

To the best of our knowledge, our work is the first to show consistent improvement in terms of quality over the standard greedy algorithm for influence maximization, as confirmed by our thorough experimentation using several different benchmark networks and different ways of assigning the influence probabilities to the edges.

Within MULTISENSOR, the computation of spheres of influence has been used to provide CSI (Consistency of Sphere of Influence) scores for Twitter users, indicating the reliability/consistency of the propagation cascades generated by this user. The global influence index (GIN) has also been updated, so as to use the size of the sphere of influence instead of the expected cascade size as was done in previous versions of the SMAP service; the former providing a more robust measure.

2.6 Community detection

The analysis of social media as complex networks, by means of community detection, has provided insight into the role of individual users within their community, and into the position of each community in the social network. The community detection problem, in general, is defined as the identification of groups of densely connected users in a social media network, given the network structure. The MULTISENSOR community detection framework has been presented in previous deliverable (D3.3), where several SoA community detection techniques have been reported, in static or evolving networks. A plethora of community detection algorithms have appeared in the literature (Fortunato, 2010; Malliaros and Vazirgiannis, 2013; Harenberg et al., 2014), however, only a few of them are large-scale algorithms and directly applicable in large social media graphs (Papadopoulos et al., 2012).

In MULTISENSOR, the social media network is topic-based and related to a list of hashtags, such as #energy_policy, #homeappliances, etc. Communities may be static (extracted for a single day) or dynamic, in the context of evolving networks. The detection of dynamic communities has been also tackled in the SocialSensor project (FP7-287975), where a community detection algorithm is applied on each graph snapshot, defined by the network of user mentions (@user). The method that is used for the detection of communities on a graph snapshot is the Louvain method (Blondel et al., 2008).

The Louvain method is based on the modularity maximization and involves two phases that are repeated iteratively. In the first phase, each vertex forms a community and for each vertex i the gain of modularity is calculated by removing vertex i from its own community and placing it into the community of each neighbour j of i. The vertex i is moved to the community for which the gain in modularity becomes maximal. In case the modularity decreases or remains the same, vertex i does not change community. The first phase is completed when the modularity cannot be further increased. In the second phase, the detected communities formulate a new network with weights of the links between the new nodes being the sum of weights of the links between nodes in the corresponding two communities. In this new network, self-loops are allowed, representing links between vertices of the same community. At the end of the second phase, the first phase is re-applied to the new network, until no more communities are merged and modularity is maximized.

Contrary to the modularity maximization approach, we adopt the expected codelength minimization of the Infomap method (Rosvall and Bergstrom, 2008; Rosvall et al., 2010; Bohlin et al., 2014). The inventors of the Infomap method showed that the problem of finding a community structure in networks is equivalent to solving a coding problem. In general, the goal of a coding problem is to minimize the information required for the transmission of a message. Initially, Infomap employs the Huffman code (Cover and Thomas, 2012) in order to give a unique name (codeword) in every node in the network and then minimizes the Shannon information (Cover and Thomas, 2012) required to describe the trajectory of a random walk on the network. In the following, we show the superiority of codelength minimization and the Infomap community detection method, compared to other approaches, in terms of performance.

The community detection module is freely available on Github[1] and has been implemented in R, using the igraph[2] package (version 1.0.1), under the General Public Licence (GPL) v2 or greater. It constructs a graph of crawled Twitter mentions, with nodes the corresponding Twitter IDs, based on given hashtags, such as #foodmanufacturing, #yogurt, etc. The graph is then clustered into communities using the Infomap community detection algorithm. Finally, the module outputs a JSON file, which keeps the network structure as a list of links among Twitter IDs, for visualisation purposes, and lists the Twitter IDs within each community. The URL of each Twitter ID is also kept, in order to associate the community detection module with other services, such as the influence scores, which are assigned on each Twitter user.

The most central nodes are “key-players” in the Twitter network of mentions and they are identified using a novel centrality measure, namely Mapping Entropy Betweenness (MEB) (Gialampoukidis et al., 2016). For example, using the daily posts related to the hashtags #home_appliances, #dishwasher, #Siemens and #LG, we get the network of Twitter users, presented in Figure 11. Each community is marked with another colour and the Twitter user who is considered the most central, according to MEB centrality, has the largest vertex size.

