Toward indicative discussion fora summarization *

Mike Klaas
University of British Columbia
klaas@cs.ubc.ca

Abstract

Summarization of electronic discussion fora is a unique challenge; techniques that work startlingly well on monolithic documents tend to fare poorly in this informal setting. Additionally, conventional techniques ignore much of the structures that have the potential to serve as valuable features in the summarization task. We present several novel examples of such features, including the *catalyst score*, which is effective at identifying salient messages without looking at their content. We also describe and evaluate NewsSum, a prototype summarization system that is able to efficiently generate variable-length summarizations of Usenet threads.

Introduction

Information overload is a commonly-cited justification for the study of text summarization, and it is no less a justification today as it was in 1950's, when the ground-breaking work in summarization was commencing (Luhn, 1958). The domain of scientific papers has perhaps been the most-studied due to the highly-consistent and indicative structure of its documents, and the inherent value of summaries in exponentially-growing scientific literature. Other more limited and structured domains (such as news articles) have also been studied, but very little effort has been expended into less formal corpora. Discussion fora are ubiquitous examples of such, and have just as pressing a need for summarization; one of our study participants described the task of following the huge amount of content "overwhelming." The principle motivation of this project is to develop methods of summarizing a medium famously described[1] as populated by "anarchists, lunatics, and terrorists": Usenet.

Document summarization is not a new area, but only recently have discussion fora been looked at specifically. (Farrell et al., 2001) note that existing summarization tools perform poorly when applied to discussion media. This is due to the fragmented, informal nature of the medium. They correctly point out that the structure of the medium is essential to developing effective discussion summarization techniques, and present some first steps in that direction. Email summarization is examined in (Lam et al., 2002) which is a further restriction on the problem we consider. The analysis and visualization of thread structure is tackled in (Newman, 2002). Newman correctly identifies the importance to thread structure in understanding the discourse structure of a discussion.

The techniques in the current research do not focus on compression: all messages are summarized (usually lengthily), and thus are largely inadequate for summarization of fora of extremely high traffic such as Usenet. In this setting, highly-compressed, indicative[2] summaries are the desiderata. This can most easily be achieved by being judicious in the selection of messages to include in the summary, an area to our knowledge hereto untouched in the literature.

To this end, we present several features that can be used in discussion summarization, including our novel *catalyst score* that leverages thread structure. We also investigate the use of state-of-the-art term-recognition techniques to aid summarization in this medium. We present NewsSum, our summarization

* Originally written in April 2004, revised as UBC CS technical report TR-2005-04 in March 2005.

[1] Attribution unknown.

[2] *Indicative* summaries aim to suggest the content of the original document (to guide a user's selection of documents to read in depth, for instance). *Informative* summaries have as their goal the replacement of the original document, by containing the core information in document while discarding all ancilliary aspects.

engine, and use it to evaluate the usefulness of our features. The results support our claim that the catalyst score captures many important structural aspects of discourse.

Since usenet is a rather large domain (see Section 1.1), we will strive to keep tractability in mind in the algorithms and techniques we propose.

1 Discussion fora

Electronic discussion fora have been important tools for several decades, and their importance has grown recently with the huge popularity of web fora. In this section, we examine why fora present a unique summarization domain.

1.1 Archetypical example: Usenet

Usenet is a discussion medium that started among American universities in the 1960's, and was the dominant textual forum before the world-wide-web supplanted it in the mid-1990's. Although it is no longer ubiquitous among internet-savvy users, it still is a dominant medium, with approximately 3GB/day of raw text currently being posted daily across all servers.

There are several reasons why usenet (and discussion fora in general) is a unique medium in which to study summarization techniques, including:

Domain-independence Usenet newsgroups[3] exist with every imaginable subject, and thus no domain-specific cues can be leveraged.

Lack of traditional structure We cannot rely on authors introducing sections with lucid, concise, preambulatory sentences that are typical in more formal settings.

Multiple authors A single thread will be contributed to by many authors having different opinions and writing styles.

