International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 2, Issue 1, January 2014)

Discussion Summarization

N. Lalithamani[1], K. Alagammai[2], Kolluru Kamala Sowmya[3], L. Radhika[4], Raga Supriya Darisi[5], S. Shanmugapriya[6]

[1]Assistant Professor(SG), Department of Computer Science and Engineering, Amrita School of Engineering,
AMRITA VISHWA VIDYAPEETHAM, AMRITA NAGAR P.O., Ettimadai, Coimbatore - 641 112.
[2,3,4,5,6]4th year BTech, Computer Science and Engineering Students, Amrita School of Engineering,
AMRITA VISHWA VIDYAPEETHAM, AMRITA NAGAR P.O., Ettimadai, Coimbatore - 641 112.

***Abstract* — Discussion summarization is the process of condensing a text document which is a collection of discussion threads, using CBS (Cluster Based Summarization) approach in order to create a relevant summary which enlists most of the important points of the original thematic discussion, thereby providing the users, both concise and comprehensive piece of information. This outlines all the opinions which are described from multiple perspectives in a single document. This summary is completely unbiased as they present information extracted from multiple sources based on a designed algorithm, without any editorial touch or subjective human intervention. Extractive methods used here, follow the technique of selecting a subset of existing words, phrases, or sentences in the original text to form the summary. An iterative ranking algorithm is followed for clustering. The NLP (Natural Language Processing) is used to process human language data. Precisely, it is applied while working with corpora, categorizing text, analyzing linguistic structure. Thus, the quick summary is aimed at being salient, relevant and non-redundant. The proposed model is validated by testing its ability to generate optimal summary of discussions in Yahoo Answers. Results show that the proposed model is able to generate much relevant summary when compared to present summarization techniques.**

***Keywords* — Bi-Type graph model, clustering, discussion summarization, ranking, score calculation, TCC approach**

I. Introduction

There is always a necessity to know what is happening around the world and the dimensions in which people think over these issues. But, with this never-ending inflow of information, it gets difficult for any human being to devote some time to go through any discussion which runs a few pages or has a few set of comments. A summary is thus a shorter version that contains the key information of a document or a set of documents.

According to Mani and Maybury [13], text summarization is “the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks).”

The main goal of a summary is to paraphrase the main ideas from the threads of discussion. Summaries can be produced either from a single discussion or multiple discussions [9] pertaining to the same topic. Similarly the task of producing summary from many documents is called multi-document summarization [10]. Hence, in a nutshell, Discussion Summarization aims at presenting an extractive summary of a thematic discussion by clustering and ranking the discussion threads based on their similarity.

Clustering is the process of grouping a set of documents into clusters of similar documents. So, documents within a cluster should be similar. Clustering is the most common form of unsupervised learning. There is no labeled or annotated data in unsupervised learning. Clustering the given data according to similarity is the major task in text summarization. This paper proposes TTC (Two Tiered Cluster) with Soft clustering approach. In Soft clustering, each document (sentence in this case) can belong to more than one clusters.

Ranking is the process of retrieving sentences that are relevant to the discussion based on scores.

The dataset considered here is from the domain-Yahoo Answers (http://www.answers.yahoo.com). The data is retrieved from the former website using YQL (Yahoo Query Language) and pattern module. The data retrieved is pasted in a text file for further use. The TTC approach is followed here involves clustering (grouping) the contents based on the topic which is referred as TC (Topic Cluster) and in turn these TCs are clustered based on similarity which is known as SC (Similarity Cluster). The similarity of the sentences is obtained by the use of Word Net imported form NLTK (Natural Language Tool Kit).

The ranking of sentences within the clusters and across the clusters is done with the help of Bi-Type Graph [1]. Thus, the summary contains highly ranked sentences based on the desired length mentioned by the users. Here we propose an algorithm that ensures that the order of these sentences is preserved.

A short, quick, terse, relevant and comprehensive summary would solve this problem, thereby reducing user's time and effort to go through the entire discussion. The main contributions of this paper are:

1. An effective clustering is done by grouping the contents into a two-tiered clustering approach.
2. Ranking each sentences with the help of a Bi-Type graph model while considering the users rating the comments.