[1] https://github.com/MKLab-ITI/multisensor-community-detection

[2] http://igraph.org/r/

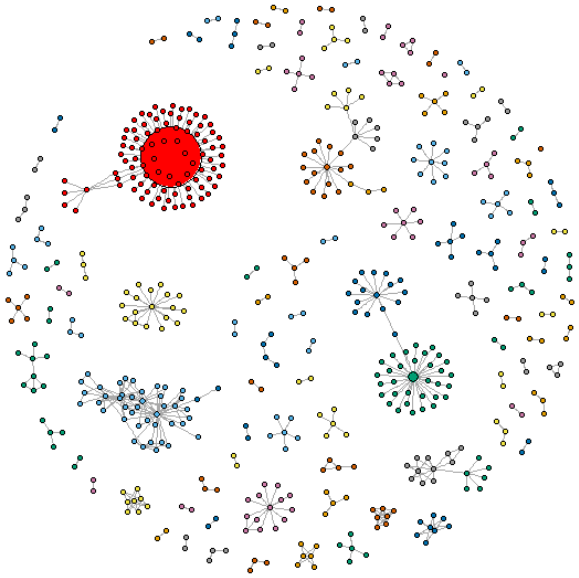

Figure 11: Network of Twitter users in UC2 - Household Appliances (6th February 2016)

2.6.1 Experiments

Experiments are divided into two categories, one for the comparison of small-scale networks and another for medium and large-scale experiments, where only Louvain and Infomap are able to provide a community detection result, due to the high computational cost of all other considered methods. We use the most prominent evaluation measures in community detection, namely Normalized Mutual Information (Danon et al., 2005) and Rand (Rand, 1971).

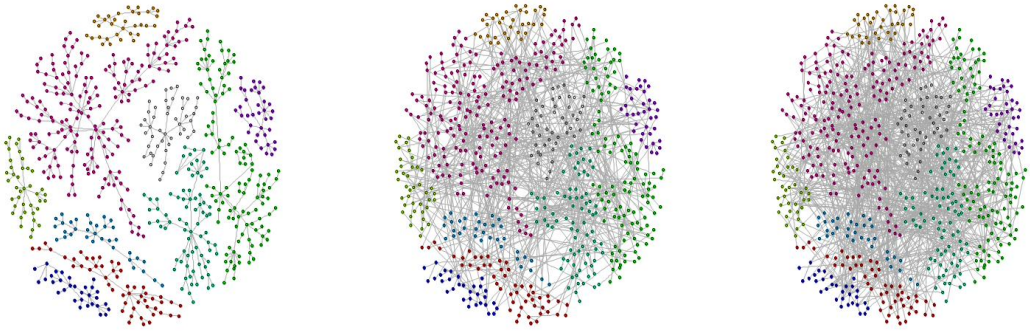

Figure 12: LFR benchmark network in three states: no mixing, moderate mixing and very mixed communities

The datasets have ground truth community structure, i.e. annotated vertices by a group they belong to. For comparison we selected several public datasets of various sizes, up to 1M nodes, which are available online[3,4]. The **Karate Club** Dataset (Zachary, 1977) is Zachary's

[3] https://networkdata.ics.uci.edu/index.php

[4] https://snap.stanford.edu/data/index.html

karate club social network of friendships between 34 members of a karate club at a US university in the 1970, in which there was a conflict that led to the split of the club into communities. The **American College Football** Dataset is the network of 115 American football games during regular season Fall 2000, as presented in (Girvan and Newman, 2002).

The network has nodes indicating to which conferences they belong to, with ground-truth split into 12 communities (Atlantic Coast, Big East, Big Ten, Big Twelve, Conference USA, Independents, Mid-American, Mountain West, Pacific Ten, South-eastern, Sun Belt, Western Athletic). The **Amazon** Network Dataset describes an Amazon product network with 334,863 nodes and 925,872 edges, into 75,149 communities, the **DLBP collaboration** Network Dataset describes a DBLP collaboration network with 317,080 nodes and 1,049,866 edges, into 13,477 communities, and the **YouTube** Network Dataset describes a part of YouTube's online social network with 1,134,890 nodes and 2,987,624 edges, into 8,385 communities (Yang and Leskovec, 2015). Moreover, we have constructed 2 benchmark networks, according to the LFR framework for evaluating community detection algorithms (Lancichinetti et al., 2008), namely **LRF benchmark650** and **LRF benchmark21K**, in order to test the algorithms in their robustness to identifying very mixed communities (Figure 12). The **LRF benchmark650** has 650 vertices with ground-truth split into 10 communities, minimum community size 20, community size power-law fit beta=1.89 (p-value=0.16>0.05), degree distribution power-law fit gamma = 3.54 (p-value=0.29>0.05) and maximum degree = 13. The **LRF benchmark21K** has 21,226 vertices, 200 communities, minimum community size 10, community size power-law fit beta=2.00 (p-value=0.70>0.05), degree distribution power-law fit gamma = 3.33 (p-value=0.13>0.05) and maximum degree = 52.

