Diploma Thesis

for the topic

Measuring topic diffusion in blog communities

---

submitted at the	Faculty of Engineering of the University of Rostock
presented by:	Marc Wessely born at 14.01.1980 in Teterow marc.wessely@uni-rostock.de
Matrikel-No.:	200984
Diploma Studies of:	Computer Science
Period of Work:	6 Months
Censor:	Prof. Dr. rer. nat. PD Clemens H. Cap, University Rostock
Mentor:	Dipl. Inf. Martin Garbe, Universität Rostock
Chair:	Information - and Communication Services

Rostock, September, 30th 2009

Abstract

As companies hold competitive advantages by innovations nowadays only for decreasing periods of time, the importance of their reputation is growing. Therefore discussions of new topics which could harm or leverage the reputation, so called issues have to be observed, identified and analyzed before they reach a critical mass audience.

Weblogs as socially connected media channels offer possibilities to observe networks of individuals, as they link to eachother by means of citations and blog-to-blog links. By analyzing these linking structures one can discover communities in which certain topics are discussed by publishing, citating and commenting articles. This flow of information is also known as diffusion, a process described in the following.

The main objective of this work is to compare and evaluate different approaches for modelling and measuring topic diffusion within blog communities. Modelling the diffusion process should allow prediction of diffusion cascades and diffusion rates. Companies can decide based on that information how to deal with issues, using detected opinion leaders as strategic communication partners. The compared models have to be implemented and applied within discovered communities. The results in form of diffusion cascade estimations will be compared to real diffusion cascades in the dataset.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Economic success of companies depends on competitive advantages by innovations. In times of globalization and growing dynamics of markets this advance is hold only for constantly decreasing periods of time. That's why reputation, the public opinion regarding a company, becomes more important established by trust and creditability towards a company's stakeholders as e.g. customers or business partners.

1.1 Reputation

There are two kinds of reputation, functional and social reputation. While functional reputation is an indicator for the compentency and success of a company, social reputation indicates its moral integrity in the society [Mar05]. Competency is the cognitive expectation of the public regarding a companies performances in brands, products and services. Integrity means to fulfill normative social ethnic expectations. Both types of reputation evaluate the level of adjustment of the companies market activities to the expectations of existing or potential stakeholders. On the other hand a company tries to define its identity, to cultivate an unique profile. The secret of a positive reputation is based on the balance between adjustment embodied in expectation management and demarcation through identity management [Mar05]. Consequently companies look for mechanisms, tools and strategies for directly or indirectly influencing the public opinion according to its own interests. The classical strategy for influencing attitudes by communication is marketing, in detail communication political instruments as advertisement, public relation campaigns or sponsoring. These aim to convince potential customers and other stakeholder groups that the particular company has a high competency in their products, fits the common social ethnic values and provides services with an unique customer value.

Also other groups of interest try to influence the public, which is permanently influenced by mass medias as television, print media and with growing importance the internet. For ensuring its positive reputation, companies want information about:

- groups of interest they dont know
- expecially new medias they are not experienced enough with
- relevant events of communication they dont have in their field of vision

1.2 Issuetracking

Events of communication in selected media channels, so called issues which:

- have a positive or negative effect on the reputation
- are of public interest
- and construct relations to a companies stakeholders

have to be observed. These issues have to be identified and analyzed before they reach a critical mass audience. The tool which deals with early detection of issues and evaluates them according to their potential of influence on reputation is issue monitoring. Issuemonitoring tries to get the complete picture of all groups of interest which matter to the public, related to the company. The risks and dynamics on reputation which result from the development and expansion of such issues are the subject of issue tracking. Issuetracking analyzes mechanisms of development and expansion of public awareness regarding a certain issue in time, by following an issues lifecycle. Public awareness of an issue spreads through a social system over time as a social process of influence on adoption of communication behaviours. This process is also known as diffusion in which certain individuals act as opinion leaders to spread issues to the rest of the social system. [RR03]

This thesis takes a look at issue diffusion modelling in the blogosphere based on an experiment suited for the Volkswagen AG. By monitoring selected blog communities which produce content which relates to VW, issues are detected as groups of blog articles dealing with similar contents. The temporal and social structures in these groups are analyzed to model and predict issue lifecycles, to support a corporate issuetracking.

1.3 Issuetracking in the Blogosphere

The internet provides contents which are less costly to monitor, once certain software is available. For monitoring issues especially news portals and weblogs are useful sources, as they provide novel information. While news portals representing mainly the print media in the internet, weblogs represent individuals or communities of bloggers who describe and comment current events in a mainly opinion oriented manner.

1.3.1 Possibilities

Weblogs as socially connected media channels offer also possibilities to observe networks of individuals, as they link to eachother by means of citations and blog-to-blog links. By analyzing these linking structures one can discover communities in which certain topics are discussed by publishing, citating and commenting articles. Those topics which are relevant for a particular company to take notice of, are in the focus of this thesis. Tracking these issues in the blogosphere, especially in certain blog communities opens up new possibilities for a corporate issuetracking as:

- monitoring a large set of unknown media channels (36.4 million active blogs)
- analyzing diffusions in a highly growing social system (doubles every 5.5 Months)

1.3.2 Challenges

On the other hand there are also challenges, which have to be considered as:

- anonymity preferred in the internet
- global publishing and reach
- heterogenous topic landscapes
- spontanous and alternating patterns of diffusion
- communities of interest instead of localized networks

Modelling and measuring issue diffusion aims to determine the potential of an issue to become interesting for the public opinion. Every blog participating in this process has a certain influence on the issue diffusion in its community. Measuring this influence, by reference numbers of innovativeness and social status, allows detecting topic specific opinion leaders in blog communities. Companies can decide based on that information how to deal with those issues, using detected opinion leaders as strategic communication partners to spread information to improve or stabilize their reputation.

1.4 Aims and Objectives

The main objective of this thesis is to develop a general applicable methodology for detection, modelling and measuring topic diffusion within blog communities. That methodology should allow prediction of diffusion cascades and diffusion rates independent from data sources and application scenarios. It will be applied to a dataset of selected blogs in the topic circumference of the Volkswagen AG, evaluating its impact to support a corporate issuetracking.

In detail two basically different models, a network and a statistical model, are compared regarding their ability to map real diffusion processes. These will be analyzed for distinct characteristcs such as network effects or termporal interdependencies. The compared models have to be implemented and applied within discovered communities. A key question is therefore how to localize a diffusion network for creating a representative dataset. Furthermore the optimal training data has to be examined from observed topic diffusions. The results in form of diffusion cascade estimations will be compared to real diffusion cascades in the dataset, to evaluate their level of convergence. Detecting topic specific opinion leaders by measurements of innovativeness and centrality is thereby a sub objective and a by product of the discussed models.

Finally the the results from evaluation will be summarized, considering the ability of the developed methodology to fit best the requirements of a corporate issuetracking system for the blogosphere. Based on that conclusions a proposal is given for improving such a system by better data management and a greater variety of applied models for more exact diffusion predictions in blog communities respective to its known characteristica.

1.5 Structure

This thesis is separated in a couple of chapters whose content will be described now in the following. Chapters 2 and 3 are containing basics and state of the art approaches for information flow modeling and machine learning. Chapter 4 deals with the conceptional design, and chapter 5 with the implementation of the framework for comparing models of topic diffusion. Chapter 6 concentrates on the evaluation of the results. Finally Chapter 7 gives a summary and an outlook for further development. Additionally there is an appendix containing the used data model, implementation details and an declaration of authenticity of contents.

Chapter 1, Introduction: This Chapter includes setting the economic context of the thesis, motivation for dealing with topics as corporate reputation related issues and the thesis aims and objectives.

Chapter 2, Machine Learning Approaches: This Chapter introduces basic concepts of several learning approaches for classification of data items into predefined or unknown categories.

Chapter 3, Diffusion Modelling: In this Chapter sociological insights on diffusion of new ideas and its elements are explained. Especially a focal view on network effects of diffusion is done to point out the relevant influence factors. Basic concepts of network and dynamic modells are described, which are neccessary for understanding the concept chapter. Moreover recent approaches for diffusion modelling in the blogosphere are presented and discussed to point out the positioning of this thesis in the research landscape.

Chapter 4, Conceptional Design: Includes the concept for measuring topic diffusion in blog communities. It is seperated into two main parts: topic detection and tracking, and diffusion modelling and measuring. The topic tracking section deals with data aquisition, data preparation and topic tracking. The diffusion modelling section deals with model selection, model fitting and evaluation of diffusion estimations. The content of all of these subsections is structured into requirements, inputs, outputs, decision criterias and used methods.

Chapter 5, Prototype Implementation: Contains the implementation details of the framework used for data aquisition, topic tracking, diffusion modelling and evaluation. Thereby the architecture of the framework will be described with special focus on the developed components which are embedded in multi-tier architecture layers.

Chapter 6, Evaluation: This Chapter will present the results from the application of different topic tracking methods on the dataset, and from the comparison of diffusion rate estimations of the applied diffusion models. Different measurements for model fitting and evaluation will be discussed and validated.

Chapter 7, Conclusions and Future Work: Summarizes the results from evaluation, considering the ability of the different analyzed diffusion models to map real diffusion processes. Based on that conclusions a proposal is given for future development of a more accurate framework allowing more exact diffusion predictions

Chapter 2

Machine Learning Approaches

Topic Detection and Tracking is a subject from the field of textmining. Topic Detection deals with detection of groups of items in a large dataset, so called clusters which belong together according to similar or same attributes. These attributes are called features, they describe significantly the content or matter of the items. As this thesis deals with detection of topics in a large dataset of blog articles, features are those words in the articles contents which describe their semantic best. A topic is hereby a concept which groups similar articles according to features which are occuring in their contents in similar quantities. Once those topic clusters are detected from articles which have been published in the beginning, articles which have been published in the following can be assigned if they match the required similarity. Topic Tracking is this linear process of assigment of articles to topic clusters by classification based on content similarity.

Classification as the opposite approach to clustering aims to assign items to allready defined classes. Here classification is used to decide to which topic cluster an article should be assigned, that decision can also result in a new cluster. There mainly four different approaches for classification existing which will be tested regarding their performance for this experiment and explained therefore in the following. Each of these approaches is based on a certain model which needs to be trained with the so called "'Looking Back Window"', the list of allready defined topic clusters with their assigned articles.

2.1 Instance based Learning

Instance based algorithms like k-nearest Neighbors try to locate fitting clusters by analyzing the documents most similar to the classified one. If e.g. the five nearest documents are taken into account, the method prefers the cluster for classification where the major-

ity is associated to. So if two are in one cluster and the rest in dinstinct other clusters the article will be classified to this topic.

2.2 Centroid based Learning

Centroid oriented approaches as e.g. Roccio build an average feature vector from all feature vectors of articles contained in the topic cluster. This centroid is then compared to the document feature vector rearding its similarity using the cosine similarity measurement. The topic cluster whose centroid has the highest similarity with the articles feature vector is classified to fit in the cluster. The minimal condition is that this similarity exceeds a predefined threshold.

2.3 Bayesian Learning

An article can be assigned to a cluster according to a propability to fit to this cluster. Naive Bayes as an example algorithm for this approach, calculates conditional propabilities out of the normalized term frequencies of the topic and document feature vectors. The topic which gets the highest propability will be chosen as the documents topic.

2.4 Decision Tree Learning

Chapter 3

Diffusion Modelling

Diffusion theory attempts to explain how new ideas and practises spread within and between communities. According to Rogers [RR03], diffusion is the process by which a new idea is communicated through certain channels over time among the members of a social system. Empirical research confirmed that new ideas spread through interpersonal contacts largely consisting of interpersonal communication [RG43, Val95]. Most diffusion studies focus on understanding the factors which lead members of a communitiy to adopt a new idea while others do not. Further, selected studies try to make out why some people adopt behaviour early, whereas others wait a certain amount of time before accepting the new practise.

This work concentrates on measuring the diffusion of topics in blog communities, which is influenced by the ability of a blogger to persuade another blogger: to write a new article in a given topic, to citate an originating blog article, to link to the originating blog by means of trackbacks, comments or blogroll links [JKFO06]. By modelling the information flow as adoption behaviours in time [SCHT07], the measurement of success is the rate of adoption. That means the amount of blogs which adopt the new topic by topic related writing or linking behaviours within an observed period of time.

3.1 Diffusion Networks

Diffusion networks have been studied in by several researchers [RR03] [Val95]. It has been observed that blogs form communities by their linking behaviour. The membership of a blog in a community is determined by the probability to link more to blogs within the community than outside [WFI94]. Within these communities blogs tend to preferentially link to blogs which are more authorative and deal with similar topics [RR03].

By analysing these preferential attachment occuring by link proximity the community of a blog can be discovered. The measurement for link proximity is modularity, which has to be greedily optimized to decide if a blog belongs to a community or not [CNM04]. As blogs are more likely to be influenced to topics by blogs that deal with similar topics and have influenced them before, the diffusion of new topics happens more likely within communities due to their topical homogenity [RR03].

3.2 Rate of Adoption

Within diffusion networks certain individuals are inevitely more open to adoption than others. Blogs can be classified regarding to their adoption behaviour. Rogers found five adopter categories: innovators, early adopters, early majority, late majority and laggards [RR03]. Moore [Moo02] stated that there is a chasm beween the early adopters and the early majority in terms of a time lag until they adopt. This majority has a higher treshold in terms of adoption willingness. This treshold is related to their adoption decision process [SCHT07].