The rest of this paper is organized as follows. Section II reviews related work in text summarization methods. Section III (Proposed Model) defines the new clustering and ranking approaches and their application to discussion summarization. Section IV presents experiments and evaluations, whereas section V presents discussion on the result of the proposed model. Conclusions are presented in Section VI.

II. Related Work

There exist many approaches and algorithms for a general text summarization: extractive, abstractive, aided and maximum entropy [14]. Generally, extractive and abstractive summarization methods are used by many researchers for text summarization.

The extractive based summarization ensures that the terms, phrases or sentences are picked from the original text and presented in the summary whereas, the abstractive summarization is based on the semantics of the words or phrases. For the later, the summary is generated by the use of machine learning techniques. This paper focuses on extractive based summarization approach [15].

There exist numerous approaches to find the similarity between two sentences. Few of them are as follows:

A. Maximum Marginal Relevance (MMR)

MMR [17] is a widely used approach as it selects the most relevant sentences at the same time avoiding redundancy. This method is known for its simplicity and efficacy in text summarization. Shasha and Yang [16] had modeled a model in which the sentences with the highest MMR scores are iteratively chosen for the summary until the later reaches a predefined proper size.

In this paper, we have come up with an idea to present a summary of flexible size. In other words, the user has the privilege to specify the number of lines of summary.

B. Centroid Score

Another method to evaluate similarity measure is centroid score which calculates the distance between a sentence and the entire document. This method is similar to Cosine Similarity.

C. Cosine Similarity

This method is also a commonly used similarity measure. In this approach, each sentence is represented as a vector space model. The cosine similarity [8] is described mathematically as:

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q} \bullet \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|} \bullet \frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

Here, q_i is the tf-idf weight of term i in the query d_i is the tf-idf weight of term i in the document, cos(q,d) is the cosine similarity of *q* and *d* or, also as the cosine of the angle between *q* and *d*.

In our experiments, we found that the usage of cosine similarity can be very effective in the process of ranking. This method is suitable for finding out the weights of the edges between a pair of nodes. Note that, a node can be either a word or a sentence which are represented as W_{SS}, W_{ST}, W_{TS}, W_{TT} have been discussed later in section III (Proposed Model).

D. Corpus-Based Semantic Similarity

The similarity measures that we discussed till now are all based on simple lexical matching, that is, only the words that occur in both contribute to the similarity. This type of literal comparison cannot always capture the semantic similarity of text.

This paper uses the above model to evaluate the similarity measure. The mathematical formulae for this model are discussed in section III.

Ranking is an issue when we use extractive summarization. This happens when the comments are of the same discussion or theme as there is a possibility that some information might be repeated. In order to remove the redundancy, effective clustering of similar sentences should be done. The effective clustering which is followed here is achieved with the help of CBS approach [1]. The clustering results from the CBS approach is used to select the sentences from the original text to generate the summary.

Cosine, Centroid score and CBSS (Corpus Based Semantic Similarity) are few algorithms that are used to find the similarity between any two sentences. Only the words which are present in both the sentences contribute for computing the similarity measure in Cosine and Centroid scores, whereas in case of CBSS algorithm, all the terms are given weightage as per their respective tf(term frequency) and idf(inverse document frequency) values[16]. So, a modified version of CBSS algorithm is used here.

III. Proposed Model

The following are the steps that are performed in order to obtain a summary of the discussion threads on a particular theme (refer *Fig. 1*):

A. Retrieval of the data from Yahoo Answers
B. Clustering the sentences.
C. Ranking the sentences.
D. Selecting and Reordering the sentences.

For better results, this paper uses Ranking and Clustering by mutually and simultaneously updating each other as shown in the *Fig. 1*.

