D4.1.2 Multi-Lingual Summarisation of Stream Media Software - v2

Dominic Rout (USFD), Kalina Bontcheva (USFD), Ian Roberts (USFD)

Abstract.
FP7-ICT Strategic Targeted Research Project (STREP) ICT-2011-287863 TrendMiner
Deliverable D4.1.2 (WP4.1)

This deliverable describes the prototype interactive summarisation algorithms for stream media, developed in task 4.1. The first step is to select a time period, followed by topics and entities of interest, and then produce a summarised subset of relevant tweets, that meet those constraints. Methods developed in WP2 and WP3 are used in order to derive entities and topics. Work presented here builds upon the summarisation methods, described in the first version of this deliverable [GB12], alongside a new task of ranking content interactively.

Keyword list: summarisation, centroid, TextRank, topic models

Project	TrendMiner No. 287863
Delivery Date	November 4, 2013
Contractual Date	October 31, 2013
Nature	Prototype
Reviewed By	Francesco Bellini (Eurokleis)
Web links	`http://demos.gate.ac.uk/trendminer/summarization/`
Dissemination	PU

TrendMiner Consortium

This document is part of the TrendMiner research project (No. 287863), partially funded by the FP7-ICT Programme.

DFKI GmbH
Language Technology Lab
Stuhlsatzenhausweg 3
D-66123 Saarbrcken
Germany
Contact person: Thierry Declerck
E-mail: declerck@dfki.de

University of Southampton
Southampton SO17 1BJ
UK
Contact person: Mahensan Niranjan
E-mail: mn@ecs.soton.ac.uk

Internet Memory Research
45 ter rue de la Rvolution
F-93100 Montreuil
France
Contact person: France Lafarges
E-mail: contact@internetmemory.org

Eurokleis S.R.L.
Via Giorgio Baglivi, 3
Roma RM
00161 Italy
Contact person: Francesco Bellini
E-mail: info@eurokleis.com

University of Sheffield
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Contact person: Kalina Bontcheva
E-mail: K.Bontcheva@dcs.shef.ac.uk

Ontotext AD
Polygraphia Office Center fl.4,
47A Tsarigradsko Shosse,
Sofia 1504, Bulgaria
Contact person: Atanas Kiryakov
E-mail: naso@sirma.bg

Sora Ogris and Hofinger GmbH
Linke Wienzeile 246
A-1150 Wien
Austria
Contact person: Christoph Hofinger
E-mail: ch@sora.at

Hardik Fintrade Pvt Ltd.
227, Shree Ram Cloth Market,
Opposite Manilal Mansion,
Revdi Bazar, Ahmedabad 380002
India
Contact person: Suresh Aswani
E-mail: m.aswani@hardikgroup.com

Changes

Version	Date	Author	Changes
0.5	1.10.2013	Kalina Bontcheva	deliverable outline created
1.0	20.10.2013	Dominic Rout	complete draft
1.1	30.10.2013	Kalina Bontcheva	abstract, introduction, and general edits

Executive Summary

This deliverable describes the prototype interactive summarisation algorithms for stream media, developed in task 4.1. The first step is to select a time period, followed by topics and entities of interest, and then produce a summarised subset of relevant tweets, that meet those constraints. Methods developed in WP2 and WP3 are used in order to derive entities and topics. Work presented here builds upon the summarisation methods, described in the first version of this deliverable [GB12], alongside a new task of ranking content interactively.

Data collection, annotation, and preliminary algorithm evaluation are also included in this deliverable, since no other suitable human-annotated social media datasets currently exist. Therefore, these form an important novel contribution to research on summarising social media streams.

This deliverable investigated several different content ranking approaches and the one found to be most successful (centroid) uses topic models rather than direct word features. These topic models are derived using a method first proposed in WP3, although we modified the approach to use LSI rather than LDA.

The best performing centroid method with topic models has now been made available for integration in the use case prototypes. Usage documentation is also included in this deliverable.