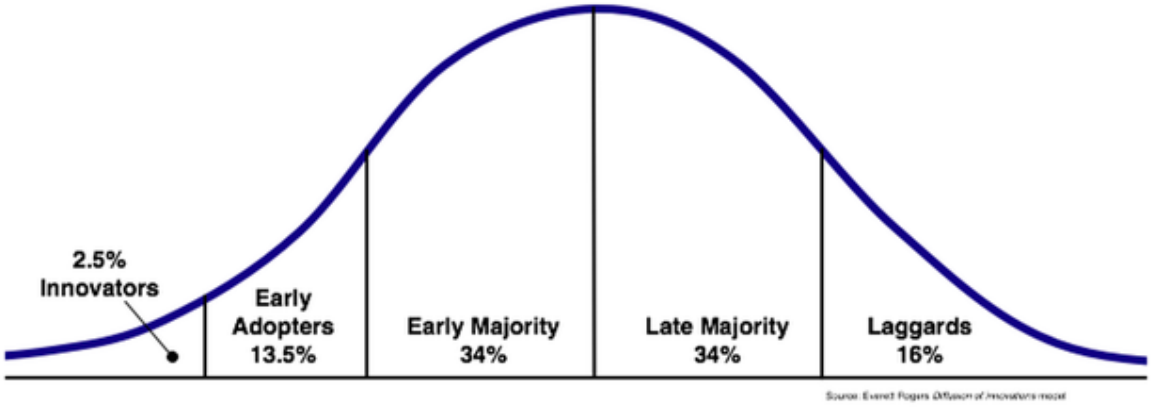

Figure 3.1: Adoption Lifecycle

In blog communities there are two different groups related to their adoption decision: leaders and followers. Leaders are denoted to be highly innovative and authorative. They relate mainly to information outside of the community such as other networks or mass medias, contribute novel information [ZXJ08] and influence others in adopting their opinions. Followers in contrast relate to information mainly to their network peers [RR03] to overcome uncertainty in decision making. That implies that they observe their network peers for a longer time until they decide to adopt.

3.3 Applied Models

The following two subsections will describe the two models which have been applied in the discussed experiment of this thesis. The first is a relational network model and the second is a dynamic model based on continues-time markov chains.

3.3.1 Linear Threshold Model

Diffusion network models aim to mathematically simulate the spread of information by weighting the influence. There are two kinds of diffusion network models: relational and structural network models. Relational network models concentrate on the direct and indirect one individual has in its network to explain patterns of adoption behaviours. Structural network models look at the positions individuals have in their network and how individuals with same positional attributes influence eachother in adoption.

The Linear Threshold Model [Val95], as a relational network model takes a look at diffusion as a social process by modelling the adoption behaviours of blogs in relation to those of their direct neighbors in the network. The network exposure is the fraction of a blogs personal network, the direct linked blogs, that have allready adopted a certain topic at a determined point of time. That approach is based on insights from sociology that there are two different types of individuals in social networks related to adoption behaviour.

- Opinion leaders which are innovative, adopt new ideas early and have a high social status.
- Followers, the majority of individuals imitating the adoption behaviour of others by observing their personal network.

Opinion leadership is a concept rooting in sociology based upon the two-step-flow-hypothesis, firstly introduced by paul lazarsfeld et. al. [PFL44][p. 151ff]. That means basically that information spread in social networks from highly influential individuals to the rest of the social network. These opinion leaders are very innovative and have a high social status in their community. That means they are the first adopting an idea. They gather their information and are influenced mainly through other medias or individuals outside the network.

In opposite opinion followers are influenced by those leaders. They are imitators, that means that they observe their network peers for their adoption decision, and have a less social status. followers make adoption decisions due to network or contagion effects. Contagion means that the adoption decision of an individual is based on the observed network exposure. Each individual has a threshold regarding this exposure, the fraction of adopters to direct neighbors that need to exist until when he is adopting. These thresholds, exposures at time of adoption are the core of this model.

3.3.2 Continous-Time Markov Chain

Diffusion Networks evolve over time. Dynamic factors in the web are essential and cannot be ignored. Based on the fact that bloggers and users are not only interested in the pages with high authority scores, but also those who provide recent information. But these studies ignored the diffusion rate, which means the amount of blogs adopting a new topic within a time period. Furthermore it measures how likely information flows from one blog to another within a limited time period. This likeliness can also be interpreted as preferences or thresholds of adoption related to a certain piece of information.

The markov chain based approach from based on the publication from Song et. al. [SCHT07] focusses on how efficiently the information can diffuse among the users in the network. The model is based on Continuous-Time Markov Chain (CTMC), in which both the probability (how likely) and the rate (how fast) for the information to flow from one individual to another is captured.

A Continuous-Time Markov Chain models markov processes which satisfy the Markov property and takes values from a discrete state space. This preliminary has been made as no causal dependencies of adoption behaviours can be stated. The modelled processes of adoption are mapped in the CTMC as a transition between states from a discrete state space. Therefore these processes are discrete processes in continues-time.

Markov Property

The Markov property states that at any times $t + s > t > 0$, the conditional probability distribution of the process at time $t + s$ given the whole history of the process up to and including time t, depends only on the state of the process at time t.

Markov Process

A markov process is the jump process between two states within a certain interval. Such a process is the homogenous form of a so called poisson process and has the following properties:

- Seldomness, adoptions do seldom occur simultanously
- Homogenity, adoptions occur with a constant rate per time unit
- Memorilessness, the number of adoptions within one period of time independent from the number of adoptions occuring within another period of time

As the markov process fits these properties, the intervals of delay between two state transitions are poisson distributed. This distribution can be approximated by the exponential distribution.

Transition Propabilities

The output is a matrix of transition propabilities which will be learned in the training phase of the model, based on observations of temporal interdependencies in the publishing data of a topic cluster. Using that matrix the further diffusion of a new topic can be predicted with maximum likelyhood estimation for transitions to not yet activated states. Adoption cascades will here consists of the newly activated blogs with their staying time measured in days.

3.4 Diffusion Modelling in the Blogosphere

In the last decade a couple of researchers have studied the impact of diffusion modelling in the blogosphere from different perspectives. They aim to model the spread of information using different mathematical devices, preliminaries and fulfilling different scientific and economic interests. These can be categorized in four different approaches:

- Networks Models view diffusion as information propagation through social networks
- Dynamic Models concentrate on temporal patterns of diffusions
- Epidemics Models map diffusion like the spread of infectious deseases
- Game Theoretic Models picture adoption behaviours as coordination games

3.4.1 Network Models

There are two general network approaches: threshold models and cascade models.

In the basic Linear Threshold Model proposed by Kempe et.al. [KKT03] each node becomes activated if the sum of of the weights of the active neighbors exceeds its threshold. In the Independent Cascade Model proposed by Goldberg et. al. [GLM01] every node gets a single chance to activate each of its neighbouring nodes and succeeds with a certain probability.

This probability is independent of the history of the previous adoption behaviours of its neighboring nodes. Kempe et. al. generalized that model in a later publication [KKT05]. In the General Cascade Model the assumption of independence is eleminated.

Java et. al. [JKFO06] used the basic Linear Threshold Modell but applied the algorithms to blog graphs based on links between blogs instead as Kempe et. al. to citation networks. They took a simplistic view on the blogoshere and converted the blog graph into a directed influence graph. An influence graph thereby is a weighted, directed graph with edge weights indicating how much influence a source blog has on its destination.

Starting with that graph they identified a target set of leaders so that information spreaded from that blogs causes a large number of followers to adopt the new idea.

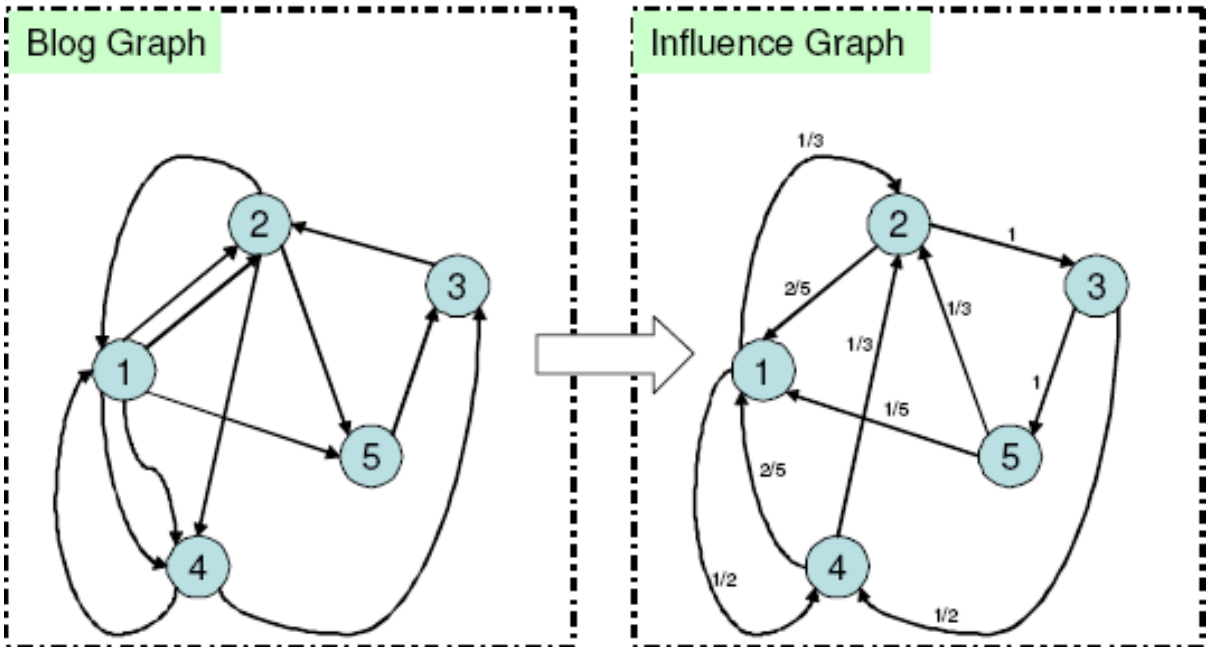

Figure 3.2: Influence Graph

They used pseudo random threshold values and the number of parallel directed edges (citations) as weights for the influence graph. Several heuristics have been used to identify the target set as PageRank, HITS and greedy search which locally maximizes the influenced node set. They also cared about the effect of spam blogs which can decrease the efficiency of the influence model to predict good target sets. Using PageRank as the heuristic for identifying the target minimizes at best the effect of spam blogs.

Independent Cascade Model In the Independent Cascade Model bloggers are connected by a directed graph where each edge is weighted with a copy probability, which represents the influence of bloggers to infect others in subsequent time steps. Whereby this probability is independent from history of other neighbours written in that topic. Additionally they introduce a readership probability for the event, that a blogger reads another blog on a given day. The process of propagation begins with a blogger reading an article in a given topic with the readership probability, so a time delay is chosen from an exponential distribution. Then with the above mentioned copy probability the blogger will choose to write about it. In that process there is only a single chance for a topic to propagate along a given edge. Using that probabilities a transmission graph is generated given a community of blogs and a timeout interval. They compute then for each topic and network peer the probability that the topic traverses the edge in between, and update the copy and readership probabilities based on the posterior probabilities above. This

procedure is done until the convergence due to the timeout interval.

3.4.2 Dynamic Models

Differently than in the Linear Threshold Model proposed by Kempe et.al. [KKT03], Song et. al. model this treshold in their as a time delay on infomation flow paths [SCHT07]. Their diffusion rate based information flow modell is based on the foundation of Continous-Time Markov Chain, which is a continous time stochstic process. They model each node in the diffusion network as a state and weight the edges between nodes with conditional transition probabilities. The threshold is modelled as the delaying time information stays in a node before making a transition to others according to the transition probabilities. Furthermore they assume this time delay to follow an exponential distribution estimated by the expected value of the observations of the staying time of each node in the network. The transitional propabilities are estimated based on the inter-state transition and the time delay. Finally they compute the inter-personal diffusion rate from both parameters of the model to generate a transition rate matrix for the network.

3.4.3 Epidemic Models

Epidemical Models take the approach of modelling information flow based on the analogy to the spread of desease in social networks [GGLNT04][p. 2]. Classical desease propagation models in epidimiology are based upon the cycles of desease in a host [Bai75].

Early studies of epidemics took place on homogenous networks, in which contacts are chosen randomly from the entire network. In models of small-networks called communities [CNM04] only a constant fraction of the entire network will be infected based on a transmission probability known as the epidemic threshold. Epidemic spread in many real networks, including networks defined by blog-to-blog links, follow power law distributions. For the blogspace many topics propagate without becoming epidemics. Refinements are to modify the transmission model by relating the probability of infection to the distance of the initiator [WHAT03] or to better model power-law networks involving clustering coefficients, the propability that two neighbors of a node are themselves neighbors [WS98].

SIR Models A person is first susceptible (S) to the desease, if then exposed to the desease by an infectious contact, the person becomes infected (I) with some propability.

The desease then stays in the host until he recovers (R). Thus SIR models deseases in which recovered host are never susceptible to the desease.

Girvan et. al presented a SIR model with mutation [GCNS02], in which a node is immune to any strain which is sufficiently close to a strain which he was previously infected. In analogy to the blogspace that could be similar topics.

Gruhl et. al. [GGLNT04] described the individual propagation of information using cascade and epidemic modells. The present the Independent Cascade Model from Goldberg et. al. [GLM01] in the SIR framework, extending it later to the SIRS framework to allow multiple postings from authors. They describe cascade modell with temporal states at which a set of nodes writes about a topic, at each successive state other nodes can be infected. This process goes on until no new articles appear for some number of time steps, referred as the timout interval.

SIS Models SIS models the situation in which a host maybe becomes susceptible again. In the blogosphere SIS might be interpreted as becoming infected from topics by reading blog posts of a friend. Other network peers might also read her blog post and become infected as well. In the blogoshere, however, many topics propagate without becoming epidemics, thats why such a model would be inappropriate. One refinement is to consider a more accurate model of power-law networks. Klemm et. al. [EK02] have demonstrated a non-zero epidemic threshold under the SIS model in power-law networks produced by a certain generative model. Leskovec et. al. presented a very simple SIS model using states for a blogs desease status [JL07].

3.4.4 Game Theoretic Approaches

The spread of information in the blogosphere can also be modelled as a game in which individuals get payoffs for adopting a particular meme or topic. That payoff can manifest in form of additional linkings according to the increased social status of the adopting blog in its community. Morris [Mor00] e.g. considers blogs as players from a particular type. This type is whether to be an adopter or a non-adopter. Each player i receives a positive payoff for each of its neighbors that has the same type as i. Morris and Young explore the question of whether adopters can "'take over"' the graph, if every blog chooses to become a non-adopter with probability increasing as the number of i's neighbors that are non-adopters increases.

