Natural Language Engineering **16** (2): 161–192. © Cambridge University Press 2010
doi:10.1017/S135132491000001X

Automatic summarisation of discussion fora

ALMER S. TIGELAAR, RIEKS OP DEN AKKER
and DJOERD HIEMSTRA
Database and Human Media Interaction Groups, University of Twente, The Netherlands
e-mail: a.s.tigelaar@cs.utwente.nl, infrieks@cs.utwente.nl, hiemstra@cs.utwente.nl

(*Received 2 April 2009; revised 27 November 2009; accepted 11 January 2010*)

Abstract

Web-based discussion fora proliferate on the Internet. These fora consist of threads about specific matters. Existing forum search facilities provide an easy way for finding threads of interest. However, understanding the content of threads is not always trivial. This problem becomes more pressing as threads become longer. It frustrates users that are looking for specific information and also makes it more difficult to make valuable contributions to a discussion. We postulate that having a concise summary of a thread would greatly help forum users. But, how would we best create such summaries? In this paper, we present an automated method of summarising threads in discussion fora. Compared with summarisation of unstructured texts and spoken dialogues, the structural characteristics of threads give important advantages. We studied how to best exploit these characteristics. Messages in threads contain both explicit and implicit references to each other and are structured. Therefore, we term the threads *hierarchical dialogues*. Our proposed summarisation algorithm produces one summary of an hierarchical dialogue by 'cherry-picking' sentences out of the original messages that make up a thread. We try to select sentences usable for obtaining an overview of the discussion. Our method is built around a set of heuristics based on observations of real fora discussions. The data used for this research was in Dutch, but the developed method equally applies to other languages. We evaluated our approach using a prototype. Users judged our summariser as very useful, half of them indicating they would use it regularly or always when visiting fora.

1 Introduction

This research focuses on finding a way to provide useful automatically generated summaries of discussions held on Internet fora. From a user's perspective it would be helpful to be pointed to relevant messages in such discussions. This would not only save reading time, but also make it easier to learn from the discussion content, lower the threshold for participation and improve the overall quality of contributions (Farell 2002). Existing forum search facilities can assist in finding discussions with interesting topics. Frequently, the title of a discussion clearly represents the main question or statement. Nevertheless, to find the actual answer(s), or the most relevant reactions, the entire discussion needs to be read. Instead of this tedious process, we postulate that it would be highly useful to obtain a summary of a discussion automatically.

The term *thread* refers to the structuring of messages in a discussion. This structure also appears in other types of researched Internet communication such as e-mail (Lam *et al.* 2002; Rambow *et al.* 2004; Wan and McKeown 2004) and newsgroups (Lang 1995). However, few researchers have focused on discussion fora in this context (Farell, Fait-weather and Snyder 2001; Klaas 2005; Weimer, Gurevych and Mühlhäuser 2007). This is important since different types of Internet communication yield threads with different characteristics, for example: e-mail threads tend to have few participants and, according to Dalli *et al.* (2004), these threads are also relatively short. Studies exist that exploit discussion fora for different purposes than summarisation. For example, semi-automation of grading, automatically finding unanswered questions and thread topic detection (Feng *et al.* 2006; Kim *et al.* 2006a; 2006b).

The central task in this research also relates to document summarisation. However, methods that work well for the traditional single-document summarisation task appear to fare poorly for threads. Treating a thread as one monolithic document simply does not work (Klaas 2005). At first glance our interests seem to match with multi-document summarisation. Nevertheless, the traditional multi-document summarisation task is defined as summarising multiple monologue documents covering the same subject. This significantly differs from the task at hand that targets threads which are essentially dialogues formed by message 'exchange'.

People use fora for many other purposes than pure discussion (Baldwin, Martinez and Penman 2007). We focus exclusively on threads of the following types in this paper:

- *Problem–solution*: A main question is posed, replies are posted and (optionally) follow-up questions are asked.
- *Statement–discussion*: An (opinion) statement is voiced, replies are posted and stances are clarified and revised.

For problem–solution threads the ideal output would be a clear problem definition and one or more possible solutions. This is similar to the concept of conversation focus as presented by Feng *et al.* (2006). For statement–discussion threads, the main statement and the major stances of authors in the discussion should be output.

Statement–discussion threads are generally *wide*. As follow-up to the initial post, many authors respond to give their insights. Problem–solution threads are usually *narrow* and involve more contributions of the initial author to refine the problem and work towards the solution. Nevertheless, both type of threads have a main focus which the summarisation process should capture.

Farell *et al.* (2001) specifically focus on creating a summary of each message in a thread individually as opposed to generating one large, combined, summary. In this research we concentrate on the latter instead. For generating one summary we adopt the term *monolithic* (as opposed to manifold). The summaries that we generate use only text that already exists. Therefore, they are *extractive*. They are also not generated based on a user query, but rather based on the available data. As such, their content is determined by the data combined with algorithms and heuristics, making them *data-focused*. Generating indicative summaries is not so useful given that we already know that a user is interested in a specific thread. However, making full-coverage informative summaries is difficult (Hovy 2004). Therefore, we restrict ourselves to finding the main focus of a discussion. The summaries produced as a result of this we term *semi-informative*. Finally, threads consist of *hierarchically* arranged messages that express a *dialogue*. With all these terms defined we can now state the prime focus of this research: the creation of *monolithic extractive data-focused semi-informative* summaries of *hierarchical dialogues*.

Threads are not static. They develop over time as new messages are posted. We are interested in threads that have already developed over some time and that consist of at least several posts. Threads may also contain images, but with the exception of emoticons we do not consider such multimedia content in this research. Furthermore, we detect references to external sources contained in messages, but do not give them preferential treatment. Finally, the developed prototype supports both Dutch and American English, but evaluations were done using only Dutch fora. However, keep in mind that many important techniques discussed in this paper are for the main part language independent.

We define the following research questions:

(1) What are the structural characteristics of threads and how can these be exploited for automatically building summaries of threads?
(2) What is the performance and usefulness of a thread summarisation system?

We answered these questions by developing a method and implementing it in a prototype system that we evaluated. We regard this system as a black box and define its primary inputs and outputs as follows:

- Input: a thread and the size of the desired summary.
- Output: a summary of the input thread with the desired size.

The insights presented in this paper are based on real data. We manually examined forty threads on a technical discussion forum.[1] Half of these were of statement–discussion type, the other half of problem–solution type. We also used threads from several other fora for refinements, fine-tuning and checks[2] and to prevent making the derived heuristics too specific to technical fora.

The difference of the summarisation approach in this paper, over general summarisation, is in the exploitation of characteristics specific to threads. Our most important contributions are as follows: (1) an approach to find and weight the referential structure of a thread; (2) a summarisation algorithm specifically for threads.

The rest of this paper is arranged as follows: related work is discussed in Section 2; Section 3 describes how threads are structured, how this structure can be found and weighted and how surface message characteristics can be exploited; Section 4 treats several technologies used in the final summarisation process; this is followed by Section 5 which describes the actual summarisation process; the prototype and its

[1] `http://gathering.tweakers.net`

[2] Notably, `http://www.stand.nl/forum` and `http://forum.fok.nl`

evaluation are presented in Section 6; the paper concludes with Section 7. The appendices (A and B) show how the algorithm is applied to two example threads that were also used during evaluation, including example summaries.

2 Related Work

Farell *et al.* (2001) were the first to write about summarisation in the specific context of discussion groups. They found that human summarisers use the structural discourse relationships between postings to guide their choice of sentences when creating a summary. Based on this finding, they developed a hierarchical discussion summarisation algorithm that uses term salience. Sadly, they did no concrete user evaluation, that specifically measures summarisation performance, in their initial and follow-up paper. They seem to focus mostly on issues surrounding the user interface and interaction (Farell *et al.* 2001; Farell 2002).

The work of Klaas (2005) is close to our research. He argues that traditional summarisation does not apply to threads. Instead, according to him, we have to look for other approaches that exploit characteristics unique to threads. Similar to Farell *et al.* (2001) he identifies salient terms in messages. His use of the thread referential structure, by recursively scoring messages, is interesting. In this paper, we build upon this idea. Klaas also performed a small user evaluation. One of his results is that even for summaries with a high compression ratio, smaller than 5 per cent, users can still identify the thread's topic. User perception of his system was mixed. Some users felt his system produced consistent results, others were unsure or thought that the output was 'garbage-filled'.

There have also been studies that specifically analyse on-line discussions, but not in the context of summarisation. Kim *et al.* (2006b) focus on finding a way to semi-automate grading based on the quality of discussion participation. Their corpus consists of discussion threads from university undergraduate computer science students. In a related paper they use speech acts to find threads with unanswered questions and confusions (Kim *et al.* 2006a). Instructors can use this information to help determine where to focus their attention. Some interesting findings are as follows: about 95 per cent of all threads start with a question post and an answer directly follows that question in 84 per cent of all cases. They also found that acknowledgements are usually found at the end of a thread (73 per cent). Nevertheless, they use a domain specific corpus which consists predominantly of threads that have only two messages: question-answer pairs. Such threads are less interesting to summarise, since the need for a summary becomes more pressing as the thread length increases.

3 Discovering, weighting and filtering the data structure

Before we can proceed, we first need to understand our data. A thread consists of two parts. First, an outer structure: the relation between messages. Second, an inner structure: the content of the messages. In this section we explore how we can determine the outer structure and exploit surface characteristics of the inner structure.

a_1: Has anyone finished Children of Dune? Is it good?

b_1: *In message a_1, Alpert wrote:*
> Has anyone finished Children of Dune? Is it good?
I really enjoyed reading this book. I can recommend it. It's well written.

c_1: *> Is it good?*
The book is really hard to read. The beginning is too confusing.

a_2: Okay, that's useful advice *Ben* and *Carlos*. I liked the earlier books. So I think I'll check it out.