Contents

Chapter 1

Introduction

In June 2012 the widely popular microblogging Twitter service had 500 million users, posting millions of tweets daily[1]. The unprecedented volume and velocity of incoming information has resulted in users starting to experience Twitter information overload, perceived as "seeing a tiny subset of what was going on" [Dou10]. In the context of Internet use, previous research on information overload has shown already that high levels of information can lead to ineffectiveness, as "a person cannot process all communication and informational inputs" [Bea08].

Automatic methods for summarisation of incoming tweets are thus required, to enable professional Twitter users to keep up more easily with key relevant information.

In the context of news and longer web content, automatic text summarisation has been proposed as an effective way of addressing information overload. However, research on microblog summarisation is still in its infancy. Unlike carefully authored news text or scientific articles, microposts pose a number of new challenges for summary generation, due to their large-scale, noisy, irregular, and social nature.

The summarisation of a Twitter timeline can be defined as identifying which tweets are of interest to the given user and should thus be included in the summary. In this deliverable, the prioritisation of content within a user's home timeline is treated as a ranking task.

This deliverable describes the algorithms for the summarisation of multi-lingual streaming media, as well as their implementation. We also present some preliminary evaluation results. Work presented here builds upon the summarisation work, described in the first version of this deliverable [GB12], alongside a new task of ranking content interactively.

Our approach assumes an interactive summarisation scenario, where a set of tweets have been pre-selected by a user, both in terms of covering a short time span and ad-

[1] http://techcrunch.com/2012/07/30/analyst-twitter-passed-500m-users-in-june-2012-140m-of-them-in-us-jakarta-biggest-tweeting-city/ (Visited October 30, 2013).

dressing the same topic. In our experiments, we use short, non-overlapping windows of content at most a few hours long. Interactive, 3D term clouds (similar to those described in D4.1.1) are used to drive a topical filtering process, which will be described in more detail in Chapter 2.

For experimentation and evaluation purposes, the task of ranking political tweets according to relevance for the SORA use case (WP7) is used. The efficacy of several tweet summarisation methods is investigated, in order to give greater prominence to more useful or subjectively interesting content. Early evaluation shows that our methods can significantly outperform simply showing posts in reverse-chronological order, placing more important posts within easier view of a reader and thus reducing the overall effort required by the political analyst.

1.1 Task Definition

From an algorithmic perspective, the summarisation of social media streams consists of four distinct steps:

1. Collect a window of contiguous posts from a user's incoming media stream.
2. Separate the contents of that window into discrete textual units.
3. Derive automatically a number of linguistic and other features for each such unit.
4. Use these features to construct a summary of the salient information in that window.

Classical multi-document summarisation makes the assumption that all given documents are about the same topic or event [NM11]. However, as argued above, social media timelines are diverse, not just in terms of topics, but are also in terms of interestingness and relevance to the user. Therefore, we argue that a key step in summarising topically diverse social media streams is to filter out irrelevant content.

The focus of our experimentation is on the problem of ranking microposts by interestingness to a given user (political analyst in the case of WP7). Given the incoming social media stream, the goal is to score tweets according to how 'interesting' or 'salient' they are to the particular user. These scores are then used to filter out the irrelevant tweets.

A sentence is commonly the smallest unit of text that can be considered for inclusion in an extractive summary. Since tweets are very short and rarely contain multiple sentences, a single tweet could be considered equivalent to a sentence. This leads naturally to our formulation of the problem as ranking and filtering entire tweets by relevance.

The task has been described variously as "tweet recommendation" [YLL12], "personalised filtering" [KOSP11] and "tweet ranking" [UC11, DJQ+10]. The goal could have been defined in a number of other ways. For example, one might attempt to extract the

most useful snippets of conversation from the window of tweets, or the most interesting URLs shared, however, driven by user requirements, we concentrate on the reranking of social media posts by salience.

1.2 Relevance to TrendMiner

Given that TrendMiner is targetted at large volumes of streaming media, windowing by time, filtering by keywords or classifying by sentiment are not guarenteed to yield subsets of tweets that can be feasibly inspected manually. Prioritising important content is necessary in assisting analysis of relevant text and allowing a sense to be gained quickly of the key points from relevant items.