The information propagation can also be based on selfish decisions by individuals to learn a particular information. In that case the individual which ones to learn has a cost by establishing a link to the one providing the information and gathering a profit through this linking behaviour. The research in this field explores properties of the social network which forms under this scenario [HS03].

3.4.5 Positioning of this Thesis

This thesis takes a look at two different diffusion modelling approaches from the economic context of issuetracking. That means for a specific corporate scenario topic diffusions should me measured to evaluate their relevance for reputation dynamics. Other works were more interested on the implications for a better focussed marketing [KKT05, KKT03, JKFO06, JKF+07, GGLNT04] or more technical interests as link prediction or link-spam detection [EFL04, NC08, JL07].

Due to the restriction of the experiment as a corporate scenario, the discovering and localizing of an enclosed community for monitoring issues was important. Related works as e.g. Meme-tracking and the Dynamics of the News Cycle [JL09] had more large scaled datasets due to their technical capacities and widely focus.

Most of the researchers in this field concentrated on one particular approach [GGLNT04, KKT05, KKT03, JKF+07, SCHT07, WS98] or tried to integrate different approaches into one model [MG09]. In contrast this thesis compares two models based on network and dynamic approaches of diffusion modelling.

The majority of network approaches like the threshold models based on Granovetter [Mar87][p. 1420-1443] or the cascade models based on Kempe et. al. [KKT05] aim to determine influential nodes in social networks to reach a maximized target set. These opinion leaders are a a by product but not the focus of the applied network model of Valente [Val95]. Its aim is to estimate the overall amount of adoptions within a predefined period of time.

The dynamic models in this field are all based on continues-time markov chains. Song et. al. [SCHT07] aims to estimate inter-personal diffusion rate which is the time needed for information to spread from one blog to another. Götz et. al. [MG09] integrate network and dynamic approaches into one model to simultanously modelling topological and temporal patterns of diffusions. Differently the applied model aims to estimate average rate of adoptions and adoption cascades using observed times of adoption.

Chapter 4

Conceptional Design

This chapter describes the conceptional design of the above mentioned general applicable methodology for measuring topic diffusion within blog communities. From aquisition of information related to company specific events of communication to measuring and prediction of topic diffusions several preprocessing and modelling steps have to be passed. Each of these steps extract more useful information for detection and evaluation of issues regarding their potential on reputation. Thats why the neccessary requirements have been stated specifically for each of these steps or levels of information processing. On each of these levels the generated information becomes the input for the subsequent level. Based on that information certain criterias are getting applied to decide which methods or models have to be chosen to ensure optimal results for the next level of knowledge creation. Finally this sequence of optimal decision should result in most precise predictions of real diffusion cascades and rates.

Therefore a complex integrative framework had to be designed and developed allowing the integration of all required processes needed for topic diffusion modelling. Using that framework an experiment was set up monitoring a community of blogs for 2 months, between begin of May and end of June 2009, in the topic circumference of the Volkswagen AG. The resulting dataset was used for analyzing the topic diffusion processes occuring within topic related groups of blog articles, to investigate the impact of the analyzed models for supporting a corporate issue tracking for that company.

The presented concept is seperated into two main parts: Tracking Topics, and Modelling and Measuring Topic Diffusion. This structure has been chosen to clearly point out that these are the two basic aspects of this thesis. Topics, more concrete issues have to be first detected and tracked before their diffusion, the development of the public awareness to that issue can be modelled, measured and finally predicted.

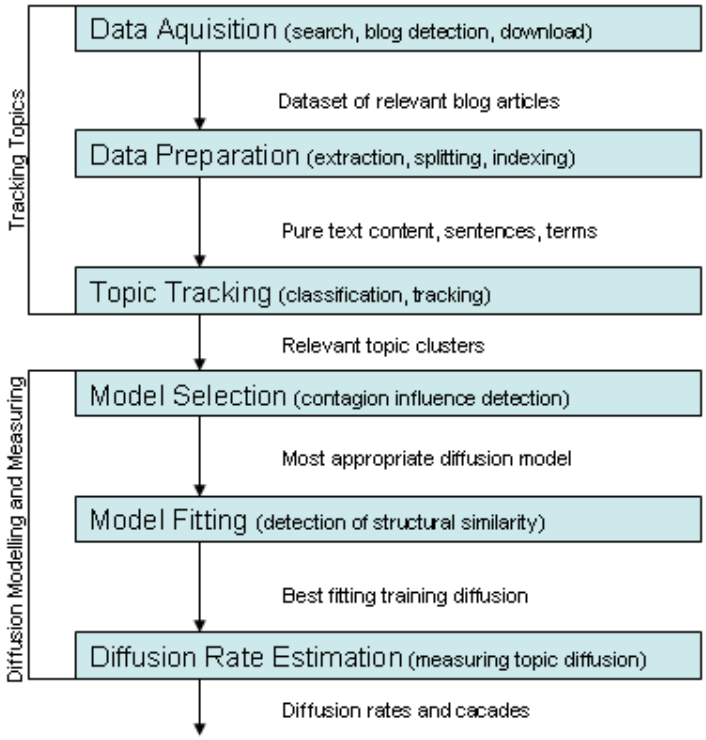

Figure 4.1: Conceptional Design

The Tracking Topics section therefore includes subsections for all steps neccessary for tracking issues as aquisition of the required dataset of blog articles, the extraction, cleaning and indexing of their contents and the clustering into groups of articles with similar contents. Modelling and Measuring Topic Diffusion encloses selection of the right model, evaluation of the best fitting training data for that model and finally the estimation of diffusion rates and cascades and their evaluation of preciseness.

4.1 Tracking Topics

Detecting and tracking topics in blog articles requires the retrieval of those articles from selected blogs, which produce relevant contents for the company, issues are tracked for. The data aquisition section will deal with the concepts and realization of this information retrieval task. The contents of the retrieved articles in form of HTML pages has to be cleaned from unneccessary elements to extract the pure contents. This pure text then has to be splitted and again cleaned from special characters to extract the most significant words of the articles. The subtasks of this content cleaning and term extraction will be explained in the data preparation section. The occuring terms will be counted, as these term frequencies are needed for the process of topic tracking, described in the topic tracking subsection.

4.1.1 Dataset Aquisition

Monitoring and tracking issues for the Volkswagen AG requires localizing a blog community dealing with relevant contents relevant for Volkswagen. Search engines keep track of those weblogs which are active in publishing articles. There are two general types of search engines: universal and specific ones. Universal search engines as Google or Yahoo monitor all kind of websites. Specific search engines restrict their monitoring on certain types of websites. Blog- and news- specific search engines are best for delivering that information. They offer the service to query recent articles matched by certain keywords, which are the input for this subtask.

The results from news- search engines had to be considered because they include many blog articles as well as blog- search engines include many results from news portals. Thats why after gathering articles, the publishing sources had to be seperated into a group of news portals and weblogs by classification. The resulting community of blogs has been monitored throughout the time period of the specified experiment. Monitoring means here the process of periodical downloading of recent blog articles. Additionally the network structure of the blog community had to be gathered and stored, by extracting links from those articles. That data was used to analyze the network density of those interconnected blogs publishing in a certain topic. This measurement will be used later for the selection of the best fitting diffusion model estimating diffusion rates.

Requirements

Selecting the required diffussion network in form of a blog community means to verify the relation of publishing blogs to the company, issues should be detected for. Such verification can happen only by matching a blogs published contents with those the company is dealing with. The following section will point out in detail how such matching has been done. The resulting community has to be enclosed so that no blogs will be added throughout the experiment and that only blogs are belonging to it.

Once this enclosed community of blogs has been aggregated, it had to be decided how these blogs have to be monitored. Every day of the monitoring period a crawling allgorithm had to check on each blogs for new published articles, which requires information where to find these articles. The location of the publishing lists of each blog containing that information had to be stored in front of the monitoring period.

The network data extracted from blog articles has to be stored that it pictures real communication patterns of the observed communities. All relevant links which appeared in the article or on the blog itself have to be used the map blogs to their network peers. The number of link occurences between blogs function hereby as a measurement of the intensity of communication between those blogs.

Inputs

Generally it can be stated that for data aquisition in issue tracking the domain of media channels and the scope of interest has to be known and fixed. This input is dependent on the scenario it is applied to. As this the underlying experiment deals with issue tracking for the Volkswagen AG it is a general corporate scenario. That means that all articles which appear in the blogosphere in the circumference of this company should be tracked. Therefore the scope of interest was set to the company name itself. The keywords describing this scope have been used to query news- and blog- specific search engines in March 2009, whose results serve as input for subsequent data aquisition steps.

Outputs

The final output of this information retrieval are the blog articles in form of HTML pages, the blog community and the network data. The articles will be stored in binary form enhanced with metadata as the articles name, url, summary, language and the publishing date which is required for analyzing the temporal structure of topic clusters. For each blog of the gathered community the name, url, local origin, date of gathering and the url of its publishing list has been stored. The network data will be stored in form of blog to blog mappings associated with counts for their linking intensity.

Criterias for Decisions

On this level four important decisions had to be made, regarding the selection of fitting search engines, the keywords describing the scope of interest, the dataformat of the publishing lists and the classificator used for weblog detection.

Search Engines The search engines where selected due to two criterias: novelity of information and amount of listed sources. Every search engine observes a certain amount of sources in the internet. Google, Alta Vista and Technorati keep track of the most sources while ensuring that their search results are recent articles. Furthermore they provide search results sorted by their publishing date or even per day. That open up the possibility to scrape those results daily from the result pages.

Chosen Keywords The chosen keywords were "'Volkswagen"' and "'VW"' to ensure that all articles dealing with topics anyhow related to Volkswagen can be queried. The publishing lists of weblogs, the so called newsfeeds are provided in several dataformats. Rss (Real Simple Syndication) has been chosen as the dataformat in the version 2.0. That is basically an xml based data format for providing information about new content updates from websites. The criteria here was its widely spread among the blogosphere, as well as the simple structure of the xml files. Using a simple parser is easy to extract the needed information listed above.

Weblog Detector The classificator used for detection of weblogs among the retrieved sources is a simple naive bayes classificator. It was chosen as its learning model requires only few training data to deliver satisfying classification results.

Used Methods

This section explains in detail how: weblogs were detected in the set of relevant sources, weblog articles were retrieved and links were extracted from these articles.

Weblog Detection Separation of weblogs from newssites was done in two steps, firstly by a weblog detector component and secondly by manual selection. As mentioned before the weblog detector was developed as a naive bayes classificator. The training data of this classificator consisted of 100 manual selected weblogs and 100 manual selected news articles. Weblog characteristic words like e.g. Blog, Tag, Pingback or Trackback were counted in contents of these articles. These number of occurences were cumulated over the amount of training documents for each feature. The resulting weights were smoothed using La-Place Smoothing with the amount of training documents and the cumulated feature counts. That mathematical operation resulted in certain baysian prior probabilities for each feature, to appear in a typical blog or news site using.

These two vectors of conditional propabilities were stored in form of a JSON encoded document. JSON (Javascript Object Notation) is a very simple, easy to parse data format. Using a JSON parser these values were extracted as input data for the naive bayes classificator. Based on that prior propabilities, baysian posterior probabilities were calculated to decide whether the classified website matches a typical news or weblog site. That was done by inference using the term frequencies of above mentioned weblog specific features and the prior probabilities calculated before. Finally all sources which were gathered via the search machines, were classified as a blog or a news site.

In the second step all news sites and their articles were removed from the dataset. The remaining weblogs were again manually analyzed to be proper classified. Each of these websites was verified to deal with event related contents in an opinion focused manner, is structured and designed typical for blogs and has links to other blogs. After postselection 680 blogs were resulting, which are the input of the second data aquisition step described in the following.

Weblog Article Retrieval Almost every blog is exporting nowadays newsfeeds, a collection of entries for recently published articles on that respective blog in a xml based format like rss or atom. Rss or Really Simple Syndication was firstly developed and is distributed broadly. A newsfeed is mapped in rss to xml as a channel element containing a couple of item elements. These items contain elements for the title, description, publishing date and link of the referenced blog article. Here especially the date attribute is valuable as it provides a detailed temporal resolution needed for diffusion modelling.
Atom has some more attributes allowing more detailed information about articles. As every newsfeed is supporting rss in its version 2.0 but not all atom, this format was chosen when retrieving newsfeed items of a particular blog. Using a simple SAX xml parser from the jdom library the article information was extracted. In a second step the article content was retrieved from the blog itself using the url of the mentioned newsfeed items. The resulting HTML page was then stored and passed through to further preprocessing explained in the next subsection.

Link Extraction Blog-to-blog links and citations had to be extracted from the article HTML pages, using a java HTML parser component library, to constitute a representative network graph needed for modelling the diffusion processes. From the list of all links found in an article only those were stored which were pointing out of the origined blog and exclusively to a blog of the previously retrieved dataset. As many blogs are hosted by blog software providers like blogger.com or blogspot.com only links to exact urls of blogs or their articles were taken into account, but not those to blog services nor to bookmarking or search services. Whenever a links was found in an article to another blog or its articles, a weight for this linkage has been increased to count the intensity of communication between these blogs. Finally the used link extraction allgorithms found in articles of 450 of the selected 680 blogs connections to other blogs. For each of these 450 blog it has been stored to which other blog they link and how often.

4.1.2 Data Preparation

Data preparation means the extraction and generation of information out of blog article contents serialized in a chainover of predefined processes. Each process generates cleaner and more detailed information of subelements of the extracted text.

At first the article contents have to be cleaned from design elements for extracting pure text contents. This preprocessing step will be explained more detailed as it is the base for all following processes. It determines the quality of the outputs, which are dependent upon eachothers results. Then subsequently elements of these texts as paragraphs, sentences and significant words so called terms need to be extracted and indexed to quantify their appearance in articles. These steps will be described in the following subsections, with focus of the relevance of their outputs for tracking topics.