Low signal-to-noise ratio Newsgroups are full of spam, nonsense, and a significant amount of off-topic banter. The magnitude of this effect varies enormously across newsgroups.

Significant internal structure *References*[4] give us an automatic relationship among posts in a thread, which can give insight to the important posts in a thread.

[3] Newsgroups are the logical separation of topics on usenet.

[4] The References header in a Usenet post contains the key(s) of parent messages in the thread.

1.2 Structure

In this section, we outline a few of the unique features that discussion fora afford to the task of their summarization. This intuition is valuable in developing the set of features to use.

1.2.1 Positional and lexical cues

In a formal, structured corpus (such as a collection of academic papers), position is one of the most important cues for sentence selection (Edmundson, 1969). First and last sentences of paragraphs, leading sentences after section demarcations, and sentences in introductory and concluding paragraphs are often used. Usenet, conversely, has very little formality. Even paragraphs are rare; messages are generally too short to warrant this kind of attention from authors.

This doesn't mean that all hope need be abandoned. While messages are generally unstructured, authors sometimes include a self-summarizing, concise opening sentence. This can also be exploited in the context of quotation, as we discuss below.

Finally, although discussion messages lack many of the cue phrases that characterize extract-worthy sentences in formal corpora, there are discourse-related clues unique to the domain. The most clear example is the presence of a question, which is highly relevant to understanding the crux of the discussion and can be important for putting replies in context.

1.2.2 Quoting

The standards for quoting replied-to messages are simple, and relatively well adhered-to on Usenet. This allows us to easily distinguish quoted content from new content in a message. Beyond this, however, the selection of quoted material might be useful in sentence extraction. If the author intersperses his reply with quotations, the first sentence in reply to each block is likely to be a highly compressed summary of the author's response to that material. This might also help preserve consistency in a summary; if we have extracted a sentence s in a post, replies that quote s will be more likely to follow logically from s.

1.2.3 Multiple authors

Although the heterogeneity of discussion text is exacerbated by having each thread composed by multiple authors with different styles and goals, the multiplicity of authorship in a group can also aid in summarizing the discussion. In particular, it allows author profiles to be developed that allow an a priori evaluation of the relevance a message's worth. This is particularly valuable in threads where other features fail to strongly distinguishing among messages, allowing this measure of "reputation" to break ties.

Beyond this use, however, authorship can also be used as a local feature.[5] An author contributing only once to a large thread is likely not to have put forward the most important point in the thread. Similarly, the most prolific contributors probably represent the major viewpoints espoused.

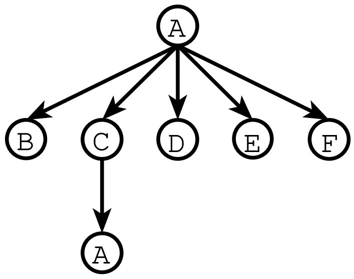
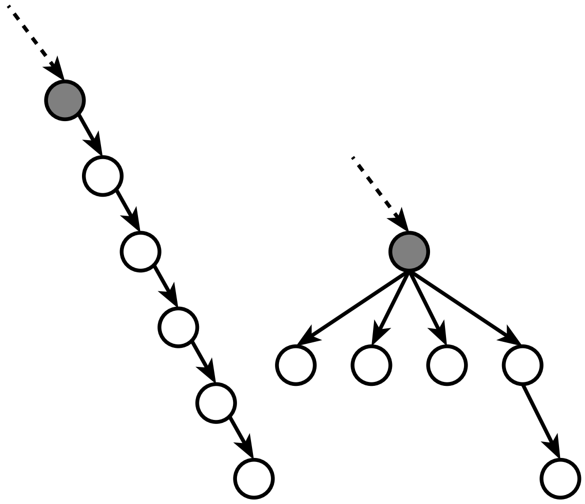


Figure 1: The typical format of a *Question-Response-Thanks* thread. Each author is represented by a unique capital letter in this example.

