Query-based summarization of discussion threads

Suzan Verberne
Leiden Institute for Advanced Computer Science, Leiden University
s.verberne@liacs.leidenuniv.nl

Emiel Krahmer
Tilburg School of Humanities, Tilburg University

Sander Wubben
Tilburg School of Humanities, Tilburg University

Antal van den Bosch
Centre for Language Studies, Radboud University & Meertens Institute

(Received)

Abstract

In this paper we address query-based summarization of discussion threads. New users can profit from the information shared in the forum, if they can find back the previously posted information. However, discussion threads on a single topic can easily comprise dozens or hundreds of individual posts. Our aim is to summarize forum threads given real web search queries.

We created a data set with search queries from a discussion forum’s search engine log and the discussion threads that were clicked by the user who entered the query. For 120 thread–query combinations, a reference summary was made by five different human raters. We compared two methods for automatic summarization of the threads: a query-independent method based on post features, and Maximum Marginal Relevance (MMR), a method that takes the query into account. We also compared four different word embeddings representations as alternative for standard word vectors in extractive summarization.

We find (1) that the agreement between human summarizers does not improve when a
query is provided; (2) that the query-independent post features, as well as a centroid-based baseline outperform MMR by a large margin; (3) that combining the post features with query similarity gives a small improvement over the use of post features alone; (4) that for the word embeddings, a match in domain appears to be more important than corpus size and dimensionality. However, the differences between the models were not reflected by differences in quality of the summaries created with help of these models.

We conclude that query-based summarization with web queries is challenging because the queries are short, and a click on a result is not a direct indicator for the relevance of the result.

1 Introduction

User-generated content in online forum communities is a valuable source of information. For example, it has been shown that patients are better informed if they participate in online patient communities (van Uden-Kraan et al., 2009). This is not only true for patients that post messages themselves, but also for ‘lurkers’ (i.e. forum users that do not post but only read) (van Uden-Kraan et al., 2008). New community members can profit from the information shared in the forum, if they can find back the previously posted information. However, discussion threads on a single topic can easily comprise dozens or hundreds of individual posts, which makes it difficult to find the relevant information in the thread (Bhatia & Mitra, 2010). This has motivated the development of text mining methods for disclosing the information in forum communities, combining free text search with information extraction and summarization (van Oortmerssen et al., 2017). Automatic summarization can pivot information finding in long threads by reducing a thread to only the most important information, which can be helpful for patient communities, but also for many other kinds of discussion forums.

Following previous work in thread summarization (Bhatia et al., 2014; Verberne et al., 2017), we take an extractive summarization approach (Hahn & Mani, 2000): extracting salient units of text from a document and then concatenating them to form a shorter version of the document. We approach the thread summarization task
as a post selection problem: selecting the most relevant posts from the thread and showing them in their original order, hiding the non-selected posts in between (Verberne et al., 2017). In this paper, we add an important aspect of user interaction for thread summarization: a search query for which a forum thread was retrieved. Our motivation for moving to query-based thread summarization is because discussion forums are often accessed through keyword queries.

Previous work in the context of the Document Understanding Conference (DUC) showed that having a question to focus the summary may help to improve agreement between the human reference summaries (Dang, 2005). However, instead of using the elaborate queries developed for the DUC tasks (consisting of a title and a description), we address the problem of thread summarization for real user queries. We use queries entered in the search engine of a large, open discussion forum. These queries are short and underspecified, like queries entered in general web search engines and social media search engines such as Twitter search (Teevan et al., 2011). Some examples from the query log on which we based our experiments that illustrate the query types, are (translated to English): ‘samsung’, ‘trampoline’, ‘friends rotterdam’, ‘baby 7 months’, ‘marathon’, ‘involuntary childlessness’, ‘threesome’ and ‘irregular work hours’. This type of real user queries are much shorter, and contain less information than the DUC-style topics.

Thus, in this paper we address the problem of query-based summarization with short user queries and we evaluate existing methods for that problem. A commonly used method for query-based summarization, especially in the context of web retrieval, is Maximal Marginal Relevance (MMR) (Carbonell & Goldstein, 1998). In previous works, MMR was successfully used for extractive summarization of meetings by Murray, Renals and Carletta (2005), a task that is similar to discussion thread summarization, as meetings also consist of turns from different participants to the discussion. However, as opposed to the task that we address, the task addressed by Murray et al. was query-independent. In this paper, we evaluate MMR for query-dependent extractive summarization of discussion threads using short user
queries, and compare it to a common query-independent method based on generic
post features such as length, position and centrality.

In addition, we follow up on work that shows the value of word embeddings as
text representations in automatic summarization (Denil et al., 2014; Nallapati et al.,
2016a; Zhang et al., 2016). Word embeddings are vector representations of words
that represent the contextual semantics of words in a large corpus (the neighboring
words, approximating the meaning of a word). Word embeddings can be learned
from raw text without supervision. They became very popular with the release
of word2vec in 2013, a word embedding toolkit that can efficiently train vector
space models on large corpora (Mikolov et al., 2013). In this paper, we compare
word embeddings to standard word-based vectors for query-based discussion thread
summarization.

Our research questions are:

RQ1 How do human summaries of discussion forum threads change when a short
user query is provided as focus of the summary?

RQ2 How does Maximum Marginal Relevance (MMR) perform on forum threads
with real user queries?

RQ3 Can we improve over MMR by including forum-specific summarization fea-
tures in the model?

RQ4 Does the use of word embeddings instead of standard word vectors as text
representations lead to better summaries?

The contributions of this paper compared to previous work are: (1) we evaluate
the inter-rater agreement for query-based summarization with user queries from a
query log; (2) we show that MMR fails for query-based discussion thread summa-
ritization with real user queries; (3) we prove the robustness of an extractive sum-
marization approach for discussion summarization based on generic post features;
(4) we confirm the value of word embeddings as text representations in extractive
summarization.

The organization of this paper is as follows: In Section 2 we discuss related re-
search. In Section 3 we present the data set that we constructed for our experiments,
Query-based summarization of discussion threads

followed by a description of our methods in Section 4. In Section 5 we analyze the collected reference summaries in order to answer RQ1 and RQ4, and in Section 6 we present the experimental results for answering RQ2 and RQ3. We answer and discuss our research questions in Section 7 and formulate our conclusions in Section 8.

2 Related work

In this section, we first discuss previous literature on methods for extractive summarization (Section 2.1), followed by more specific literature on query-based summarization (Section 2.2) and summarization of discussion forum threads (Section 2.3), and finally related work on supervised learning with multiple human reference summaries (Section 2.4).

2.1 Methods for extractive summarization

In extractive summarization, the most salient units of text from the original document are concatenated to form a shorter version of the document. This is different from abstractive summarization, where the text is re-written (abstracted) into a shorter text. The identification of the most salient text units in extractive summarization can be approached as a selection problem, a classification problem, or a ranking problem (Das & Martins, 2007). In the selection approach (Carbonell & Goldstein, 1998; Nallapati et al., 2016a), units are selected one by one in descending order of relevance, while taking into account the previously selected units. In the classification approach (Kupiec et al., 1995; Nallapati et al., 2016a), each unit is classified independently of the other units as either relevant or non-relevant. In the ranking approach (Gong & Liu, 2001; Toutanova et al., 2007; Svore et al., 2007; Metzler & Kanungo, 2008; Amini & Usunier, 2009; Shen & Li, 2011; Sipos et al., 2012; Dlikman & Last, 2016) each unit is assigned a relevance score, then the units are ranked by this score, and the most relevant units are selected based on a threshold or a fixed cutoff (a predefined number of units or words). In this paper, we evaluate both a selection approach (selecting posts one by one in descending
order of relevance) and a ranking approach (scoring all posts, rank them by the score and evaluate the ranking).

For most document types, the summarization units are sentences (Gupta & Lehal, 2010). In the case of conversation summarization the units are utterances (Marge et al., 2010; Liu & Liu, 2008; Murray et al., 2005; Penn & Zhu, 2008), and for discussion thread summarization the units typically are posts (Bhatia et al., 2014; Verberne et al., 2017). Most methods for extractive summarization select sentences based on human-engineered features. These include surface features such as sentence position and length (Radev et al., 2004), the presence of title words, the presence of proper nouns, content features such as word frequency (Nenkova & McKeown, 2012) and the presence of prominent terms (Lin & Hovy, 2000). Recently, deep neural network (DNN) based approaches have become popular for extractive summarization. The motivation is that creating unit representations with the use of DNNs can “avoid the intensive labor in feature engineering” ((Zhang et al., 2016):1) by using low-dimensional vector representations (Cheng & Lapata, 2016; Zhang et al., 2016).