Dataset / Method	Karate Club		American College Football		LFR benchmark[5] 650	
	NMI	**Rand**	**NMI**	**Rand**	**NMI**	**Rand**
Edge Betweenness	0.5178	0.6844	0.8788	0.9632	0.2517	0.8317
Fast Greedy	**0.8255**	**0.9020**	0.6977	0.8807	0.2519	0.8319
Label Propagation	0.7071	**0.9020**	0.8792	0.9755	0.3298	0.8439
Louvain	0.6872	0.7736	0.8903	0.9688	0.2502	0.8320
Walktrap	0.6956	0.7861	0.8874	0.9705	0.2986	0.8241
Infomap	**0.8255**	**0.9020**	**0.9242**	**0.9847**	**0.3317**	**0.8449**

Table 3: Evaluation in small-scale networks

[5] Evaluation measure averaged with respect to the mixing parameter $\mu \in [0,1]$

Dataset (size)	LFR benchmark (21,226)		Amazon (334,863)		DBLB collaboration (317,080)		Youtube (1,157,828)	
Method	**NMI**	**Rand**	**NMI**	**Rand**	**NMI**	**Rand**	**NMI**	**Rand**
Louvain	0.2373	0.8100	0.5089	0.9796	0.4083	0.9874	0.6488	0.9618
Infomap	**0.3360**	**0.8139**	**0.5970**	**0.9820**	**0.5721**	**0.9947**	**0.7628**	**0.9937**

Table 4: Evaluation in medium and large-scale networks

The baseline methods we have selected are the most prominent community detection approaches, namely the Edge Betweenness of Girvan and Newman (Newman and Girvan, 2004), Fast Greedy (Clauset et al., 2004), Label Propagation (Raghavan et al., 2007), Walktrap (Pons and Latapy, 2006) and Louvain (Blondel et al., 2008). The results for the small- and large-scale experiments are reported in Table 3 and Table 4, respectively. The comparison in large-scale experiments is shown is presented only for the Louvain and the Infomap method, due to the high computational cost of all other methods considered.

The superiority of Infomap is shown in all considered networks, where we observe an improvement of Normalized Mutual Information and Rand index up to 16% and 13%, respectively, when compared to the Louvain community detection method, a fact that shows that we have achieved our goals (presented in D1.2) to the highest expectation.

Infomap shows great robustness, when compared to the other methods, in merging communities together by increasing the mixing parameter μ, as demonstrated in Figure 13 and Figure 14. In particular, in Figure 13 it is shown that Infomap competes with Label Propagation, in terms of the best performance, for medium and large mixing values, in the LFR benchmark 650. However, when NMI and Rand are averaged over all values of the mixing parameter $\mu \in [0,1]$, Infomap outperforms the Label Propagation method in both NMI and Rand scores. In Figure 14, we observe similar behaviour even for a significantly larger dataset (LFR benchmark 21K), verifying the superiority of Infomap in all cases examined.