Finally, authorship provides insight into the structure of many common types of threads. In technical newsgroups, a common type of thread seen is the *Question-Response-Thanks* thread (see Figure 1). If the last post in these types of threads is made by the original poster (OP), it is often a general "thank-you" for the help received, or an explanation of how the problem was eventually resolved. An non-tread-terminating post by the OP might be a followup question, or request for clarification in an answer received. Another example of structure revealed by authorship are *disputes*, which are often characterized by exchanges in which two aggravated parties respond primarily to each other in the thread; this authorship relationship could be used to detect this problem and other discussion fora phenomena.

1.2.4 Threading

Threading is the most important structural element to consider when summarizing a discussion corpus. Recent research has shown that simply considering the text in a thread as a monolithic unit and using conventional summarization tools produces inadequate summaries (Farrell et al., 2001). Farrel et al. also point out that taking the structure of the thread into account is crucial for obtaining coherent and consistent summaries. Their main approach is to summarize each message individually and include every message in the summary. This approach often doesn't not achieve a very high compression of the thread.



(a) A deep subthread

(b) A shallow subthread

Figure 2: Two examples of indicative thread structure. Both shaded post have the same number of descendants, but the post in *(b)* is probably more important in the subtread than the shaded post in *(a)*, as its descendants are less deep.

Since our primary goal is to intelligently select representative messages within a thread, it is natural to look to threading structure as a tool in this task. In a large discussion, messages with more responses are likely to contain a unique (or controversial) insight, and are important for both their content and context provided to subsequent posts. Two obvious measures that come to mind are the number of direct children a message has, and the total number of messages that appear under the message in the thread. Neither are completely satisfactory on their own, however. While we want to reward messages with a higher total number of children, doing so will overemphasize the importance of messages earlier in the thread. Also, it is clear that the number of direct children is important. Figure 2 illustrates the situation of two subthreads with the same number of message but with importantly different structure. In 2(b), the shaded message is critical to the subthread, having directly spawned most of the messages in the subthread. In 2(a), the shaded message is probably still the most important message in the subthread, but its importance to the remainder of the thread is mostly indirect.

We combine the two measures into a single weight

[5] Ie., one used within a given thread, as opposed to over the entire group.

called the *catalyst score* that takes both into account. This weight is defined recursively as the number of direct children a post possesses, plus the discounted catalyst score of each child message. This will be described formally in the next section.

2 The NewsSum system

We have implemented a summarization system for Usenet discussion[6] which is capable of producing an arbitrary-length overview of a given thread regardless of group. NewsSum incorporated many of the features described in Section 1.2 and others besides; this section will describe the system in detail.

2.1 Outline

The main steps of the algorithm are as follows:

1. Retrieval of messages; preliminary formatting.
2. Part-of-Speech (POS) tagging the corpus.
3. Extraction of multi-word terms.
4. Threading messages.
5. Scoring messages and sentences.
6. Summarizing threads.

Typically steps (1)-(3) are performed in some batch manner per time interval (fi., daily). (4)-(6) are currently implemented in one large step in our system, but it would be trivial to separate (6) so that multiple summaries could be generated without having to rescore and thread the group.

As mentioned before, our corpora are collections of messages from Usenet groups. We selected ten groups of various domains and type (ranging from informal chat groups to technical and scientific) and collected just under half a gigabyte of textual data in all. Each message is stripped of its nonessential headers and stored in its own file.

POS tagging is done by concatenating all content from the messages in a group into a single file (that is, after having stripped headers, quotation, signature, etc). We then use a trained TreeTagger to determine Part-of-Speech tags for the corpus (Schmid, 1994). Its principal advantage over available Brill taggers (Brill, 1992) is that trained parameter files for common English are already available to download. Figure 3 shows the output at this stage.

[6] Often referred-to as "news", though bearing little relation to the informational broadcasts typically associated with that noun.