Most DNN-based methods for extractive summarization use Convolutional Neural Nets (CNN) to create abstract unit representations in the form of word embeddings (Denil et al., 2014). Some methods use word embeddings together with position information to represent units (Zhang et al., 2016). These unit representations are then used for calculating document centrality (sometimes called salience or representativeness), measuring how well the unit represents the information in the complete document. A number of methods combine salience with diversity in order to reduce redundancy (Yin & Pei, 2015; Tsai et al., 2016). Fewer works use Recursive Neural Networks (Cao et al., 2015a) or Recurrent Neural Networks (RNNs) (Nallapati et al., 2016b; Nallapati et al., 2016a), or a combination of CNNs and RNNs (Cheng & Lapata, 2016).

The most important findings of the work with DNNs are (a) that word embeddings are successful text representations for the estimation of document salience (Denil et al., 2014; Nallapati et al., 2016b; Nallapati et al., 2016a), but: (b) the improvements over traditional methods are small and only significant for some
datasets (Tsai et al., 2016); (c) when increasing the number of layers in the network (which is only sensible for large datasets), the performance increases marginally and might even decrease because of overfitting (Zhang et al., 2016); and (d) the models suffer from domain adaptation issues when tested on a different corpus (Nallapati et al., 2016a). Following up on those findings in this paper, we experiment with the use of word embeddings for the estimation of document centrality (representativity) and query relevance.

2.2 Query-based summarization

Many previous methods for query-based summarization are directed at multi-document summarization (Cao et al., 2015b; Li & Li, 2014). In the literature on automatic summarization, two distinct motivations are given for query-based summarization. The first is information finding: Query-based document summaries make the result list of a search engine more tailored towards the information needs of the user (Park et al., 2006; Pembe & Güngör, 2007). It was shown that users of search engines perform relevance judgments more accurately and more quickly when they are presented with a summary of the retrieved documents (Tombros & Sanderson, 1998; Nenkova & McKeown, 2011).

The second motivation is the development of higher quality reference summaries: the idea is that having a question to focus the summary can help to improve agreement in content between reference summaries (Dang, 2005). In the context of the Document Understanding Conference (DUC), query-based summarization was part of a shared task in 2005 with 32 participating automatic summarization systems (Dang, 2005). The systems had to answer a list of complicated questions, compiling the answer from collections of 25 to 50 texts. For each topic, four human-written summaries were created as reference (Hovy et al., 2006). In 2006, the goal of the shared task moved to real-world complex question answering. The task was to compose a “fluent, well-organized summary such that the target length does not exceed 250-words and that the summary of the documents should answer the questions in the topic statement.” ((Mohamed & Rajasekaran, 2006):1)
The most-referenced method for query-based summarization is Maximal Marginal
Relevance (MMR) (Carbonell & Goldstein, 1998). MMR combines query-relevance
with information novelty (diversity). The goal is to minimize redundancy while
maximizing query relevance when selecting text units for extractive summarization.
Text units are chosen according to a weighted combination of their relevance to
the query and their novelty with respect to the text units that have already been
selected. MMR is defined as:

\[
\text{MMR} = \arg \max_{T_i \in T \setminus S} \left[ \lambda (\text{Sim}(T_i, Q)) - (1 - \lambda) \max_{T_j \in S} \text{Sim}(T_i, T_j) \right]
\] (1)

\(T\) is the set of all text units in the document that might be selected for the extractive
summary; \(S\) is the set of previously selected units, and \(Q\) is the query. \(\text{Sim}(T_i, Q)\)
and \(\text{Sim}(T_i, T_j)\) are both similarity measures, possibly computed in the same way
(e.g. cosine similarity). \(\lambda\) defines the relative weight of both similarity components.
Thus, MMR optimizes the weighted linear combination between the similarity of
a text unit to a query (positively) and the similarity between the text unit and
the most similar previously selected text unit (negatively), the latter component
being a measure of diversity. MMR has been shown to be a successful method for
query-based conversation summarization in previous work (Murray et al., 2005).

There is some previous work addressing query-based summarization of user-
generated content. Schilder and Kondadadi (2008) present a method for query-
based summarization of customer reviews, where the query refers to a specific
product. One previous paper addresses query-based summarization of discussion
forums (Hussain et al., 2014). Like us, they approach the problem as a post rank-
ing task, but the main difference is that they propose a two-step approach, where
the first step is the retrieval of relevant posts and the second step is further sum-
marizing the retrieved posts. For the summarization, they identify relevant word
features by applying Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation
(LDA). The authors do not provide an experimental evaluation, only showing
precision scores for four queries, ranging from 62% to 78%.
2.3 Summarization for discussion forum threads

Over the last decade, some research has been directed at the summarization of forum threads (Zhou & Hovy, 2005; Zhou & Hovy, 2006; Tigelaar et al., 2010). The majority of the recent work addressed summarization of comment threads on news websites (Ren et al., 2011; Llewellyn et al., 2014; Giannakopoulos et al., 2015; Kabadjov et al., 2015; Aker et al., 2016; Barker et al., 2016). In the past decade, abstractive summarization techniques have been successfully applied to tasks related to discussion thread summarization, such as summarizing email threads (Zajic et al., 2008), summarizing spoken and written conversations and meetings (Mehdad et al., 2014; Oya et al., 2014), and Twitter topic summarization (Zhang et al., 2013). In our work, we choose to use extractive summarization, because in the retrieval of discussion threads we consider it to be of importance that the individual opinions of the forum members are not rephrased or aggregated.

The work that is most related to our work are the papers by Krishnamani, Zhao and Sunderraman (2013) and Bhatia and Mitra (2014). Krishnamani et al. use topic models for representing forum posts, combined with post clustering on the basis of proper nouns, author and date. They evaluate their method on both the DUC 2007 benchmark for multi-document summarization and on their own forum data and compare their results to the results of centroid-based summarization (MEAD (Radev et al., 2004)). While on the DUC data their method outperforms MEAD, it is not consistently better on the forum data: for short summaries the centroid-based baseline gives better results than their experimental setting (Krishnamani et al., 2013).

Bhatia et al. take a feature-based approach in selecting the most relevant posts from a thread, thereby particularly investigating the use of dialog acts in thread summarization. They evaluate their method on two forums: ubuntuforums.org and tripadvisor.com, and find that for both datasets incorporating dialog act information as features improves results (Bhatia et al., 2014).

More recently, Condori and Pardo (2017) have compared extractive and abstractive methods for opinion summarization, focusing on product reviews. They found
that summaries produced by extractive summarization methods have better informativeness than summaries produced by abstractive methods, although the extractive methods led to more redundancy in the summaries. Evaluation with human judges showed that in terms of the utility of the opinion summaries, the extractive methods were better than the abstractive ones (Condori & Pardo, 2017).

2.4 Supervised learning using multiple reference summaries

Reference summaries created by humans are commonly used for the evaluation of summarization systems (Dang, 2005). Summarization is an inherently subjective task: human summarizers tend to disagree to some extent on the information that should be included in the summary (Marge et al., 2010; Liu & Liu, 2008; Penn & Zhu, 2008). The agreement between human raters on the content of an extractive summary can be measured using the proportions of selected and non-selected units, and the percentage of common decisions (selected/non-selected). Agreement is then calculated in terms of Cohen’s κ (Radev et al., 2003) for two raters or Fleiss’ κ for multiple (more than two) raters (Landis & Koch, 1977). In related work, κ scores between 0.1 and 0.3 are reported for the summarization of conversations (Marge et al., 2010; Liu & Liu, 2008; Penn & Zhu, 2008) and a κ of 0.219 for discussion thread summarization specifically, which indicates fair agreement (Verberne et al., 2017).

The way we approach the problem of low inter-rater agreement is by having multiple raters create reference summaries. This was also done in the context of DUC, where multiple reference summaries per topic (at least four and up to nine) were created (Dang, 2005). With five human summarizers per thread, we use voting as a measure for relevance: a post selected by (almost) all summarizers is more relevant than a post selected by zero or few summarizers. We use the number of votes for a post as dependent variable in a linear regression model (LRM) predicting the relevance of that post in a thread. The approach taken by Parthasarathy and Hasan (2015) on extractive speech summarization is similar to our method in that respect: They make use of multiple different reference summaries by assuming that
Fig. 1. A screenshot of three messages in a thread on the Viva forum (left) with the translation to English (right). The bottom message contains a quote of an earlier post.

