HELSINKI
INSTITUTE FOR
INFORMATION
TECHNOLOGY

Combining Topic Models and Social Networks for Chat Data Mining

Ville Tuulos and Henry Tirri

July 4, 2004

HIIT
TECHNICAL
REPORT
2004–13

Combining Topic Models and Social Networks for Chat Data Mining

Ville Tuulos and Henry Tirri

Helsinki Institute for Information Technology HIIT
Tammasaarenkatu 3, Helsinki, Finland
PO BOX 9800
FI-02015 TKK, Finland
http://www.hiit.fi

HIIT Technical Reports 2004–13
ISSN 1458-9478
URL: http://cosco.hiit.fi/Articles/hiit-2004-13.pdf

Combining Topic Models and Social Networks for Chat Data Mining

Ville H. Tuulos and Henry Tirri
Complex Systems Computation Group,
Helsinki Institute for Information Technology
P.O. Box 9800, FIN-02015 HUT, Finland.
{firstname}.{lastname}@hiit.fi

Abstract

Informal chat-room conversations have intrinsically different properties from regular static document collections. Noise, concise expressions and dynamic, changing and interleaving nature of discussions make chat data ill-suited for analysis with an off-the-shelf text mining method. On the other hand, interactive human communication has some implicit features which may be used to enhance the results.

In our research we infer social network structures from the chat data by using a few basic heuristics. We then present some preliminary results showing that the inferred social graph may be used to enhance topic identification of a chat room when combined with a state-of-the-art topic and classification models. For validation purposes we then compare the performance effects of using this social information in a topic classification task.

1. Introduction

The Internet is gaining more and more popularity as a medium for real-time communication. Increasing amount of information is produced by active discussions instead of static web pages. A popular IRC (Internet Relay Chat) channel directory, SearchIRC.com, lists currently 624,563 public discussion channels with over 1.2 million chatters. In addition to IRC channels, a vast number of web chats and other forms of instant messaging exist, forums which are popular especially among new chatters.

Naturally most of the public discussions are casual chitchats which are neither meant to be searched nor analyzed. Large-scale indexing of discussions is ethically problematic, and in many cases also unfruitful from the search point of view. However, there's a remarkable category of channels and chat forums which could benefit from search engines suited specifically for on-line chats. For instance many open-source projects have channels for developers and user support. Having this valuable information archived and analyzed with good search facilities could make their work more effective.

Typical keyword-based searches are far from optimal for this type of data. Of all the corpora of present natural language, chats have probably the worst spelling and grammar and the most level of noise. In order to ease typing, chatters tend to use acronyms and other concise expressions extensively which make simple inverted indices without semantic disambiguation very inaccurate. In addition, given a particular query, the inherently dynamic nature of discussions makes it non-trivial to identify relevant pieces of information. On the other hand the chat data is typically free from irrelevant structural, medium-related information which needs to be cleaned up, such as HTML mark-up tags in web pages.

We see chat search an interesting and important challenge for information retrieval. In our research we are focusing on utilizing state-of-the-art probabilistic models for topic identification in the context of chat data. By learning a topic model using a corpus of channels we may condense information about common subjects of discussions, and then use the model to enhance results of chat search and analysis.

However the document clustering and dimensionality reduction methods used for topic identification are typically based on the assumption that the corpus is a static set of discrete documents, written in decent language. This assumption clearly does not apply directly to the chat data. One might try to fit the data to the assumption using e.g., a sliding window over the discussion, and then considering each window as a document. As we will see, this approach doesn't take into account the peculiar nature of chat language, which tends to distort the analysis results.

In this paper we focus on an approach trying to remedy the problems discussed above. It is clear that the chat discussion has a lot of structure beyond considering the individual single-line responses in isolation. Here we will utilize an implicit feature of chats, namely social relations between the chatters, to filter out probably irrelevant parts of

discussion. The idea is analogous to web-graph analysis, like Hubs and Authorities [15] and PageRank [18], which try to discriminate highly relevant web sites from uninteresting ones. However, instead of aiming at ranking we use the relevance scores to enhance the topic models.