Music/NN can/MD be/VB medieval/JJ &/CC SCA/NP ./SENT Oh/UH make/VVP sure/RB you/PP have/VHP your/PP$ favorite/JJ drink/NN on/IN hand/ NN too/RB .../: LET'S/NP HAVE/VH FUN/NN !/SENT
Lydia/NP Aeyalweard/NP New/NP to/TO this/DT group/NN ,/, so/RB forgive/VV me/PP if/IN this/DT has/VHZ been/VBN covered/VVN ./SENT

Figure 3: Part of the output of POS tagging for rec.music.folk, formatted as input for ATR.

2.2 Term recognition

Using key words (or *terms*) as a feature for summarization has been used as early as Luhn's initial foray into the field (Luhn, 1958). In that example, the frequency of words were taking as indication of their likelihood to be terms, and thus relevance to sentences. As Edmundson noted, this is not a very good feature. The technique, however, has survived to current research. Farrell et al. use a related measure call *salience* which is similar to Luhn's term, but includes a factor which rewards terms that better discriminate among groups (Farrell et al., 2001). They argue that terms help distinguish content among messages and between groups, and we agree with that standpoint. However, research in term recognition has advanced considerably in the past half-century, and we will use a state-of-the-art term-recognition algorithm for our summarization package. Not only have these methods been extensively-evaluated and have met with great success, but also recognize multi-word terms, which is increasingly important as most unique concepts are described by multi-word phrases. Multi-word terms display less polysemy across domains as well. See (Frantzi, 1998) for a general overview of the various methods for term recognition.

2.2.1 The *C-value* method

We use the *C-value* method of (Frantzi et al., 2000). This technique combines linguistic and statistical criteria to determine a *termhood* measure for a sequence of words that represents the likelihood that that sequence is a domain term.

A *string* is a set of words found in a document. The *C-value* method restricts the list of candidate terms to strings of words corresponding to set of possible POS tags. The simplest example is a string of nouns, but more complex linguistic criteria are also considered. The statistical component of *C-value* is the calculation of the *termhood* of candidate terms. Candidate terms are rewarded for having high frequency in a corpus and (weakly) for having longer length, but are penalized for occurring frequently as subsets of other candidate terms (which is indicative that it is the *parent* strings which are the true terms). However, a term that is found in many different parent terms is itself likely to be a domain-specific term.

The termhood (*C-value*) for string a is defined to be:

$$C(a) = \begin{cases} \log_2 |a| \cdot f(a) & T_a = \varnothing \\ \log_2 |a| \left(f(a) - \frac{1}{|T_a|} \sum_{b \in T_a} f(b) \right) & \text{else} \end{cases}$$

Where $|a|$ is the length in words of string a, and T_a is the set of terms that are lexical supersets of a. If $T_a = \varnothing$, then a is a *root* term.

C-value	Term
20.833333	STAINLESS STEEL
12.600000	FRACTURE MECHANICS
10.000000	WEB SITE
8.000000	GOOD LUCK
7.924812	THANKS IN ADVANCE
7.000000	CAST IRON
7.000000	SECANT MODULUS
6.339850	PRINCIPLE THAT INDIVIDUALS
6.339850	HIGH CARBON STEEL
6.000000	SPARK PLUG
6.000000	COPPER TUBING

Figure 4: Head of list of terms found by the *C-value* method for sci.engr.metallurgy. Many domain-specific terms are extracted, but the list also contains many phrases that are a result of the corpus being a discussion fora (e.g. WEB SITE).

2.3 Features and scoring: Messages

In this section we make explicit the features we select for our system and how they are combined numerically. We used a relatively austere set of features– since we are not using feature-selection techniques from machine learning, a large feature set would be highly sensitive to fine tuning of the weights. By keeping the number of features limited, the effect of each is more easily measurable.

2.3.1 Length factor

We measure the length of a post by the number of new complete sentences it contains. Most messages are quite sort (between 3-4 sentences on average), but longer messages are not rare. We would like to reward longer messages, but we cap the maximum length considered for this feature to reduce the weight placed on FAQs and other strangely long posts (we arbitrarily chose a 100-sentence limit). Also, the gain from length should decrease as the post lengthens, which means that a sublinear function should be used. We experimented with both log and square root and found the latter gave a better trade off between rewarding long posts while having diminishing returns.