Text units that are selected by more raters are of higher importance than text units selected by fewer raters (Parthasarathy & Hasan, 2015).

3 Data

3.1 The Viva forum dataset

In this study, we used data from the Viva forum\(^1\), a Dutch discussion forum with a predominantly female user community. The Viva forum has 2.8 Million registered users and 12 Million page views per month, which makes it one of the largest Dutch-language web forums.\(^2\) Figure 1 illustrates what messages on the forum look like.

We created a reference data set for query-based thread summarization based on a sample of threads summarized in a previous summarization study (Verberne et al., 2017).\(^3\) In that study, we presented human raters from the Viva target group

---

\(^1\) [http://forum.viva.nl](http://forum.viva.nl)

\(^2\) [http://www.sanoma.nl/product/viva-online/](http://www.sanoma.nl/product/viva-online/)

\(^3\) That data is available at [http://discosumo.ruhosting.nl](http://discosumo.ruhosting.nl)
with a discussion thread of between 20 and 50 posts (median length: 34 posts) and asked them to select the most important posts. Each thread was summarized by ten different raters. We also asked the raters how useful it would be for this thread to have the possibility to see only the most important posts (scale 1–5). 106 threads were summarized in total: 100 randomly selected threads that have at least 20 posts, and 8 additional threads from category ‘Digi’ – comprising technical questions – that have at least 20 posts, to be sure that our sample included problem-solving threads.

For the current follow-up study with query-based summarization, we obtained the query logs of the Viva forum search functionality. The query log contains over 1.5 Million queries entered between May and December 2015. The average query length in the log data is 1.5 words, which is shorter than written queries in web search engines (3.2 words, according to (Guy, 2016)) and comparable to queries in Twitter search (1.6 words, according to (Teevan et al., 2011)).

There are 546,949 clicks on pages in the Viva query log. 329,653 of these lead to forum threads; the others are other pages in the Viva domain, such as blog posts and articles. From the 106 threads that were summarized in our previous study, we only kept the threads for which at least half of the participants had given a usefulness score of $\geq 3$, indicating that it would be useful for a thread to have the possibility to see only the most important posts. From the selected threads, we removed the threads without clicks in the query log. The result is a set of 42 threads with at least one unique query. Most threads had multiple queries connected to them in the query log; the total number of unique thread-query combinations is 120. We included all these combinations in our sample. Although this sample seems small at first sight, it is large in comparison to other manually created reference data sets for automatic summarization, especially because each thread-query pair is summarized by five human summarizers. Table 1 shows a number of example query-thread combinations.

4 Search bar at the bottom of this page: http://forum.viva.nl/
Table 1. Example queries with titles of clicked threads in the Viva query log. The queries and titles have been translated to English for the reader’s convenience. ‘peach1990’ is a forum user name.

<table>
<thead>
<tr>
<th>Query</th>
<th>Title of clicked thread</th>
</tr>
</thead>
<tbody>
<tr>
<td>activities with toddler</td>
<td>Am I strict enough? Toddler troubles..</td>
</tr>
<tr>
<td>Toddler</td>
<td>Am I strict enough? Toddler troubles..</td>
</tr>
<tr>
<td>peach1990</td>
<td>Am I strict enough? Toddler troubles..</td>
</tr>
<tr>
<td>end of relationship</td>
<td>End of relationship and pregnant</td>
</tr>
<tr>
<td>no relationship</td>
<td>End of relationship and pregnant</td>
</tr>
<tr>
<td>relationship</td>
<td>End of relationship and pregnant</td>
</tr>
<tr>
<td>relationship over and now</td>
<td>End of relationship and pregnant</td>
</tr>
<tr>
<td>pregnant</td>
<td>End of relationship and pregnant</td>
</tr>
<tr>
<td>pregnant relationship</td>
<td>End of relationship and pregnant</td>
</tr>
<tr>
<td>nub</td>
<td>Nub theory 12 week scan</td>
</tr>
<tr>
<td>nub theory</td>
<td>Nub theory 12 week scan</td>
</tr>
</tbody>
</table>

3.2 Reference summaries by human raters

We recruited raters in the Viva forum target group (female, 18–45 years old) via the research participant system of Radboud University as a form of targeted crowdsourcing. All raters were familiar with the Viva forum. Each query–thread combination was summarized by five different raters. The raters decided themselves how many threads they wanted to summarize. At their first login they were presented with one example thread to get used to the interface. After that, they were presented with a randomly selected thread from our sample. They were paid a gift certificate if they completed at least ten threads. Figure 2 shows a screenshot of the post selection interface. On top of the screen the query is shown in a search box-like format with the word ‘Zoekvraag’ (search query) on the left. The left column of the screen shows the complete thread; the right column shows an empty table. By clicking on a post in the thread on the left it is added to the column on the right.

5 Targeted crowdsourcing is a form of crowdsourcing in which workers are selected who are likely to have the skills needed for the target task, instead of open recruitment on a crowdsourcing platform (Chowdhury et al., 2014; Chowdhury et al., 2015).
Fig. 2. A screenshot of the post selection interface. The query (‘Zoekvraag’) is on top of the screen, and stays visible when scrolling. The left column (‘Volledige topic’) shows the full thread while the right column (‘Jouw selectie’) shows the rater’s selected threads (with the first post always selected). In the blue header, the category and title of the thread are given. Each cell is one post, starting with the author name and the timestamp.

(in the same position): by clicking it in the right column it disappears again. The opening post of the thread was always selected.

The instructions in the left column read: “Please select the pieces of text (by clicking them one by one) that you think are the most important for the thread, given the search query. You can determine the number of selected posts yourself, but try to create a concise summary in which there is not too much redundant information.” The instruction text in the right column reads: “By reading your selection of posts you can check whether you created a good summary of the topic. You can remove posts from your selection by clicking on them. Click on the ‘Submit selection’ button if your selection is final. If you did not select any posts, please explain in the comments field why.”

It is possible that the thread is not relevant to the query, because a click on a result does not necessarily indicate its relevance (Dupret & Liao, 2010). Therefore, we also asked the raters to assess the relevance of the thread for the query, with the
following explanation (in Dutch, below the search box with the query): “This query was entered in the search engine of the Viva forum by a forum visitor. The thread you see below was clicked by her. It could be that the thread was not as relevant as the visitor thought; for that reason we ask you to indicate the relevance of the thread for the query at the bottom of the screen.” The relevance was assessed by the raters on a scale of 1–5, 1 meaning ‘completely irrelevant’ and 5 meaning ‘highly relevant’. This is similar to how relevance assessments are traditionally created in the context of evaluating information retrieval systems (Alonso & Baeza-Yates, 2011; Kekäläinen & Järvelin, 2002). One caveat is that it is sometimes difficult for raters to judge the relevance of the results if they were not the original poster of the query; the raters can only rely on the surface form of the query, without any context.

4 Methods

In this section we describe two methods that we implemented: one method using query-independent post features (Section 4.1), and the query-dependent method Maximum Marginal Relevance (MMR) (Section 4.2). In Section 6.1 we list the baselines to which we compare our methods.

4.1 Method 1: query-independent post features (PF)

With five raters per thread, each post receives between 0 and 5 votes. We argue that the number of votes for a post is an indicator of its relevance: a post that is selected by all five raters can be expected to be more relevant than a post that is selected by only one or two raters. Thus, we train a linear regression model (LRM) in which the number of votes for a post is the dependent variable and a set of post features are the independent variables. The post features that we used as independent variables are taken from the literature on extractive summarization (Weimer et al., 2007; Tigelaar et al., 2010; Bhatia et al., 2014; Verberne et al., 2017). The most commonly used feature types in the literature are the position of the post in the thread, the representativeness (centrality) of the post for the thread, the prominence of the
Table 2. Word2vec models that we compare for the representativeness features. The Wikipedia model was pre-trained the Viva models were trained by us. All models were trained with a window size of 11 and a minimum count of 5 for words to be included in the model.

<table>
<thead>
<tr>
<th>Model name</th>
<th>Corpus</th>
<th>Corpus size</th>
<th>Number of dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiki320</td>
<td>Dutch Wikipedia</td>
<td>392 Mw</td>
<td>320</td>
</tr>
<tr>
<td>Viva320</td>
<td>Viva forum sample</td>
<td>22 Mw</td>
<td>320</td>
</tr>
<tr>
<td>Viva200</td>
<td>Viva forum sample</td>
<td>22 Mw</td>
<td>200</td>
</tr>
<tr>
<td>Viva100</td>
<td>Viva forum sample</td>
<td>22 Mw</td>
<td>100</td>
</tr>
</tbody>
</table>

author, the readability of the post, and the popularity of the post. All these features are language-independent, meaning that they can be extracted for any discussion forum thread without knowing the language of the thread.