Fig. 1. A fictive example thread corresponding structurally to Figure 2. The authors are Alpert (a), Ben (b) and Carlos (c). The left column shows message identifiers; the column on the right the actual message text. Italicised phrases indicate usage in structure discovery.

3.1 Discovering the outer structure of threads

Many fora display flat lists of messages. The only explicitly encoded information in such lists is usually the date and time of each posted message. This throws away a lot of structural information. Authors in a thread usually reply to (one or more) specific messages. We broadly distinguish between two types of references:

(1) Explicit mentioning of author names, for example: 'but as *Mohinder* said . . . '
(2) Quote blocks: literal citations of earlier messages that usually include explicit references to the source message (Carenini, Ng and Zhou 2007).

Using these references we can find the relations between messages. Based on the example thread in Figure 1 the conceptual task is illustrated in Figure 2. We want to go from Figure 2(a) to discovery of the relations as depicted in Figure 2(b). Thus transforming a linear temporal message chain, based on message metadata, into a *directed acyclic graph* by using semantic information contained within the messages. Implicit temporal ordering still exists in Figure 2(b), namely that c_1 was posted after b_1 depicted by placing c_1 to the right of b_1. Time is thus visually represented by top–bottom and left–right ordering which maps onto an earlier–later scale.

To detect reply structure, based on explicit mentioning of author names, we first collect the names of all the authors in the thread. The next step is finding all candidate matching words in a post. We do this by first looking for exact matches. This leaves misspellings of author names for which we employ the Ratcliff/Obershelp (RO) algorithm to perform loose matching (Ratcliff and Metzener 1988). This algorithm is rather expensive: cubic in worst case, normally quadratic and linear in best case. Since false positives can occur, we need to restrict ourselves to candidate words in each post that likely match. For this we consider words Part-of-Speech tagged as proper nouns, words following an @-sign (copes with '@nickname' references) and words before the word 'wrote'. If an author name is mentioned in a post, we assume that post refers to the last post made by that author (Schuth, Marx and de Rijke 2007). This can cause false positives if an author name is used in a different context

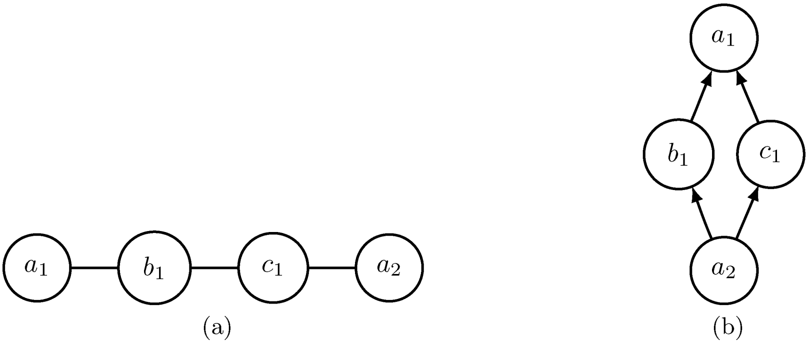

Fig. 2. Conceptual thread structure discovery. Nodes represent messages. In the node name each character represents a unique author and each number a unique post by that author. Edges in (a) represent temporal links (c_1 was posted after b_1) whereas edges in (b) indicate references (a_2 refers to b_1 and c_1).

which really refers to a different person. However, we have never seen this happen in any of our reference threads. Author names on fora tend to be very distinguishable.

Quote blocks are recognisable by either '>' marks or HTML blockquote tags. Quotations are usually made up of fragments of the source message. Quote blocks can appear in several forms: with an explicit link to the source message (most common), and/or with explicit mentioning of the author name and without any references. The approach described in the previous paragraph covers author names. The presence of explicit links provides a more detailed reference and thus always overrides mentioning of author names. To find the source of quote blocks without an explicit link, we compare each line in a quote block to the lines in preceding messages. Lines in messages are traversed bottom to top; messages chronologically from most recent to first post. We also conduct this comparison using the RO algorithm. Once a line in the quote matches a line in a preceding message, we create a reference to that message. The quote count of the source line is also increased, this count is used later during actual summarisation. This process of creating references continues until either all lines in the quote match or no more messages remain to match against. When looking at previous posts, we skip quote blocks in those posts.

We can apply the described structure discovery methods to Figure 1. The link between message b_1 and a_1 is explicit. Even if the message identifier a_1 was not explicitly mentioned in b_1, we could have found the link by parsing the author name: '*Alpert* wrote:'. The link between message c_1 and a_1 is implicit. We compare the text in c_1's quote block, 'Is it good?', to previous messages. We skip the quoted text in message b_1 and find matching text in a_1. Hence, a link between a_1 and c_1 is created. Finally, message a_2 mentions two author names: Ben and Carlos. These are Part-of-Speech tagged as proper nouns. Based on this we create links to the most recent messages of Ben and Carlos which are b_1 and c_1. These messages are found by comparing the author names in message a_2 to the author name metadata of each message. This completes the transformation from (a) to (b) in Figure 2.

There is still another situation to consider. We regularly observed no quoting at all in replies to original texts. So, what assumptions should we make for messages without such references? We studied several threads and made the following observations:

- If a message of the initiating author of the thread refers to no other message, then he always responds to the most recently posted message. When these are messages of himself they usually elaborate or report on advancements made towards solving a problem.
- Messages of other authors almost always follow-up to the last message of the initiating author in the thread. Alternatively, they respond to the message they themselves were last quoted in or referred to by name. Some exceptions to this heuristic exist, like people responding to an older message of the initiating author, usually to give extra suggestions. However, we can not automatically detect these exceptions without actually interpreting the text. The heuristic appears to cover most of the cases quite well.

To discover the structure of threads we use a combination of clues (explicit mentioning of author names and quote block recognition) and rules based on the above two observations. At the end of the discovery process, all messages in a thread, except the first post, have at least one reference to an other message.

3.2 Weighting messages within threads

We now know which messages refer to which other messages. This knowledge can be exploited to find the relative importance of the individual messages in the discussion. Kim *et al.* (2006b) found that the number of responses to a message indicates its importance. Some additional phenomena are as follows: the beginning of the thread is important since it contains a description of the actual problem or statement; the end of the thread too because it is populated by working solutions and summaries of preceding posts; and messages posted by the initiating author are informative (even if they are leaf nodes). We derived the *Positional Message Relevance* (PMR) formula for this, based on insights by Klaas (2005) and our own observations:

$$pmr(m) = \#P_m + \#C_m + A_m + 2 \cdot \left(H(m) - \tfrac{1}{2} \cdot HT\right) + \sum_{c \in C_m} pmr(c) \qquad (1)$$

where P_m is the set of parent messages that message m points to; C_m the set of child messages that point to m; A_m is 0 normally and -1 if the message is a leaf whose author is not the thread initiator; $H(m)$ the height in the tree (relative to the first post at height one) at which m is; and HT the height of the entire thread. The height of a message $H(m)$ is always that of the longest path between the first post and message m. The purpose of the height term is to slightly discount messages earlier in the thread and to give some extra weight to those at the end of the thread. This can cause PMR values to drop below 0 in which case we fix them to zero. The extra multiplier 2 was determined by experiment to be useful for helping representation of messages near the end of the thread. We think it would be good to automatically tune this parameter, possibly by using the average width of the thread or a branching factor.

Besides the relative height, the fourth term in the formula, the sum of the other parts of the formula equate to 0 for normal leaf nodes. The exceptions to this are as follows: the leaf message was posted by the thread initiator (in which case 1 is added) or a leaf node has multiple parent nodes (each extra parent adds 1 to the weight of the node). This captures the fact that strongly referring leaf nodes are interesting for summarisation, since they are usually already partial summaries by themselves.

Besides the factors represented in the PMR there are other phenomena to consider. Especially in longer discussions, those of statement–discussion type, some authors post more messages than others. They have a higher degree of *participation*. Linked to this, some authors contribute more substantial amounts of text. They have a higher *talkativity*. Authors with a high participation degree and talkativity usually have a much larger steering influence on the discussion. Hence, it is important that their contributions are well represented. This idea, applied to spoken discussions, is also treated in the work of Rienks (2007).

To cope with this, we calculate the participation degree of each unique author in the thread. This is expressed as the proportion of messages of a specific author relative to the total number of messages a thread consists of:

$$participation\,(a) = \frac{\#\,\{m \in M \mid isauthor\,(a,m)\}}{\#M} \tag{2}$$

where M is the set of all messages; m is a specific message; and the *isauthor* function determines if author a is the author of message m.

Similarly, we also calculate the talkativity for each unique author in the thread. We define this as the number of words an author contributed to the total number of words a thread consists of (excluding quotations):

$$talkativity\,(a) = \frac{\sum_{m\epsilon M|isauthor(a,m)} length\,(m)}{\sum_{m\epsilon M} length\,(m)} \tag{3}$$

where *length(m)* is the length in words of message *m*.

During assignment of summarisation weights to messages we would like to give extra weight to authors that exhibit both a high talkativity and a high participation degree. The ‘maximised’ case, both having value 1, would only be true for a thread with one author, which we do not consider in this research. However, we should assign more weight to authors that lean towards this case. Therefore, we express the weight preference in a simple weighted combined function (with $\beta = \frac{1}{2}$):

$$pt\,(a) = \beta \cdot participation\,(a) + (1 - \beta) \cdot talkativity\,(a) \tag{4}$$

We term this the Participation–Talkativity (PT) factor. Just like the PMR, we use it later for message weighting.