This paper concentrates on combining topic models with social networks. Highly interesting aspect of time-series analysis and topic tracking is discussed elsewhere [3], [7]. Although in this paper the combination is used developed for topic identification of a chat channel, the techniques discussed may be utilized in other chat-related tasks like user profiling and signal separation, where the task is to separate interleaved threads of discussion. These interesting other aspects are left for further research.

2. Related work

To our knowledge there's no previous work on combining automatically inferred social networks and topic models. However, there are some published studies about both of the subjects separately. Due to the lack of publicly available large chat data sets, the field hasn't yet gained great interest as opposed to the currently prevailing research on static document corpora.

The simplest approach for document classification, as well as topic identification for a channel, is to use a term vector or bag of words to represent a document [17]. Dot product between a pair of vectors (documents) may then be used to measure their "semantic" similarity. This approach is used for topic identification in IRC data for example by MIT's Butterfly agent [10]. Obviously in most real contexts, a naive approach like this doesn't yield high classification accuracies. A somehow refined approach is taken by ChatTrack [2] which uses predefined concepts, and the well-known TF·IDF term weighting scheme [17] to aid classification.

A more sophisticated model, Independent Component Analysis [11], is used to identify and track chat channel topics in [16] and [3]. This approach is comparable to our use of topic models.

However, all the approaches discussed above focus on building models based on the (textual) content of the documents analyzed. As we discussed earlier, chat (or any "threaded" document data such as email) also has information in the response structure inherent to any discussion. This type of response structure is often addressed by modeling it by social networks, which are widely studied and utilized in different fields. In the context of chat, most of the studies about inferring social relationships automatically among chatters are focused on visualization and user interface aspects, like Chat Circles [9]. In [1] social graph analysis was used to partition newsgroup users into opposite camps with respect to the topic discussed. Interestingly enough like our context, this also is a problem where pure statistics based approach performs poorly without response structure, as the vocabulary used by the opposing parties tend to be nearly identical and thus does not provide enough information to classify discussants into separate camps. Our use of social networks is inspired by PieSpy [19] which uses a few basic heuristics to infer relationships of chatters in IRC channels.

3. Data

The data consists of transcripts of multiple discussion channels $C \in \mathcal{C}$. A channel is a sequence of U utterances, $u_1, ..., u_U$. Each utterance is a sequence of words, $\mathbf{u} = (w_1, ..., w_D)$. Possibly overlapping sets of words form topics. We assume that the number of topics, K, is fixed beforehand. This assumption can be justified either by the specific context where the channels are already identified, or it can be estimated using the standard probabilistic methods (for example similarly as the number of components is estimated in mixture modeling). Thus each word in an utterance is thought to be produced by a topic with certain probability, as discussed in section 5.

Each utterance is has an attached information of a channel participant $p \in \mathcal{P}_c$ generating the utterance. Additionally we know the exact moment t when it was uttered. Thus we may directly observe a channel as a sequence of triplets:

$$C = ((t_1, u_1, p_1), ..., (t_U, u_U, p_U)).$$

4. Social networks

We know that each utterance u_i in a discussion is generated by one individual chatter p_i. We also know that seldom discussions are only interleaved monologues of chatters. Instead many utterances are intentionally *targeted* at the certain chatter. Types of chats vary from cocktail-party style mixtures of dialogues and small group chats to almost lecture-type sessions, where only one speaker is active and the others listen. However in contrast to the real-world discussions, social conventions and conversational etiquette are much looser in chats, yet not non-existent. As a result, nature of discussions and relations between the chatters keep changing all the time.

As Internet relayed chats lack non-verbal aspect of communication, like eye contact and physical proximity, targeting of the utterances must be accomplished using mostly verbal cues. Practically all the chats show an user-definable nickname with each utterance. Thus a reply to an utterance may be conveniently targeted at a specific individual including his/her nickname to the reply. Quite established convention is to prefix a targeted utterance with the recipi-

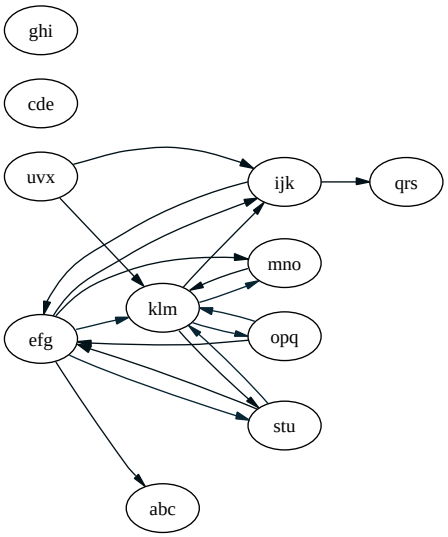

Figure 1. Example of an automatically inferred social network

ent's nickname and separate it from the actual content either with a comma or a colon.