Requirements

For grouping articles into topic clusters their contents, precisely words in these contents describing their semantic are of central use. These words will be extracted from the contents, which need to be cleaned to reduce the occurence of unneccessary HTML elements within subsequent processing steps. As cleaner the extracted text are, the better features can be extracted later to ensure semantic correctness of the detected clusters. This extraction of pure contents should be done without or with the fewest loss of contents. These texts will be splitted into paragraphs and sentences. These elements should be splitted in the way they naturally appear in the text. Thats why characters of seperation as well as unneccessary characters need to be determined. While extracting terms from those sentences, their part-of-speech and their baseform has to be detected correctly. Topic tracking methods will use this information for grouping and selecting article features.

Inputs

The input of this level of data processing is the output of the data aquisition, the dataset of blog articles in form of HTML pages. These have been compressed and stored in binary form to minimize needed storage space. The storage makes it possible to clearly seperate data aquisition from data preparation. That means if the downloading and storage of an articles HTML page succeeded completely, it will be again decompressed so that the methods described in the following can be applied to their contents. The stored pages can also be used for a data preprocessing at a later time point, with improved methods of data extraction to increase the quality of this information.

Outputs

The final output of the data preparation of blog articles are the particular subelements of their contents. These are paragraphs, sentences and the relevant words of that respective article called terms. In fact the most important output are the terms, whose occurences in the document, the so called term frequencies are getting counted for the document itself and the overall document collection.

Only those words have been chosen to be terms, whose length is greater than three characters and which are either adjectives or verbs or nouns. That means words with a fewer length or words which are not belonging to the mentioned word classes have been ignored while extracting terms. The frequencies of the most significant terms will be used lateron in the topic tracking methods to select the features denoting an articles semantic. As described above those features will grouped according to their baseform and part-of-speech tag by same methods to reduce grammitical complexity to the matter of interest. Therefore this metadata of the extracted terms is an essential output as well.

The other subelements of the articles text are stored to map the structure of these texts in the database. This information could be used to identify blocks of text with different topical alignment. Furthermore this information could be used to assign articles to many topics, which would model more realistic semantics than assigning one article solely to one topic. Due to simplicity of modelling and tracking of issues this information and explained procedure has been not used within this thesis.

Criterias for Decisions

The decisions which has to be made for preparating data for topic tracking, were related to the choice of the used methods. These are the allgorithms for content extraction, paragraph and sentence splitting and the term extraction with part-of-speech tagging and baseform reducing.

Content Extraction As described in the chapter dealing with basics of topic tracking, content extraction is a quite challenging task. Its results are very essential for all following data preprocessing steps but also for the processes of topic tracking and diffusion modelling. Thats why the selected method has to extract pure contents with high precision and recall.

Precision means that the allgorithm extracts to a high extend only those contents which are text elements. Recall means that the loss of text elements from the document is minimal after extraction. Secondly the allgorithm had to be easy to implement as content extraction is an important task but not the focus of this thesis. Speed of extraction was also a decision criteria as the content extraction was executed online directly after downloading of recent blog articles. Fortunately Jyotika Prasad and Andreas Paepcke from Stanford University released in October 2008 a new method, called "'CoreEx"', for extracting core contents from news sites which has extraordinary good results for precision (97 percent), recall (98 percent) and speed of extraction (15 ms). The details of this content extraction method will be decribed in the next subsection.

Paragraph and Sentence Splitting After extracting pure texts, these had to be splitted into paragraphs and sentences. Splitting means to detect characters to seperate two pieces of text from eachother. The so called delimiters had to be fixed and will be stated in the following section. Additionally special characters had to be removed as they carry no semantic information. It had to be decided which language tool to use for that task. Regular expressions have been chosen for removing these characters, as they are easy to specify and to apply by certain string manipulation methods. Finally abbreviations, specific for the articles language, had to be recognized as they include charcters used for splitting texts. One recognized they had to be ignored and removed while splitting the texts. It had to be decided how to do the matching of these sequences of characters. The relevant abbreviations have been stored in a list, so that it can be imported by the splitting component during startup. This option has been chosen as it is the simplest possibility to match words with these abbreviations in each step of text analysis using list operators.

Part-of-Speech Tagging For determining the part-of-speech, means the type of the word a convenient component had to be chosen. This task comes from natural language processing and is quite complex and difficult to implement. Therefore a component library from standford university has been integrated which provides that service. This library also delivers the neccessary language model files required to properly tagging words. It was chosen because it was simple to integrate in the developed issue tracking framework. Furthermore it delivered satisfying results with the test data resulting from monitoring news- and blog- specific search engines in March 2009.

Baseform Reducing Reducing baseforms is also a subject of natural language processing. For simplicity also this methods have only been integrated from already implemented tools. The source code of a baseform reducing component based on decision tree learning delevloped by the university of leibzig has been integrated into the framework. The trees neccessary for determination of baseforms have been also taken from this library. The decisive criteria were as well the good results from testing with the dataset of articles from March 2009.

Term Extraction Extracting terms from splitted and cleaned sentences is a simple task of tokenizing a text string. As programming language provides such a service no external library had to be integrated. For ensuring the significance of those terms and manual filtering has been applied. This task could have been also automated using list of so called stopwords and part-of-speech tags for irrelevant words. Unfortunately the tight development plan of these features didnt allow the implementation of such methods.

Used Methods

This section explains in detail how: pure contents of weblog article were extracted, those contents were splitted into paragraphs and sentences, significant words were extracted and the part-of-speech as well as the baseform of these words were determined.

Content Extraction Pure contents of weblog article HTML pages has been extracted using an algorithm based on a publication from Jyotika Prasad and Andreas Paepcke from Stanford University. As their algorithm was implemented for extracting pure contents from news pages, it has been reimplemented to adjust it to extraction of weblog article pages. It is based on a simple heuristic technique that uses a Document Object Model (DOM) tree representation of each article page, where every node in the tree represents an HTML node in the page. By analyzing the amount of text and number of links in every node, it calculates a heuristic measure to determine the node which contains the main content. This measurement is a basically a text- to link- count ratio. These counts of text and link nodes will be summed up walking up the DOM tree recursively from a text node up to the root of the document.

Whenever a link is found, the link count will be increased for that node. This value will be added then to the link count of its parent node. This basic scoring function has two drawbacks:

1. it favors small nodes with no links over larger nodes with a few links, this behavior is undesirable as mostly the large nodes contain the main content
2. the algorithm fails when the main content is contained in a set of nodes, instead of being held under one node

Weblogs, as modern designed pages structure their content not by html elements as tables but with the use of several containers. These are positioned with the help of style information. Thats why the content is seperated among many nodes, mostly even there are advertisements in between. For extracting pure contents from weblog article pages with high precision, these drawbacks have to be solved.

The first drawback is solved by a weighted scoring function. Therefore a new element is added capturing the fraction of the total text of the page that is contained in the scored node. In that way the node with much text will be preferred for selection. Additionally weights are introduced for the text-link count ratio and the pagetext fraction which has a much higher value thereby to support the descision for a text rich node.

But still in case the content is distributed over several nodes, the algorithm fails so that a subset selection has to be introduced selecting only some of the children of a highly scored node. Only those children will be selected whose text-link ratio is above a predefined threshold. The pure text content of each of these selected nodes is stored as a paragraph of the processed article page. With these important modifications the algorithms is able to achieve the above mentioned performance measures fitting nicely the requirements for task. The concrete implementation of their algorithm in java will be described in detail in the implementation chapter downwards.

Sentence Splitting The pure text extracted from the article pages has to be splitted into paragraphs for mapping a texts structure in the database. The used method of content extraction is able to identify multiple nodes containing pure text. Each of these text blocks will be stored as paragraphs. These paragraphs are still containing unneccessary tokens such as dates, numbers, special characters, dots or question marks. These tokens need to be removed as they contain no semantic information needed for definition of article specific features. As noted above regular expressions have been used to detect those charcters in the text and to replace them with empty strings. In a second step characters of seperation, so called delimiters as e.g. question marks, exclamation marks and dots have been used to split these cleaned text pieces into sentences. In that way the set of tokens analyzed for significant terms has been reduced. The association of sentences and paragraphs and their terms will be stored later to identify paragraphs as semantic distinct blocks of information of an article.

Part-of-Speech Tagging Part-of-Speech Tagging is a task originated in Natural Language Processing. Words of a text will get classified considering not only the definition of the word but also the context it is embedded in. Context means here what kind of words are located next to it. The used classifier is based on decision tree learning, a learning approach described in the topic tracking basics chapter. The implemented issue tracking framework uses the implementation of this classifier from the stanford university. This so called MaxentTagger takes a word and outputs a tag string. This machine learning algorithm needs training data in form of model text files for the used language. These text files contain trees of words associated with part-of-speech tags for test contents, for the english and the german language. The decision tree based classifier determines the tag within these trees which has the highest similarity to the term which has to be tagged.

Baseform Reducing The identified part-of-speech tags of a term are used to define a words wordclass. Based on that class unneccessary prefixes and suffixes are getting removed to result in the words baseform. For building feature vectors, needed for measuring content similarity, term frequencies means their occurences in an article text are counted. Baseform reducing is neccessary so that not words that semanticly same are counted as differently.

The baseform reducing task is performed by a component from the university of leibzig. For ensuring exact integration of this functionality the part-of-speech tagging has been not used from the above mentioned MaxentTagger, as the component implements its own tagging algorithms. These are also based on decision tree learning but using a different implementation technique.

Term Extraction Terms have been extracted from splitted sentences in two steps. At first a sentence has been splitted into a list of tokens using certain string tokenizer functions. The resulting tokens have been identified as terms and filled with their part-of-speech tag and their baseform. The methods used to determine a terms part-of-speech tag and baseform will be described in the following paragraphs. Secondly these terms have been manually filtered, by removing those terms from the database which were too short or not in the required word classes as described in the output section above.

4.1.3 Topic Detection and Tracking

Topic Detection deals with detection of groups of items in large datasets. These so called clusters belong together according to similar or same attributes. These attributes or features describe significantly the content or matter of the items. As this thesis deals with detection of topics in a large dataset of blog articles, features are those words in the articles contents which describe their semantic best.

A topic is hereby a concept which groups similar articles according to features which are occuring in their contents in similar quantities. At first topic clusters are detected from articles which have been published in the beginning. Then subsequently articles which have been published afterwards can be assigned if they match the required similarity. Topic Tracking is this linear process of assignment of articles to semantically meaningful clusters, called topics [BNJ03], by classification based on content similarity.

Requirements

The classifiers used for assignment of articles to topic clusters use certain distance measures to decide if two articles are contentwise similar or not. These distance measures compare the vectors of normalized and weightes feature frequencies of that articles. A key question is thereby which features to select, as this determines significantly the results of the applied classifiers. Only relevant and significant terms, which:

- have a length greater than three characters
- are an adjective, verb or noun
- carry semantic information (no abbreviations or senseless character sequences)

have been chosen as features for comparing articles published in a certain period of time. These have been weighted using a scheme explained in the following subsections for ordering them by their significance. This scheme needs to consider the qunatity of their appearance and their uniqueness of appearence in blog article contents. It has been also used for weighting the above mentioned document vectors. The selection of the used classifier determines the quality and quantity of resulting topic clusters, as these vary in complexity and preciseness of classification.

Their output have to match best semantically correctness with continueity of published articles. More precisely the articles of those clusters need to deal with the same topic, and the gaps between their publishing dates need to be minimal. This requirement is neccessary to ensure that the applied diffusion models can measure reasonable topic diffusions within these clusters. Finally speed is an important requirement for the tracking algorithms, which sunsequently assign articles to topic clusters. In the experiment of this thesis this has been done offline, means after all articles of the experiment have been retrieved and analyzed. The fast tracking allowed many iterations of improvement of the algorithms. For online tracking of issues that is even more important as it determines the delays between the processing of article contents.

Inputs

The terms extracted on the previous described level of data preparation and their metadata as part-of-speech-tags, baseforms and frequencies are the input for feature selection and topic tracking methods on this level. For erecting document vectors of term frequencies for each article this data has to be queried from the database, normalized and weighted. Part-of-speech-tags and baseforms offer possibilities of grouping and filtering of terms as features while querying.

Outputs

Diffusion models analyze the temporal and topological structure of the spread of information within a social networks. Analyzing topic diffusion processes requires therefore mapping blogs, publishing in a particular topic, to articles published at a certain date. That means those clusters have to be determined which match the required criteria of semantic correctness and continuity with satisfying measures. These measures will be described in the next subsection. Those clusters will be used by diffusion models as training data as well as topics whose diffusion will be predicted.

Criterias for Descisions

Detection and tracking of topics is a very complex task and includes therefore a variety of decisions determining the quality of its output. These are: the learning approach and classifiers chosen for classifying articles into topic clusters, the parameters of the tracking algorithms, the method for selecting features and the concrete measures of similarity and continuity for determination of the clusters with the highest quality.

Learning Approaches There are four learning approaches, explained in detail in the chapter of machine learning models, which have been analyzed regarding their impact for tracking topics: baysian learning, decision tree learning, learning based on artificial neural networks and instance-based learning.

- Bayesian learning, which uses conditional probabilities, requires minimal training data to produce good classification results but is limited to predefined categories. It contains no concept for opening up new categories which is neccessary for detection and tracking of new topics.

- Decision tree learning uses trees describing the structure of the data it should detect categories in. The aim of topic tracking is to detect unknown patterns of topical alignment of articles. Such required a priori information is therefore not available, thats why it could not be applied to this task.

- Learning based on artificial neural networks ,as a supervised learning approach, calculates a target function supervising its classification results in every learning step. These approaches like e.g. support vector machines produce very exact clusters but are very heavy to compute and therefore slow in execution. As speed of execution had been an important requirement this approach has not been chosen.

- Instance-based learning as an unsupervised learning approach, learns categories as instances mapped as points in an multidimensional euclidean vector space. In every learning step only categories have to be updated whose instances have changed while tracking. It also includes a concept for opening up new cetegories as decribed downwards. Thats why instance-based learning has been chosen to be the learning approach for classification in this thesis.