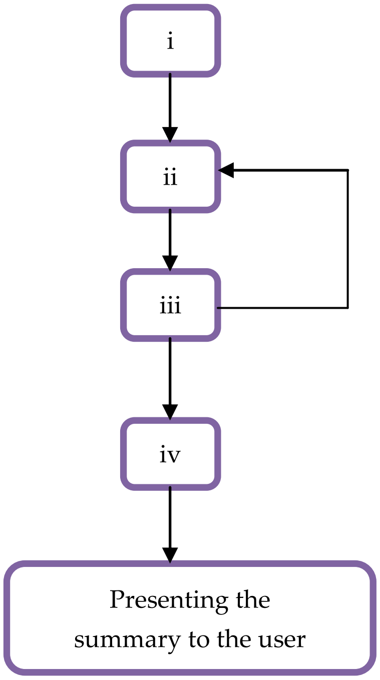

Fig. 1. Proposed Model

A. Retrieval of the data from Yahoo Answers:

Each discussion in Yahoo Answers has a question followed by a set of comments on any particular topic. Thus, every discussion is uniquely identified by a discussion id – qid(attribute). This qid is used in YQL to retrieve the data in the XML format, from which the data obtained is parsed to normal text using pattern module (in NLTK).

B. Clustering the sentences

For clustering, a novel approach is proposed which is known as TTC algorithm. Initially, it clusters the contents topic-wise and later each of these clusters is in turn clustered based on the similarity of sentences. In this algorithm, there are two types of clustering - theme clustering and similarity clustering.

Theme clusters are those which are formed by the sentences or comments that belong to the synonymous discussions. Similar clusters are those that contain the sentences which has the same meaning. A modified Corpus Based Semantic Similarity algorithm is used to find the similarity between two sentences. According to Shasha Xie and Yang Liu, the native Corpus Based Semantic Similarity algorithm is [16]:

$$sim(T1, T2) = 0.5 * (\mathrm{X} + \mathrm{Y})$$

$$\mathrm{X}=\frac{\sum_{w\in\{T1\}}(maxSim(w,T2)*idf(w))}{\sum_{w\in\{T1\}}(idf(w))}$$

$$\mathrm{Y}=\frac{\sum_{w\in\{T2\}}(maxSim(w,T1)*idf(w))}{\sum_{w\in\{T2\}}(idf(w))}$$

$$\mathrm{maxSim}(\mathrm{w},\mathrm{T}_i) = \mathrm{max}_{(wi\in\{Ti\})}\{\mathrm{sim}(\mathrm{w},\mathrm{w}_i)\}$$

The blue rectangles in the *Fig. 2* depict the theme clusters, the ones in red are for the similarity clusters and the green rectangles represent the sentences.

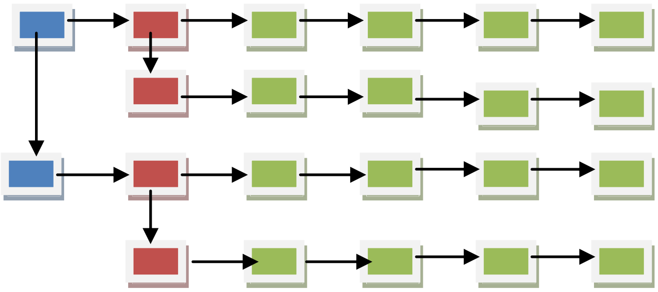

Fig. 2. Visual Representation of Clusters

For each word *w* in segment (sentence) *T1*, we find a word in segment (sentence) *T2* that has the highest semantic similarity to *w* (*maxSim*(*w*, *T2*)). Similarly, for the words in *T2*, we identify the corresponding words in segment *T1*.

The similarity score of the two text segments is then calculated by combining the similarity of the words in each segment, weighted by their word specificity (i.e., idf values). The value of $sim(w,w_i)$ is 1, if the word *w* and w_i is present in the sentence T_i.

The modified maxSim function proposed in this paper is:

maxSim(w,Ti) = a+b

a=1 , if w in Ti
0, otherwise
b=1 , if w_{j-1} and w_{j+1} in Ti
0, otherwise

In the above, the value of variable *a* is either 1 or 0 depending on the existence of the word in the sentence (1 if present and 0 if not). Similarly, variable *b* is 1 or 0 depending on the existence of the words adjacent (left or right) to the word w_i in the sentence T_i. Thus, this modified *maxSim* function results in better clustering of similar sentences than the native *maxSim* function.