However, targeting doesn't have to be this explicit. Actually chatters tend to use this explicit convention intuitively only when the recipient might be ambiguous. Quite often timing serves as an implicit cue for the intended recipient: If one reacts quickly to an utterance and the channel is otherwise rather inactive, quite probably the reaction is a reply. Maybe most importantly, in many cases the semantic content of an utterance is enough to make its recipient unambiguous.

Having the chat data, an interesting question raises: Would it be possible to automatically predict intended recipients for utterances? From the above discussion it should be clear that a perfect algorithmic solution is beyond our capabilities. However, at least explicit verbal cues and distinctive differences in timings should be detectable. As a result, we may visualize these who-talked-to-who relations as a directed graph. We call this graph a social network since it faintly resembles the small-scale social behavior of the chatters. We will show that these graphs may be utilized purely in a technical manner to enhance workings of e.g., a chat search engine.

4.1. Heuristics

We aim at extending the observed utterance triplets with heuristically estimated target information. Some utterances are deliberately directed at a certain participant $r \in \mathcal{P}_c \cup p^\lambda$, where p^λ denotes a missing target. Thus we may reformulate channel sequences as 4-tuples

$$\widehat{C} = ((t_1, u_1, p_1, r_1), ..., (t_U, u_U, p_U, r_U)).$$

Note that only channel participants p and their utterances u are directly observable. Possible target r of an utterance is implicit and must be inferred using some heuristics, as explained below.

For each item of sequence $\delta \in [1...U]$, we may extract a subset of the channel $\widehat{C}_\delta = ((p_1, r_1), ..., (p_\delta, r_\delta))$ i.e., only the participants and their targets who have been active up to this moment. This induces a graph, namely a *social network*, based on $\widehat{C}_\delta$. Let the graph be denoted by $\mathcal{G}_\delta = (V, E)$. Set of vertices V correspond to the set of channel participants, $V = \mathcal{P}_c$ and set of edges E represent who-talked-with-who -relationships, induced by $\widehat{C}_\delta$. Actually the graph $\mathcal{G}_\delta = (\mathcal{P}_c, \widehat{C}_\delta)$ is a multigraph, since $\widehat{C}_\delta$ may contain multiple identical elements. Elements with $r = p^\lambda$ are ignored.

We use a few rather simple heuristics to infer the social network based on observed utterances u, their timings t and the corresponding set of chatters $\mathcal{P}_c$. The heuristics are the same used by PieSpy[19] with some minor refinements. The heuristics are as follows:

Explicit reference is a nickname in the beginning of an utterance. However not an exact match is needed. Quite commonly chatters shorten or simplify complex nicknames so we use a regular expression to find matches which take this habit into account.

Immediate reaction happens when a line uttered after a longish silent period of time gets a reply within a certain short time span. We set the minimum silent period threshold to 120 seconds and the maximum reply delay to 20 seconds.

Dialogue is a moderately fast-paced sequence of utterances by two chatters. We set minimum sequence length to 5 lines which must occur within 180 seconds.

In our experience, these seemingly simple heuristics seem to capture coarse relations between the chatters quite decently. However there's clearly room for further research and theoretically better justified approaches. An example graph, automatically inferred using the above heuristics is shown in Figure 1. A node denotes a chatter and an edge existence of an utterance targeted at the pointed chatter. The figure is inferred from a chat excerpt of about 200 utterances. The actual nicknames of the chatters are removed to protect their privacy. One can easily notice that the graph has some distinctive characteristics. In particular, it is far from random.