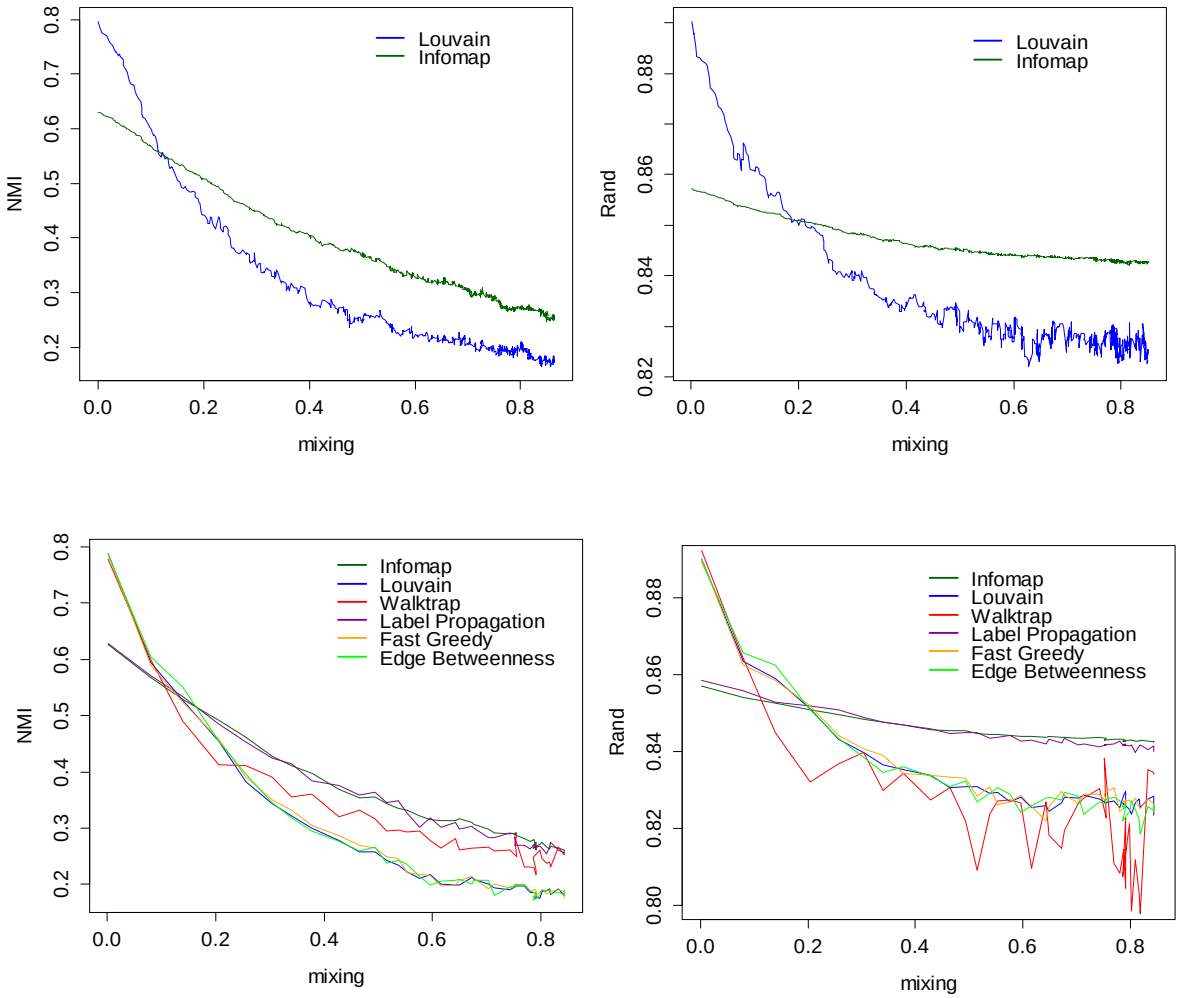

Figure 13: The performance of the LFR benchmark 650 network in various mixing states