C. Ranking the sentences

Ranking is the process of identifying the most important sentences (based on the score) which could form the summary. The ranking algorithm uses Bi-Type graph model as its key feature in determining the score, thereby rank of each sentence. Here, Xiaoyan Cai and Wenjie Li's proposed algorithm [1] is modified such that the ratings of the comments given by the different users on Yahoo Answers is considered. The original graph model states that:

$$r(s_i)=L+M$$

$$L = \sum_{i=1}^{m} W_{ST}(i,j) * r(t_j)$$

$$M = \sum_{i=1}^{n} W_{SS}(i,j) * r(s_j)$$

$$W = \begin{pmatrix} Wss & W_{ST} \\ W_{TS} & W_{TT} \end{pmatrix}$$

$$r(t_i) = P+Q$$

$$P = \sum_{i=1}^{n} W_{TS}(j,i) * r(s_j)$$

$$Q = \sum_{i=1}^{m} W_{TT}(i,j) * r(t_i)$$

The set of edges that connects the vertices is *E*.

An edge can connect any combination of sentences and words. *W* is the adjacency matrix in which the element represents the weight of the edge connecting two vertices (vertices are terms and sentences). *W* can be decomposed into four blocks, i.e. W_{SS}, W_{ST}, W_{TS} and W_{TT} each representing a sub-graph of the textual objects indicated by the subscripts [1].

$W_{ST}(i,j)$ is the cosine similarity between the sentence s_i and the term t_j. Thus, the value of $W_{ST}(i,j)$ is between 0 and 1. If $W_{ST}(i,j)$ is near 1, it means the sentence s_i and the term t_j are semantically similar. Else if, $W_{ST}(i,j)$ is near 0, it means the sentence s_i and the term t_j are semantic different. $W_{SS}(i,j)$ is the cosine similarity between the sentences s_i and the term t_j is equal to as the relationships between terms and sentences are symmetric. The $W_{TT}(i,j)$ value is the measure of cosine similarity between the terms t_i and t_j.

The initial values of $r(s_j)$ and $r(t_i)$ are modified in this paper as:

$r(s_j)$ = Number of Likes + 1
$r(t_i)$ = Term Frequency

The Number of Likes here refers to the ratings given by the user for a certain comment. Here, scalar value of 1 is added to $r(s_j)$ as an AOSF (Add One Smoothening Factor) because the Number of Likes may be none. In this way, the value of *P* is prevented from being 0.

D. Selecting and Reordering the sentences

In this module, we concentrate on content of the summary. At this stage, each of the sentences would have been given a score with the help of the ranking algorithm which is mentioned above. Now, the top-scored N sentences are retrieved, where N is the number of lines to be present in the summary as specified by the user.

Since we have the sentences which are to be present in the summary, now the challenge lying ahead is about the arrangement of these sentences. This issue is solved with the help of TC which contains the set of comments in an order. Compare each of the selected sentences from above with the sentences in the TC and get their relative positions in the summary.

So, the ordering of the sentences in the final summary is preserved. The summary obtained from the above module is written in a file and presented to the user.

IV. Analysis Of Results

Sample input file for the discussion summarization's proposed model is given in *Fig. 3*. The input file contains the text from the discussion form which is followed by a number (number of likes given by users).The data, in other words, the text as well as the numbers are retrieved from the parsed version of discussion's xml data. The number (number of likes), which is highlighted in blue in *Fig. 3*, is appended at the end of every comment which belongs to particular discussions.

For this particular input file, the domain is related to Superstitions. Here, we take the support of a dictionary with all keywords under the topic, Superstitions.

If a black cat crosses your way, it's bad luck for you. Turn around, take another route. Take another path and turn around. This shows that from very ancient times, Indians knew what we were coming to. So-called VIPs (whose importance is only to their own immediate family and friends) with black cat security are known to create traffic problems wherever they go with their cavalcades. So if you spot, or even sense any of them, it is best to turn around and take another route. 10.

Sneezing before doing something good/big is a bad omen. I have not heard of such a belief, but let us think of what it might signify. Good/big things usually involve large gatherings. In these days of H1N1 and similar viruses that can spread through air, if an infected person sneezes, it can be considered a bad sign for others in the gathering. 8.

In early Egyptian times, Black cats were iconic character in Animal world. 5.

It was until then status of Cats started getting associated with witches in Europe. 0.

Fig. 3. Content of the input file

The next step is to cluster the content topic-wise, which is the first process in TTC that is, to generate TC. Considering the input file in *Fig. 3*, the number of clusters formed is two. They are *cat* and *sneeze*.