Table 1. Example topics

Programming	School	Family	Chat	Iraq	Politics	Q&A	Gravity
C	physics	her	channel	iraq	government	question	force
use	math	she	u	war	vote	ask	field
it	book	women	banned	bush	liberal	answer	magnetic
program	stuff	woman	op	use	democratic	help	gravity
compiler	problem	child	ask	nukes	right	book	move
book	good	children	talking	korea	republican	read	object
code	course	man	go	weapons	political	one	point
windows	class	parents	discussion	saddam	party	try	acceleration

5. Topic Model

This paper focuses on short sequences of utterances aka utterance snippets in contrast to time-series. A typical application could be a chat search engine which is given a few example utterances as a query. The engine should estimate latent topic(s) in the query and return channels or archived snippets having similar content. Note that as the query snippet gets shorter, estimation of the topics gets harder rather steeply. Thus based on a single, typically very brief, utterance it's practically impossible to do any reliable inference. Actually an equivalent query estimation problem is faced by conventional topic-based search engines (see e.g. [6]).

Having this setting in mind, we may regard a short sequence of utterances as a bare bag of words. With this respect the setting is equivalent with static document collections. Thus we may use phrases "utterance snippet" and "document" interchangeably in this context.

Let J denote the total number of lexemes (words) in our lexicon. Bag of words representation for a document D_i is a J-dimensional vector $\mathbf{w}_i$, where the jth component of $\mathbf{w}_i$ gives the number of occurrences of word w_j in the document. In the *multinomial PCA* (mPCA) approach [4] the document collection is modeled by assuming that the words are generated from K probability distributions, where K is a much smaller number than the number of lexemes J. Thus we expect that the words have some redundancy given the topic of discussion. Each of these K probability distributions can be represented as a J-component vector where the jth component gives the probability for occurrence of the word w_j in the context of this topic. As these probability distributions define which words occur together with high probability, we may concisely call them "topics". However one should be careful with the topic-intuition: Certainly not all the topics in this statistical sense have a meaningful semantic interpretation. Some of them might model an abstract subject of discussion whereas some others correspond solely with syntactical constructs of language.

Let Ω denote a $J \times K$ matrix, where the jth column gives the probabilities for term w_j in each of the K topic distributions. Now, intuitively it makes sense that a textual document may contain text from several topic distributions, i.e., a single document can be related to several different topics. Correspondingly a snippet of utterances contains probably many different threads and levels of discussion. In the mPCA approach this is modeled by assuming that the text generating probability distribution for each document is a weighted linear combination of all the topic distributions. More formally,

$$\mathbf{w}_i \sim \text{Multinomial}(\boldsymbol{\theta}_i, \Omega, L_i), \tag{1}$$

where L_i denotes the number of words in document D_i, and $\boldsymbol{\theta}_i$ gives the mixing coefficients of the text generating probability distribution corresponding to document D_i. The prior distribution for the vectors $\boldsymbol{\theta}_i$ is usually assumed to be the Dirichlet distribution, the conjugate distribution of the Multinomial.

Intuitively, the components of the the vector $\boldsymbol{\theta}_i$ reveal to what extent document D_i addresses each of the topics. Consequently, as discussed in [8], mPCA can be seen as a multi-faceted clustering method, where each document belongs to each cluster (topic) with some probability. On the other hand, the model can also be viewed as a dimensionality reduction scheme: for those familiar with standard principal component analysis or equivalently with latent semantic analysis [17] in the document modeling context, it is evident that the above model is a discrete equivalent for the standard PCA with the Gaussian data generating function replaced by the multinomial.

In summary, so far we have three different representations for a snippet of utterances $D_1, \ldots, D_I$. First, they can be seen as strings of words. Second, ignoring the ordering of the words, they can be thought of as word count vectors $\mathbf{w}_1, \ldots, \mathbf{w}_I$. Third, they can be treated as topic probability vectors $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_I$.

The third model is especially suitable for the chat data analysis since it compresses relevant aspects of the discussion to a concise representation. Subsequent analyzes may now be built on top of this representation which should be more robust than word-level approaches since it doesn't rely solely on occurrence of exact forms of individual words. Thus we may be able to infer the topics reliably although the

snippet might contain unseen expressions and typos. Moreover the model may be seen to contain a form of implicit semantic disambiguation as ambiguous words may be compensated by co-occurring non-ambiguous words. We illustrate some example sets of words having high probabilities in selected topics in table 1. For details about the data and the model used see section 7.