Feng *et al.* (2006) also model the relevance of contributions of specific authors in a discussion in their research on conversation focus. However, they use speech acts combined with a graph based algorithm called HITS (Kleinberg 1999). Specifically, they rate an author’s value based on the positivity of the reactions on his posts. This approach works well on a manually annotated corpus. However, reliably finding

Table 1. *Formatting characteristic formulas.* m *is a message (a list of sentences);* s *is a sentence within a message (a list of words); and* w *is a word within a sentence*

$f_{nc}(m) = \frac{1}{\#m} \cdot \sum_{s \in m} iscapitalised(s)$	[*m*]issing capitalisation.
$f_{ac}(m) = \sum_{s \in m} \frac{1}{\#s} \cdot \#\{w \in tail(s) \mid isallcaps(w)\}$	All [*CAPITALISED*]!
$f_{np}(m) = \frac{1}{\#m} \cdot \sum_{s \in m} hasnopunctuation(s)$	missing punctuation[]
$f_{re}(m) = \frac{1}{\#m} \cdot \sum_{s \in m} hasrepeated!(s)$	repeated exclamation[*!!!*]
$mfs(m) = 1 - \left(\frac{1}{4} \cdot f_{nc}(m) + \frac{1}{4} \cdot f_{ac}(m) + \frac{1}{2} \cdot f_{ns}(m) + \frac{1}{2} \cdot f_{re}(m)\right)$	

the polarity direction and type of post automatically is a difficult task with much domain specificity. Our approach is more simplistic, but scales better because it is completely automatic. The PMR is intended to take care of determining the less relevant information at the message level rather than at the author level.

3.3 Filtering based on message surface characteristics

Readability is an important aspect of texts that can be used for filtering messages. A wide variety of metrics exist for this. These are frequently based on the word count per sentence and number of syllables per word. Determining the latter automatically is difficult for a machine. Several alternative formulas exist that use the number of characters in words as opposed to the number of syllables (DuBay 2004).

We calculate the average length of words based on such character counts and the average length of sentences based on word counts within a post. Conceptually, our approach is close to that of Coleman and Liau (1975), with the exception that no single combined score is created. It remains somewhat dubious to compare these Average Word Length (AWL) and Average Sentence Length (ASL) statistics among posts because the messages differ in length. Nevertheless, we assume that the AWL and ASL can be used to measure the (un)importance of a single message with respect to the other messages in a thread. This targets posts that are unusually terse or very lengthy with respect to word usage and sentence length. These posts suffer from low readability and users are less likely to read them compared with the other messages.

The quality of a post is hard to define. In fact, properly rating post quality would require exhaustive semantic knowledge. Intuitively, certain formatting characteristics of a message can be indicative of poor quality. The suggestion of using such surface features was initially inspired by comments made by Steven de Jong, a moderator of the NRC weblog (`http://weblogs2.nrc.nl/discussie/`). He independently observed the same phenomena as Weimer *et al.* (2007). They investigated the effectivity of several types of surface features for a good versus bad classification task of forum posts. Weimer *et al.* based their research on a corpus of human rated posts. We use four surface features similar to theirs, shown in Table 1. We calculate a grand global score, termed the *Message Formatting Score* (MFS), based on these four characteristics. An MFS of 1.0 indicates a well-formatted message whereas an MFS of 0.0 indicates a very poorly formatted one.

Table 2. *Performance of preprocessing tools*

Tool	Test data	Evaluation result
Tokeniser	De Limburger	99.5 per cent accuracy
Part-of-Speech Tagger	Spoken Dutch Crps.	97.5 per cent accuracy
Partial Parser	Alpino Treebank	87.5 per cent precision, 95.6 per cent recall
Named Entity Recogniser	CoNLL2002	86.9 per cent precision, 42.1 per cent recall

4 Technologies

4.1 Preprocessing

We build upon several foundational Natural Language Processing (NLP) technologies and implementations thereof. We, in particular, use parts of the Natural Language Toolkit (NLTK) (Bird, Klein and Loper 2008). Other NLP system components are briefly mentioned in this section. The performance of each is summarised in Table 2.

We use the tokeniser from Kiss and Strunk (2006) included in the NLTK. It has intelligent handling for some tricky issues in tokenisation. Kiss and Strunk show that the accuracy of this tokeniser on a Dutch newspaper corpus, named The Limburger, exceeds 99 per cent. For Part-of-Speech tagging we use a Hidden Markov Model trigram tagger from the Hammer Tagger Toolkit with some extensions specific to web fora content (Stegeman 2007). We created taggers based on the Spoken Dutch Corpus (SDC) for Dutch and on the Brown corpus for US English (Francis and Kûcera 1979; van Eynde 2004). We projected the tags of these corpora onto a self-developed unified tagset consisting of twenty-six tags. Our tests revealed the accuracy of the SDC tagger to be 97.5 per cent, on par with state-of-the-art tagging performance (Manning and Schütze 1999). We do partial parsing using the built-in regular expression based chunker of the NLTK. Chunking rules for Dutch and US English were written manually based on an existing grammar previously created by one of the authors for other purposes. We tested our partial parser against the Dutch Alpino Treebank corpus (Bouma, van Noord and Malouf 2000). The performance is around 87 per cent precision and 95 per cent recall.

We perform Named Entity Recognition (NER) to find the names of people, organisations and locations using gazetteer lists augmented with some manual heuristics inspired by Bogers (2004). We evaluated this against the CoNLL2002 newspaper data (Sang 2005). NER precision is about 87 per cent and recall 42 per cent. Finally, we also do additional semantic tagging to recognise URL's, e-mail addresses, dates, times, quantities and emoticons. Lam *et al.* (2002), who use regular expressions to find semantic units, served as inspiration for our approach.

4.2 Anaphora resolution

An anaphor is a specific type of reference pointing back to an antecedent in a text. When making extractive summaries, sentences are 'cherry picked' out of a larger body of text. However, sentences frequently depend on the preceding sentences by referring to them. This makes the summary unreadable if those previous sentences are not included in it. This phenomenon is known as *dangling anaphora.*

The concept of anaphora is frequently confused with coreference which is a related, but different, concept. An anaphor *depends* on its antecedent for correct interpretation. A coreference essentially stands alone and thus needs no additional context to be understood (van Deemter and Kibble 2000).

We are interested in *indicative* anaphoric resolution. This means instead of replacing an anaphoric reference, we augment it with a likely antecedent. We always place this 'suspected' antecedent behind the anaphoric expression, in parentheses, since this still leaves the possibility for a user to detect wrong anaphoric resolutions. An example:

I bought a videocard today. The *card [videocard]* is noiseless.

Anaphora resolution consists of two parts: the identification of an anaphoric expression and finding a noun phrase it refers to. We focus on finding three types of anaphora: pronouns (like: it, he, she), generalisations (videocard→card, big house→house) and acronyms (Free Software Foundation→FSF). Naturally, we do not resolve references within the same sentence.

We walk through the entire text, accumulating all noun phrases in sentences. These are used as candidates for anaphoric resolution. To detect anaphoric expressions we employ the following strategies:

- Using a lists of common pronouns in the third person: he, she, they, et cetera. For example: '*Niels* got cabin fever. *He* ran around frantically'.
- Partial matching between (suffixes of) heads of noun phrases (Hoste and van den Bosch 2007). For example: 'The stinging behaviour of *worker bees* is very effective. But, the *bees* who do . . . '
- Detection of acronyms build from the previously seen noun phrases (Hoste and Daelemans 2005). For example: 'Sponsored by the *British Broadcasting Corporation.* The *BBC* has . . . '

We then score anaphoric expressions against the candidates using heuristics: distance restrictions (Mitkov 1999); weighting based on definiteness and the presence of prepositions (op den Akker *et al.* 2002); number agreement (Jurafsky and Martin 2000); the position of the noun phrase in the sentence and the presence of surrounding parentheses. We do not consider word gender.

Novel is the usage of thread specific characteristics in the anaphoric resolution. We allow references to cross post boundaries. The first several lines of a post may refer anaphorically to the last lines of a post that chronologically precedes it. Additionally, the first lines may also refer to one or more other previous 'parent' posts as determined by the discovery of the outer thread structure. For resolving pronouns in the first (I, me, mine) and second (you, your) person, sometimes called deictic references, we also exploit the thread structure. Self references (first person) resolve to the name of the poster of the message. Second person references resolve to the author of the post the current post is a reply to, provided only one such message exists.

4.3 Sentence type detection

Detecting the type of sentences in a post is useful for a variety of purposes. Firstly, for filtering out less relevant sentences, and secondly, for finding links between sentences across messages (Farell 2002). For filtering we restrict our detection to a few basic sentence types:

- Openers: 'dear . . . '
- Closers: 'thanks!'
- References: 'http://. . . '
- Lists: '1. x, 2. y, . . . '

These types of sentences are assigned a lower priority during summarisation. Besides these, a major important group of sentences are questions. We detect these based on the presence of question marks and sentences constructed in the interrogative form, looking for words such as who, where, when, et cetera (McKeown, Shrestha and Rambow 2007). Although this does not capture all question sentences, and may even capture some that are not, we believe it is a good starting point.

4.4 Question–answer linking

We use a small question typology for classifying question sentences based on Webclopedia (Hovy, Hermjakob and Ravichandran 2002). It specifies for several types of questions what kind of answer one would expect. For a 'who' question we would anticipate the answer to be a person or an organisation, for 'when' we would expect some specific time or date. This information allows us to create a link between a question sentence and a possible answer.

The identification of question and corresponding answer blocks is a more objective task than that of summarisation (McKeown *et al.* 2007). Kim *et al.* (2006a) showed that, on specific corpora, a significant portion of forum posts consists of questions (36 per cent) and answers (43 per cent) . The usefulness of linking questions and answers was established by Zechner (2002). He concludes that while such linking does not significantly affect the informativeness of dialogue summaries, it does significantly increase the coherence.