We denote the length factor of a message m as $f_L(m)$ and define w_L to be the weight for this factor.

2.3.2 Uniqueness factor

One factor we noticed time after time in our experiment was that the more carefully an author appropriately trimmed quoted material, the more relevant the reply was to the original post. Thus, we introduced a uniqueness factor that reward posts with carefully-chosen quotations.

We define the uniqueness factor $f_U(m)$ of a message m to be the ratio of *original* sentences (i.e., unquoted material) to total sentences in the post. A side effect of this feature is the the main message (first post) in a thread will receive a small bonus from this feature. w_U is the feature's weight.

2.3.3 Term factor

A multi-word term's *C-value* gives us a numeric estimate of a term's usefulness. However, this number is primarily meant to be used as a comparative measure among terms, not as a quantity that necessarily signifies anything meaningful on its own (Frantzi et al., 2000). Thus, to use the *C-value* directly could be dangerous. Indeed, when using this factor directly, we found that the total term factor could vary enormously without there being significant differences in the quality of the messages.

Initially stymied by this perplexing issue, we soon found that the *logarithm* of total *C-value* was quite useful. Messages benefit from containing salient terms, but are not overly rewarded if they happen to contain many terms.

We define the term factor $f_T(m)$ of m to be $\log(\sum_{s \in m} C_s)$, where C_s is the sum of the *C-value*s of any terms found in sentence s. w_T is the weight of the term factor.

2.3.4 Catalyst score

Here we formalize the catalyst score informally described earlier. We have found it to be a particularly useful feature in discriminating important messages in a thread. It is defined as follows:

$$f_C(m) = |D_m| + \gamma \sum_{c \in D_m} f_C(c)$$

Here, D_m is the set of direct descendants to m (messages which reply directly to it). The catalyst score of leaf nodes is 0. $\gamma \in [0, 1]$ is a discount factor which is a system parameter. We have found through experimentation that choosing γ to lie in the range $(0.5, 0.7)$ works well. This is sufficiently high to put slightly more weight on earlier messages in a thread, while not precluding later messages from being selected.

Consider the threads in Figure 2. Although both spawned subthreads of the same size, the catalyst score of the shaded message in 2(a) is 2.3056, while the score of the message in 2(b) is 4 (assuming $\gamma = 0.6$).

As before, w_C is this factor's weight.

2.3.5 Message score

Combining all the features, we arrive at the following expression for the relevance score of a message:

$$w_L f_L(m) + w_U f_U(m) + w_T f_T(m) + w_C f_C(m)$$

w_L, w_U, w_T, w_C, and γ are system parameters, which can be set manually or learned.

2.4 Features and scoring: Sentences

Summarization of individual messages is not our focus, primarily since existing tools tend to do a good job on a single document. We have, however, implemented a simple summarizer which uses the following features:

2.4.1 Length factor

As in messages, length is a criterion for determining relevance. This is the logarithm of the number of words in the sentence.

2.4.2 Term factor

Multi-word terms are also used to score sentences within a message. As before, the term factor is the logarithm of the sum of the *C-value*s of all the terms found in the sentence.

2.4.3 Bonuses

Sentences are given a bonus for being the first sentence of original content in a message, as well for being terminated with a question mark, which will be particularly important to putting replies in context.

2.5 Summarization

When summarizing, the desired number of messages and total sentences are given are parameters, denoted n_m and n_s respectively. First, the n_m highest-scoring messages are selected. As long as $n_s > n_m$, they are allocated each one sentence of summarization. If $n_s > n_m$, then the remaining sentences are allocated to the message in proportion to their relevance score.

The messages selected are output in the order they appear in the original thread–this is critical to aid coherence. In our pilot system, the depth of the message is shown with a number of brackets (eg. `> > >`). For each message, the system picks the a_m highest scoring sentences, where a_m is the sentence allocation mentioned in the previous paragraph. Again, these sentences are returned to the order in which they appear in the original message and output. Finally, if two consecutive messages output were not adjacent in the original thread, the system outputs the number of messages it chose to skip.