There are multiple ways to estimate the model parameters, $\boldsymbol{\theta}$ and Ω. For details of the estimation procedure, see e.g., [5]. Since we use $\boldsymbol{\theta}$ as an alternative document representation and not just to cluster words together, we have to be careful with the estimation method. In our experience Gibbs sampling tends to produce better estimates than, say, the mean field method.

6. Socially enhanced topic model

We are now ready to present the following hypothesis: Topic-wise relevance of a chatter may be approximated using characteristics of the corresponding social network. By topic-wise relevance we mean the proportion of content words in chatter's and its neighborhood's utterances which are in accordance with the "true" topic of the current discussion. In other words, by using the social networks one may discriminate sources of noise from the actual signal – by reducing the noise and amplifying the signal. The intuition here is roughly the same as with web-graph analysis.

Due to preliminary nature of this study we are interested in the following basic questions: Which features of social networks might be beneficial having the above task in mind, and how this auxiliary information could be utilized. We will present our attempt to study the former question in section 6.1 and then the latter in 6.2. The tracks are not independent since the empirical results obtained with the graphs affected the design of the sampling model. The corresponding empirical settings are explained in section 7.

6.1. Graph features

We studied effects of the following features to the topic identification accuracy. The features are used to give a relevance score to each of the chatters in the examined channel. The features are extracted from a graph based on the full excerpt of the channel activity. Each of the scores was scaled to range $[0..1]$ which was then used to weight individual utterances.

Indegree The total number of utterances directed at a chatter. In Figure 1. chatter "klm" has the highest indegree, 5. Rationale is that the relevant utterances get replied more probably than the irrelevant ones. We set cut-off range to $[0..20]$ edges.

Outdegree The total number of targeted utterances by a chatter. In Figure 1. chatter "efg" has the highest outdegree, 5. This measures basically targeted loquaciousness of a chatter. Again, the cut-off range was $[0..20]$.

Complementary outdegree Inverse of the previous or measure of silence. In figure 1. many chatters have complementary outdegree of zero, e.g., "abc". This was taken into account so as to see whether a bad feature could worsen the results.

PageRank Originally PageRank[18] was meant to remedy problems of the bare indegree score for the web. Rationale is that even though a web page, or equivalently a chatter, may get referred by many others, only the referrals of authoritative sources really matter. We were interested to see whether this holds for this kind of social networks also.

Note that the scores aren't mutually exclusive. For instance in Figure 1. chatter "klm" has both high indegree and outdegree.

Using the above measures each chatter is given a weight between $[0..1]$. To get some baseline results for further theoretical work on how the weights should be embedded to the topic model, we tested two different approaches to modify the data itself to stress the relevant aspects. In filtering point of view this is natural, as we may concretely get rid of noise using the weights. Another approach is to "bias the sample" so that the relevant snippets of utterances would outnumber the irrelevant ones and thus ease the parameter estimation in the topic model. We compared the following weighting schemes:

Filtering Each utterance of a chatter is cloned multiple times. Number of clones is determined by $\lfloor weight_c \cdot F \rfloor$, where $weight_c$ is the relevance weight for the chatter and F is a global multiplication factor. Typically we used $F = 20$. Intuition behind this approach is that we increase probability of content words in a snippet of utterances in price of irrelevant words.

Biased sample As the topic model is based on a corpus of snippets, we may affect the results by biasing the sample. We get a weight for a snippet by taking the average of the weights of utterances within it. As with the filtering approach, we may now clone each snippet according to its weight. Rationale is that the corpus-level probabilities of irrelevant snippets get smaller.

6.2. Utterance Sampling

The above setting tries to illustrate how different characteristics of social networks affect topic estimation. As

results in tables 2 and 3 show, the differences are moderately small. Moreover simple features like indegree and outdegree perform well when compared to more sophisticated metrics such as PageRank. Thus we may formulate a more rigorous utterance expansion method based on these features.

The bag of words assumption allows us to ignore permutation of words. Thus instead of looking at the utterances *per se*, we construct a pool of timestamped words for each $p \in \mathcal{P}_c$. In practice the pool is a set of word-timestamp pairs uttered by a chatter. Consider a tuple (t_i, u_i, p_i, r_i) in a channel sequence. We don't use u_i directly to estimate the topic, since it would be probably too scarce for that. Instead, we may sample arbitrarily many words from the combined pool of p_i and its neighborhood $N(p_i, \mathcal{G}_i)$.