The procedure for finding question-answer pairs is as follows:

(1) For a specific message *m* find out what messages it refers back to using the thread structure.
(2) In the referred messages identify all question sentences that have an expected answer type. For example the sentence '*Where* did the Kahana go?', would have answer type 'location'.
(3) Find all sentences in message *m* that contain a noun(phrase) with a semantic tag (person, date, et cetera) that matches the answer type in the previous step. For example 'The ship went to *an Island*', where the italicised expression has the semantic tag 'location'.
(4) Increase the question-answer score of both the set of sentences found in the previous step, as well as the original sentence that contained the question.

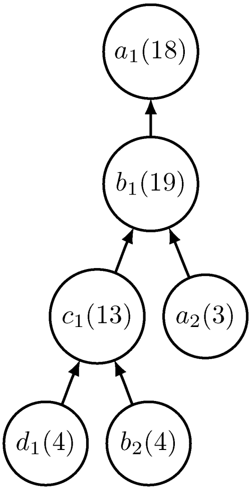

Fig. 3. An example thread. Messages are represented by nodes, arrows indicate references. In the node name each character represents a unique author and each number a unique post by that author. The PMR of each node is shown between parentheses behind the node name.

We prefer sentences with a high question-answer score for inclusion in the summary. One might argue that for summarisation purposes it would be best to always include both the question and answer part of a pair. However, doing this is tricky since there can be multiple pairings. It would also conflict with weighting at the message level. We did not investigate this further.

5 Summarisation process

The summarisation algorithm consists of several steps. These are as follows: *filtering*, *distributing message weight*, *distributing sentence weight* and finally *selecting sentences*.

Message filtering is done based on the average sentence and word length. If either of these differs sufficiently from the average in the thread none of their content is included in the summary. This is based on z-scoring with z between -3 and $+3$. Additionally, we filter out messages with a poor, below 0.40, Message Formatting Score (see Table 1, Section 3.3). The first message of a thread is never filtered out.

The remaining messages in the thread are each assigned a weight which determines how much of their content will be included in the summary. The weight of a message is determined based on the PMR (equation 1, Section 3.2) and PT factor (equation 4).

Figure 3 shows an example thread and Table 3 shows the results for the steps that we describe for this thread. We work from PMR/PT values on the left to the final sentence weights on the right of the table. The PMR values for each message derive from the thread structure. The PT values for each author are fictive. In the example the PT and normalised PT are the same, since *all* messages are selected.

We first normalise the PMR over all selected messages of the thread:

$$pmr^{norm}(m) = \frac{pmr(m)}{\sum_{n \in N} pmr(n)} \tag{5}$$

Table 3. *Example thread sentence assignments. Shows how we calculate the number of assigned sentences for each message of the thread in Figure 3. Messages are listed in the first column, each subsequent column shows the results of a formula described in this section. Note that pt^{norm} is a value assigned to each unique author and is repeated for each author in the table. The last two columns show the number of assigned sentences assuming a thirteen sentence summary. The* select *column shows select* n *sentences out of message length* m *as* n/m. *Since the left values in this column do not add up to thirteen, an extra pass is made that removes sentence weight from messages. The rightmost column, labelled* r, *shows the final sentence count assignments*

Message	*pmr*	pmr^{norm}	pt^{norm}	$pt^{msgnorm}$	*weight*	*select*	*r*
a_1	18	0.30	0.46	0.07	0.22	3/13	3
b_1	19	0.31	0.30	0.05	0.23	3/10	3
c_1	13	0.21	0.08	0.08	0.17	3/4	3
a_2	3	0.05	0.46	0.39	0.16	3/10	2
d_1	4	0.07	0.16	0.16	0.10	2/8	1
b_2	4	0.07	0.30	0.25	0.13	2/5	1

where N is the set of all *selected* messages ($N \subseteq M$ where M is the set of all messages in the thread) and m is the message to calculate the pmr^{norm} for.

Next, we normalise the PT of each author over all selected messages as follows:

$$pt^{norm}(a) = \frac{pt(a)}{\sum_{b \in \{author(n) | n \in N\}} pt(b)} \tag{6}$$

where a is the author we want to calculate it for; the *author* function gives the author of a message and b iterates over all authors of the *selected* messages.

We use both of the preceding formulas to calculate the PT at the *message level*, rather than at the author level. This PT factor is distributed inversely with respect to the PMR of the messages posted by the author. Meaning that the PT weight for a specific author shifts towards messages that have a low (or zero) PMR from that author. We express this formally as:

$$pt^{msgnorm}(m, a) = pt^{norm}(a) \cdot \left(1 - \frac{pmr^{norm}(m)}{\sum_{n \in N | author(n) = a} pmr^{norm}(n)}\right) \tag{7}$$

where a is the author of message m. This formula is applied only if more than one message of an author has been selected for inclusion in the summary.

Finally, we calculate the weight for each message based on both the normalised PMR and PT as follows:

$$weight(m) = \beta \cdot pmr^{norm}(m) + (1 - \beta) \cdot pt^{msgnorm}(m) \tag{8}$$

where m is the message to calculate the weight for. We set the value for β to $\frac{2}{3}$, determined to yield good results by experiment. Thus, the PMR has more influence on the final message weight than the PT factor.

The total number of sentences that the summary should consist of is an input to the summariser. The calculated message weights can now be used to calculate for each message what ‘slice’ it contributes to the summary ‘pie’. We do this as follows:

$$select\,(m) = \lceil weight\,(m) \cdot sentences \rceil \tag{9}$$

where *m* is a message and *sentences* is the number of requested summary sentences.

While the message weights always sum to one, the number of sentences must be a whole number. For this reason we apply the *ceiling* function. Since this can cause the sum of sentence weights to exceed one, a second pass is made over the final sentence weight values. In this pass we remove sentence weight from messages that already have a low weight. The *r* column in Table 3 shows the result. We penalise messages, with the same weight, posted earlier more than those posted later. The effect is a bias both towards messages with a high weight, increasing coherency, and towards messages at the end of the thread, increasing coverage.

One remaining problem is that short messages may get too many sentences assigned. For example: a message consisting of only two sentences gets five assigned. To handle this, a final pass is made that removes excess sentence weight. The excess sentences are then assigned to the messages from highest to lowest weight until there are no more sentences to redistribute. If at this point there is still sentence weight left, the resulting summary will be shorter than requested.

We know now *how many* sentences we can select from each message for our summary. However, we still need to decide *which* sentences we will select. We use the following priority heuristics for this: firstly, sentences in a message are selected that are marked as either being an answered question or an answer to a question. Secondly, the longest question sentence, in sense of character length, is preferred. Lines cited in one or more separate messages have third priority: the more a particular line is cited, the more it is preferred over other cited lines. Sentences with the same citation count are considered in bottom-to-top order. If no other heuristics are applicable, the system falls back on the top–bottom interleave: from the sentences that remain we simply pick the first one, than the last one, than the second one, et cetera. Finally, the low priority lines are considered in long to short order. This includes the greeters, list items, et cetera.

Both Klaas (2005) and Farell *et al.* (2001) demand that some part of each message should be included in a summary. We take a different approach and allow the system to exclude all content of a message. Sentences are always included in the summary in their original message order to increase coherence. Resolved anaphoric references are included to further aid in understanding the context.

6 Evaluation

To evaluate the summarisation method presented we implemented our ideas described as a prototype system. The inputs to this prototype system are as follows: the thread URL; the language (Dutch or English); the compression ratio as a percentage or the absolute number of sentences to retain; and finally: the forum type which refers to the forum software used at the mentioned thread URL. We transform forum content, from the HTML page(s) a thread consists of, to an internal data structure.

It is quite difficult to evaluate summaries. We chose a user evaluation as we believe the actual target user group should judge the created summaries. We conducted the evaluation via an on-line webform. Confronting users with just a thread and the summariser output for that thread did *not* seem like a good approach. We reasoned that this does not encourage people to thoroughly read the thread. We concluded that it would be better to first *actively engage* users with its content. Therefore, each participant in the evaluation first had to make some assignments with regard to the thread content. This was really only to 'prime' the users sufficiently as to enable them to better judge the machine generated summaries.

The exact evaluation procedure was as follows: after an introductory screen each user was confronted with the same two threads in turn. First, a problem–solution thread consisting of twelve messages and hundred unique sentences (see Appendix A). Second, a statement–discussion thread consisting of eight messages and thirty unique sentences (see Appendix B). These two test threads were not used during development of the system and were from two different fora. We showed them unaltered, with their original mark-up from the source page, to recreate a familiar environment as close to real usage as possible. We requested participants to make two small assignments for each thread:

(1) Select the five most important messages in the thread and rank them in order of importance.
(2) Create your own extractive summary by pasting a selection of sentences in a textbox: twelve for the first thread and ten for the second. Percentagewise this roughly conforms to the lower (15 per cent) and upper bounds (35 per cent) for a useful summary according to Hovy (2004).

After these priming assignments we confronted the users with a machine generated summary of each thread. They were requested to judge various aspects of these summaries such as the coverage and coherence. Next, we asked users several opinion questions regarding the usefulness of automatic summarisation and various features. Finally, users were asked some demographic questions that closed the evaluation.

Eighteen people participated in the evaluation. This makes it hard to draw statistically significant conclusions, but is enough to give a general impression. Participants were predominantly male (72 per cent) and nearly all in the 21–30 age range. Half of the respondents read threads in webfora at least once a week, but only 17 per cent posted with this frequency. Respondents mostly visited between one and four different fora at least once a month. Most respondents claim to have above average Internet searching skills (∼90 per cent). However, they are less confident about their own Internet *fora* searching skills (∼50 per cent above average). This supports the idea that there is a difference between these tasks.

The primary objective of the evaluation was to find out how humans *judge* the machine generated summaries. Comparing human and machine summarisation performance was not a goal. Nevertheless, we did record the message orderings and extractive summaries created by the participants. This data can give some insight into the relation between real human summarisation and the judgements.