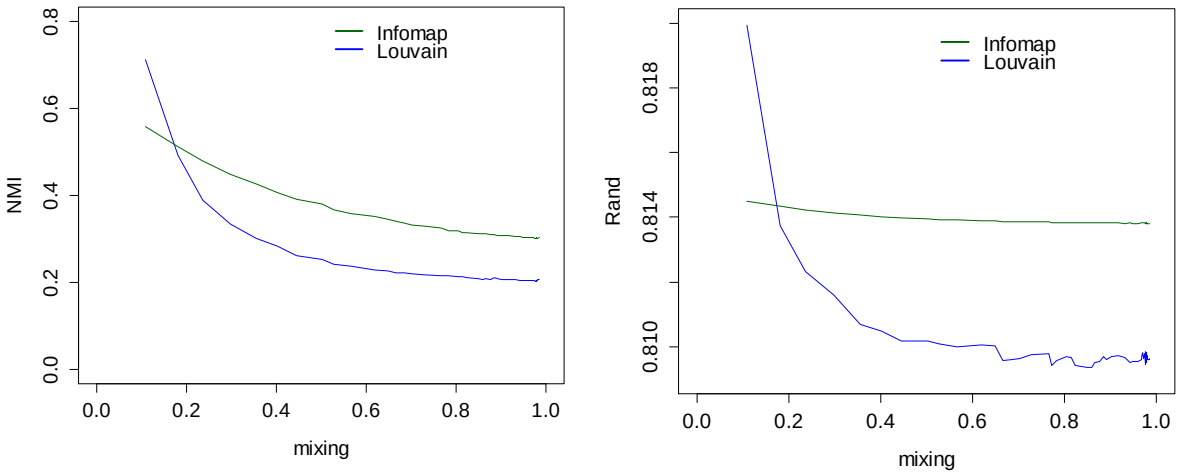

Figure 14: The performance of the LFR benchmark 21K in various mixing states

3 CONCLUSIONS

This document reported on the technological advances for task T3.4 (Information propagation and social interaction analysis). More specifically, it discussed the advanced techniques and modules developed for the above task and provided performance results. It described algorithms for analysing the network of interactions among social media users, the accurate detection of influential users, and the evolution of dynamic communities of information within the network.

The "Information propagation and social interaction analysis" module that implements several services, namely the contributor analysis, the influential user detection, and the community detection services, was evaluated. The contributor analysis service, evaluated in the First Prototype Evaluation Report (D8.3), was assessed positively for both understandability and relevance by the majority of the judges. Also, more than 70% of the judges agreed (or strongly agreed) that the information provided was easy to visualize and understand, whereas more than 50% of the judges reported that the provided information was relevant. This approximates the "Good" assessment of the contributor analysis module that we anticipated to receive. Finally, the superiority of the proposed community detection approach, using the Infomap algorithm, was demonstrated in several networks. The improvement of NMI and Rand is up to 16% and 13%, respectively, when compared to other community detection approaches, showing that we have achieved our goals (presented in D1.2) to the highest expectation.

The code for the implementation of the "Information propagation and social interaction analysis" module is available at:

https://gitlab.bigdata.eurecat.org/ioannis.arapakis/MULTISENSOR-user-and-context-centric-content-analysis.git.

4 REFERENCES

Abowd, G. D. (1999). Software Engineering Issues for Ubiquitous Computing. In Proceedings of the 21st International Conference on Software Engineering (pp. 75–84). New York, NY, USA: ACM. http://doi.org/10.1145/302405.302454

Agirre, E., Cer, D., Diab, M., Gonzalez-agirre, A., and Guo, W. (2013). sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics.

Aho, A., Garey, M.R, and Ullman, J.D (1972). The transitive reduction of a directed graph. Siam Journal on Computing, 8(3):410–421.

Arapakis, I., Cambazoglu, B. B., and Lalmas, M. (2014). On the Feasibility of Predicting News Popularity at Cold Start. Social Informatics: 6th International Conference, SocInfo 2014, Barcelona, Spain, November 11-13, 2014. Proceedings, 290–299. http://doi.org/10.1007/978-3-319-13734-6_21

Arapakis, I., Lalmas, M., Cambazoglu, B. B., Marcos, M.-C., and Jose, J. M. (2014). User engagement in online news: Under the scope of sentiment, interest, affect, and gaze. Journal of the Association for Information Science and Technology, 65(10), 1988–2005.

Ashok, V. G., Feng, S., and Choi, Y. (2013). Success with style: Using writing style to predict the success of novels. Poetry, 580(9), 70.

Balahur, A., and Steinberger, R. (2009). Rethinking Sentiment Analysis in the News: from Theory to Practice and back. Proceeding of the 1st Workshop on Opinion Mining and Sentiment Analysis (WOMSA), 9.

Bandari, R., Asur, S., and Huberman, B. A. (2012). The Pulse of News in Social Media: Forecasting Popularity. CoRR, abs/1202.0.

Barzilay, R., and Lapata, M. (2008). Modeling local coherence: An entity-based approach. Computational Linguistics, MIT Press, 34(1), 1–34.

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society. Series B (Methodological), 289–300.

Benjamini, Y., and Hochberg, Y. (2000). On the adaptive control of the false discovery rate in multiple testing with independent statistics. Journal of Educational and Behavioral Statistics, 25(1), 60–83.

Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., and Jurafsky, D. (2004). Automatic Extraction of Opinion Propositions and their Holders. Proceedings of the Association for the Advancement of Artificial Intelligence Spring Symposium on Exploring Attitude and Affect in Text (AAAI), 22–24.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. Journal of Machine Learning Research, JMLR.org, 3, 993–1022.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In Proceedings of the 45th Annual

Meeting of the Association of Computational Linguistics (pp. 440–447). Prague, Czech Republic: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P/P07/P07-1056

Blondel, V. D., Guillaume, J. L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10), P10008.