Let D denote number of words in the expanded utterance. We assume that part of the words are generated by p_i itself and the rest by its neighbors. Let $\nu \in [0..1]$ denote a free parameter measuring proportion of words generated by p_i. Thus exactly νD words will be generated by p_i. Rest of the words will be generated by neighborhood $N(p_i, \mathcal{G}_i)$ according to empirical frequencies of edges. We define the following probabilities:

$$P(p_j|E = in, N(p_i, \mathcal{G}_i)) = \frac{n(E = in, p_j)}{\sum_{k \in N(p_i, \mathcal{G}_i)} n(E = in, p_k)}$$

$$P(p_j|E = out, N(p_i, \mathcal{G}_i)) = \frac{n(E = out, p_j)}{\sum_{k \in N(p_i, \mathcal{G}_i)} n(E = out, p_k)}$$

i.e. the probability that an utterance by p_i will be directed at its neighbor p_j or that an utterance by p_j will be directed at p_i. Again, we are not sure which of the directions should be given the highest weight, so we let it modifiable by a parameter $\kappa \in [0..1]$ defining the mutual proportions of the edge directions. Thus in total $D(1-\nu)[\kappa P(p_j|E = in, N(p_i, \mathcal{G}_i)) + (1-\kappa)P(p_j|E = out, N(p_i, \mathcal{G}_i))]$ words will be produced by neighbor p_j. The generative process for each expanded utterance is as follows:

1. Choose a number of words D.
2. For each neighbor $p_j \in N(p_i, \mathcal{G}_i))$:
   (a) choose number of words by this neighbor

$$D_j = D(1-\nu)[\kappa P(p_j|E = in, N(p_i, \mathcal{G}_i)) + (1-\kappa)P(p_j|E = out, N(p_i, \mathcal{G}_i))]$$

   (b) For each of the D_j words:
      i. Choose a word w from p_j's pool $\bigcup_{k=1}^{i} u_k^j$ i.e. the words uttered by p_j this far.
3. Go to 2b and generate νD words for the p_i itself, then quit.

In step 2b we have to take into account that words in the pool expire in time due to topic evolution. Currently we let the words faint exponentially i.e. probability for a word to be selected is distributed according to an inversely exponential distribution.

7. Empirical Evaluation

Evaluation of unsupervised language models is a non-trivial task. The language itself doesn't contain any objective ways to measure, say, accuracy of topic identification. Some auxiliary information is needed for objective evaluation. In our context the most practical choice is the channel name although it's not the most interesting one from real application point of view.

Note that this setting is somehow artificial. Knowing the nicknames of chatters per each channel beforehand eases enormously prediction of an unseen snippet if the nicknames are shown. However, identifying the channel *per se* is not very interesting. Instead, we want to show that by just looking at the content with aid of the social networks, we can give accurate estimates about the topics. This could be applied for instance to track topic changes in time within a channel. Unfortunately we do not have the needed auxiliary information, which in practice someone would have to tag manually, to evaluate this setting directly. Luckily the results obtained with the chosen channel classification setting generalize to the tracking case also.

7.1. Data

We collected some 650 megabytes of chat data between October 30th and November 26th 2003 with our Irchiver data collection bot [12]. Guaranteeing privacy of chatters was of utmost importance even though we collected data only from public channels. We provided all the details where and how the data is to be used. Also we agreed that the dataset won't be distributed elsewhere. This formed the basis for our corpus which was later accompanied with some other similar sets of data.

The data consists of lines, each line having an utterance and chatter's nickname, timestamp and the channel name. All the channels are in English. For evaluation we selected six channels with distinctive topics: Bible, C++, Politics, Philosophy, Windows2000 and Physics. However due to casual nature of the discussions, topics vary from time to time also within a channel. For each channel an excerpt of 200,000 consecutive utterances were extracted starting from the beginning of each channel log.