Table 4. *Human–human agreement and human-machine agreement for various aspects. Ranking is measured with the Kendall's tau-b $K_\tau b$ correlation (range* $[-1, +1]$*). Content overlap is measured with various ROUGE variants (range* $[0, 1]$*). For ranking all 18 human summaries were used, whereas for overlap only 16 were used for each discussion (two outliers were removed)*

Discussion	Metric	μ Human–human	μ Human–machine
1	$K_\tau b$	0.557	0.374
	ROUGE-1	0.526	0.375
	ROUGE-2	0.357	0.203
	ROUGE-3	0.319	0.176
2	$K_\tau b$	0.584	0.683
	ROUGE-1	0.640	0.592
	ROUGE-2	0.502	0.409
	ROUGE-3	0.469	0.371

The machine summarisation algorithm produces a weight for each message in each thread. Based on this score, messages can be ranked by importance. The participants directly ranked the messages in each thread. We wanted to find out the degree of resemblance between the machine ranking and the rankings produced by the participants. We used Kendall's tau-b correlation score to compare the human and machine rankings. This score is defined as follows (Agresti 2002):

$$K_\tau b(C, D, T_1, T_2) = \frac{\#C - \#D}{\sqrt{(\#C + \#D + T_1) \cdot (\#C + \#D + T_2))}} \tag{10}$$

where C is the set of concordant pairs between two rankings R_1 and R_2; D is the set of discordant pairs; T_1 is the number of ties only in ranking R_1 and T_2 is the number of ties only in ranking R_2. The score is in the range $[-1, +1]$. A negative score indicates the two rankings are inversely related. The maximum positive score denotes the two rankings are identical. Whereas a score of 0 means no correlation exists.

We first compared the rankings for each discussion among humans by leaving one ranking out. This ranking was used to calculate the average Kendall's tau-b score against the remaining human rankings. The same approach was used to compare humans and the machine. This time comparing against the machine ranking instead of the human one left out. We only asked participants to rank the five most important messages. Therefore, the remaining messages, seven for the first discussion and three for the second, are all tied at rank six. Results are shown on the first row of Table 4 for each discussion.

For both discussions the machine ranking positively correlates with the human rankings on average. Especially for the second discussion the correlation is quite strong. The humans agree more with the machine ranking for this discussion than with each other. However, this discussion was also shorter than the first. We also observed that many people agree on what the most important message is in the thread. However, the agreement becomes lower as the rank decreases.

The machine produced an extractive summary for each of the two threads. We asked participants to do exactly the same. We wondered how similar the machine produced summary of each thread is to the human summaries. All summaries have the same length restriction. Therefore, we are comparing the recall of the machine with that of the human participants. Hence, we used the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) overlap metric to compare the extractive summaries, which we interpreted as follows (Lin 2004):

$$ROUGE\,(c, R, n) = \frac{\sum_{r \in R} \#\,(gram\,(r, n) \cap gram\,(c, n))}{\sum_{r \in R} \#gram\,(r, n)} \tag{11}$$

where c is the candidate summary; R the set of reference summaries and n the n-value for n-gramming. The $gram\,(a, n)$ function yields the n-grams for text a.

To compare the extractive summaries we did two leave-one-out tests for each discussion. First, in the human–human test, one human summary was left out and used as candidate summary, the remaining summaries formed the reference set. This process was repeated for each human summary and the resulting ROUGE scores were averaged. The outcome gives an impression of the extent to which humans agree with each other on the summary content. The second test was the human-machine test which also leaves each human summary out once. However, here the machine summary was used as candidate instead of the left-out human summary.

Table 4 shows the results for each discussion in the second to fourth rows. Note that we only used sixteen of the eighteen human summaries, since two summaries did not meet the length and content criteria we set. Scores shown are for ROUGE-1 (unigram), ROUGE-2 (bigram) and ROUGE-3 (trigram) overlap (values of n of 1, 2 and 3 in (equation 11)). If the scores seem high, remember that participants were asked to create an extractive summary and not to rephrase content, in essence 'pasting' sentences of the original discussion.

The first discussion seems more troublesome for humans than the second. A similar pattern can be observed for the machine. However, the machine performs much worse on the first discussion. Still, in the second discussion it achieves about 84 per cent of the human score on average, which is good. However, keep in mind that the second discussion thread was smaller and that its summary was allowed to be larger relative to its total size.

We made several observations regarding the human summaries: participants ordered the sentences in a non-chronological way with respect to the order in the original thread. The automatic summariser does not do this. Similar to the message ordering, we saw agreement among participants. This time on the few most important sentences in the thread based on the sentence selection frequency. However, frequently selected messages in the ordering task did not get a proportionally higher amount of selected sentences in this second task. This supports the idea that the PMR alone is not enough to distribute sentence weight. Hence, the addition of the PT factor is justified.

So, what are the judgements of the participants themselves: what did they think of the machine generated summary for each of the two threads? This is shown in Figure 4. We asked users to grade the machine summaries, as if judging summaries created by students, on a ten-point scale. A one is very bad, a six is sufficient and a ten is considered excellent. The summary of the first thread was rated with a 5.89

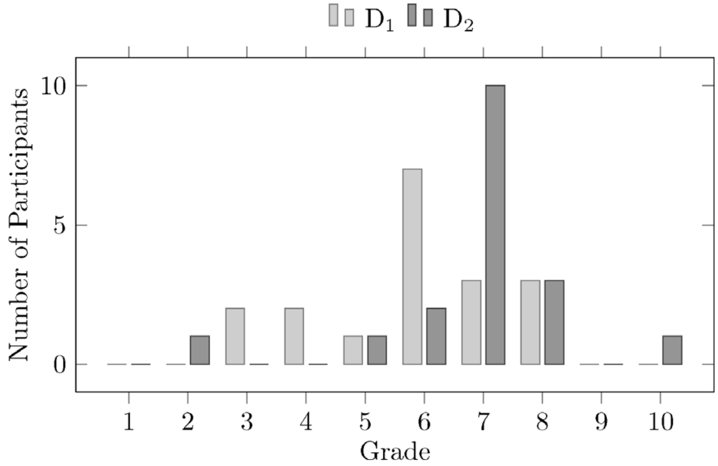

Fig. 4. Automatic summary grades ($n = 18 \cdot 2$). Grades were given for the first discussion D_1 and second discussion D_2 on a scale from one (low) to ten (high). A six is considered sufficient.

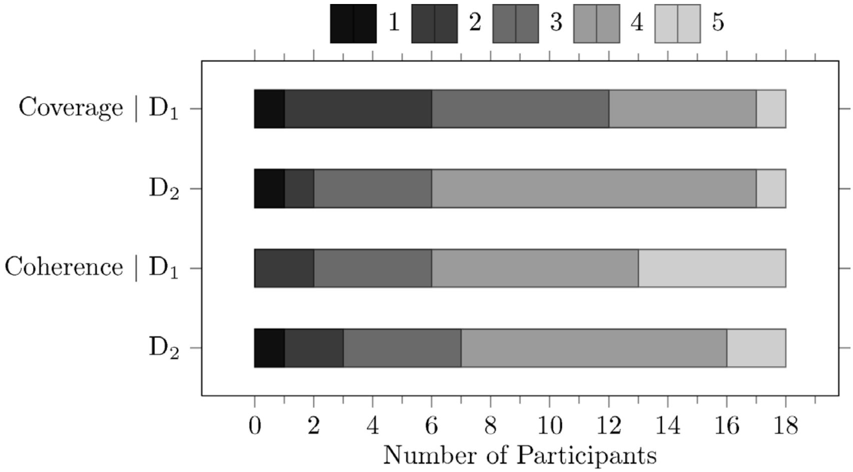

Fig. 5. Automatic summary user judgements ($n = 18 \cdot 4$). Coverage and coherence were judged for discussions D_1 and D_2 on a scale from one (bad) to five (good).

average score (median 6), the second was rated with a 6.83 (median 7). Note that for most users it took about 15 minutes to manually make an extractive summary of a thread. The developed prototype does this in several seconds.

We asked users to rate the coverage and coherence of the summaries on a five-point scale. Figure 5 shows that users rated coherence higher than the coverage.

Most respondents, 95 per cent, indicated an automatic thread summariser as either a useful or very useful tool. Half of the respondents would use the summariser every now and then, the other half would use it regularly or always when visiting fora. This is promising and shows there is genuine interest in the developed technology as well as a potential user base.

7 Conclusion

This paper shows a promising way to build automatic summaries of threads. We used a combination of heuristics and traditional Natural Language Processing technologies. Core parts of the presented methodology rely on existing metadata in threads. We have developed heuristics for recognising the referential structure between messages in threads, for filtering messages and for recognising the importance of contributions of authors. Finding sentence types and linking questions and answers aided sentence selection. Finally, we combined everything into one summarisation algorithm, implemented in a prototype system.

The performance of modules of the prototype system, that we could evaluate on external datasets, is on par with the state-of-the-art performance for Dutch. Results of the evaluation of the prototype system show promise. Little can be concluded about the thread types, this requires more quantitative evaluation. The grades can be subjectively interpreted as acceptable. However, we can objectively see from Kendall's Tau-b and ROUGE scores that the summariser does a fair job with message ranking and sentence selection compared with humans. Even though user judgements scores show that the coverage could be improved, the scores for coherence are high. This suggests that presenting the summary as a dialogue with anaphoric references expanded is a good approach.

Nevertheless, room for improvements exists and several aspects deserve more attention. In our opinion, the ideas behind question-answer coupling are sound, but a more generic, less domain data dependent, solution would be useful. Speech act analysis may be a good starting point for this. For sentence selection the fallback interleaving heuristic is a bit basic. The approach used for message filtering works well and correctly identifies messages with poor formatting characteristics by using the Message Formatting Score (MFS). However, the usage of Average Word Length and Sentence Length is not so fruitful in hindsight. The MFS almost always already identifies messages picked out by deviations of these length statistics.