The algorithm evaluates each comment and compares each term with the keywords in dictionary on Superstitions.

This is how it gets to the conclusion of having two topic clusters.

If a black cat crosses your way, it's bad luck for you. Turn around, take another route.

Take another path and turn around.

This shows that from very ancient times, Indians knew what we were coming to.

So-called VIPs (whose importance is only to their own immediate family and friends) with black cat security are known to create traffic problems wherever they go with their cavalcades.

So if you spot, or even sense any of them, it is best to turn around and take another route.

In early Egyptian times, Black cats were iconic character in Animal world.

It was until then status of Cats started getting associated with witches in Europe.

Fig. 4. Contents of *Cat* topic cluster

Sneezing before doing something good/big is a bad omen.

I have not heard of such a belief, but let us think of what it might signify.

It was until then status of Cats started getting associated with witches in Europe.

I have not heard of such a belief, but let us think of what it might signify.

Good/big things usually involve large gatherings.

In these days of H1N1 and similar viruses that can spread through air, if an infected person sneezes, it can be considered a bad sign for others in the gathering.

Fig. 5. Contents of *Sneeze* topic cluster

The second process in TTC is to generate similarity clusters SC, for each sentence in TC. In the *Fig. 4* and *Fig. 5*, various colors are used to represent similarity clustering. In other words, the sentences belong to the same cluster have same colors.

After the clustering of the content, the next task is to rank the sentences in the SC based on score as specified in section III.

This shows that from very ancient times, Indians knew what we were coming to.
So if you spot, or even sense any of them, it is best to turn around and take another route.
It was until then status of Cats started getting associated with witches in Europe.
I have not heard of such a belief, but let us think of what it might signify.
Good/big things usually involve large gatherings.
Sneezing before doing something good/big is a bad omen.
If a black cat crosses your way, it's bad luck for you.
In these days of H1N1 and similar viruses that can spread through air, if an infected person sneezes, it can be considered a bad sign for others in the gathering.
In early Egyptian times, Black cats were iconic character in Animal world.
So-called VIPs (whose importance is only to their own immediate family and friends) with black cat security are known to create traffic problems wherever they go with their cavalcades.

Fig. 6. Content ranked in order

The *Fig.6* represents the sentences in order of their rank.

Cat: If a black cat crosses your way, it's bad luck for you. This shows that from very ancient times, Indians knew what we were coming to. So if you spot, or even sense any of them, it is best to turn around and take another route. It was until then status of Cats started getting associated with witches in Europe.

Sneeze: Sneezing before doing something good/big is a bad omen. I have not heard of such a belief, but let us think of what it might signify. Good/big things usually involve large gatherings.

Fig. 7. Content of the summary

The *Fig. 7* represents the summary of N=7 sentences, which is of the length specified by the user and in the order of TC.

Thus, the summary obtained is free from the redundant data and only contains the user specified top relevant sentences.

V. Discussions

Including the stop words in the process of ranking and clustering might mislead the results. So for better results, we are not considering the stop words in ranking or clustering.

Redundant data or similar sentences might have a bad influence on ranking. So, we have identified the similar sentences before ranking and ordering. This makes the summary more efficient in the aspects of relevancy.

VI. Conclusion And Future Enhancements

This paper focuses on two major functions that is, Clustering and Ranking. A TTC approach is used in this paper which helps in effective clustering of the sentences. A Bi-Type graph model is followed for ranking, same algorithm is used for both ranking within the cluster and across the cluster. Clustering and Ranking within the cluster is done simultaneously. Users rating are considered during ranking process both within and across the clusters.

The proposed model can be extended to deal with the sentences consisting short-hand representations of words (informal chatting language) by introducing a dictionary for the same. A couple of machine learning algorithms can be used to detect sentences which does not carry any information.

References

[1] Xiaoyan Cai and Wenjie Li, "Ranking Through Clustering: An integrated approach to multi-document summarization", IEEE Transactions on Audio, Speech and Language Processing, Vol. 21, No. 7, July 2013.