7.2. Evaluation of the graph features

Each channel was splitted to 8000 snippets of 25 utterances totaling to 48,000 "documents". We preprocessed the snippets by removing the 100 most frequent words. Also any references to the nicknames of chatters were removed from the utterances so that the topic model couldn't predict the channel topic just by looking at the participants. The topic model gives us topic distributions per each document. Each topic of the distribution can be seen as generating a number of words in each document. Formally, for the document D_i we get a K-dimensional vector $\mathbf{u^i}$ suitable for classification with

$$u_k^i = L_i \boldsymbol{\theta}_k^i, \tag{2}$$

where L_i is word count for document D_i. The vectors were normalized to unit norm. We employed the SVM^{light} V5.0 [14] classifier with default settings. We noticed that if the snippet length is increased to 50 or even 100 utterances, we obtain classification accuracies of 95%-98% even with the basic topic model. This indicates that the classification task is meaningful indeed. However benefits of the combined model become more evident when the snippets aren't so abundant of topical cues.

We compared how the different heuristics and weighting schemes affect to the per-channel classification accuracy. As the SVM^{light} classifier works with binary classes, we trained a classifier for each channel separately. Firstly for each channel a social network was inferred, using one of the scoring methods. Then for each scoring and weighting scheme a topic model was learned from all the data, totaling to 10 different topic models. Equation 2. was used to obtain 10 sets of vectors suitable for classification from topic distributions.

For validation we used an approach closely resembling 10-fold cross-validation, but which has balanced test sets. Ordinary 10-fold crossvalidation would use 10% of data for testing and 90% for learning. However due to our multiclass setting only $1/6$ of the training and test samples would be positive and $5/6$ negative for each of the classifiers. Thus a classifier could obtain $5/6$ classification accuracy just by guessing.

We modified the setting so that we splitted the positive samples to 10 non-overlapping sets. One of the sets was taken as a test set at time and the rest were used for training. Then we took a random permutation of the negative samples and picked the first samples for testing so that the test set would contain as many negative and positive samples. The training set was filled with the remaining negative samples so that 90% of the data would be used for training. With this scheme the default accuracy for a classifier is 50%. Due to the random aspect of the validation, we repeated each step ten times and calculated the average. In total, classification accuracies presented in the tables 1. and 2. are averages of 100 runs per class. We observed only slight variation between the runs.

Table 2. Filtering: Accuracy per class

Channel	mpca	IDeg	ODeg	CDeg	PRank
Bible	83	89	88	83	88
C++	87	90	88	87	87
Philosophy	86	89	87	82	87
Physics	85	89	89	85	87
Politics	83	86	87	83	81
Win2000	90	91	91	86	91

Table 3. Biased sample: Accuracy per class

Channel	mpca	IDeg	ODeg	CDeg	PRank
Bible	83	88	87	86	86
C++	87	90	90	88	88
Philosophy	86	87	88	86	87
Physics	85	89	89	87	79
Politics	83	85	81	85	87
Win2000	90	89	91	90	91

7.3. Results

Classification accuracies per each class are shown in Table 2. and Table 3. Table 2. contains the results for the filtering scheme and Table 3. for the biased sample scheme. Column "mpca" shows the basic topic model without any use of social networks. Ideg stands for indegree scoring scheme and Odeg for outdegree, CDeg for complementary outdegree and PRank for PageRank correspondingly.

Differences in the results are not large but some trends are distinctive. Firstly, filtering scheme seems to work slightly better than biased sample in this setting. However in both cases sensible use of the social networks tends to improve the results. As we hypothesized, complementary outdegree worsens the results, but maybe not as much as one would expect. Not surprisingly, indegree seems to be the clear winner of the scoring schemes. People that get replied a lot probably have some sense in their talks. PageRank which enhances indegree in the web context doesn't shine here as much. Maybe authoritative persons are not keen to refer each other extensively. As the small difference between the outdegree and indegree show, the scores actually overlap heavily. That is, talkative chatters get replied a lot.

All in all, the general trend seems to favor usage of the social networks. We gained confidence that a simple measure like indegree could indeed serve as a scoring method. Yet our weighting schemes, filtering and sample biasing, are insufficient to produce large differences to the results.

Then on the other hand we explicitly did not want to overfit our methods to this rather straightforward classification task. If that were the case, one would use a discriminative topic model instead of a totally unsupervised MPCA.