Bohlin, L., Edler, D., Lancichinetti, A., and Rosvall, M. (2014). Community detection and visualization of networks with the map equation framework. In Measuring Scholarly Impact (pp. 3-34). Springer International Publishing.

Bouayad-Agha, N., Casamayor, G., Mille, S., and Wanner, L. (2012). Perspective-oriented generation of football match summaries: Old tasks, new challenges. ACM Transactions on Speech and Language Processing (TSLP), 9(2), 3.

Chen, L., Wang, W., Nagarajan, M., Wang, S., and Sheth, A. P. (2012). Extracting Diverse Sentiment Expressions with Target-Dependent Polarity from Twitter. In Proceedings of the 6th International Association for the Advancement of Artificial Intelligence Conference on Weblogs and Social Media (AAAI/ICWSM).

Chierichetti, F., Kumar, R. and Vassilvitskii, S. (2010). Finding the Jaccard median. Proceedings of the ACM-SIAM Conference on Discrete Algorithims (SODA).

Clauset, A., Newman, M.E. and Moore, C., (2004). Finding community structure in very large networks. Physical review E, 70(6), p.066111.

Cover, T. M., and Thomas, J. A. (2012). Elements of information theory. John Wiley & Sons.

Dai, W., Xue, G., Yang, Q., and Yu, Y. (2007). Transferring naive bayes classifiers for text classification. In Proceedings of the 22nd Conference on Artificial Intelligence (AAAI), 540–545.

Dang, H. T. (2005). Overview of DUC 2005. Proceedings of the Document Understanding Conference, 2005, 1–12.

Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. (2005). Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment, 2005(09), P09008.

Feng, L., Jansche, M., Huenerfauth, M., and Elhadad, N. (2010). A Comparison of Features for Automatic Readability Assessment. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters (COLING) (pp. 276–284). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1944566.1944598

Flesch, R. F. (1979). How to write plain English: A book for lawyers and consumers.

Fortunato, S. (2010). Community detection in graphs. Physics Reports, 486(3), 75-174.

François, T., and Fairon, C. (2012). An AI readability formula for French as a foreign language. Proceedings of the Conference on Empirical Methods in Natural Language (EMNLP), 466–477.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1), 1.

Gamon, M. (2006). Graph-based Text Representation for Novelty Detection. Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, 17–24. Retrieved from http://dl.acm.org/citation.cfm?id=1654758.1654762

Gao, J., Pantel, P., Gamon, M., He, X., Deng, L., and Shen, Y. (2014). Modeling interestingness with deep neural networks. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Gialampoukidis, I., Kalpakis, G., Tsikrika, T., Vrochidis, S. and Kompatsiaris, I. (2016). Key player identification in terrorism-related social media networks using centrality measures, European Intelligence and Security Informatics Conference (EISIC 2016), August 17-19, Uppsala, Sweden (accepted for publication)

Gildea, D., and Jurafsky, D. (2002). Automatic Labeling of Semantic Roles. Computational Linguistics, MIT Press Linguistics, 28(3), 245–288.

Girvan, M., and Newman, M. E. (2002). Community structure in social and biological networks. Proceedings of the national academy of sciences, 99(12), 7821-7826.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In Proceedings of the Twenty-Eight International Conference on Machine Learning (ICML).

Godbole, N., Srinivasaiah, M., and Skiena, S. (2007). Large-Scale Sentiment Analysis for News and Blogs. International Conference on Weblogs and Social Media (ICWSM), 7, 21–24.

Goker, A., and Davies, J. (2009). Information retrieval: Searching in the 21st century, John Wiley.

Goyal, A., Bonchi, F. and Lakshmanan, L.V (2010). Learning influence probabilities in social networks. Proceedings of WSDM.

Grafstein, A., and Bailin, A. (2001). The linguistic assumptions underlying readability formulae. Language and Communication, Pergamon, 21(3), 285–301.

Gunning, R. (1952). The Technique of Clear Writing. McGraw-Hill.

Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., ... and Samatova, N. (2014). Community detection in large-scale networks: a survey and empirical evaluation. Wiley Interdisciplinary Reviews: Computational Statistics, 6(6), 426-439.