Because of the successful results with word embeddings as representation of text units in extractive summarization (Denil et al., 2014; Nallapati et al., 2016a; Zhang et al., 2016), we added representativeness features to the feature set that are based on word2vec representations. An advantage of using the word embeddings instead of the literal words is that the similarity between a post and the title will be non-zero even if none of the words overlap. For example, the cosine similarity between the literal word vectors for the following example title and post is zero, while the cosine similarity between the word2vec representations is relatively high, due to semantic similarity between the individual words:

- title: “Op date gaan terwijl je niet helemaal lekker in je vel zit…”
- Going on a date while you are not feeling well.
- post: “Laat even weten hoe het gegaan is Avonturiertje? Fijne avond”
- Let us know how it went Avonturiertje? Have a good evening

We compared a number of word2vec models for this purpose, which are listed in Table 2. We used one pre-trained model that was trained on the Dutch Wikipedia.
by Tuikens, Emmery and Daelemans (2016).\(^6\) In addition, we trained three models on the Viva corpus, using gensim with the same parameters as the Wikipedia model (min_count=5, window=11, negative=15),\(^7\) only varying the number of dimensions. The model \texttt{Viva320} has the same number of dimensions as the pre-trained Wikipedia model; we chose the dimensionality of 100 (\texttt{Viva100}) because that is the default value in gensim word2vec, and the dimensionality of 200 (\texttt{Viva200}) as middle way between 100 and 320. When training the word2vec models on the Viva corpus, we excluded the threads that are in our sample in order to prevent overfitting.

We used each of these models with two different methods for computing the similarity between a post and the thread (or title) as measure for the post’s representativeness: cosine similarity and word mover’s distance (Wan & Peng, 2005; Kusner \textit{et al.}, 2015). For computing the cosine similarity we took the following steps:

1. get the word vector from the word2vec model for each of the non-stopwords in the post;
2. average these word vectors into one vector representing the post (Kenter \textit{et al.}, 2016);
3. get the word vector from the word2vec model for each of the non-stopwords in the thread (or title);
4. average these word vectors into one vector representing the thread (or title);
5. measure the cosine similarity between the two average vectors.

For the word mover’s distance, we used the implementation by Kusner, Sun, Kolkin, and Weinberger (2015).\(^8\) We converted distance to similarity by using 1 minus the word mover’s distance as similarity measure. Before evaluating the models in the

---

\(^6\) Available from \url{https://github.com/clips/dutchembeddings} (Tuikens \textit{et al.}, 2016)

\(^7\) An explanation of the parameters can be found at \url{https://radimrehurek.com/gensim/models/word2vec.html}.

\(^8\) \url{http://vene.ro/blog/word-movers-distance-in-python.html}
Table 3. Post features

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>absolute position in the thread</td>
</tr>
<tr>
<td>Position</td>
<td>relative position in the thread (post position divided by no of posts)</td>
</tr>
<tr>
<td>Popularity</td>
<td># of responses (quotes) to the post</td>
</tr>
<tr>
<td>Representativeness</td>
<td>cosine sim between post and thread (term counts vectors)</td>
</tr>
<tr>
<td>Representativeness</td>
<td>cosine sim between post and title (term counts vectors)</td>
</tr>
<tr>
<td>Readability</td>
<td>word count</td>
</tr>
<tr>
<td>Readability</td>
<td>unique word count</td>
</tr>
<tr>
<td>Readability</td>
<td>type-token ratio</td>
</tr>
<tr>
<td>Readability</td>
<td>relative punctuation count (no of punctuation marks divided by total no of characters)</td>
</tr>
<tr>
<td>Readability</td>
<td>average word length (# of characters)</td>
</tr>
<tr>
<td>Readability</td>
<td>average sentence length (# of words)</td>
</tr>
<tr>
<td>Author prominence</td>
<td>proportion of posts in thread by author of current post</td>
</tr>
<tr>
<td>Representativeness</td>
<td>cosine sim between post and thread (word2vec representations)</td>
</tr>
<tr>
<td>Representativeness</td>
<td>cosine sim between post and title (word2vec representations)</td>
</tr>
</tbody>
</table>

In the context of the automatic summarizer, we first do a separate (intrinsic) evaluation of the models (Section 5.3).

All post features are listed in Table 3. We standardize feature values by converting them to their z-value using the mean and standard deviation per feature (a standard procedure for regression analysis). For evaluation, we use five-fold cross validation by splitting the set of threads in five partitions, and in five runs train on four partitions and test on the fifth. We then report means and standard deviations for the evaluation scores over the raters (micro averages, weighted by the number of threads summarized by each rater). Note that in this feature set the query is not taken into account, thus the posts get the same prediction from the linear regression model (LRM), independent of the added query.

The regression analysis also allows us to see the contribution of each of the post features to the post ranking. We will evaluate that in Section 5.3.
4.2 Method 2: Maximum Marginal Relevance (MMR)

We implemented MMR as method for query-based summarization because it is the most-referenced method for query-based summarization in the literature and it has been shown to be a successful method for query-based conversation summarization (Murray et al., 2005), which is a similar task to ours. MMR is an unsupervised method, which might be at a disadvantage compared to Method 1. Using a supervised approach has advantages over using an unsupervised approach, against the costs of creating training data. On the other hand, MMR has the claimed advantage of using the query, as opposed to our feature-based approach (LRM). Given the success of MMR with unsupervised query-based summarization in prior work we expect the method to be fit for our problem.

The central components of MMR are the similarity functions $\text{Sim}(T_i, Q)$ and $\text{Sim}(T_i, T_j)$ (see Equation 1). We compare two different similarity metrics for these functions:

1. the cosine similarity between the word vectors (tf-idf term weights);
2. the cosine similarity between the average vectors over the word embeddings from the best performing word2vec model from Section 4.1

We tokenize the queries by splitting on non-alphanumerical characters and lower-cased them. Because the queries in our data are short (1.5 words on average) and we do not want to remove any potentially relevant information we do not remove stopwords or apply lemmatization.\(^9\)

We select posts from the thread by iteratively going over the unselected posts in order of posting (from top to bottom in the thread), finding the post with the Maximum Marginal Relevance (MMR), and add it to the set of selected posts, with its MMR score. If two or more posts have equal MMR, we order them by position in the thread. We also investigate the effect of varying MMR’s parameter $\lambda$. Note

\(^9\) Our implementation can be found at https://github.com/DISCOSUMO/query-based_summarization

\(^{10}\) In fact, almost all queries are singular noun lemmas, e.g. ‘relatie’ ‘relationship’, ‘badkamer’ ‘bathroom’, ‘marrakech’, ‘burn out’, ‘nieuwe macbook’ ‘new macbook’

that MMR is a language-independent method, just like method 1; it can be applied to discussion forum threads without knowing the language of the forum.

4.3 Combining query information with post features

We will evaluate PF (post ranking based on query-independent post features) and MMR (post selection based on MMR) separately and use the best performing of the two as the basis for our experimental setting in which we combine query information with post features. In the next section, we will first analyze the collected reference summaries.

5 Analysis and comparison of the reference summaries

All raters belonged to the target group of the Viva forum: 19 women (average age = 22 years, SD = 3.3). They together created 600 (120 × 5) reference summaries. The median number of selected posts per thread combination is 5 (mean 6.0), with a large standard deviation (4.8). The highest number of posts selected for a thread is 28, the lowest is 0. In the following subsections, we will investigate and discuss: the relation between query and post selection (5.1), a comparison and analysis of the inter-rater agreement with and without query (5.2), and the features of the selected posts (5.3). The statistics of the reference set are summarized in Table 4.