Figures 5 and 6 show a thread and the resulting summary generated.

2.6 Performance

The summarizer (which is the system not including the Part-of-Speech tagger and multi-word term extractor), was implemented in approximately 800 lines of perl and tested on a 500 mHz Linux server with 256 MB of RAM. Even on such an archaic system, the summarizer performs with excellent speed, processing a newsgroup with 2100 messages in 668 threads in 27.87s.[7] This rate is sufficient to summarize over 6 million messages daily, which exceeds the requirements to summarize all discussion on usenet by two fold.

To achieve this, it is necessary to reduce the complexity of checking each sentence for all potential terms in $O(1)$. This can be accomplished by taking every adjacent subsequence of the words in the sentence (of which there are a constant number, since there is maximum sentence length, and maximum term length), and check against a hash table containing the terms. This is the only part of the system that requires any algorithmic thought (but it is necessary: a naïve implementation is about 20 times slower).

3 User evaluation

We conducted a small user study to evaluate various aspects of our system. In this section we present it and our conclusions.

3.1 Evaluation of summarization systems

Firmin and Chrzanowski highlight many unique issues in evaluating summarization systems, such as pointing out that the agreement even among expert abstractors can be low ($< 50\%$) (Firmin and Chrzanowski, 1999). This renders comparison of computer-generated to human-generated summarizations difficult to meaningfully quantify. This is

[7] Times do note include the tagging and term extraction steps, which need not be performed daily.

even more true in discussion fora, where similar information gets often-repeated in a thread. Thus in this study we will not attempt to directly-compare user- and machine-generated summaries.

3.2 Description of user study

We use both extrinsic and intrinsic methods to evaluate NewsSum. For intrinsic tests, both the acceptability of a summary and the comparison between two summaries of the same thread are indicated by the participant. Usenet threads are annotated with a one-line *subject line* (eg. What am I looking for? in Figure 6); we will use this in an extrinsic test. The participants are shown a set of summaries and will be asked to determine what subject line the thread corresponds to from a set drawn randomly from threads in the newsgroup.

When comparing two summaries, participants were asked to choose one they preferred as a *replacement* for reading the original thread, and one they preferred as a tool for deciding which threads to peruse further. Users were also asked to qualitatively evaluate the system.

The seven participants in this study were all computer science or mathematics graduate students familiar with computers. When asked to rate themselves on their use of electronic discussion fora, two users identified themselves as non-users, two users identified themselves as power-users, and the remaining listed thier skill as intermediate.

3.3 Multi-word terms as features

Participants were shown three different summary pairs where used all the features in NewsSum, and the other omitted the term factor. Threads were selected on the basis of the two approaches producing different summary (it was very common that both were quite close, if not identical).

Only in one thread (A) was a significant difference found: the term-factor summary was preferred in 78% of the cases. Participants had much more difficulty distinguishing the summaries in the other two cases (we conjecture that this is due to being from more technical groups). 4 of the 7 participants reported no difference between the two summaries for thread B, and the remaining three mildly preferred the non-term-based summary (although this was statistically insignificant). In thread C, the term-based summary was preferred in 7 of 14 cases, the non-term-based in 5 of 14, and neither in the remaining two.

As we note later, participants chose the same summary as preferred for both indicative (tool for choosing interesting reading) and informative (text replacement) summaries. Nothing statistically significant emerges.

The comments on these threads are revealing. Almost invariably, it was remarked that the term-based summary provided more information, was more "on-topic", and provided fewer sentences that were distractive. However, the term-based summary was also found to be less cohesive and more "jumpy".