Hatzivassiloglou, V., and Wiebe, J. M. (2000). Effects of adjective orientation and gradability on sentence subjectivity. Proceedings of the 18th Conference on Computational Linguistics (COLING), 1, 299–305.

Jo, Y., and Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM), 815–824.

Kempe, D., Kleinberg, J. and Tardos, E (2003). Maximizing the spread of influence through a Social Network. Proceedings of the SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).

Klein, D., and Manning, C. D. (2003). Fast Exact Inference with a Factored Model for Natural Language Parsing. In Advances in Neural Information Processing Systems 15 (NIPS), 3–10.

Kohlschütter, C., Fankhauser, P., and Nejdl, W. (2010). Boilerplate detection using shallow text features. Proceedings of the Third ACM International Conference on Web Search and Data Mining, 441–450.

Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. Physical review E, 78(4), 046110.

Liu, B. (2010). Sentiment analysis and subjectivity. Handbook of Natural Language Processing, CRC Press, Taylor and Francis Group.

Liu, B. (2012). Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, Morgan and Claypool Publishers, 1–167.

Louis, A., and Nenkova, A. (2011). Text Specificity and Impact on Quality of News Summaries. Proceedings of the Workshop on Monolingual Text-To-Text Generation, 34–42. Retrieved from http://dl.acm.org/citation.cfm?id=2107679.2107684

Louis, A., and Nenkova, A. (2013). A corpus of science journalism for analyzing writing quality. Dialogue and Discourse, 4(2), 87–117.

Louis, A., and Nenkova, A. (2014). Verbose, Laconic or Just Right: A Simple Computational Model of Content Appropriateness under Length Constraints. EACL, 636.

Malliaros, F. D., and Vazirgiannis, M. (2013). Clustering and community detection in directed networks: A survey. Physics Reports, 533(4), 95-142.

McLaughlin, G. H. (1969). SMOG grading: A new readability formula. Journal of Reading, JSTOR, 12(8), 639–646.

McNamara, D. S., Crossley, S. A., and McCarthy, P. M. (2009). Linguistic features of writing quality. Written Communication.

Meena, A., and Prabhakar, T. V. (2007). Sentence Level Sentiment Analysis in the Presence of Conjuncts Using Linguistic Analysis. Proceedings of the 29th European Conference on Advances in Information Retrieval (ECIR), 4425, 573–580.

Mehmood, Y., Bonchi, F. and Garcia-Soriano, D. (2016). Spheres of Influence for More Effective Viral Marketing. Proceedings of the 2016 ACM SIGMOD Conference.

Mei, Q. (2010). Contextual text mining, University.

Mihalcea, R., and Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, 233–242.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems (NIPS), 3111–3119.

Moghaddam, S., and Ester, M. (2012). On the Design of LDA Models for Aspect-based Opinion Mining. Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM), 803–812.

Nenkova, A., Chae, J., Louis, A., and Pitler, E. (2010). Structural Features for Predicting the Linguistic Quality of Text. Proceddings of the Empirical Methods in Natural Language Generation (EMNLP), 5790, 222–241. http://doi.org/10.1007/978-3-642-15573-4_12

Newman, M. E., and Girvan, M. (2004). Finding and evaluating community structure in networks. Physical review E, 69(2), 026113.

Okazaki, N., Matsuo, Y., and Ishizuka, M. (2005). Improving chronological ordering of sentences extracted from multiple newspaper articles. ACM Transactions on Asian Language Information Processing (TALIP), 4(3), 321–339.

Papadopoulos, S., Kompatsiaris, Y., Vakali, A., and Spyridonos, P. (2012). Community detection in social media. Data Mining and Knowledge Discovery, 24(3), 515-554.

Pavlick, E., and Tetreault, J. (2016). An Empirical Analysis of Formality in Online Communication. Transactions of the Association for Computational Linguistics, 4, 61–74.

Peng, W., and Park, D. H. (2011). Generate Adjective Sentiment Dictionary for Social Media Sentiment Analysis Using Constrained Nonnegative Matrix Factorization. In Proceedings of the 6th International Association for the Advancement of Artificial Intelligence Conference on Weblogs and Social Media (AAAI/ICWSM).

Peterson, K., Hohensee, M., and Xia, F. (2011). Email formality in the workplace: A case study on the Enron corpus. Proceedings of the Workshop on Languages in Social Media, 86–95.