The summarisation approach in its entirety has been tested only on a limited number of threads of specific types. To what extent this generalises to other threads is an open question. However, this research paints a clear picture of what is important in forum discussions. We conclude that creating automatic summaries of on-line discussions in Internet fora in the way presented in this paper is a promising idea. People that use fora can use it as an effective time saving technology. A good next step would be investigating this further by creating annotations on larger datasets with many participants. The outcome can be used for refining the heuristics and, possibly, for finding new patterns.

Acknowledgements

Thanks goes to Ruben Wassink for his thorough review of the thesis that forms the basis for this paper. We also thank Maarten Fokkinga, Marco Gerards, Paul Bakker, Dirk Heylen, Mariët Theune, Anton Nijholt, Maurice van Keulen and Gert Bolmer for their comments.

References

Agresti, A. 2002. *Categorical Data Analysis*, p. 68, 2nd ed. New York: Wiley-Interscience.

op den Akker, R., Hospers, M., Kroezen, E., Nijholt, A., and Lie, D. 2002. A rule-based reference resolution method for dutch discourse analysis. In *Proceedings of International Symposium on Reference Resolution in NLP*, Alicante, Spain, pp. 59–66.

Baldwin, T., Martinez, D., and Penman, R. B. 2007. Automatic thread classification for linux user forum information access. In *Proceedings of ADCS*, Melbourne, Australia, pp. 72–9.

Bird, S., Klein, E., and Loper, E. 2008. Natural language processing in python. `http://nltk.sourceforge.net/index.php/Book` (Draft Version 0.9.2).

Bogers, T. 2004. *Dutch Named Entity Recognition: Optimizing Features, Algorithms, and Output*. Master's thesis, University of Tilburg.

Bouma, G., van Noord, G., and Malouf, R. 2000. Alpino: wide-coverage computational analysis of Dutch. In *Proceedings of CLIN*, Tilburg, The Netherlands, pp. 45–59.

Carenini, G., Ng, R. T., and Zhou, X. 2007. Summarizing email conversations with clue words. In *Proceedings of WWW*, Banff, AB, Canada, pp. 91–100.

Coleman, M., and Liau, T. L. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology* **60**(2): 283–84.

Dalli, A., Yunqing, X., and Wilks, Y. 2004. FASIL Email summarisation system. In *Proceedings of COLING*, Geneva, Switzerland, pp. 994–1001.

van Deemter, K., and Kibble, R. 2000. On coreferring: coreference in MUC and related annotation schemes. *Computational Linguistics* **26**(2): 629–37.

DuBay, W. H. 2004. The principles of readability. Technical Report, Impact Information. `http://www.impact-information.com/impactinfo/readability02.pdf`.

van Eynde, F. 2004. *Part of Speech Tagging en Lemmatisering van het Corpus Gesproken Nederlands.* Centre for Computerlinguistics, Catholic University of Leuven.

Farell, R. 2002. Summarizing electronic discourse. *International Journal of Intelligent Systems in Accounting, Finance & Management* **11**: 23–38.

Farell, R., Fait-weather, P. G., and Snyder, K. 2001. Summarization of discussion groups. In *Proceedings of CIKM*, Atlanta, GA, pp. 532–34.

Feng, D., Shaw, E., Kim, J., and Hovy, E. 2006. Learning to detect conversation focus of threaded discussions. In *Proceedings of HLT-NAACL*, New York, pp. 208–15.

Francis, W. N., and Kûcera, H. 1979. Brown corpus manual. `http://icame.uib.no/brown/bcm.html`

Hoste, V., and Daelemans, W. 2005. Learning Dutch coreference resolution. In *Proceedings of CLIN'04*, Leiden, The Netherlands.

Hoste, V., and van den Bosch, A. 2007. A modular approach to learning Dutch co-reference resolution. In *Proceedings of WAR I*, Bergen, Norway, pp. 51–75.

Hovy, E. 2004. *The Oxford Handbook of Computational Linguistics: Text Summarization*, chapter 32, pp. 583–98. Oxford, UK: Oxford University Press.

Hovy, E., Hermjakob, U., and Ravichandran, D. 2002. Qtargets used in webclopedia. `http://www.isi.edu/natural-language/projects/webclopedia/Taxonomy`

Jurafsky, D., and Martin, J. H. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, p. 340. Upper Saddle River, NJ: Prentice-Hall.

Kim, J., Chem, G., Feng, D., Shaw, E., and Hovy, E. 2006a Mining and assessing discussions on the web through speech act analysis. In *Proceedings of ISWC*, Athens, GA.

Kim, J., Chem, G., Feng, D., Shaw, E., and Hovy, E. 2006b Modeling and assessing student activities in on-line discussions. In *Proceedings of AAAI EDM*. Boston, MA.

Kiss, T., and Strunk, J. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics* **32**(4): 485–525.

Klaas, M. 2005. Toward indicative discussion fora summarization. Technical Report UBC-CS TR-2005-04, University of British Columbia.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* **46**(5): 604–632.

Lam, D., Rohall, S. L., Schmandt, C., and Stern, M. K. 2002. Exploiting e-mail structure to improve summarization. In *Proceedings of CSCW (Interactive Posters)*, New Orleans, LA.

Lang, K. 1995. Newsweeder: learning to filter netnews. In *Proceedings of ICML*, Tahoe City, CA, pp. 331–39.

Lin, C.-Y. 2004. Looking for a few good metrics: ROUGE and its evaluation. In *Proceedings of NTCIR Workshop*, Tokyo, Japan, pp. 1765–76.

Manning, C., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*, p. 371. Cambridge, MA: MIT Press.

McKeown, K., Shrestha, L., and Rambow, O. 2007. Using question-answer pairs in extractive summarization of email conversations. In *Proceedings of CICLing*, Mexico City, Mexico, pp. 542–50.

Mitkov, R. 1999. Multilingual anaphora resolution. *Machine Translation* **14**(3–4): 281–99.

Rambow, O., Shrestha, L., Chen, J., and Lauridsen, C. 2004. Summarizing email threads. In *Proceedings of HTL/NAACL Short Papers*, Boston, MA, pp. 105–8.

Ratcliff, J. W., and Metzener, D. M. 1988. Gestalt: an introduction to the Ratcliff/Obershelp pattern matching algorithm. *Dr. Dobbs Journal*, 7, p. 46.

Rienks, R. 2007. *Meetings in Smart Environments: Implications of Progressing Technology*. Ph.D. thesis, University of Twente.

Sang, E. T. K. 2005. Language-independent named entity recognition. `http://www.cnts.ua.ac.be/conll2002/ner/`

Schuth, A., Marx, M., and de Rijke, M. 2007. Extracting the discussion structure in comments on news-articles. In *Proceedings of CIKM/WIDM*, Lisbon, Portugal, vol. 123, pp. 97–104.

Stegeman, L. 2007. Hammer tagger. `http://wwwhome.cs.utwente.nl/~infrieks/stt/stt.html`

Wan, S., and McKeown, K. 2004. Generating overview summaries of ongoing email thread discussions. In *Proceedings of COLING*, Geneva, Switzerland, pp. 549–56.

Weimer, M., Gurevych, I., and Mühlhäuser, M. 2007. Automatically assessing the post quality in online discussions on software. In *Proceedings of ACL Demo and Poster Sessions*, Prague, Czech Republic, pp. 125–28.

Zechner, K. 2002. Automatic summarization of open-domain multiparty dialogues in diverse genres. *Computational Linguistics* **28**(4): 447–485.

Appendix A: First discussion

This discussion is started by author *w* who is currently considering dropping out of university and looking for a job with his previously obtained diploma. He posts to a forum to get the opinions of others on the possibility of jobs with his diploma. This is a thread of *Problem–Solution* type. The original thread title is ‘Working in ICT with senior secondary vocational education (in ict system administration)’.

Messages

The following listing shows all source message in chronological order. The original thread was in Dutch, so the messages were translated to English. In the translation as much of the original wording and casing is preserved as possible, this includes subtle errors. In the original experiment the discussion was shown in full mark-up as used on the source forum. This listing merely shows the message text.

w_1 I completed my senior secondary vocational education ICT system administration last year. I am studying on the polytechnic university, but this is not going so well. My question is: How much work is there at the level of ICT system administrator? Are there people who started working after their senior secondary vocational education?

– What kind of employers are feasible (peak, call2 ?)

– What is a normal starter salary?

– What possibilities are there for further schooling while working?

Who started working after their senior secondary vocational education and how do you like it? What is your job? What does a working day look like? (extra pay?)

n_1 For salary and secondary employment conditions look at the topic 'What salary does an ICT worker get on average? (part 7)'.

I think that it should be easy to find a job, put your CV on monsterboard or something like that, plenty of people who will invite you for a job interview (even when I started with the polytechnic university after finishing my senior secondary eduction, employers were keen on hiring system administrators).

i_1 *cut recruitment*

Welcome to this forum. As you can read in the Work & Income - Policy: any form of recruitment / head hunting is forbidden on this forum.

m_1 Currently the market is good for job seekers. I put my cv on nvbank two weeks ago and my phone was ringing off the hook.

If you join a secondment company, you will likely be working at a customer for some time (depending on your education level and experience). Sometimes the duration is six months, sometimes shorter and naturally: sometimes longer. Often your employer will offer a range of courses and exams (keep into account your student debt though).

What a working day looks like depends on the customer:) Getting out of bed, working and going home, same as any other job: P Salary differs for each employer.

There are plenty of positions for junior system administrators, help desk and workplace management. Take a look at the job advertisement sites.

r_1 *[Cites message w_1 partially: 'I completed my senior secondary vocational education in ICT system administration last year.']*

Question: What certification did you obtain in those 4(?) years?