The result that the term factor improves the percentage of on-topic sentences extracted is encouraging, as reducing perceived "clutter" in a summary is critical in the user's estimation of the summary's worth. A possible explanation for the lack of coherency that appears to come with using the term factor is that the term list was truncated so that only the 400 highest-scoring terms were considered. Despite seeming like a reasonable number, a thread tended to average only one or two posts with a non-zero term factor. This causes somewhat of a discontinuity in the scoring, where certain messages are selected despite not being optimal in terms of the context they provide. It is worth investigating if using the entire term list would provide a larger spectrum of term factors (and thus a smoother distribution among messages selected).

3.4 Length of summaries

The same participants were each shown two summaries from each of threads D and E. All summaries were generated using all features. One summary consisted of 10 messages with 10 sentences extracted total (one per message in our system), which we will refer to as the 10-10 summary. The other consisted of 5 messages with 10 sentences extracted (distributed according to the algorithm in section 2.5)—this is the 5-10 summary.

Nothing can be drawn from the statistical analysis. The 10-10 summary was preferred in 9 of 28 cases, and the 5-10 summary was preferred in 11 of 28 cases.

Several users commented that the 10-10 summaries provided less context and was more jumpy, although one user thought that the 5-10 summary was more jumpy than the 10-10 summary. Some participants felt that the having more sentences per post provided more useless information. Users felt that having only one sentence for the main post in the thread particularly hindered comprehension.

These results seem to indicate that the decision of sentence allocation should be made on a per-message basis, using a metric of information gained by the context provided.

3.5 Other quantitative results

For each of the first three threads, the participants were asked to select the most likely subject line out of a list of approximately fifteen. The correct subject was identifies in 87% of cases. 5 of 7 participants were able to identify all subject lines correctly. It is not surprising that a thread summary provides a good tool for subject identification. Authors often choose a paraphrase of their subject as their introductory sentence. It is good to note that even in these high-compression summaries ($< 5\%$), it is still simple for readers to identify the subject matter of the thread.

Another interesting result is that in 13 of 35 comparisons, the participants did not have the same preference for the summary more useful for as indicative compared to informative. This affirms that different techniques may be needed to provide summaries for different uses.

3.6 General qualitative comments

Four users felt that the system produced generally consistent results, two were unsure, and one felt that the output was "garbage-filled".

Participants were generally ambivalent about including the number of messages skipped between summaries messages. Most simply skipped over them, not finding them helping. Two users strongly preferred their inclusion, however, which indicates that this feature should be user-tunable.

The opinion on the usefulness of the summaries was quite varied. Both non-users and power-users of discussion fora were unsure about the usefulness, the latter due to their already impressive reading speed and skill at identifying important messages themselves. Several users in between strongly felt that there was a high potential for time savings when using a system like this, however.

Several users also felt that the system would be useful as a tool to highlight important messages and sentences in a traditional full view of the thread, which is an interesting idea.

4 Discussion and future work

NewsSum is very much a prototype with the goal of exposing the usefulness of the discussion fora features that we have identified. There is significant potential for improvement when creating a polished summarization system. First, it is difficult to evaluate the interactions among features, or know if our weight selections were appropriate. Ideally, the weights should be learned using a corpus expert-tagged with message relevance for many groups. This technique has met considerable success in monolithic summarization and we believe has great potential here as well. This would also allow the inclusion of various other features without fear of over-specification (using an appropriate regularizer assures that irrelevant features do not receive any weight).

The system also suffers from aesthetic blight. It is difficult to appropriately convey the positions of messages in threads using purely textual output; a GUI could help the user better orient the summarized messages in the original thread. Such a system could also integrate the summaries with conventional discussion-browser software, in which case a direct evaluation of time savings and usefulness could be performed.

Finally, NewsSum could very much use a system to automatically select the appropriate number of messages and sentences to extract from each thread, and a means of selecting which threads to summarize. Our preliminary investigation into this is inconclusive. It would not be surprising if the right answer depended strongly on the shape of the thread as well as the number of messages it contains.

4.1 Conclusion

We have identified many pertinent features unique to the problem of summarizing discussion fora, and shown that our catalyst measure and the use of multi-word terms are useful as features in this problem.