[2] S. Fisher and B. Roark, "Query-focused summarization by supervised sentence ranking and skewed word distributions," in Proceedings Document Understanding Conference, 2006

[3] Xiaoyan Cai and Wenjie Li,"Mutually reinforced manifold-ranking based relevance propagation model for query-focused multi-document summarization", IEEE Transactions on Knowledge And Data Engineering, Vol. 25, No. 8, August 2013.

[4] Hien Nguyen, Eugene Santos, and Jacob Russell, "Evaluation of the impact of user-cognitive styles on the assessment of text summarization" IEEE Transactions on Systems, Man and Cybernetics, Vol. 41, No. 6, November 2011.

[5] Chien Chin Chen and Meng Chang Chen, "A content anatomy approach to temporal topic summarization" IEEE Transactions on Knowledge And Data Engineering, Vol.24, No. 1, January 2012.

[6] Elias Iosif and Alexandros Potamianos, "Unsupervised semantic similarity computation between terms using web documents" IEEE Transactions on Knowledge And Data Engineering, Vol. 22, No. 11, November 2010.

[7] Davide Falessi, Giovanni Cantone, and Gerardo Canfora, "Empirical principles and an industrial case study in retrieving equivalent requirements via natural language processing Techniques" IEEE Transactions on Software Engineering, Vol. 39, No. 1, November 2013.

[8] Manning, C.D., Raghavan, P. and Schütze, H. (2009) An Introduction to Information Retrieval. Cambridge, England: Cambridge University Press.

[9] X. Wan, “Towards a Unified Approach to Simultaneous Single-Document and Multi-Document Summarizations,” Proceedings 23rd International Conference Computational Linguistics, pp. 1137-1145, http://portal.acm.org/ citation.cfm?id=1873781.1873909, 2010.

[10] X. Wan, “An Exploration of Document Impact on Graph-Based Multi-Document Summarization,” Proceedings Conference Empirical Methods in Natural Language Processing, pp. 755-762, http://portal.acm.org/ citation.cfm?id=1613715.1613811, 2008.

[11] Celikyilmaz and D. Hakkani-Tur, “Discovery of topically coherent sentences for extractive summarization,” in Procedings 49th Association for Computational Linguistics Conference 2011, 2011, pp. 491–499.

[12] H. Lin and J. Bilmes, “A class of sub modular functions for document summarization,” in Proceedings 49th Association for Computational Linguistics Conference, 2011.

[13] Mani and M. T. Maybury, Advances in Automatic Text Summarization. Cambridge, MA: MIT Press, 1999.

[14] Automatic Summarization online: http://www.en.wikipedia.org/wiki/Automatic_summarization.

[15] L. Antiqueris, O. N. Oliveira, L. F. Costa and M. G. Nunes, “A complex network approach to text summarization,” Information Sciences, vol. 175, no.5, pp. 297–327, February 2009.

[16] Shasha Xie, Yang Liu, ” Using Corpus And Knowledge-Based Similarity Measure In Maximum Marginal Relevance For Meeting Summarization, ” The University of Texas at Dallas, Richardson, TX, USA.

[17] Jaime Carbonell and Jade Goldstein, “The use of MMR, diversity-based reranking for reordering documents and producing summaries,” in Proceedings of Special Interest Group in Information Retrieval, 1998.

[18] A. Nenkova, “Automatic Text Summarization of Newswire: Lessons Learned from the Document Understanding Conference,” Proceedings 20th National Conference on Artificial Intelligence (AAAI), pp. 1436-1441, 2005.

[19] J. G. Corbonell and J. Goldstein, “The use of MMR, diversity-based reranking for reordering documents and producing summaries,” in Proceedings 21st Special Interest Group in Information Retrieval Conerenc., 1998, pp. 335–336

[20] V. Qazvinian and D. R. Radev, “Scientific paper summarization using citation summary networks,” in Proceedings 17th Computational LinguisticsConference, 2008, pp. 689–696.

[21] D.R. Radev and K.R. McKeown, “Generating Natural Language Summaries from Multiple Online Sources,” Computational Linguistics, Vol. 24, No. 3, 1998, pp. 469-500.

[22] Summarizing Text for Intelligent Communication Symposium (Dagstuhl, Germany, 1993); http://www.ik.fh-hannover.de/ik/projekte/Dagstuhl/Abstract