Table 4. Statistics of the reference set

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of raters</td>
<td>19</td>
</tr>
<tr>
<td>Mean (stdev) age</td>
<td>22 (3.3)</td>
</tr>
<tr>
<td>Number of reference summaries</td>
<td>600</td>
</tr>
<tr>
<td>Median (stdev) number of selected posts per thread</td>
<td>5 (4.8)</td>
</tr>
<tr>
<td>Mean (stdev) relevance score assigned to a thread given the query</td>
<td>2.8 (0.9)</td>
</tr>
</tbody>
</table>
5.1 Relation between query and post selection

The average relevance score assigned to a thread given the query was 2.8 on a 5-point scale, with an average standard deviation per thread of 0.9. Half of the query–thread combinations (59 out of 120) received a mean relevance score of at least 3 from the raters. This indicates that for half of the queries, the clicked thread was considered irrelevant, which confirms previous research that a click on a result is not necessarily an indicator for relevance (Dupret & Liao, 2010), and that it is sometimes difficult for raters to judge the relevance of results for someone else’s query.

There is a significant positive relationship between the relevance score assigned by a user and the number of posts selected by that user (Kendall $\tau = 0.234, p < 0.0001$). Thus, the more relevant the thread is for the query, the more posts are selected for the summary.

Posts that contain the literal query get selected significantly more often than posts that do not contain the query: $\chi^2(1, N = 25056) = 123.19, p < 0.0001$, but the association is weak: Cramer’s $V = 0.07$, where 0 means that there is no association between variables.$^{11}$

5.2 Inter-rater agreement (RQ1)

We investigated the agreement between human raters on which posts should be included in the summary given a query. In previous work, we reported inter-rater agreement in terms of Cohens $\kappa$, Fleiss’ $\kappa$, the Jaccard similarity coefficient (Verberne et al., 2017).$^{12}$ In order to calculate Cohen’s $\kappa$ in a pairwise fashion, we computed the agreement between each pair of raters for each thread. If both raters

---

$^{11}$ N is the total number of posts in the data times the number of raters times the number of unique queries for the thread.

$^{12}$ Fleiss’ $\kappa$ measures inter-rater agreement for multi-rater data, and Cohen’s $\kappa$ for separate pairs of raters. The former seems more appropriate as we have 5 raters per thread, but the latter would be conceptually correct because we do not have the same 5 raters for each thread. Here we report both Fleiss’ $\kappa$ and Cohen’s $\kappa$. 
selected 0 posts, we set $\kappa=1$. We also report the Jaccard similarity coefficient, which is calculated as the size of the intersection divided by the size of the union of the two sets of selected posts. If both raters selected 0 posts, we set Jaccard=1. The three previously used agreement metrics all measure agreement on the granularity level of complete posts. One problem of that approach is that it does not take the content of a post into account. Two raters might select two different posts, but the two posts might be very similar. Both $\kappa$ and Jaccard result in 0 for the selection of two distinct but highly similar posts. We therefore added ROUGE-2 as an additional agreement metric, computing the overlap between two human summaries on the granularity level of word bigrams instead of complete posts (see Section 6.1 for a more elaborate explanation of ROUGE-2).

The results for the four metrics are in Figure 3. Four different samples are compared (bars from left to right): (1) all threads summarized without query in previous work (Verberne et al., 2017); (2) the subset from those threads that also are present in the sample for query-based summarization; (3) the same threads summarized with query; and (4) the subset from those threads for which the human summarizers gave a mean relevance assessment of at least 3.

One problem with the $\kappa$ measures for inter-rater agreement is that they are ill-suited for data with a large class imbalance (Xu & Lorber, 2014; Powers, 2012). In post selection, we observe a large class imbalance because the majority of posts is not selected. This effect is the most visible in the results for Fleiss' $\kappa$. In computing the chance agreement, Fleiss' $\kappa$ uses the total number of post selections (by all raters) and the total number of non-selections (by all raters). These numbers are 9239:28971 for the query-independent post selection data from the previous study. With a chance agreement of 0.633 and a measured agreement of 0.713, Fleiss' $\kappa$ was 0.219. In the query-dependent case, the class imbalance becomes worse, i.e. 3673:45654, giving a chance agreement of 0.862. With a measured agreement of 0.873, Fleiss' $\kappa$ is only 0.075. If we only consider the threads that were judged as relevant to the given query (the rightmost bar) then the agreement is slightly better, but Cohen's $\kappa = 0.191$ at the most.
Fig. 3. Inter-rater agreement in terms of Fleiss $\kappa$, Cohen’s $\kappa$, Jaccard similarity coefficient, and ROUGE-2, for four thread sets: the data published in Verberne et al. 2017 (post selection without query), a subset from that data that contains only the threads that are also in the query sample; the current data with queries; and the current data for the threads with a mean relevance score of at least 3 for the given query according to the human summarizers.

The class imbalance effect causes the $\kappa$ statistics to be less suited. Therefore, we will further focus on the results in terms of the Jaccard coefficient and ROUGE-2. Jaccard is a measure for the size of the overlap between the summaries of the two raters, on the post level; a coefficient of 0.25 means that on average 25% of the posts selected by the two raters overlap. ROUGE-2 is a measure for the size of the overlap on the word bigram level; a score of 0.25 means that on average 25% of the word bigrams overlap between two raters. The results for both metrics indicate that adding a query to the summarization task does not lead to a higher agreement between the human summarizers. The direct comparison between the second and third bar, which include the same set of threads without and with query, shows
that the agreement for the with-query setting is not higher – in fact, it is lower – than the agreement for the without-query setting.

We investigate this in more detail by analyzing the dispersion of Jaccard values between pairs of summaries (recall that Jaccard is a pairwise overlap measure, between two individual summaries). We compare three sets of Jaccard values:

- the Jaccard values between pairs of summaries created for the same thread by different raters, without a query given
- the Jaccard values between pairs of summaries for the same thread by different raters, with the same query given
- the Jaccard values between pairs of summaries for the same thread by different raters, one with query and one without query

The dispersions of these three sets of Jaccard values are in Figure 4. The mean Jaccard for between query and noquery is almost equal to the mean Jaccard for with query. This suggests that the differences between two individual summaries cannot be explained by the presence of the query; the individual differences between human raters are larger. On the other hand, we see that the mean Jaccard for without query is higher than for with query. A paired t-test on mean Jaccard values per thread indicates that this difference is significant with $p = 0.01$.13

It is surprising that the agreement between human summarizers decreases when a query is added to the task. It could be that the query has a different effect for different raters: some raters might be influenced more by the query than others. Another cause could be that the query confuses the raters because many queries are short and underspecified (e.g. ‘pregnant’, ‘toddler’): some raters might have summarized the thread independently of the query because they judged the query as irrelevant, whereas others might have tried to summarize the aspects covered by the query.

It was argued in the context of DUC that having a question to focus the summary

13 We can do a paired test on these values because the same set of threads is included in the comparison.
can help to improve agreement in content between the model summaries (Dang, 2005), but unfortunately, none of the previous work on query-based summarization reports inter-rater agreement scores. Our findings contradict the assumption that the addition of a query will improve inter-rater agreement. This is most likely related to the nature of our query data compared to the DUC topics: we have short queries (mostly 1 or 2 words) from a query log, while DUC involves well-formed explicit questions, consisting of a title and a narrative, formulated by the task organizers.

Fig. 4. Dispersion of the Jaccard similarity index between individual summaries for the same thread, either both without query, both with query, or one with and one without query.

5.3 Analysis of post features for human-created summaries

Comparison of representativeness models We compared word vectors and word embeddings from several word2vec models as representations of posts and threads in the representativeness feature, with 2 different similarity metrics. The results of this comparison are in Table 5. The models all significantly correlate with the human post ranking, and they all significantly correlate with each other. The cor-
Table 5. Evaluation of models for representativeness: 5 different representations for posts and threads (words or word embeddings trained on two different corpora) and 2 different similarity metrics (cosine similarity or 1 minus the word mover’s distance). All correlations are significant with \( p < 0.0001 \).

<table>
<thead>
<tr>
<th>Post/thread representation</th>
<th>Cosine similarity</th>
<th>1-wmd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word vectors</td>
<td>0.317</td>
<td>0.255</td>
</tr>
<tr>
<td>Word2vec_Wiki320</td>
<td>0.321</td>
<td>0.291</td>
</tr>
<tr>
<td>Word2vec_Viva320</td>
<td><strong>0.345</strong></td>
<td>0.296</td>
</tr>
<tr>
<td>Word2vec_Viva200</td>
<td><strong>0.345</strong></td>
<td>0.292</td>
</tr>
<tr>
<td>Word2vec_Viva100</td>
<td>0.335</td>
<td>0.293</td>
</tr>
</tbody>
</table>

The correlation with the human post ranking is stronger for the word embeddings models than for the standard word vectors, and cosine similarity gives stronger correlations than word mover’s distance. The results indicate that it is possible to get better representativeness models with word embeddings compared to standard word vectors. Table 5 also shows that the Viva models outperform the larger pre-trained Wikipedia model, which indicates that domain is more important than corpus size in training meaningful word2vec models.