Instance-based learning has been chosen also due to its simplicity of implementation and calculation of content similarities. That simplicity implies its speed of execution, which in detail will be listed in the evaluation chapter. From this learning approach two classifiers have been tested: k-nearest neighbor and roccio, these will be described in the next subsection. The clusters produced by the chosen classifiers showed acceptable quality of semantic and continuity, whose precise measures are described downwards. Therefore the other analyzed approaches of learning topic clusters havent been evaluated, as this thesis focuses on diffusion measuring.

Feature Selection The features selected in the beginning of the experiment had less significance, due to insufficient filtering while term extraction. Thats why many features (400) have been selected in the beginning to ensure that every article can be classified. This increased the size of the feature vectors in the space, so that the distances between respective instances increased as well. As the similarity of article is measured in instance-based learning models by the distance of those feature vectors, the tracking produced very many clusters with very few articles within. Therefore k-nearest neighbors, an instance-based classifier explained downwards, has been chosen in the beginning due to its simple approach producing less exact but large clusters.

Chosen Classifiers Once the interrelation between the amount of selected features and the resulting distances of feature vectors has been discovered, all insignificant terms have been removed manually from the database. This process has been described in the data preparation section above. Finally the amount of features could be reduced to 100, as the quality of chosen features increased tremendously. Based on these features the k-nearest neighbors classifier produced very large and very few clusters. The reason for this behavior is that k-nearest neighbors always prefers the cluster containing the article with the closest distance. The shrinking of the vector space which resulted in closer distances of instances. So rarely the case happened that an article could not be assigned to any topic cluster, which would have resulted in creation of a new one. Thats why most articles have been classified into one large cluster.

Therefore roccio, the second classifier has been tested to improve the clusters quality. Roccio works on instead of single instances on average feature vectors, the so called centroids which represent the semantic centre of a particular cluster. The roccio classifier measures the distance of the feature vector to the centroid of topic clusters with the cosine similarity. The cosine similarity allows to normalize and weight feature vectors according to its scalability. The results from tracking articles based on this classifier are much more balanced than the results from k-nearest neighbor. The concrete numbers of these results will be presented lateron in the evaluation chapter.

Tracking Algorithms As described above tracking articles in topic clusters is a linear process. That means the dataset of articles ordered by their publishing date will be processed sequentially. The amount of topic clusters and their associated articles for the period of observation, the so called looking-back-window, increases exponentially with the processed period of publishing. Handling that complexity requires restriction of the training data. The training data are those clusters, which are taken into account at each step of classification. A time window has to be specified, which restricts the looking-back-window to those clusters and articles published within. This time window has been set to seven days for this experiment. This period was chosen as has been assumpted that a lifecycle of a topic, in which no blog has published any article for one week, is finished. These topics are not useful for assignment of recent published articles. This assumption matches also the requirement of continuity, that means that at least one article has been published per day within a particular topic.

The requirement of semantic correctness has been ensured by specifying certain thresholds for content similarity used by the roccio classfier. Two thresholds have been fixed, one for assignment of articles to topic clusters and the other for merging similar topics. During the tracking two topic clusters can become very similar due to convergence of their semantic core. If the cosine similarity of centroids of two topic clusters are measured to be higher than 0.95, they are merged. That means that all articles of one cluster will be assigned to the other. The threshold for assignment of articles is 0.9. These numbers have been evaluated to fit best for semantic correctness. Values less than these result in to broad topic landscapes while higher values result in too slim ones.

The amount of features determines the proximity of article feature vectors in the vector space. If too few features are selected for a time window, some feature vectors may be null vectors. Null vectors occur if an article contains none of the selected features. Such articles can never be assigned to any existing topic cluster, as their scalar product with other feature vectors will always be zero. On the other hand too many features increase the dimensionality of the vector space. In such high dimensioned vector space instances are located too far from each other to be grouped into senseful clusters. Therefore a vector size of 100 features has been evaluated as the optimal choice for high quality topic clusters with acceptable size.

Used Methods

This section explains in detail how: features were selected from weblog articles, document vectors of term frequencies were erected, weighted and normalized for those articles and articles have been classified and tracked.

Feature Selection Comparing documents regarding their semantic similarity, requires features which are significant for the document collection within a defined period of observation. That means for a fixed time frame, measured in days, the most significant words have been selected as the features of the collection. More specifically 100 baseforms of terms have been chosen, which had the highest occurences in documents and appeared in the fewest documents of the looking-back-window. These are the documents, published in the last 7 days, looked back from the date of tracking articles. These features were ordered by a weighting scheme called "'Term-Frequency Inverse-Document-Frequency"' (TFIDF) which combines these two requirements into one measurement. That term frequencies have been stored during the preprocessing step of term extraction explained in the previous section. The document frequency, as the amount of documents a term appears in, had to be queries from the database.

Erecting Document Vectors Once the features of a particular looking-back-window are known, their frequencies can be extracted from documents for every tracked article. As this information was stored during the data preparation, these frequencies can be simply queried from respective table of the database. The queried frequencies are normalized using the sum of the frequencies of all features found in that article. Additionally they are weighted with the inverse document frequency, which counts how many documents of the collection contain a specific term. In that way these feature vector of articles contain 100 term frequencies which are weighted with their significance for this particular document. A term might appear frequently in a document but if this term is also appearing often in other documents its significance for describing the documents semantic is less than one more exclusively appearing often in a particular document. The scheme used for this weighting as mentioned above is called TFIDF.

Classification Classification aims to assign items to predefined categories. In the field of topic tracking the process of classification is more dynamically. That means that not only predefined categories are used, but also new categories are detected. The classification is based on a certain learning approach used for learning categories and their measurements of characterization.

The instance-based learning approach has been chosen in this thesis for classifying articles into topic clusters. It is a unsupervised learning approach, which means that no target function supervises its learning results. This approach subsequently updating the learned model without complex recalculations of a target function, which reduces its preciseness but increaes its speed of learning. Based on the learned knowledge articles can be assigned to topics treated as categories. If an article cannot be assigned to any existing topic cluster, a new cluster or category is created. That happens if the similarity of the classified article with any topic cluster never exceeds the predefined threshold of 0.9.

Two classifiers, k-nearest neighbors and roccio have been applied to the dataset. Both classify articles as instances, according to their distance relation to already clustered articles in an multidimensional euclidean vector space. An instance is mapped thereby as a point in that vector space. The differences of both classifiers are described as follows:

- k-nearest neighbor tries to locate fitting clusters by analyzing the documents most similar to the classified one. If e.g. the five nearest documents are taken into account, the method prefers the cluster for classification where the majority is associated to. So if two are in one cluster and the rest in dinstinct other clusters the article will be classified to this topic. The classifier uses the eucledian distance to measure the closeness of the instances mapped into the vector space.
- Centroid oriented approaches as e.g. Roccio build an average feature vector from all feature vectors of articles contained in the topic cluster. This centroid is then compared to the document feature vector rearding its similarity using the cosine similarity measurement. The topic cluster whose centroid has the highest similarity with the articles feature vector is classified to fit in the cluster. The minimal condition is that this similarity exceeds a predefined threshold.

Tracking Tracking topics in a dataset of blog articles means to detect new clusters or to assign articles to existing ones. Within a period of two months, from begin of may until end of june 2009, this has been done within a continuesly shifting time window of seven days. The selection of features, loading of the looking-back-window and updating of the applied learning model has been done only once for this time window. These has been verified to be the most time consuming processes. The updated or newly created topic clusters have been parallely inserted or updated in the database using article to topic associations. As described above existing topics will also be merged if their centroid's cosine similarity exceeds a threshold of 0.95.

4.2 Measuring Topic Diffusions

Topic diffusions occur within communities of blogs through the social process, in which bloggers try to persuade other bloggers to write new articles in a given topic. Thereby these bloggers citate an originating blog article, link to the originating blogs by means of trackbacks or blogroll links, and comment written articles [JKFO06][p. 1]. This information flow can be modelled as adoption behaviours in time [SCHT07][p. 1]. Adoption behaviours are those events caused by influential blogs, resulting in writing and linking actions by influenced blogs. This adoption has been detected and tracked by the methods described in the previous section. Their output, the relevant topic clusters of the experiment will be analyzed for diffusion patterns. Their interpretation will lead to further decisions regarding the modelling and measuring of those topic diffusions.

This thesis compares two different approaches for modelling and measuring topic diffusions regarding their impact to support a corporate issue tracking. The main objetive is to measure the potential of those topics to gather public awareness, expressed by the rate of adoption. That means the amount of blogs which adopt the new topic by topic related writing and linking behaviours within an observed period of time. The applied models estimate such rates and diffusion cascades, a list of articles published in a topic within a certain period of time. These estimations have to be compared with real occuring diffusion to verify the preciseness of prediction. Therefore the best fitting model and optimal training data has to be chosen to ensure optimal prediction results.

This section decribes the general applicable methodology used for selecting the right diffusion model and to determine its optimal training data. For each of these tasks two approaches and measurements are getting applied and calculated for both models. The results of their predictions of diffusion rates and cascades of particular topic diffusions will be used for evaluation of these approaches. The aim is to determine which of these methods has to be prefered generally for each task, tracking issues in the blogosphere. The sections describing these tasks contain subsection for requirements, inputs, outputs, criterias for decisions and used models or methods. The measured topics will be ordered according to their predicted diffusion rate. As the models deliver the most influential blogs acting in analyzed topic diffusions, these will be also listed in the evaluation chapter.

4.2.1 Model Selection

Topic diffusions have two basic aspects: the social process and its temporal structure. Therefore diffusion processes in blog communities are determined by network effects within the community and external factors, which influence the process from outside the community. Network effects are effects of social influence by networks peers also known as contagion. They can be analyzed directly using the network data aquired about the observed blog community at the step of data aquisition.

External factors as e.g. mass medias, print medias, marketing campaigns cannot be measured directly as their influencers act outside the focal view of the experiment. The temporal structure of the topic diffusions can be used instead to recognize required diffusions patterns. Two diametrically opposite models have been compared focused on the both types of influencing factors.

The model to analyze network effects has been chosen to be the the linear threshold model from thomas valente. It models diffusion as a linear social process, where each individual observes its network peers adoption behaviours. The adoption decision of an individual is determined by its own threshold of adoption and its social pressure for adoption. A markov-chain based model has been applied for analyzing external effects by statistically modelling the temporal interdependencies of publishing behaviours of observed blogs. These observation lead to probabilities for certain adoptions based on a fitting probability density function. For ensuring optimal estimation results the model which fits best the underlying data from topic clusters has to be selected. That selection has been made according to the availibility of sufficient network data for the part of the blog community which publishes in a certain topic. Two measurements have been calculated and tested to indicate the influence of network effects on the observed topic diffusion. In that way it has been verified whether network or external effects dominate a particular topic diffusion.

Requirements

The main requirement of this task is that the model is selected, whose preliminaries and generated knowledge fits best the real occuring adoption behaviours in the modelled topic diffusion. The correctness of the selection decision has to be verified:

- matching the model preleminaries with the charistica of the topic diffusion
- comparing the real occuring diffusion patterns with the models captured ones

That implies that measurements for these properties of the best fitting model have to be determined and calculated. Then these measurements will be used to rank both models. The resulting decision will be evaluated later with the preciseness of predictions.

The applied models need to have the same inputs and outputs, which will be described in the next subsections. These have to map the semantic, temporal and social interdependencies occuring in the observed topic diffusions. Especially the temporal resolution of publishing timestamps has to be correct and comparable.

Inputs

For modelling topic diffusions three kind of input data is neccessary: the topic clusters which are analyzed, the network data of blogs publishing in these topics and the temporal data in form of publishing dates of clustered articles. The topic clusters need to be semantically correct and have to provide a satisfactory continuity in their lifecycle. The citerias for these two properties have been presented in the last two sections. These clusters are the direct output of the process of topic tracking. The network data as the linked blogs relevant for a certain topic diffusion will be queried from the database. These have been stored during the data aquisition. The relevant blogs publishing in a topic can be queried easily as the publishing blogs from each article are stored in the database as well. Finally every article has a certain publish date, which is used to sort the topic clusters by those timestamps. The requirement for these timestamps has been stated above.

Outputs

The output of this level of information processing is the selected model. Due to the interpretation of the calculated measurements of model comparison this will be either the linear threshold model from thomas valente or the markov-chain based model.

Criterias for Descisions

Deciding which model to choose for modelling and measuring topic diffusions requires a measurement of influence detection. That means a measurement which decides if a topic diffusion is dominated by network or external effects.

The approach taken for the selection decision concentrates primarily on the level of influence coming from network effects in that respective topic. This influence can be measured using two methods:

1. calculating the link intensity of subsequently publishing blogs
2. calculating the network density of all blogs published

in that observed topic diffusion. Both methods have been chosen because the fit the preliminaries of the linear threshold model. This model states that diffusion occurs in a linear sequence, therefore blogs of all articles of the observed topic need to be linked together. Only then the diffusion can be suspected to be dominated by network effects. If there are no links between those blogs other external factors influenced these blogs to publish. A neccessary condition for selecting the network model is therefore that the majority of subsequentially publishing blogs are linked together. The measurement tested will be the ratio of the amount of the subsequentially publishing, interconnected blogs to all blogs publishing in that topic, defined as:

$$\frac{SLB}{N}$$

, where SLB (subsequentially linked blogs) is the amount of blogs which have subsequentially published and are linked together. N are all blogs which have published in the observed topic diffusion.

The second method calculates the degree of interconnection between all publishing blogs. That is neccessary as the linear threshold model analyzes personal networks of publishing blogs to make decisions about their adoption behaviours. A personal network of a blog are those blogs it is directly linked with. The quantity of these links is measured by the out degree, the links going out from one blog to others. A personal network composed of numerous connections between direct linked blogs is referred to be dense or integrated [Val95][p. 40]. A blog with a dense personal network is not likely to receive much information from the outside [Val95][p. 40]. These dense personal networks are therefore not correlated with a fast diffusion, as diffusion happens within and throughout personal networks of individuals. In contrast a dense overall network indicates that there is a lot of communication in the network, this fascilates the information flow in the network [Val95][p. 42]. An overall network are all blogs observed for a topic diffusion.