The study you mention is from 'before my time';)

r_2 *[Cites message m_1 partially w/o quote mark-up: 'There are plenty of positions for junior system administrators, help desk and workplace management. Take a look at the job advertisement sites.']*

The junior system administrators are currently situated India and Poland... It will be difficult to join them as a new colleague I am afraid :|

Via secondment it should be easy to get a job though ...

a_1 Put your cv on-line and you will find out. I also have the same education, and my phone was ringing off the hook. Everywhere I went for an interview I arrived and so I could pick from among 4 jobs. Salary is around 1,600–1,800 or so.

Whether further schooling is possible depends on the company, you can discuss it with them. For example, they pay for my certificates.

w_2 *[Cites messages r_1 partially: '[. . .] What certification did you obtain in those 4(?) years?'. The study you mention is from 'before my time';)]*

No certification. Yes, I completed itil and ecdl as subjects at school.

r_3 Do you know if you want to learn more or want to work? Difficult choice . . . An academic degree opens up a lot more opportunities, especially later in your career . . .

A first job in ICT is not really a challenge for you I think.

Do you have a job in mind that you think you'd like?

u_1 I started with the same education as you at TTP. It is possible for me to pursue MCSE, Cisco, et cetera. degrees.

If you want I can put you in contact . . .

c_1 *[Cites message w_1 in full]*

– What kind of employers are feasible (peak, call2 ?)

Plenty of secondment companies are eager to get new employees. Don't you worry, finding a job in ICT with such a diploma will be easy.

– What is a normal starters salary?

between 1,600 and 1,700 is not unrealistic

– What possibilities are there for further schooling while working?

Every decent company offers its employees possibilities for training and certification to improve their job performance and to enable promotion to a higher job. If you are going to work in the executive branch of ICT (meaning: without the word 'manager' in your job title) you will always need to keep learning. It is unavoidable.

who started working after their senior secondary vocational education and how do you like it?

3 years ago I went into industry with my senior secondary vocational diploma. Started at a service desk, one and a half years ago, and I like it a lot.

What is your job?

After one and a half year service desk I was promoted to chief network administrator

What does a working day look like?

Get up, drive to work, drink coffee, work a little bit, lunch, work a little bit, dinner, work a little bit more, drive home, enjoy your free evening without homework:)

h_1 *[Cites message w_1 in full]*

– What kind of employers are feasible (peak, call2 ?)

Indeed, via secondment companies it should be easy to find a job, it is not infeasible to start there even though your salary will not be that high. Don't pay too much attention to the vacancies on monsterboard where they demand a whole list of software knowledge and experience, just put

your cv on monsterboard and keep your phone on for a day. Guaranteed that, as ict-er of your level, your phone will be ringing constantly due to companies that want to recruit you.
– *What is a normal starters salary?*
I started last year with 1,400 gross as workplace manager, it has become 1,500 and starting next month it will be 1,600 if everything goes as planned.
– *What possibilities are there for further schooling while working?*
As much as you want, every decent company offers enough possibilities to keep learning, usually in the form of certificates, etc.
who started working after their senior secondary vocational education and how do you like it?
I tried university for a year, but I dropped out because I wanted to relocate and did not like to start afresh at a new school, etc., and so I started working and I do not regret it.
What is your job?
Workplace management at a large energy supply company.
What does a working day look like?
Drive to work, park the car, start laptop, get coffee, check e-mail, check the forum, read fok. Start planning, etc., check what incidents there are that day which I need to take care of. Solve incidents, etc., after that update / close the tickets and so on, thereafter continue until the day is over. On days like these I also regularly go to the coffee machine and have a friendly chat with people that are on the way there. It generally is a relaxed job. Sometimes it can be rather stressful, but that does not occur regularly. I will be here for another two months, I'll try to get some new certificates during this time so that I can transfer to more interesting assignments after this one.
(extra pay?)
Travel allowance if you use your own car to drive.

Properties and structure

Figure 6 shows the discovered structure of the thread. The discussion is 'wide', meaning that the first post yields many reactions. Most positional relevance is assigned to the first post w_1, the deeper part of the discussion between w and r and the chronologically final posts in the thread. Although r_2 cites m_1 in the original thread, this is not recognised, because the cited paragraph in r_2 is not marked up as citation. Message r_2 is linked to w_1 as fallback. Table A1 shows how the *pmr* values in Figure 6 were determined. See equation (1) in Section 3.2 for the *pmr* formula.

Table A2 shows other properties of authors and messages in the first thread. The numbers shown are based on the Dutch version of the thread, not the English translation. However, the MFS and pmr^{norm} are exactly the same for the English version, since these are language independent measures. The pt^{norm} is approximately the same, since it is partially based on the number of words in a message which may

Table A1. *Calculation of* pmr *values for the first discussion. Each term in the* pmr *formula is shown as a column. The final column is the sum of the other columns and represents the final* pmr *value as shown in Figure 6. Messages with the same term values are shown in one row*

Message(s)	$\#P_m$	$\#C_m$	A_m	$2 \cdot \left(H(m) - \frac{1}{2} \cdot HT\right)$	$\sum_{c \in C_m}$	*pmr*
w_1	0	8	0	$2 \cdot (1-2) = -2$	23	29
n_1, i_1, m_1, r_2, a_1	1	0	−1	$2 \cdot (2-2) = 0$	0	0
r_1	1	1	0	$2 \cdot (2-2) = 0$	13	15
w_2	1	2	0	$2 \cdot (3-2) = 2$	8	13
r_3,u_1	1	0	−1	$2 \cdot (4-2) = 4$	0	4
c_1	1	1	0	$2 \cdot (2-2) = 0$	3	5
h_1	2	0	−1	$2 \cdot (3-2) = 2$	0	3

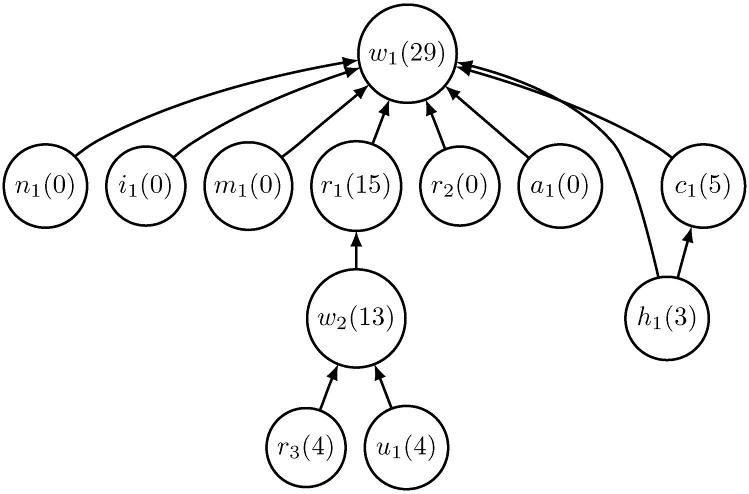

Fig. 6. First discussion thread structure.

differ slightly in the translation. The *pmr*^{norm} values are simply normalised values of those in the rightmost column of Table A1. We have omitted the Average Word Length and Average Sentence Length because they are specific to Dutch.

Summary

The requested summary size was twelve sentences. The result is shown below: source message identifiers on the left, the chosen sentence(s) from each message, numbered, on the right. Anaphoric references are shown between square parentheses.

Title: Working in ICT with senior secondary vocational education (in ict system administration)

w_1 01. I [*w*] completed my senior secondary vocational education ICT system administration last year.

02. I [*w*] am studying on the polytechnic university, but this is not going so well.

Table A2. *Author and message properties for the first thread including sentences assignments. The* pt^{norm} *values apply to authors and sum to one. The other metrics apply to individual messages. The last two columns show the number of assigned sentences assuming a twelve sentence summary. The* select *column shows select* n *sentences out of message length* m *as* n/m. *Since the left values in this column do not add up to twelve, an extra pass is made that removes sentence weight from messages. The rightmost column, labelled* r, *shows the final sentence count assignments. The structure of this table is the same as Table 3 in Section 5*

Author	pt^{norm}	Message	mfs	pmr^{norm}	$pt^{msgnorm}$	*weight*	*select*	*r*
w	0.14	w_1	0.75	0.40	0.07	0.29	4/11	4
		w_2	0.75	0.18	0.07	0.14	2/2	2
n	0.07	n_1	0.61	0.00	0.07	0.02	1/2	0
i	0.05	i_1	0.75	0.00	0.05	0.02	1/2	0
m	0.08	m_1	1.00	0.00	0.08	0.03	1/7	0
r	0.18	r_1	0.63	0.21	0.06	0.16	2/3	2
		r_2	0.90	0.00	0.06	0.02	1/3	0
		r_3	0.99	0.05	0.06	0.05	1/4	0
a	0.05	a_1	0.96	0.00	0.05	0.02	1/6	0
u	0.05	u_1	0.92	0.05	0.05	0.05	1/3	0
c	0.16	c_1	0.71	0.07	0.16	0.10	2/15	2
h	0.22	h_1	0.72	0.04	0.22	0.10	2/22	2

03. My [*w's*] question is: how much work is there at the level of ICT system administrator [secondary vocational education ICT system administration]?

04. who started working after their senior secondary vocational education and how do you like it?

r_1 05. Question: What certification did you [*w*] obtain in those 4(?)

06. The study you mention is from 'before my [*r's*] time' ;)

w_2 07. No certification.

08. Yes, I completed itil and ecdl as subjects at school.

c_1 09. Don't you [*w*] worry, finding a job in ICT with such a diploma will be easy.

10. If you [*w*] are going to work in the executive branch of ICT (meaning: without the word 'manager' in your [*w's*] job title) you [*w*] will always need to keep learning.

h_1 11. – What kind of employers are feasible (peak, call2 ?)

12. Indeed, via secondment companies it should be easy to find a job, it is not infeasible to start there even though your salary [a normal starters salary] will not be that high.