The best performing models are the 200-dimensional and the 320-dimensional Viva forum model. We will use \texttt{Word2vec$_{\text{Viva320}}$} in our summarization experiments. The differences between the models as listed in Table 5 do not necessarily translate to significantly different summaries. We will investigate the effect of using the best word2vec model compared to standard word vectors in Section 6.2.

**Analysis of the linear regression model** Table 6 shows the ranking of the post features that significantly predict the number of votes for a post, according to the linear regression analysis. As representativeness feature, we used the cosine similarity between the word2vec representations from \texttt{Word2vec$_{\text{Viva320}}$}. The feature ranking is mostly similar to the feature ranking previously found in Verberne et
Query-based summarization of discussion threads

Table 6. Post features that are significant predictors for the number of selected votes, sorted by the absolute value of the regression coefficient $\beta$; the independent variable with the largest effect (either positive or negative) is on top of the list. *** indicates a $p < 0.001$.

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>$\beta$ coef</th>
<th>signif.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Popularity</td>
<td># of responses (quotes) to the post</td>
<td>-1.07</td>
<td>***</td>
</tr>
<tr>
<td>Position</td>
<td>absolute position in the thread</td>
<td>-0.32</td>
<td>***</td>
</tr>
<tr>
<td>Readability</td>
<td>unique word count</td>
<td>0.25</td>
<td>***</td>
</tr>
<tr>
<td>Representativeness</td>
<td>w2v cosine sim between post and thread</td>
<td>0.15</td>
<td>***</td>
</tr>
<tr>
<td>Readability</td>
<td>type-token ratio</td>
<td>-0.13</td>
<td>***</td>
</tr>
<tr>
<td>Author prominence</td>
<td>proportion of posts in thread by author of post</td>
<td>-0.12</td>
<td>***</td>
</tr>
<tr>
<td>Readability</td>
<td>average word length (# of chars)</td>
<td>0.09</td>
<td>***</td>
</tr>
</tbody>
</table>

al., (2017) for query-independent summarization. This suggests that the characteristics of the selected posts are independent of the presentation of a query; posts in the beginning of the thread, posts that are relatively long, and posts that are representative of the thread tend to be selected more often.

One coefficient that stands out negatively is the number of responses (quotes) to the post. The number of responses for a post is counted as the number of later posts that literally quote part of the post (using the quote function of the forum). In the data for query-independent summarization this coefficient was also negative, but much smaller (0.08). We had a more detailed look at our sample in order to find out what types of posts are quoted in later posts. Only 176 out of the 1571 posts (11%) in our data are quoted at least once, most of them exactly once. The majority of the quoted posts is relatively short, and quoted as a whole. 73 contain a question mark (e.g. “Maybe it is much more busy in a group practice?”); 40 contain an emoticon (e.g. “Ahhhhhh, please let me get my crystal ball?! ;))”, and 48 quote a previous post themselves. Questions and jokes are typically not the type of posts that are relevant for a (query-based) summary because the implicit goal of the summary is to answer the query. ‘Responses being responded to’ constitute a form of conversation (discourse) that is not effective to select for the summary.
because the messages are meaningless without context. Thus, the type of messages that get quoted in the forum are typically messages that are not relevant for the (query-based) summary, and therefore the coefficient for the number of responses is negative.

6 Results of the automatic summarization methods

We will first explain our evaluation method (Section 6.1), then evaluate and compare the two methods (Section 6.2) and finally discuss and evaluate combinations of the methods (Section 6.3).

6.1 Evaluation method

Since our methods generate a score per post, we can consider our extractive summarization approach as a ranking problem (see also Section 2.1). Transforming the ranking of posts into a summary requires the setting of a cutoff point, either after a fixed number of posts, or threshold-based using the output score of the algorithm. The disadvantage of both approaches is that they require the tuning of the threshold parameter. Therefore, we now choose to first evaluate the ranking using a Precision–Recall curve, increasing the number of selected posts from 1 to 20. We compare our methods to an oracle ranking: the ranking of the posts on the basis of the number of votes they received from the human summarizers when creating the reference summaries. We can consider this oracle ranking the upper limit for our automated summarizers.

Note that use of precision and recall for the evaluation of extractive summarization is a rather strict evaluation method: if the model selects a post that was not included in the reference summary, then this post is considered a false positive, even if the content of other posts in the reference summary is largely overlapping with this model-selected post. This effect is more severe in the case of texts with much redundancy. The common evaluation metric ROUGE-N adopts a more flexible approach to measuring the overlap between the automatic summary and the human reference summary by counting the textual overlap (on the n-gram or word
level) between both summaries (Lin, 2004; Murray et al., 2005; Tsai et al., 2016). ROUGE-N is recall-oriented: the number of overlapping n-grams is divided by the number of n-grams in the reference summary. This means that longer summaries tend to have a higher ROUGE-N score. For that reason, ROUGE is commonly used to evaluate summaries of the same length (Dang, 2005). Since the methods we compare do not necessarily generate summaries of the same length (because posts are of diverse lengths), we not only report ROUGE-N recall but also ROUGE-N precision. We set $N = 2$.

We report ROUGE-2 for summaries with a fixed number of posts selected by each method. As cutoff for this fixed number of posts we use the number of posts that gives the maximum F1-score in the oracle ranking – this point ($k$) denotes the optimal balance between Precision and Recall according to the human summarizers.\footnote{If both the model and the human rater selected 0 posts, we set Precision = Recall = ROUGE-N = 1.}

For this cutoff point, we also add four baselines:

- a position baseline (selecting the $k$ first posts of a thread)
- a length baseline (selecting the $k$ longest posts from a thread)
- a centroid-based baseline (selecting the $k$ posts that are the most representative for the thread in terms of cosine similarity)
- Lexrank (Erkan & Radev, 2004)\footnote{We used the Lexrank package available from https://pypi.org/project/lexrank/}

The centroid-based baseline as well as Lexrank are commonly used strong baselines in extractive summarization, among others by Krishnamani et al. (2013) for discussion forum summarization.

### 6.2 Comparison of the methods (RQ2 and RQ4)

#### 6.2.1 Precision–Recall curves

The Precision–Recall curves for the methods are in Figure 5. The oracle ranking shows a monotonically decreasing line: precision decreases from 75.2\% to 28.6\%
Fig. 5. Precision–Recall curve for all methods on query-based post selection. For the MMR methods, $\lambda = 1.0$. The word2vec model is \texttt{Word2vec Viva320}. ‘Oracle ranking’ is the ranking of the posts based on the number of votes by the human summarizers. Iso-F1=0.5 denotes the curve for all combinations of Precision and Recall for which $F1=0.5$.

while recall increases from 14.9% to 97.1%. The maximum $F1$ that is reached for the oracle ranking is 57.4%, at a cutoff of 9 posts. Therefore, we set $k = 9$ for evaluation in terms of ROUGE-2.

The curves for PF (query-independent post features) are monotonically decreasing, almost parallel to the oracle ranking, but a large margin below it. The two dashed curves with and without word2vec almost fall together. This indicates that although the \texttt{Word2vec Viva320} model performed better in the model evaluation than the standard word vectors (Table 5), the effect of this difference on the quality of the summarizations is negligible (RQ4). The curves for Lexrank and MMR are far below those for PF, and only slightly decreasing. With word2vec, MMR does not structurally improve: For a small number of posts (fewer than 9), MMR without word2vec works slightly better, while for a larger number of posts (9 and
Fig. 6. The effect of the parameter $\lambda$ in MMR on the quality of the query-based summaries, in terms of F1 and ROUGE-2, at an cutoff of 9 posts. $\lambda = 0.0$ means only the diversity component is used; $\lambda = 1.0$ means that only the query similarity component is used.

higher), MMR word2vec yields slightly better results. Although MMR and Lexrank are reported as good methods for extractive summarization in the literature, they are not successful for our data.

6.2.2 Optimizing MMR

We investigated the importance of the diversity component relative to the query similarity component in MMR by varying $\lambda$ (see equation 1) between 0.0 (only the diversity component) and 1.0 (only the query similarity component) in steps of 0.1 for all five partitions in a cross-validation setting. The results (over all partitions) in terms of ROUGE-2 and F1 for a summary length of 9 posts are in Figure 6.