Based on that observations there two types of network densities: personal network density and network density from the observed network as a whole. Personal network density is the ratio of the amount of interconnections of a blogs personal network to the links possible in that personal network. It cannot function as a measurement for influence of network effects in a topic diffusion as it captures only a part of the network. Overall network density can be a good indicator of the intensity of communication within this network influenced by social factors. Overall network density measures the ratio of the out degrees of all observed blogs to the links possible and is defined as:

$$\frac{\sum OD_i}{N(N-1)/2}$$

in which OD is the out degree of blog i and N the number of links possible in the observed network. These both measurements will be calculated for every topic diffusion which has to be modelled and measured. The selection of the right model based on this numbers will be evaluated using the preciseness of their predictions. The aim to investigate which measure fits best for this decision based on the discussed experiment.

Used Methods and Modells

This section explains in detail how: interconnection of subsequentially publishing weblogs and the overall network density of publishing blogs was calculated, the topic diffusions were modelled which were dominated by either network effects or external effects. The details of training these models and how they estimate topic diffusions will be described in the next two subsections.

Measurements of Network Effects As stated in the inputs subsection, all topic diffusions were modelled which fulfilled the requirements of relevant topic clusters as specified in the topic tracking section. For each of these clusters, ordered by the publishing date of their articles, measurements of influence detection had to be calculated.

The first measurement is the ratio of the amount of the subsequentially publishing, interconnected blogs to all blogs publishing in that topic. Therefore the properly ordered list of published articles had to be loaded from the database. Then this list is traversed and whenever a link occurs between two subsequentially publishing blogs, these are counted as SLB (subsequentially linked blogs). Finally the cumulated SLB's are divided by the overall amount of publishing blogs.

The second measurement is the overall network density. Over the same list of published articles, the out degrees of the published blogs are summed up. These can be aquired using a network graph constructed by using functions of the JUNG network library. These take the network data from the database and convert that information in a graph consting of nodes and edges. The out degrees have to be reduced to only considering those links going out to blogs publishing in the respective topic diffusion. Once these values have been summed up, the result will be divided by the amount of links possible.

Network Model The model which will be chosen in case the topic diffusion is dominated by network effects is the linear threshold model developed by thomas valente. It has been tested as a network model, because it takes a look at diffusion as a social process by modelling the adoption behaviours of blogs in relation to those of their direct neighbors in the network. That means it measures the degree to which an individual is exposed to a particular topic through its personal network. Network exposure is the fraction of a blogs personal network, the direct linked blogs, that have allready adopted a certain topic at a determined point of time.

The network effect this model captures is named contagion. That means that the adoption decision of an individual is based on the observed network exposure. Each individual has a threshold regarding this exposure, the fraction of adopters to direct neighbors that need to exist until when he is adopting. These thresholds, exposures at time of adoption are the core of this model. Training this model means to calculate thresholds. Predicting with this model means to interpret thresholds.

The model has been implemented using the JUNG java library. It operates upon a network graph constituted from the list of blogs as its vertices and the collected links as its edges. At first the model will be trained with a representative topic cluster, chosen according to two measurements decribed in the next section. These determine the topic cluster whose diffusion structure is most similar to the one the model should measure. After training the model procceds an iterative prediction algorithm starting with the blog, which published at last in the topic diffusion the model should measure.

Dynamic Model The model which will be chosen in case the topic diffusion is dominated by external effects is a dynamic model based on a Continuous-Time Markov Chain (CTMC). It is a dynamic model as it focusses on modelling temporal dependencies of continues-time processes. The processes which have to be modelled here, are the sequence of adoption behaviours of blogs. More precisely the events of adoption of a topic by a blog in terms of writing or linking actions, influenced by another blogs adoption. These are allowed to occur at any time, even adoption can occur at the same time. But no causal dependencies of adoption behaviours are allowed, as the training data of the model does not provide that information in case the trained topic diffusion is dominated by external effects.

As the preliminaries of a CTMC fit exactly the requirements for a dynamic model capturing external effects of topic diffusions, the applied model has been implemented based on it. For information flow modelling in the blogosphere, the states are blogs from the discrete space of blogs publishing in the observed topic diffusion. The event of adoption, blog i writes an article in a particular topic after blog j, is modelled as a state transition between the states representing the blogs with a probability that the event occurs in a certain time period.

The transition probability is therefore the probability per time unit, that the CTMC makes a transition between two states. The time between this transition is the state staying time. That means the amount of time the CTMC stays at one state before it jumps to another. The adoption processes, means adoption events occuring within certain staying times, are markov processes which fulfill the requirements of a homogenous poisson process 3.3.2. The time period of a poisson process is following a poisson distribution, which can be approximated using a exponential distribution function.

Thats why the staying time between a transition of the CTMC has been assumpted to be exponential distributed. The density function of the exponential distribution is used therefore for calculating probabilities for state transitions of observed intervals. This function takes a rate λ as a parameter which stands for the average rate of adoption, that means the average frequency of publications, for the observed period of time. The model will be trained generating a transition probability matrix using the described distribution. Based on that matrix the cascade of blogs, which will publish in the future, have been estimated.

4.2.2 Model Fitting

Models which predict future developments of processes need to be trained to learn the knowledge they need for that prediction. Diffusion models therefore need to be trained with enclosed topic diffusions to learn their specific knowledge to predict diffusions of new topics. That means whenever a new topics future diffusion has to be predicted, the used model has to be trained with a fully observed topic diffusion of the past. The accuracy of those models relies on the convenience of the knowledge generated. That means that the learned knowledge and therefore the structure of used training data, fits the structure of the new topic diffusion. This section deals with methods how to determine the best fitting training data, in form of past topic diffusions, for a new topic diffusion. Two approaches will be introduced, each specific for the applied diffusion model.

Requirements

The requirement for a proper model fitting is that the chosen training data allows the selected model to predict most precise diffusion rates and cascades. The evaluation of this preciseness will be decribed in the next subsection. The models map the structure of the trained topic diffusion using their specific mathematical devices. They can only predict further diffusions using that applied structure. That leads to the requirement of training data to be structural similar to the estimated one.

Inputs

Determination of best fitting training data for a diffusion model requires: the selected model and the list of topic diffusions which could function as training data. The selection of the required model was described in the last subsection in detail. The ground dataset for selection of training data are those topic clusters which are relevant in terms of semantic correctness and continuity. Their collection and evaluation has been described in the topic tracking subsection. The selection of those clusters from this ground dataset, which are useful as training data will be decribed lateron.

Outputs

The result of model fitting methods is the best fitting topic cluster for a given diffusion model and a given topic cluster whose further diffusion should be predicted. The evaluation of this selection will listed in the evaluation chapter.

Criterias for Descisions

The definition of this structural similarity for training data is dependent on the model used for measuring diffusions. The reason is that the selection of the model, was also made according to the structure of a particular analyzed topic diffusion. These topic diffusions were decided to be dominated by either network or external effects. As explained in the last subsection that implies a different network structure of the blogs involved in publishing. Determining for a particular topic diffusion which other diffusion might fit to it according to its structural similarity implies therefore also the usage of different approaches for either network or external influenced diffusions. These approaches have to restrict the topic clusters delivered by topic tracking to fully enclosed topic diffusions occuring strictly before the end of theestimated one. That means that the last article of those topic diffusions has to be published strictly before the last known article of the new topic diffusion. That requirement is neccessary that both diffusions are not merged together. That would wipe out the exactness of diffusion modelling.

General Approaches Generally it can be stated that the structure of a topic diffusion is basically a sequence or cascade of articles which have been published by certain blogs in a timeline. For simplicity only articles are considered published by distinct blogs. That means in no topic diffusion appears an article from a particular blog twice. Similarity of such cascades can be detected using two elements: same appearing blogs and the same sequence for particular blogs. That kind of general similarity measurement will be applied for the evaluation of diffusion cascade estimations of both models. In that context a general similarity measurement fits the requirements best, as both models have to be compared and evaluated in the same way to allow comparison of their estimation results. Another general approach is to measure the topical similarity of compared topic diffusions. As blogs mostly publish in certain topics, diffusions whose topics are very similar might occur in the same part of the observed community. This approach will be tested to investigate whether such semantic effects might reduce the impact of the applied diffusion model.

Model Specific Approaches For selection of model specific optimal training data this general approach is not appropriate. Here specific approaches have to be considered, to ensure structural similarity of training data specific for the needs of the chosen model. In concrete that means that one aproach has to be selected which can evaluate fitting training data dominated by network and by external effects. These chosen approaches are based upon:

1. opinion leadership, for network effect dominated training data
2. rate of adoption, for external effect dominated training data

,which will be explained in the next subsection in detail.

Opinion leadership has been chosen for recognizing similar diffusions dominated by network effects, as information is spreaded mostly by opinion leaders in such networks. The topic which future diffusion should be estimated, consists only of few articles at time of estimation. These have been published by the most innovative bloggers in the topic related part of the community. Opinion leaders are such individuals of a social network, which are influential to others in their network according to their centrality. That means basically that their published articles are often referenced by other blogs. So by detecting opinion leaders in contagion influenced diffusions, based on measurements of innovativeness and centrality, one can identify blogs spreading the new topic. This spread occurs in the personal networks of these leaders. As the observed community of the experiment is enclosed these are the same for potential training diffusions. Thats why most influential blogs can be used to infer similar diffusion structures based on a minimal set of seed blogs. These interrelations have been reported by many surveys on social networks [Val95], and have been used therefore for the development of the linear threshold model. That fact made this approach the ideal model fitting solution for that model. Additionally the related measures will be calculated in the training phase of this model anyway.

Topic diffusion dominated by external effects will be statistically modelled according to their temporal structure. That refers basically to the intervals of and between certain adoption events. A measurement for selecting optimal training data, which has a high structural similarity for such a model has to capture therefore the relation between intervals and occurring adoption events. As described in the previous subsection the average rate of adoption, as the amount of adoptions per day, is such a measure.

This rate determines the probabilities calculated by the density function of the exponential distribution. These probabilities will be used to predict the diffusion of the new topic. This model fitting criteria will only ensure that the average rate of adoption will match the estimated diffusion rate quantitywise. For ensuring also a qualitative fitting of the estimated diffusion cascade and the observed diffusion cascade of a potential training diffusion, blogs with high assigned probabilities have to be considered. That blogs are the subsequent publishing ones, a matching due to that measure will detect a seed set similar to the opinion leader approach.

Used Methods

This sections explains in detail how: topic clusters were evaluated to fit best as training data for both models and how both models have been trained.

Opinion Leadership The linear threshold model uses observations of personal network exposures of individuals to make decisions about occuring events of adoption. By calculating thresholds, the amount of adopters an observed blog is exposed at its time of adoption, the adoption behaviour of blogs has been modelled. Low thresholds indicate that blogs adopt early, which makes them innovative regarding a certain topic. Blogs in their personal network might be influenced by their adoption decision at a later point of time. That is mainly determined by the centrality of the influencers. That are the amount of blogs linking to him from his personal network, the so called in-degree. Both measures will be calculated by the model based on aquired network data. Those blogs with the lowest thresholds and the highest in-degrees can be selected as the opinion leaders of a particular topic diffusion. The blogs of the evaluated training diffusion and the new topic diffusion will be ranked according to these measures. Finally a matching algorithm will select the diffusion as training data which has the highest intersection of detected opinion leaders.

Rate of Adoption The continues-time markov chain model predicts events of adoptions based on observed delays between subsequential publishing activities of blogs. These intervals are exponential distributed with an average rate of published articles per day. Calculating that rate for a particular topic diffusion means to divide the amount of articles published by the overall observed lifetime of that topic. That is one of the neccessary tasks of training that model.

The set of possible training diffusions will be ranked according to that measure to define the ones, whose average rate of adoption has the fewest difference to the measured topic diffusion. In a second step those ranked topic diffusions will be postselected which feature same blogs with high assigned transition probabilities. The diffusion matching these both criterias best will be selected as training diffusion for the dynamic model.

Training the Network Model The training phase begins with setting of blogs as adopters based on the publishing data of the topic cluster used as training data. The selection of the optimal training data will be explained in the following section. After that primary step each adopting blog is analyzed for the network exposure that existed at his time of adoption, which will be stored as his threshold. Additionally the time period between his time of adoption and the time point his personal network reaches this threshold is stored as the threshold lag used for correction of thresholds. The threshold lag and the time of adoption of each adopting blog in measured in days from the date of the first published article in the respective topic.

Training the Dynamic Model The training phase begins with the calculation the average rate of adoption based on the publishing data of the topic cluster used as training data. The selection of the optimal training data will be explained in the following section. The average rate of adoption gets calculated by the inverse value of the sum of the observed delay intervals. After that probabilities are calculated using density function of the exponential distribution. Thereby only intervals of blogs which published subsequentially are taken into account for calculation. These will be stored in a matrix for all blogs publishing in the topic diffusion used as training data. Transitions of Blogs which could not be observed to have a event of adoption in between will get a zero assigned in that matrix. Only publishing blogs will get mapped into that transition matrix.

4.2.3 Diffusion Estimations

This subsection describes the final level of knowledge creation of the presented methodology for measuring topic diffusions. Here all results of the previous two phases of diffusion modelling, the selected model and training data will be used to predicted diffusion cascades and rates. Evaluating the prediction results from both models means to compare predicted diffusion cascades and rates with those inherent in the tracked topic clusters. That information is used to rank issues according to their reputation potential. Detected opinion leaders enhance that information.

Requirements

Requirements divide in quantitative requirements for diffusion rates and qualitative requirements for diffusion cascades. In case of rates the differences between estimations and observed diffusion rates need to be minimal. Diffusion cascades are evaluated quality wise, which means that the number of rightly predicted sequence of blogs and their intervals for adoption will be analayzed.

Inputs

Measuring topic diffusions requires the the model which have been selected to match best its data characteristics. The training data in form of the most structural similar topic diffusion and the topic diffusion which should be measured. The topic diffusion selected for prediction will be used only fractional. That means one part will be used for the modelling, and the second for the evaluation of estimations.