7.4. Evaluation of the utterance sampling

Our second test case justifies the sampling model in section 6.2. This time we didn't build topic models of utterances since we are interested purely in gains of the sampling model versus simpler sliding window approach. We took 40,000 consecutive lines of utterances from two channels, namely politics and philosophy which we discriminate using SVM. We constructed the social network based only on utterances seen before, so this case is realistic in on-line point of view.

In contrast to the previous test, in which utterance snippets were used, we performed the classification line by line. We employed SVM with radial basis function kernel, which is known of good performance in text classification tasks as reported by [13]. Our corpus contained 10894 different stemmed words. SVM was shown the bare bag of words, normalized to unit length. We used neutral (0.5) weights for κ and ν. Stratified 10-fold cross validation was used to validate the results.

The sampling model yielded classification accuracy of 87.0% on average per each line. In contrast a sliding window of 10 previous utterances yielded 91% accuracy and a window 2 previous utterances 78.7%. One should be careful in interpreting the results since the sampling model has in theory even more information available than the 10-line sliding window due to the word pools. Thus in a static classification task a (large) sliding window might be a better choice than the sampling model. However, in more dynamic cases the user is known and samples of utterances are scarce, the sampling model works well. This includes e.g. many cases in the context of search engines and especially topic estimation.

8. Conclusions

The research presented is preliminary in nature and can be enhanced in many different ways. Albeit the promising results for topic classification, more intricate incorporation of the social graph weights should improve the results. Similarly many aspects of the parameterization and the heuristics could be enhanced by more rigorous methodology. For instance, the sampling model could benefit from more sophisticated model for topic evolution.

Numerous applications could be built on top of these ideas. Here we concentrated only in improving topics by using the graphs. Another intriguing approach would be to enrich the graphs using the topics – we could e.g. color the nodes according to the interest profiles of the corresponding chatter.

9. Acknowledgements

This work was supported in part by the National Technology Agency, under the project Search-Ina-Box and by the Academy of Finland, under the projects Prose and Prima. This work was also supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

[1] R. Agrawal, S. Rajagopalan, R. Srikant, and Y. Xu. Mining newsgroups using networks arising from social behavior. In *Proc 12th International World Wide Web Conference*. ACM, 2003.

[2] J. Bengel, S. Gauch, E. Mittur, and R. Vijayaraghavan. Chattrack: Chat room topic detection using classification. In *2nd Symposium on Intelligence and Security Informatics (in review)*, 2004.

[3] E. Bingham, A. Kaban, and M. Girolami. Topic identification in dynamical text by complexity pursuit. In *Neural Processing Letters vol 17*, pages 69–83, 2003.

[4] W. Buntine. Variational extensions to EM and multinomial PCA. In *Proc 13th European Conference on Machine Learning*, 2002.

[5] W. Buntine. Applying discrete pca in data analysis. In *Proc. Uncertainity in Artificial Inelligence*, 2004.

[6] W. Buntine, P. Myllymaki, and S. Perttu. Language models for intelligent search using multinomial PCA. In *Proc. of 1st European Web Mining Forum*, 2003.

[7] W. Buntine, J. Perkiö, and P. S. Exploring trends in a topic-based search engine. In *Proc 2nd IEEE/WIC/ACM Conference on Web Intelligence*. ACM, 2004.

[8] W. Buntine and S. Perttu. Is multinomial PCA multi-faceted clustering or dimensionality reduction? In *Proc. 9th Int. Workshop on Artificial Intelligence and Statistics*, pages 300–307, 2003.

[9] ChatCircles. http://chatcircles.media.mit.edu/.

[10] N. V. Dyke, H. Lieberman, and P. Maes. Butterfly: A conversation-finding agent for internet relay chat. In *Proc Int. Conference on Intelligent User Interfaces*, 1999.

[11] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[12] Irchiver. http://cosco.hiit.fi/irchiver/.

[13] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc 10th European Conference on Machine Learning*. Springer, 1998.

[14] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[15] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc 9th ACM-SIAM Symposium on Discrete Algorithms*. ACM, 1998.

[16] T. Kolenda, L. Hansen, and J. Larsen. Signal detection using ica: application to chat room topic spotting. In *Proc. 3rd Int. Conference on Independent Component Analysis and Signal Separation (ICA2001)*, pages 540–545, 2001.

[17] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[19] PieSpy. http://www.jibble.org/piespy/.