The curves indicate that the best ROUGE-score (38.2%) and the best F1 (25.6%) are obtained at $\lambda = 1.0$. This means that query similarity alone, without the diversity component, gives the best results. This is surprising, for two reasons: (a) Since the raters were explicitly instructed to create a summary in which redundancy was avoided, it would have made sense if the diversity component would select good posts, and (b) since we found that the influence of the query on the summaries is relatively small (see Section 5), we would not expect the query similarity compo-
Table 7. Precision, Recall, F1, ROUGE-2 Recall and ROUGE-2 Precision scores for all methods, with a summary length of 9 posts. All scores are means over the 5 reference summaries and the thread-query combinations. Boldface indicates the highest scoring method per evaluation metric. Underlined scores are not significantly different from the highest score (only measured for F1 and ROUGE-N R).

<table>
<thead>
<tr>
<th>Method</th>
<th>R</th>
<th>P</th>
<th>F1</th>
<th>ROUGE-2 R</th>
<th>ROUGE-2 P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position baseline</td>
<td>43.7%</td>
<td>27.1%</td>
<td>33.4%</td>
<td>39.0%</td>
<td>39.7%</td>
</tr>
<tr>
<td>Length baseline</td>
<td>39.4%</td>
<td>23.4%</td>
<td>29.4%</td>
<td>58.4%</td>
<td>30.7%</td>
</tr>
<tr>
<td>Centroid baseline</td>
<td>49.4%</td>
<td>28.7%</td>
<td>36.3%</td>
<td>64.1%</td>
<td>33.3%</td>
</tr>
<tr>
<td>Lexrank</td>
<td>33.1%</td>
<td>21.8%</td>
<td>26.2%</td>
<td>51.2%</td>
<td>31.2%</td>
</tr>
<tr>
<td>Post feats (PF)</td>
<td>54.8%</td>
<td>34.0%</td>
<td>41.9%</td>
<td>66.0%</td>
<td>36.5%</td>
</tr>
<tr>
<td>Post feats w2v</td>
<td>54.6%</td>
<td>33.9%</td>
<td>41.8%</td>
<td>65.6%</td>
<td>36.3%</td>
</tr>
<tr>
<td>MMR, $\lambda = 1.0$</td>
<td>35.4%</td>
<td>22.8%</td>
<td>27.8%</td>
<td>39.7%</td>
<td>33.7%</td>
</tr>
<tr>
<td>MMR w2v, $\lambda = 1.0$</td>
<td>36.5%</td>
<td>23.2%</td>
<td>28.4%</td>
<td>38.2%</td>
<td>33.5%</td>
</tr>
<tr>
<td>Combined 1 (1-level LRM)</td>
<td>55.5%</td>
<td>34.5%</td>
<td>42.6%</td>
<td>66.1%</td>
<td>36.9%</td>
</tr>
<tr>
<td>Combined 2 (2-level LRM)</td>
<td>57.0%</td>
<td>35.3%</td>
<td>43.6%</td>
<td>66.1%</td>
<td>37.5%</td>
</tr>
<tr>
<td>Oracle</td>
<td>75.7%</td>
<td>46.4%</td>
<td>57.5%</td>
<td>72.6%</td>
<td>50.4%</td>
</tr>
</tbody>
</table>

The results in terms of Precision, Recall, F1 and ROUGE-2 for a summary length of 9 posts are in the upper half of Table 7. The position baseline gives the best results in terms of ROUGE-2 precision, while the length baseline outperforms the position baseline in terms of ROUGE-2 recall – an effect of the longer summaries having an advantage with recall-oriented metrics. The centroid-based baseline gives better results than the length and position baseline on almost all metrics, indicating that...
of the three important features length, position and representativeness, the latter is the most informative.

Lexrank and the MMR methods perform worse than any of the three baselines. According to a paired t-test on individual ROUGE-2 scores for thread–query–rater tuples, the difference between MMR and MMR word2vec is not significant ($p = 0.38$).

The F1 for the query-independent post features on the query-based data is 41.9%, which is close to the F1 in Verberne et al., (2017) for the query-independent summarization task (45.2%). The recall is 54.8% (stdev = 10.3 %) and the precision is 34.0% (stdev = 19.1%) This confirms the finding in Section 5.3 that the posts that are selected with and without query have similar characteristics. We also see that PF performs reasonably well in terms of ROUGE-2, compared to the oracle ranking: 66.0% Recall and 36.5% Precision. This indicates that on average, two thirds of the word bigrams from a reference summary are also contained in the automatic summary, and more than one third of the word bigrams from the automatic summary are also part of the reference summary. The difference between post features and post features with word2vec is not significant ($p = 0.82$ according to a paired t-test on individual ROUGE-2 scores for thread–query–rater tuples).

In summary, the method using query-independent post features clearly outperforms Lexrank and MMR. Both Lexrank and MMR are outperformed by the informed baselines based on position, length and centrality of the posts. There is definitely room for improvement, which we address in the next subsection.

### 6.3 Combining post features with query information (RQ3)

We saw in the previous section that MMR with query similarity as only component ($\lambda = 1.0$) gives better results than the position and length baselines. Thus, we reason that adding query relevance to the post selection model might lead to improved summaries for a subset of the threads. We implemented three different query–post similarity metrics for incorporating query relevance in the post ranking:
a) the cosine similarity between word vectors (tf-idf term weights) (‘query similarity’);

b) the cosine similarity between the average vectors over the word embeddings (‘query similarity word2vec’);

c) the cosine similarity between the query vector expanded with the 5 most similar terms from the word2vec model for each query word and the word-based post vectors (‘expanded query similarity’).\(^{16}\)

We again used the best performing word2vec model from Section 5.3 to generate the word embeddings (\textit{Word2Vec, Viva320}).

Now we have the linear regression model with query-independent post features as baseline method, and three additional features that operationalize query–post relevance: ‘query similarity’, ‘query similarity word2vec’ and ‘expanded query similarity’. We investigate two methods for incorporating these query relevance metrics in the post ranking:

1. Expanding the linear regression model from Section 4.1 with the three query relevance metrics (‘Combined 1’: 1-level LRM);

2. Learning four independent ranking functions: the ranking based on query-independent post features, and three rankings for the three query relevance metrics. Then training a second linear regression model for optimally combining the four scores in one ranking (‘Combined 2’: 2-level LRM).

\textit{Generic results.} The results for the two combination methods are in the bottom half of Table 7. The table shows that the two combination methods perform almost equally to Centroid, as well as PF and PF w2v in terms of ROUGE. The small differences between Combined 1, Combined 2, Centroid, PF and PF w2v are not significant according to a paired t-test on individual ROUGE-2 scores for thread–

\(^{16}\) The number of expansion terms is based on (ALMasri \textit{et al.}, 2016), who found that when using word embeddings for query expansion, between 5 and 10 expansion terms is optimal. Since our training corpus is smaller than theirs (21.9 Mw compared to 400 Mw, we opt voor 5 expansion terms.)
Fig. 7. Variables are significant predictors ($p < 0.05$) for the post score (number of votes) in the second-level LRM for the query-dependent data, with their beta coefficients.

query–rater tuples ($p > 0.7$ for all pairs of methods). Combined 2 seems to be slightly better than PF in terms of F1, but this difference is also not significant ($p = 0.218$ according to a paired t-test on individual F1 scores for thread–query–rater tuples between PF and Combined 2).

Figure 7 shows the trained linear regression model for Combined 2. Three of the four variables are significant predictors of the post score: the query-independent post features, query similarity and query similarity word2vec, with the query-independent post features having the largest effect. The contribution of expanded query similarity is not significant ($p = 0.109$ according to the LRM).

Effects for individual threads. We investigated the effect of adding query relevance to individual thread–query pairs. Figure 8 shows the difference between PF and Combined 2 for individual thread–query combinations. Each thread–query combination is a tick on the horizontal axis, and they have been sorted by the size of the difference. The bars represent the relative improvement, the grey area represents the F1 for PF and the black line the F1 for Combined 2.

For 29 threads (in the middle) the difference between the two methods is zero. For 21 thread–query combinations (the leftmost bars of the graph), adding the query similarity information gives an improvement larger than 0.2 in F1. A detailed look at these cases shows that they generally have no or little overlap between the query and the thread title. This leads us to hypothesizing that the query might play a larger
role for creating the summary in cases where the query asks for a one particular aspect of the thread. Examples are the query ‘adoption’ for the thread ‘Afraid for the future after second miscarriage (long IVF trajectory)’, and the query ‘fez’ for the thread ‘Istanbul or Marrakech?’ However, the individual differences between the raters are too large to draw a general conclusion on the subset of thread–query pairs for which combining the methods is profitable.