Our catalyst score also enables a less ad-hoc measure of the relevance of a specific message. We note that the first post in a thread need not necessarily be treated specially, in contrast to the opinion given in (Newman, 2002). Instead, we find the main post gets naturally included as it almost always inspires the highest level of discussion in the thread, thus has a high catalyst score. We believe that if this score were sufficiently low to not merit the first post's inclusion, then this simply indicates that it was not the most important message in the thread.

Acknowledgements

We are indebted to Lingyan Zhang for her advice and the use of her *C-value* code (Zhang, 2004), and to Giuseppe Carenini for his valuable comments and suggestions.

References

Eric Brill. 1992. A simple rule-based part-of-speech tagger.

H. P. Edmundson. 1969. New methods in automatic extracting. *J. ACM*, 16(2):264–285.

Robert Farrell, Peter G. Fairweather, and Kathleen Snyder. 2001. Summarization of discussion groups. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 532–534. ACM Press.

Therese Firmin and Michael J. Chrzanowski. 1999. An evaluation of automatic text summarization systems. In *Advances in Automatic Text Summarization*, pages 325–336.

K. Frantzi, S. Ananiadou, and H. Mima. 2000. Automatic Recognition of Multi-Word Terms: the *C-value/NC-value* Method. *International Journal on Digital Libraries*.

Kateria Frantzi. 1998. *Automatic Recognition of Multi-Word Terms*. Ph.D. thesis, Manchester Metropolitan University Dept. Of Computing & Mathematics.

Derek Lam, Steven L. Rohall, Chris Schmandt, and Mia K. Stern. 2002. Exploiting email structure to improve summarization.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159—165.

Paula S. Newman. 2002. Exploring discussion lists: steps and directions. In *Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries*, pages 126–134. ACM Press.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK. unknown.

Lingyan Zhang. 2004. Parallel automatic term extraction from large web corpora. Master's Thesis Draft.

Subject: `What am I looking for?`	Catalyst	Term	Unique	Total
`Richard`	8.56	0.00	1.00	34.39
`d2scats`	3.75	0.00	0.28	14.09
`Mitch`	1.00	0.00	0.20	5.41
`Johnny`	0.00	0.00	0.25	3.89
`Johnny`	0.00	0.00	0.80	6.82
`Johnny`	0.00	2.00	0.60	4.73
`commando`	0.00	0.00	0.40	3.41
`anebla999M`	0.00	0.00	1.00	6.00
`tdj`	0.00	0.00	0.50	4.73
`livergee`	1.00	0.00	1.00	9.00
`andy`	0.00	0.00	0.50	3.50

Figure 5: Thread from rec.music.folk summarized in Figure 6

```
> "Richard" <nogood@...  len:16  score:34.39     alloc:4
------------------------------
I don't know whether a certain music catagory lends itself to
wistful songs - perhaps. "When Somebody Loved Me" by Randy Newman
is a good example I think. I'm sure there are lots more that I
forget to recall just now. Quite possibly I might find suggestions
in the country and western, and the folk newsgroups.

> > d2scats@...      len:2   score:14.09     alloc:2
------------------------------
The saddest song known to man as far as I am concerned is "He
Stopped Loving Her Today" - George Jones There are a lot of them
on the radio right now, downright depressing if you ask me...Blake
Shelton "The Baby" Jimmy Waynes latest single. ..somebody help me
with the title...... A blast from the past, since I know more
Tanya Tucker music than anybody else, " Oh What It Did To Me" and
"Love Me Like You Used To".

...2 message(s) skipped...

> > > ~^Johnny^~ <NO.SPAM...       len:8   score:6.83      alloc:1
------------------------------
I think it's about the sixth saddest, or so.

...2 message(s) skipped...

> > anebla999MISS...   len:1   score:6.00     alloc:1
------------------------------
Never mind the bollocks by the sex pistols?

...1 message(s) skipped...

> > livergee@... (Gerry)  len:1   score:9.00    alloc:1
------------------------------
You're looking for anything by Nick Drake
```

Figure 6: Summary of thread in Figure 5