Phelan, O., McCarthy, K., and Smyth, B. (2009). Using Twitter to Recommend Real-time Topical News. Proceedings of the Third ACM Conference on Recommender Systems, 385–388. http://doi.org/10.1145/1639714.1639794

Pitler, E., and Nenkova, A. (2008). Revisiting Readability: A Unified Framework for Predicting Text Quality. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 186–195.

Pons, P., and Latapy, M. (2006). Computing communities in large networks using random walks. J. Graph Algorithms Appl., 10(2), 191-218.

Poria, S., Gelbukh, A., Hussain, A., Das, D., Bandyopadhyay, S., and S, F. (2013). Enhanced SenticNet with Affective Labels for Concept-Based Opinion Mining. Journal of Intelligent Systems, IEEE, 28(2), 31–38.

Raghavan, U. N., Albert, R., and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. Physical Review E, 76(3), 036106.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association, 66(336), 846-850.

Redish, J. C., and Selzer, J. (1985). The place of readability formulas in technical communication. Technical Communication, 32(4), 46–52.

Richards, J. C., and Schmidt, R. W. (2013). Longman dictionary of language teaching and applied linguistics. Longman London, 78.

Rosvall, M., and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences, 105(4), 1118-1123.

Rosvall, M., Axelsson, D., and Bergstrom, C. T. (2010). The map equation. The European Physical Journal Special Topics, 178(1), 13-23.

Saito, K., Nakano, R. and Kimura, M (2008). Prediction of information diffusion probabilities for independent cascade model. Proceedings of KES.

Scheible, C., and Schütze, H. (2012). Bootstrapping Sentiment Labels For Unannotated Documents With Polarity PageRank. Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), 1230–1234.

Schriver, K. A. (1989). Evaluating text quality: The continuum from text-focused to reader-focused methods. Professional Communication, IEEE Transactions on, 32(4), 238–255.

Seki, Y., Eguchi, K., Kando, N., and Aono, M. (2006). Opinion-focused summarization and its analysis at DUC 2006. Proceedings of the Document Understanding Conference (DUC), 122–130.

Shtok, A., Dror, G., Maarek, Y., and Szpektor, I. (2012). Learning from the past: answering new questions with past answers. Proceedings of the 21st International Conference on World Wide Web, 759–768.

Surdeanu, M., Ciaramita, M., and Zaragoza, H. (2008). Learning to Rank Answers on Large Online QA Collections. ACL, 8, 719–727.

Tang, R., Ng, K. B., Strzalkowski, T., and Kantor, P. B. (2003). Automatically Predicting Information Quality in News Documents. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL 2003--Short Papers - Volume 2, 97–99.

Turney, P., and Littman, M. (2002). Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus. Tecnical Report ERC-1094, National Research Council of Canada, (ERB-1094), 11.

Valiant, L.G (1979). The complexity of enumeration and reliability problems. SIAM Journal on Computing, 8(3):410–421.

Van Dijk, T.A., and Kintsch, W. (1983). Strategies of discourse comprehension (pp. 11-12). New York: Academic Press.

Watts, D.J., Peretti, J., and Frumin, M. (2007). Viral marketing for the real world. Harvard Business School Pub.

Wiebe, J., and Riloff, E. (2005). Creating subjective and objective sentence classifiers from unannotated texts. Proceedimgs International Conference Computational Linguistics and Intelligent Text Processing (CICLing), 486–497.

Wilson, T., Wiebe, J., and Hoffmann, P. (2009). Recognizing Contextual Polarity: An Exploration of Features for Phrase-level Sentiment Analysis. Journal Computational Linguistics, 35(3), 399–433.

Xia, R., Zong, C., Hu, X., and Cambria, E. (2013). Feature Ensemble Plus Sample Selection: Domain Adaptation for Sentiment Classification. Intelligent Systems, IEEE, 28(3), 10–18.

Yan, X., Song, D., and Li, X. (2006). Concept-based document readability in domain specific information retrieval. Proceedings of the 15th ACM Conference on Information and Knowledge Management (CIKM), 540–549.

Yang, J., and Leskovec, J. (2015). Defining and evaluating network communities based on ground-truth. Knowledge and Information Systems, 42(1), 181-213.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. Journal of anthropological research, 452-473

Zamanian, M., and Heydari, P. (2012). Readability of texts: State of the art. Theory and Practice in Language Studies, 2(1), 43–53.