In the next section, we discuss our results by addressing the four research questions from the introduction.

7 Discussion

RQ1 How do human summaries of discussion forum threads change when a short user query is provided as focus of the summary?

In Section 5 we observed that the presence of a query only has a small influence on the created summaries: The variation between individual raters is larger than
the variation caused by the presence of the query. One important finding was that adding a query to the summarization task did not lead to a higher agreement between the human summarizers. It was argued in the context of DUC that having a question to focus the summary can help to improve agreement in content between the model summaries (Dang, 2005). This is most likely an effect of the nature of the DUC topics: well-formed explicit questions, consisting of a title and a narrative, formulated by the task organizers. These are much more informative than real user queries in search engines. As mentioned in Section 1, our motivation for moving to query-based thread summarization is that in our line of research, the purpose of thread summarization is to facilitate information finding by forum readers, which is often initiated by a query. Instead of using the elaborate queries developed for the DUC tasks (consisting of a title and a description), we address the problem of thread summarization given real user queries. We use queries entered in the search engine of a large, open discussion forum. This type of real user queries have similar characteristics to queries entered in general web search engines and social media search engines such as Twitter search. These queries are short and underspecified, which makes it difficult to know what the intent of the searcher was (Verberne et al., 2013). As a result, the query-based summaries depend on the rater’s interpretation of the query intent, and this interpretation may differ among individual raters.

Another challenge of using queries from query logs is that a click on a result is not a direct indicator for the relevance of the result (Dupret & Liao, 2010). We found that for half of the queries the clicked thread was considered to be relevant according to our raters. This implies that it is more difficult to create query-based reference summaries for real user queries because the relevance of the thread for the query is not always apparent to the rater. Thread summarization can still be beneficial for the user who is searching for information, because users can judge the relevance of a summarized document faster than of a full document (Tombros & Sanderson, 1998; Nenkova & McKeown, 2011), provided that the part of the document that is relevant for the user is included in the summary. Our suggestion
for future data collection projects is to do the relevance assessment first, and then
only create reference summaries for the relevant query–thread combinations.

RQ2 How does Maximum Marginal Relevance (MMR) perform on forum threads
with real user queries?

Although MMR is reported as a good method for extractive summarization in
related work (Murray et al., 2005), it is not successful on our data with short
queries. By varying the weight of the diversity component relative to the query
similarity component in MMR we found that the best summaries are created by
query similarity only, without the diversity component. Still, both the centroid-
based baseline and a summarizer based on query-independent post features perform
better than MMR on most metrics. We have to conclude that MMR does not work
for this type of data, although the task is query-based summarization. We consider
this an interesting, counter-intuitive negative result.

RQ3 Can we improve over MMR by including forum-specific summarization fea-
tures in the model?

Yes. We found that none of the \( \lambda \) settings causes MMR to outperform our method of
query-independent post features. Using post features alone, or even using the rep-
resentativeness feature alone (centroid-based baseline), gives an average ROUGE-2
Recall of 66.0\% and ROUGE-2 Precision of 36.5\%. This means that on average,
two thirds of the word bigrams from a reference summary are also contained in
the automatic summary, and more than one third of the word bigrams from the
automatic summary are also part of the reference summary. This indicates that
the most important characteristics that make discussion comments relevant for the
summary, are query-independent characteristics, in particular representativeness
for the thread. This is likely to be inherent to the nature of the data: discussion
threads on the web tend to contain noise in the form of off-topic comments, in-
formal chatting, laughter (emoticons) and discourse such as good luck wishes and
thank you messages (Weimer et al., 2007). Filtering out these (often short) com-
ments apparently leads to a decent summary, independent of the query presented.
This makes a method based on post features a robust method for discussion thread summarization.

RQ4 Does the use of word embeddings instead of standard word-based vectors as text representations lead to better summaries?

It is possible to get better representativeness features with word embeddings compared to word vectors. The success of the model depends on the similarity metric, the training corpus and the dimensionality. In particular, we found that:

(a) cosine similarity leads to better correlations with human post ranking than word mover’s distance;
(b) the models trained on the Viva corpus (with the threads in our sample excluded) perform consistently better than a pre-trained model on a larger Dutch Wikipedia corpus with the same dimensionality;
(c) a higher dimensionality leads to a better model, but the differences are small.

Our findings indicate that a match in domain and genre is more important than corpus size and dimensionality in the training of meaningful word2vec models. One caveat is that although there are clear differences between the models in a direct comparison, these differences diminish when the models are exploited in the automatic summarizer: there is no significant difference between the summarization method based on post features, and the same summarization method with word2vec similarity implemented for the representativeness features. The same holds for the difference between MMR and MMR with word2vec.

8 Conclusion

In this paper we addressed query-based discussion thread summarization. We created a data set with discussion threads and corresponding queries from a large query log. For each thread-query combination a reference summary was made by five different raters. An analysis of the created summaries showed that the influence of the query on the created summary is significant but small and the inter-rater agreement for the task is lower than for comparable query-independent summaries.
Although previous work suggests that adding a query to the summarization task leads to higher agreement, this claim was never evaluated for real user queries. In our study, with real web queries, adding a query to the summarization task did not lead to higher inter-rater agreement.

We argued that query-based summarization with web queries is challenging for two reasons: First, the queries are short and underspecified, which causes the query-based summaries to depend on the rater’s interpretation of the query intent. Second, a click on a result is not a direct indicator for the relevance of the result. However, discussion thread summarization can still be beneficial for the information-seeking user, because users can faster judge the relevance of a summarized document than of a full document.

We compared two language-independent methods for automatic summarization of the threads: a query-independent method based on post features and the query-based method Maximum Marginal Relevance (MMR). We found that the query-independent post features, and even the centroid-based baseline, outperform MMR on almost all evaluation metrics. We argue that this is inherent to the noisy nature of discussion threads on the web: filtering out off-topic comments and informal chatting leads to sufficiently good summaries, independent of the query presented.

We compared a number of word embeddings representations as alternative for word vectors in extractive summarization. In a direct comparison of the models, we found clear differences in the quality of the models. A match in domain and genre appears to be more important than corpus size and dimensionality in the training of meaningful word2vec models. However, the differences between the models were not reflected by differences in quality of the summaries that are created with help of these models. For future research we recommend to always evaluate both standard word vectors and word embeddings – trained on a corpus from the topic domain – before choosing to use word embeddings for operationalization of the representativeness (salience) criterion in automatic summarization.

We judge the overall performance of our query-based summarizer to be sufficient for online use, especially given the low inter-rater agreement for the task. There-
fore, the next step would be to implement extractive summarization for a discussion forum and evaluate the user experience through A/B testing. It would also be interesting to ask the forum visitors to indicate why they clicked on a specific post given the query they entered – this would give information about the relevance criteria for post selection. From a methodological point of view, it would be interesting to investigate the options for combining MMR with other methods such as the centroid baseline to see in which cases MMR can bring additional value.

Acknowledgements

This work was carried out in the context of the project Discussion Thread Summarization for Mobile Devices (DISCOSUMO), which is financed by the Netherlands Organisation for Scientific Research (NWO).

References


Chowdhury, Shammur Absar, Calvo, Marcos, Ghosh, Arindam, Stepanov, Evgeny A, Bayer, Ali Orkan, Riccardi, Giuseppe, Garcia, Fernando, & Sanchis, Emilio. 2015. Selec-


Li, Yanran, & Li, Sujian. 2014. Query-focused multi-document summarization: Combining


Oya, Tatsuro, Mehdad, Yashar, Carenini, Giuseppe, & Ng, Raymond. 2014. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. *Pages 45–53 of: Proceedings of the 8th International Natural Language Generation Conference (INLG)*.


van Oortmerssen, Gerard, Raaijmakers, Stephan, Sappelli, Maya, Boertjes, Erik, Verberne, Suzan, Walasek, Nicole, & Kraaij, Wessel. 2017. Analyzing cancer forum discussions


Zhang, Renxian, Li, Wenjie, Gao, Dehong, & Ouyang, You. 2013. Automatic twitter topic

Zhang, Yong, Er, Meng Joo, Zhao, Rui, & Pratama, Mahardhika. 2016. Multiview Convolutional Neural Networks for Multidocument Extractive Summarization. *IEEE Transactions on Cybernetics*.