Outputs

The output of this final step of diffusion modelling are the estimated diffusion rate and cascade. Rate means here the amount of blogs which the model estimates to publish within the defined prediction horizon. Using that intervall also an average rate of adoption can be calculated. The casade is simply the sequence of publishing blogs which the applied model predicts. That allows ranking the observed and measured topic diffusions according to the diffusion rate. That ranking will deliver the most influential topic diffusions for dynamics of reputation in the blogosphere for a particular company. For each of these topic diffusions the applied model can also give hints about the opinion leaders involved in that spread of information.

Criterias for Decisions

The main decisions which had to be made for measuring diffusion and evaluation were related to the methods of estimation and the measurements used for evaluating the results of the applied and compared models. The method of estimation is bound to the selected model. Whatever model has been chosen and tested will estimate and predict future diffusions in its own way. The methods of estimation for the used network and dynamic model will be described in the next subsection in detail. Measuring topic diffusions means to measure their predicted quantity of adoptions, absolutely and relative to the prediction horizon. This rate of adoption was used as the output of both models as it will be used lateron to rank issues according to their potential to become interesting for the public opinion. Additionally these models will output cascades which will used implicitly then for detection of opinion leaders. The preciseness of the prediction of correct sequences of publishing blogs determines this detection as described in the in the model fitting subsection of the concept.

Used Methods

This sections explains in detail how: topic diffusions have been estimated using both models and how these estimations in form of diffusion rates and cascades have been evaluated according to their preciseness of prediction of real occuring diffusions.

Diffusion Estimation using the Network Model The prediction phase of the linear threshold model begins with setting the adopters of the new topic. These are the blogs which have published in that new topic. Using the generated knowledge from the training phase, in form of thresholds and threshold lags, new thresholds are calculated and existing ones updated. The prediction itself is an iterative algorithm executed over a certain amount of days, called the prediction horizon. That horizon cannot exceed the length of the lifecycle of the training topic diffusion. This algorithm subsequentially decides if and when other blogs adopt, based on the thresholds calculated before. The output at the end of that algorithm is a list of blogs, suspected to sequentially publishing in that topic diffusion within the prediction horizon. That list is referred to be the diffusion cascade, its length is the diffusion rate.

Diffusion Estimation using the Dynamic Model Based on the transition matrix created in the training phase the adoption cascade can be estimated. The topic diffusion which should be estimated, delivers the cascade up to the timepoint of prediction.

The blog which published the last article in that topic diffusion is the entry point for the prediction. Beginning with this blog, the transition probability matrix is used to find the next most probable publishing blog. That means in the row of the particular blog, the next blog is chosen which has the highest probability in that row. Then this blogs row will be searched and the next probable blog will be determined. This process is repeated until the sum of the estimated intervalls exceed the predefined prediction horizon measured in days. The intervals will be estimated using the exponential distribution, which takes the rate λ from the training topic diffusion. The resulting list represents the diffusion cascade of the estimated topic diffusion, its length is the estimated rate of diffusion.

Estimation Evaluation The outputs of the evaluated models, diffusion rates and cascades, have to be compared for each estimation of a particular topic diffusion. Diffusion rates are the amount of articles the models will estimate to publish within the predefined prediction horizon. That horizon is determined by the lifetime of the used training diffusion in days. The diffusion which has to be estimated, should be observed for its whole lifetime as well between the period of observation of the underlying experiment. The measured diffusion will be seperated in two halfs using the overall period of publishing, the above mentioned lifetime of that topic. The first half will be used for analysis for model selection and model fitting, and the second half will be used for evaluation of the predicted rates. The models are ranked according to their ability to estimate a rate whose difference is less related to the observed one. Cascades will be analyzed for the amount of blogs whose sequence of publishing is equal to the observed sequence in the second half of the measured diffusion. Based on these quantitative and qualitative measures the best predicting model will be chosen for that particular diffusion.

Opinion Leader Detection Detecting opinion leaders based on measures of innovativeness and centrality has been described in the previous subsection in detail. The centrality based on in-degrees will be computed using network data of blogs publishing in the first half of the predicted diffusion. Innovativeness is determined using the measures of the model which as been preferred according to the results of the estimation evaluation. For the network model thresholds will be used and for the dynamic model the computed transition probabilities.

Chapter 5

Prototype Implementation

This chapter describes the implementation of the issue tracking framework which concept was presented in the previous chapter. There are two main sections, one dealing with the environment which was used for development and the other with the architecture used to weave together all needed components. These components implement their own functionality but also using external libraries for repeating and outsourced tasks.

5.1 Development Environment

The prototype of issuetracking has been implemented using the eclipse integrated development environment. That environment has been chosen as it is most suitable for setting up complex java development projects. That java project was implemented using the spring framework to integrate various components into one application.

5.1.1 Integrated Development Environment

Eclipse allows integration of plugins which help the developer simplifying repeating tasks of software engineering. As the prototype was implemented using the spring framework and hibernate framework, also respective tools were used for handling these frameworks. In particular that means that a J2EE Developer plugin was used create hibernate objects out of the database scheme A.1. Also a plugin for the spring framework was integrated which simplifies the work with spring bean definition files.

5.1.2 Spring Framework

The concept of this thesis shows the complexity of the solution of issuetracking which has to be implemented by the prototype. There are several layers of information processing and each is containing various components. All these components have relations in form of data or services to eachother. For developing such highly complex system it was neccessary to think about an integration framework. Such integration frameworks allow the specification of components as xml snippets. These specifications embody a components properties, parameters and relations to other components. The framework itself runs in a container, which during startup builds all the components from associated java sources.

There are currently two important integration frameworks for enterprise applications available: the Spring Framework and the SEAM Framework. Both frameworks are open-source software and can be extended by any software developer. SEAM was developed mainly by JBOSS and focusses on entity beans suitable for application servers.

The Spring Framework was developed by rod johnson and a couple of other freelancing developers. This framework has been chosen as it has not the requirement to run on a particular application server or use certain software infrastructure. That means you can build any kind of enterprise application from very simple to highly complex application. It can run as a simple java application, junit test, on a web server or a full fledged application server even with distributed data servers. Furthermore spring allows the integration of a variety of libraries and frameworks in form of component models. The following grphic shows the full application stack of the Spring Framework with all provided functionality.

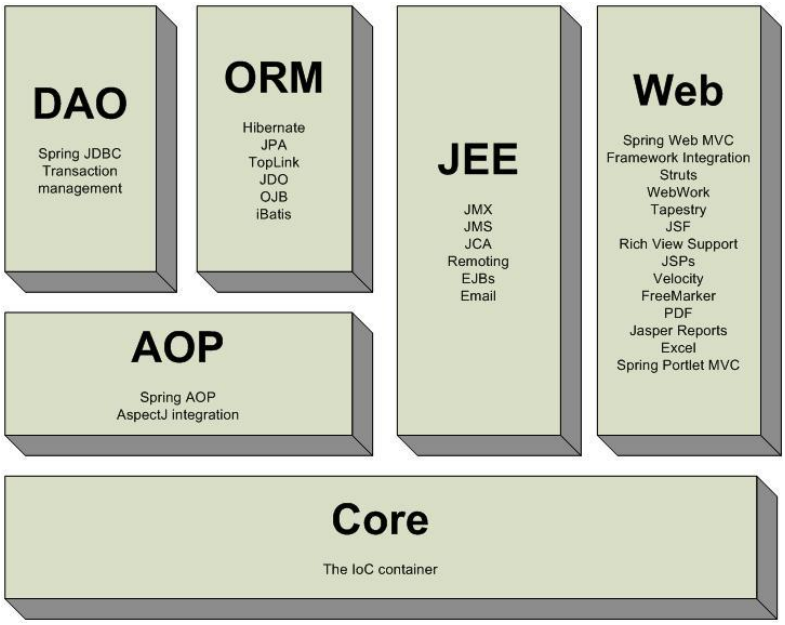

Figure 5.1: Component Stack of the Spring Framework

Dependency Injection

An important concept of the Spring Framework, which has been a selection criteria, is the dependency injection. In complex software systems every components requires links and references to others. That is neccessary as components provide functionality for eachother that the system can deliver the required services to its user. In standard applications those references or dependencies have to be wired manually. Every java class needs to care for its own required references to other objects.

With the Spring Framework its Inversion Of Control (IOC) container cares for this wiring of objects. Inversion of Control means not objects care about their dependencies and try to aquire needed objects, but the container cares for those depedencies and injects them into the objects while startup of the application. The developer only has to specify in certain xml files which components depends on eachother, the wiring happens automatically. This loosely coupling of components is the main reason why Spring has been chosen for developement of the prototype of this thesis. Fot testing parts of the system it was important to flexible and fast switch the integrated components and their dependencies. This flexible approach also simplifies extension of the prototype even when it is online.

Configuration

Beside the dependency injection another benefit of the Spring Framework its the possibility of configuration of the application, its components and their relations. This configuration takes place in certain xml files called application contexts. Each of these application context contains xml based spring bean definitions.

For every bean the resposible class has to defined. This can be a class, which has been implemented or a class from an external library. In that way developers can easily integrate new functionality without writing any line of source code. A bean can have various properties as string values and all kind of collections. These will be mapped to the java properties in the associated class. Finally every bean has references to other bean or can even contain other beans in itself.

The application context as the collection of beans will be analyzed by the IOC container of Spring, which will build all required objects. Every application context is responsible for the beans of a layer of software architecture. That is a design pattern adviced from spring to bundle all layer specific beans to one package. SUch packages or bundles will allow online exchange and updates of functionality. That has been a requirement for the developed prototype from the beginning, as such a system has to run online for a long time but has to be extended as well. Such a functionality can be implemented using springs integration for the OSGi platform.

5.1.3 Integrated Libraries

The following list shows the external libraries integrated using spring beans:

- hibernate, for mapping java objects to database entities
- quartz, for scheduling of jobs for searching articles
- jung, for mapping blogs and links into a network graph
- htmlparser, for parsing blog article html pages
- jdom xml parser, for parsing xml endoded technorati result pages and newsfeeds
- json parser, for parsing yahoo news json encoded result pages

5.2 Software Architecture

This section deals with the architecture and implementation details of the implemented prototype for issuetracking in the blogosphere. The system has been divided into layers according to the design proposal of the used integration framework.

The first layer, the data access layer, so called data access objects handle java objects enhanced with object relational mappings using hibernate annotations. These map relational structures from the underlying MySQL database into java objects using their properties and collections. The second layer, the service layer, deals with all services neccessary as decribed in the concept of this thesis.

Furthermore transactional services are needed to group data access operations which belong to unique tasks. That services will create database transactions on the fly to ensure the referential integrity of the database. Finally the presentation layer cares about visualization of the results and the user interface for the implemented prototype. The following graphic presents the described architecture in detail:

Figure 5.2: Software Architecture

5.2.1 Data Access Layer

The data access layer has been implemented using the objectrelational mapping (ORM) integration of the spring framework. Spring allows to use a certain ORM technology within its project code by use of wrapper classes. The are wrapper classes are implemented as templates which take a variety of arguments to specifiy the details of data access. For this prototype the implementation of the HibernateTemplate has been extended for the implementation of all four presented data access objects. These use a session factory to access the database during defined periods of time. The session factory has a reference to a datasource bean which represents the database itself. That contains the type of database and the access information as username and password. All data access objects contain methods handling domain objects which map certain database tables to java objects.

Domain Objects

The following paragraphs list domain objects which have been used to map database tables into hibernate annotated java objects. Each of this objects contains named queries in the hibernate query language (hql) used by the responsible data access objects, properties mapped to table attributes with getter and setter methods, index and cache specification for improving performance of data access.

Document related Entities The following list shows all domain objects which have been implemented for mapping entities dealing with the document means the article and its contents:

- Article, with its html and meta data
- Paragraph, with html snippet and pure text
- Term, with postag and word
- Term of Article, for mapping terms to articles with frequencies
- Term of Sentence, for mapping terms to sentences
- Term Frequency, for mapping document collection term frequencies

Blog related Domain Objects The following list shows all domain objects which have been implemented for mapping entities dealing with the blog means the source, its newsfeed and its linked sources:

- Source, with meta data as url and name
- Newsfeed, with url
- Linked Source, with weight and linked source id

Topic related Domain Objects The following list shows all domain objects which have been implemented for mapping entities dealing with the topics means the topic itself and its significant terms:

- Topic, with its associated articles
- Term of Topic, with the most significant topic words

Scope related Domain Objects The following list shows all domain objects which have been implemented for mapping entities dealing with the scopes means the scope itself and its representing keywords:

- Scope, with its associated keywords
- Keyword, with the word

Data Access Objects (DAO)

Data Access Objects manage the previously described domain objects using springs hibernate template. Every DAO implements methods for storage, updating and querying of needed data which will be passed through service objects by transactional services.

5.2.2 Service Layer

This subsections deals with all components which have been bundled on the service layer. At first transactional services are described which combine queries from the above described data access objects into physical and virtual transactions. Then service components are described according to the levels of information processing introduced in the conceptional design chapter.