Why were these specific sentences selected? We give the heuristic which triggered the selection for each sentence:

01	Citation count (3)
02	Citation count (2)
03	Longest question sentence
04	Question-answer score (8)

05	Longest question sentence
06	Citation count (1)
07	Interleave
08	Interleave
09	Question-answer score (2)
10	Question-answer score (2)
11	Interleave
12	Interleave

There are some surface problems with the summary: for example in the summary part of r_1 the sentence boundary is incorrectly detected, as a consequence the word ‘years’ is missing after ‘4(?)’. Also h_1 repeats a sentence from the initial post which was not marked as citation. The summary contains the main question, some clarification and some concrete answers. The difficulty with this discussion is the number of questions and answers. The main question appears to be whether there is sufficient work at the level of w. The last line in the summary provides an answer to that question. A strongly related question, expressed in sentence 04, is detected by the question-answer scoring module and an answer to it is also included.

Appendix B: Second discussion

This is a discussion started by author s who supports the stance that people who are injured (or sick) because of their own fault should pay for their expenses themselves. This is a thread of *Statement–Discussion* type. The thread title is: ‘Sick due to your own doing? Then you should pay the costs yourself’.

Messages

The following table shows all source message in chronological order. The original thread was in Dutch, so the messages were translated to English. In the translation as much of the original wording and casing is preserved as possible, this includes subtle errors. In the original experiment the discussion was shown in full mark-up as used on the source forum. This list merely shows the text of each message.

s_1 There is huge difference between sports and sporting activities, especially with respect to injuries. Participating in competitive sports, like football, during one’s free time increases the risk of injury enormously.
To put the burden for this on the employer seems antisocial to me.
The fact that people are considering to let employees pay for this burden themselves is certainly not unjustified.

p_1 And what about other risk factors? For example: smoking or drinking alcohol. Should these not also be taken into account? Does the ‘your own fault’ criterion not lead to undesired intervention of the employer in the private life of the employee?
I believe there are only two possibilities, either the employer is solely responsible or the employee. A shared burden with the ‘your own fault’ criterion is unacceptable to me.

d_1 [*Cites message* p_1]

When there is an INDUSTRIAL accident then it naturally follows that the EMPLOYER is responsible ... the premium for the insurance is thus to be paid by the boss!

In all other cases it seems to me that the EMPLOYEE insures himself for his (self chosen) risks and pays the premium for that.

With self-chosen I mean sports, reckless driving, etc. There is no intervention in private life here ... every employee can be assumed to have a healthy set of brains and should thus be very well able to avoid unnecessary risks.

p_2 [*Cites message* d_1]

But you realise that endless discussions will ensue. Can absence due to illness be blamed on excessive usage of alcohol or is a high workload the root cause? Too little movement is a proven risk factor, is this also the responsibility of the employee? Etc.

This type of discussion is futile, therefore the responsibility for all absence due to illness should clearly be assigned to one of the parties.

a_1 May we please decide ourselves how to live our lives? It's becoming gradually worse here in the Netherlands. In a short while everyone's way of live will be dictated by others.

I wonder how much provision Mr. Hermes gets from insurance companies. It seems more and more like hidden advertising. nkb.

d_2 [*Cites message* p_2]

I think no one objects if the 'decision' concerning the root cause of the absence is dictated by the family doctor. Binding for both parties and this would prevent the endless discussions so feared by you.

Not the fault of the employee, then a reimbursement of 30 per cent up to 100 per cent ... unnecessary risks taken, so the employee's own fault, then be happy with a reward (for reckless behaviour) of 70 per cent.

h_1 [*Cites message* p_2]

This sort of response makes me want to take a drink.

e_1 [*Cites message* d_2]

I thought that (family) doctors have to keep their patients's information confidential ....

Properties and structure

All citations contained complete reference to the source message. Based on this and the heuristics the thread structure was inferred as shown in Figure 7. This discussion is 'deep': most messages refer to one specific post and are only referred to once. Message p_1 is determined to be the most important in the discussion. All links in the citations in this thread are explicit and thus the thread structure is correctly found. Table B1 shows how the *pmr* values in Figure 7 were determined. See equation (1) in Section 3.2 for the *pmr* formula.

Table B2 shows other properties of authors and messages in the second thread. As for the first discussion these are based on the Dutch version, but most of them are

Table B1. *Calculation of* pmr *values for the second discussion. Each term in the* pmr *formula is shown as a column. The final column is the sum of all columns and represents the final* pmr *value as shown in Figure 7*

Message(s)	$\#P_m$	$\#C_m$	A_m	$2 \cdot \left(H(m) - \frac{1}{2} \cdot HT\right)$	$\sum_{c \in C_m}$	*pmr*
s_1	0	2	0	$2 \cdot (1-3) = -4$	19	17
p_1	1	2	0	$2 \cdot (2-3) = -2$	18	19
a_1	1	0	−1	$2 \cdot (2-3) = -2$	0	$-2 = 0$
d_1	1	1	0	$2 \cdot (3-3) = 0$	16	18
h_1	1	0	−1	$2 \cdot (3-3) = 0$	0	0
p_2	1	1	0	$2 \cdot (4-3) = 2$	12	16
d_2	1	1	0	$2 \cdot (5-3) = 4$	6	12
e_1	1	0	−1	$2 \cdot (6-3) = 6$	0	6

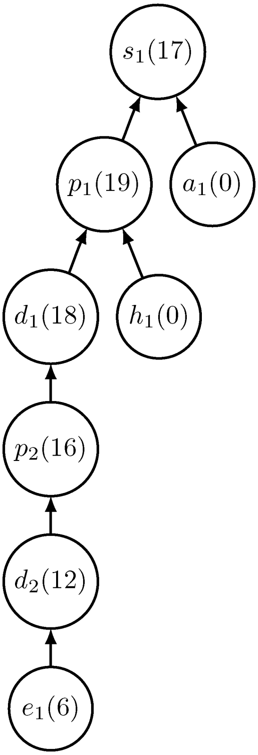

Fig. 7. Second discussion thread structure.

the same for the English version except for pt^{norm}. The pmr^{norm} values are normalised values of those in the rightmost column of Table B1.

Summary

The requested summary size was ten sentences. The result of the automatic summariser is shown below: source message identifiers on the left, the chosen sentence(s) from each message on the right. Anaphoric references are shown between square parentheses.

Table B2. *Author and message properties for the second discussion. The pt^{norm} values apply to authors and sum to one. The other metrics apply to individual messages. The last two columns show the number of assigned sentences assuming a ten sentence summary. The* select *column shows select* n *sentences out of message length* m *as* n/ m. *Since the left values in this column do not add up to ten, an extra pass is made that removes sentence weight from messages. The rightmost column, labelled* r, *shows the final sentence count assignments. The structure of this table is the same as Table 3 in Section 5*

Author	pt^{norm}	Message	mfs	pmr^{norm}	$pt^{msgnorm}$	*weight*	*select*	*r*
s	0.11	s_1	1.00	0.19	0.11	0.16	2/4	2
p	0.27	p_1	1.00	0.22	0.14	0.19	2/6	2
		p_2	0.95	0.18	0.14	0.16	2/4	2
a	0.08	a_1	0.60	0.00	0.08	0.03	1/5	0
h	0.12	h_1	0.50	0.00	0.12	0.04	1/1	0
d	0.31	d_1	0.99	0.20	0.15	0.18	2/4	2
		d_2	0.67	0.14	0.15	0.15	2/3	2
e	0.11	e_1	1.00	0.07	0.11	0.08	1/1	0

Title: Sick due to your own doing? Then you should pay the costs yourself

s_1 01. There is huge difference between sports and sporting activities, especially with respect to injuries.

02. The fact that people are considering to let employees pay for this burden themselves is certainly not unjustified.

p_1 03. And what [this burden] about the other risk factors?

04. A shared burden with the ‘your own fault’ criterion is unacceptable to me [*p*].

d_1 05. With self-chosen I [*d*] mean sports, reckless driving etc.

06. There is no intervention in private life here ... every employee can be assumed to have a healthy set of brains and should thus be very well able to avoid unnecessary risks.

p_2 07. Can absence due to illness be blamed on excessive usage of alcohol or is a high workload the root cause?

08. This type of discussion is futile, therefore the responsibility for all absence due to illness should clearly be assigned to one of the parties.

d_2 09. Binding for both parties and this [objects] would prevent the endless discussions [this type of discussion] so feared by you [*p*].

10. Not the fault of the employee, then a reimbursement of 30 per cent up to 100 per cent ... unnecessary risks taken, so the employee’s own fault, then be happy with a reward (for reckless behaviour) of 70 per cent.

Why were these specific sentences selected? We give the heuristic which triggered the selection for each sentence:

01	Interleave
02	Interleave

03	Longest question sentence
04	Citation count (2) + Sentence index
05	Citation count (1) + Sentence index
06	Citation count (1) + Sentence index
07	Longest question sentence
08	Citation count (1) + Sentence index
09	Citation count (1) + Sentence index
10	Citation count (1) + Sentence index

Many messages are cited in full one or more times in this thread. This leads to the situation where all sentences in a message, that are not a citation themselves, have the same citation count. Selection among such sentences is based on the sentence index: sentences that occur later in the message are preferred over earlier ones. This mechanism is the way that most of the sentences were selected in this thread. Although, there are some exceptions. For example, even though message p_1 contains as much as three question sentences, the first one, selected sentence 03 in the summary, is the longest question sentence that stands on itself.

The main statement really is in the thread title and there is some support for this in the sentences selected from the first message s_1. The thread is short and the summary relatively long. This increases the chance of picking good summary sentences. Nevertheless, there are still some potential pitfalls such as short sentences and sentences that add little to the discussion like the one in message h_1. The algorithm manages to prevent these sentences from being included in the summary.