Transactional Services

Transactional services are neccessary to bound associated data accesses into transactions. These operations have to be executed at once, if one fails the changes of all others have to be rolled back. A good examble is the data preparation of article content. If an article could be downloaded or its content could not be extracted also the following splitting and extraction methods will fail. Thats why this process will executed as a whole, if one fails its aborted. Spring allows definitions of transactions and transactional services using aspect oriented programming (AOP). The developer just has to sign a method or a class with a transactional annotation, and spring will wire the transaction aspect into its code. Automatically these methods will start and end a physical transaction when executed. This aspect is implemented by spring using interceptors which will start end end the transactions when called. The following graphic shows the workflow of such a service which uses itself a spring transaction manager:

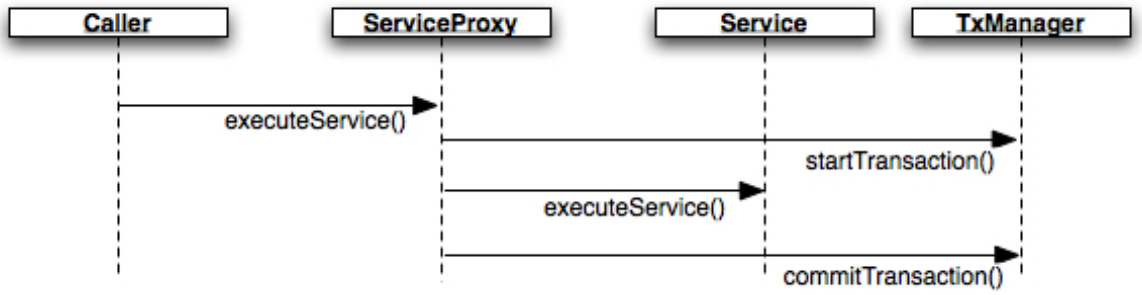

Figure 5.3: Transaction Handling

Chapter 6

Measuring Topic Diffusion

This chapter presents the results of the several steps of information processing as described in the conceptional design chapter. The first section deals with the test dataset which was aquired in March 2009 from specific search engines. The next section will present the results of tracked topic clusters, using either k-nearest neighbors or roccio as the classifier. The following three sections will deal with the results of the sub tasks of diffusion modelling as model selection, model fitting and diffusion estimation. In particular the measurements for modelling decisions will be presented. The last section will evaluate the results of all tasks to conclude the best measurements and decisions for a general applicable methodology of modelling and measuring topic diffusions.

6.1 Topic Tracking

As described in the topic tracking subsection of the concept chapter, the k-nearest neighbors and the roccio classifier have been used to track blog articles in topic clusters. The top ten clusters of these seperate tracking processes will be presented in the next two subsections. From each topic cluster the ten most significant words and the number of articles tracked will be shown.

All articles that have been published from blogs of the observed community of blogs between begin of may until end of june 2009 have been tracked. 65820 articles have been tracked after the previously described dataset has been aquired and preprocessed. That process lasted all about 13 hours.

6.1.1 K-nearest Neighbors

In this subsection the results from topic tracking using the k-nearest neighbors classifier will be presented.

6.1.2 Roccio

In this subsection the results from topic tracking using the roccio classifier will be presented. From the 65820 articles about 35000 topic clusters have been detected and tracked. That process resulted in a fairly balanced distribution of articles over topic clusters of 0.53 percent. This measurement is an indicator for the semantic correct classification of articles into topic clusters in addition to the specification of a threshold of 0.9 for required content similarity of those articles.

6.2 Model Selection

For selection of an appropriate diffusion model the 20 best tracked topic clusters, using the roccio classifier, have been analyzed for domination of network or external effects in their diffusions. The results from roccio have been chosen for diffusion modelling as they showed the best values of semantic correctness and continuity. For each inspected diffusion the previously described measurements for network effect influence detection will be presented. That means each cluster will be listed with their most significant topic words, the network density of the publishing blog network and the fraction of linked blogs subsequentially publishing.

6.3 Model Fitting

After selection of the best suited model for the topic diffusions the optimal training data has to be examined based on measurements of opinion leadership and adoption rates. These measurements have been calculated for the preselected representative 20 topic diffusions. The following subsections will present these measures for each topic cluster.

6.4 Diffusion Rate Estimation

This section will present for each of the 20 measured topic diffusions the estimated rates and predicted diffusion cascades for each applied model.

6.4.1 Linear Threshold Model

6.4.2 Continues-Time Markov Chain

6.5 Evaluation

Chapter 7

Conclusions and Future Work

Within this thesis a general applicable methodology for modelling and measuring topic diffusions in blog communities has been designed and implemented. Due to lack of time the results of the applied models could not be evaluated regarding their impact to support a corporate issue tracking.

7.1 Conclusions

Thats why the insights and results, which have been gathered during the editing of this thesis are mainly focussed on topic detection and tracking. Two classifiers have been tested with various settings of clustering thresholds, feature vector sizes and window sizes for tracking topics. The optimal values have been stated in the concept chapter already. Furthermore a dataset of 35000 topic clusters have been generated, with more than 200 clusters which allow suitable tests for diffusion modelling.

By graphical analyzing the network data of the aquired community it has been verified that a gravitational core of highly connected blogs exist surrounded by many peripheral blogs which are isolated. That implies that testing models considering networks and external effects in diffusions makes sense. The details of implementation of these models have been stated in the concept chapter as well.

7.2 Future Work

Based on the concept and using the implementation of the issuetracking prototype the dataset of topic clusters should be investigated for concrete diffusion patterns. Thereby a cleaning of features would increase the yielded results in such a evaluation. Additionally an epidemic model could be implemented and tested for cases when network and external effects in a diffusion are balanced.

Appendix A

Data Storage

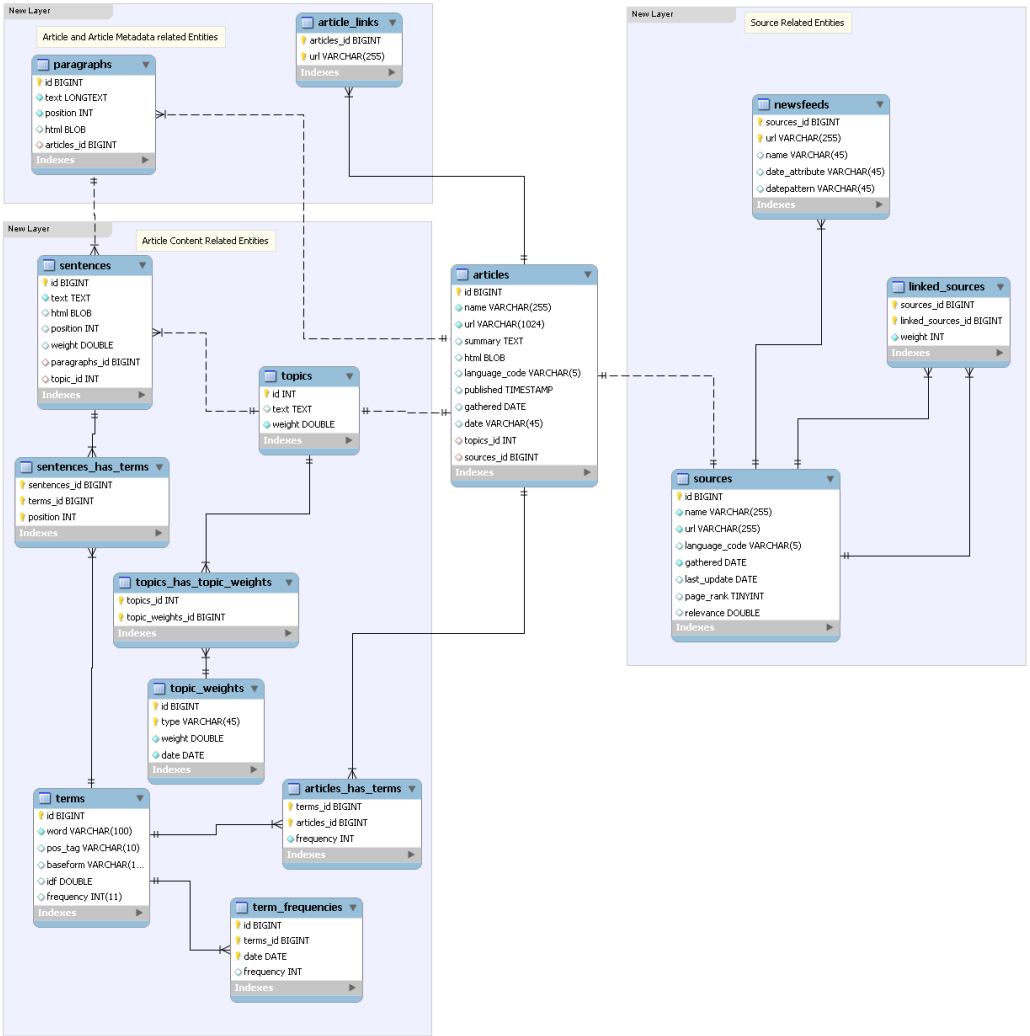

Figure A.1: Data Model

Bibliography

[Bai75] Bailey, N: *The Mathematical Theory of Infectious Disease*. Hafner Press/ MacMillian Pub. Co., 1975

[BNJ03] Blei, David M. ; Ng, Andrew Y. ; Jordan, Michael I.: Latent dirichlet allocation. In: *J. Mach. Learn. Res.* 3 (2003), S. 993–1022. – ISSN 1533–7928

[CNM04] Clauset, Aaron ; Newman, M. E. J. ; Moore, Cristopher. *Finding community structure in very large networks.* August 2004

[EFL04] Erosheva, E. ; Fienberg, S. ; Lafferty, J.: Mixed-membership models of scientific publications. In: *Proc Natl Acad Sci U S A* 101 Suppl 1 (2004), April, S. 5220–5227. – ISSN 0027–8424

[EK02] Eguiyluz, Victor M. ; Klemm, Konstantin: Epidemic threshold in structured scale-free networks. (2002)

[GCNS02] Girvan, Michelle ; Callaway, Duncan S. ; Newman, M. E. J. ; Strogatz, Steven H.: Simple model of epidemics with pathogen mutation. In: *Phys. Rev. E* 65 (2002), Mar, Nr. 3

[GGLNT04] Gruhl, Daniel ; Guha, R. ; Liben-Nowell, David ; Tomkins, Andrew: Information diffusion through blogspace. (2004)

[GLM01] Goldenberg, Jacob ; Libai, Barak ; Muller, Eitan: Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. In: *Marketing Letters* (2001)

[HS03] Haller, Hans ; Sarangi, Sudipta: Nash Networks with Heterogeneous Agents / DIW Berlin, German Institute for Economic Research. 2003. – Discussion Papers of DIW Berlin

[JKF+07] Java, Akshay ; Kolari, Pranam ; Finin, Tim ; Joshi, Anupam ; Oates, Tim: Feeds That Matter: A Study of Bloglines Subscriptions. In: *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)* Computer Science and Electrical Engineering, University of Maryland, Baltimore County, March 2007

[JKFO06] Java, Akshay ; Kolari, Pranam ; Finin, Tim ; Oates, Tim: Modeling the spread of influence on the blogosphere. In: *WWW 2006 Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2006

[JL07] Jure Leskovec, Christos F.: Cascading Behavior in Large Blog Graphs. (2007)

[JL09] Jure Leskovec, Lars B.: Meme-tracking and the Dynamics of the News Cycle. (2009)

[KKT03] Kempe, David ; Kleinberg, Jon ; Tardos, Éva: Maximizing the spread of influence through a social network. In: *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, 2003. – ISBN 1581137370, S. 137–146

[KKT05] Kempe, David ; Kleinberg, Jon ; Éva Tardos: Influential nodes in a diffusion model for social networks. In: *in ICALP*, 2005, S. 1127–1138

[Mar87] Mark, Granovetter: *Threshold models of collective behavior.* American Journal of Sociology, 1987

[Mar05] Mark, Eisenegger: *Reputation in der Mediengesellschaft.* Zürich, Schweiz : Verlag für Sozialwissenschaften, 2005

[MG09] Michael Götz, Mary McGlohon Christos F.: Modeling Blog Dynamics. (2009)

[Moo02] Moore, Geoffrey: *Crossing the Chasm.* New York : HarperBusiness, 2002

[Mor00] Morris, Stephen: Contagion. In: *Review of Economic Studies* 67 (2000), January, Nr. 1, S. 57–78

[NC08] Nallapati, R. ; Cohen, W.: Link-PLSA-LDA: A new unsupervised model for topics and influence of blogs. In: *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, Association for the Advancement of Artificial Intelligence, 2008

[PFL44] Paul Felix Lazarsfeld, Hazel G.: *The people's choice: how the voter makes up his mind in a presidential campaign.* Columbia : University Press, 1944

[RG43] Ryan, B. ; Gross, N.: The diffusion of hybrid seed corn in two Iowa communities. In: *Rural Sociology* 8 (1943), Nr. 1, S. 15–24

[RR03] Rogers, Everett M. ; Rogers, Everett: *Diffusion of Innovations, 5th Edition.* Free Press, 2003

[SCHT07] Song, X. ; Chi, Y. ; Hino, K. ; Tseng, B.: Information Flow Modeling based on Diffusion Rate for Prediction and Ranking. In: *Proceedings of the 16th International World Wide Web Conference*, 2007

[Val95] Valente, Thomas W.: *Network models of the diffusion of innovations.* Cresskill, N.J. : Hampton Press, 1995 (Quantitative methods in communication). – Thomas W. Valente. Includes bibliographical references (p. 153-163) and indexes.

[WFI94] Wasserman, Stanley ; Faust, Katherine ; Iacobucci, Dawn: *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences).* Cambridge University Press, November 1994

[WHAT03] Wu, Fang ; Huberman, Bernardo A. ; Adamic, Lada A. ; Tyler, Joshua. *Information Flow in Social Groups.* May 2003

[WS98] Watts, D. J. ; Strogatz, S. H.: Collective dynamics of 'small-world' networks. In: *Nature* 393 (1998), June, Nr. 6684, S. 440–442. – ISSN 0028–0836

[ZXJ08] Zhai, Zhongwu ; Xu, Hua ; Jia, Peifa: Identifying Opinion Leaders in BBS. In: *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on* 3 (2008), S. 398–401

Affirmation

Hereby I declare that I, Marc Wessely, have written this diploma thesis by my own. Furthermore, I confirm that no other sources have been used than those specified in the thesis itself.
This thesis, in same or similar form, has not been available to any audit authority yet.

Eidesstattliche Erklärung

Hiermit versichere ich, Marc Wessely, dass ich die vorliegende Diplomarbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Wörtliche und sinngemäße Zitate aus anderen Quellen habe ich als solche kenntlich gemacht.
Diese Arbeit, in gleicher oder ähnlicher Form, wurde bislang keiner anderen Prüfungsbehörde vorgelegt.

Rostock, October 23, 2009

.....
Marc Wessely