1.2.1 Relevance to project objectives

This deliverable presents some a description of processing techniques which aim to produce summaries of streaming media. Preliminary quantitative evaluation of those techniques is also reported.

1.2.2 Relation to other work packages

Our work should be integrated as part of a larger system as described in (WP5). We use as features in our algorithms some of the techniques developed in (WP3). From (WP2) we take linked open data disambiguated by the LODIE system, both in preparing data for our experiments and as input for the ranking algorithm.

Chapter 2

Data collection

In order to develop effective algorithms for the summarisation of streaming media, one needs to collect first a reliable evaluation data set with gold standard annotations for relevance. As argued in [RBH13], this is a non-trivial problem, which is addressed here through a carefully designed web-based interface.

In our data collection task, we work with the stream of tweets generated by users that are followed by the SORA use case partner. The SORA stream comprises mostly of Austrian political and journalist figures; it is not necessarily representative of the rest of Twitter in general, but rather it is responsive to requirements of a particular use case. The SORA timeline has been archived by the project for more than a year.

In our interactive summarisation work, the user starts off by selecting a subset of temporally and topically focused window from the incoming social media stream. We developed an annotation process that follows the same approach, and which allows users to drive their own summarisation. A similar process can then be used to generate summaries interactively, in response to user needs.

In more detail, the annotation task consists of the following steps:

1. A user choose a window of tweets from a selection of those already available.
2. Based on this selection, a number of keywords and named entities are derived and shown in an interactive word cloud. The user then chooses one or more topically related keywords and entities, and thus narrows down the set of tweets to those on the chosen topic.
3. From the set of tweets on this topic, a user selects a fixed number of highly relevant tweets.

2.1 Selecting windows

Tweets from the timeline are polled regularly, and made available for annotation via a web interface. Windows are formed by the polling process, and a window contains as many recent tweets as could be found in one request to the Twitter API, with cutoffs to prevent overlapping with previous windows. An online calendar interface is used to assist in selecting time windows (see Figure 2.1).

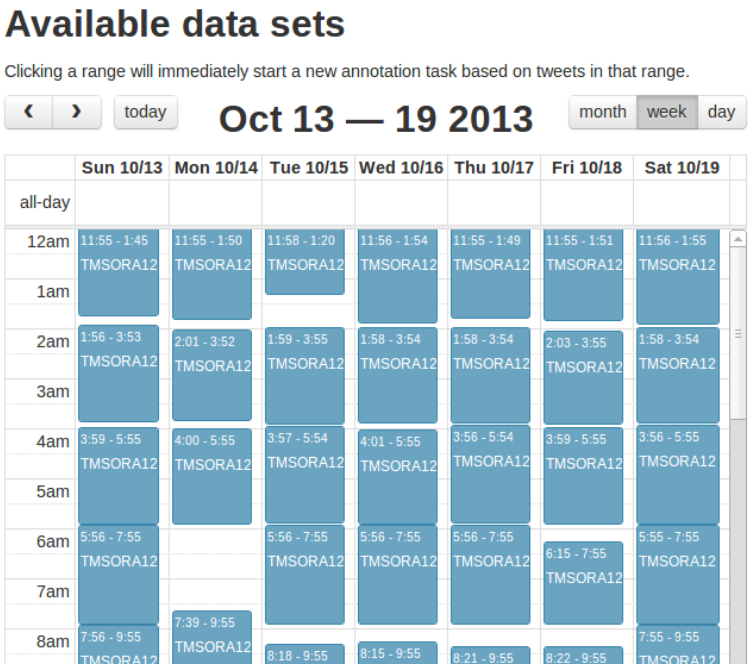

Figure 2.1: Choosing a time window to annotate

2.2 Selecting topics

Once a time window has been selected, an interactive term cloud is generated. The term cloud is populated automatically with the named entities discovered by LODIE [AGBP13] and the annotator is instructed to click on a number of entities related to the topic. In the current system, clouds are generated from the surface forms of the entities, as they appear in the documents themselves. In future work we may retrieve, using the disambiguated URIs, more canonical names such as 'rdfs:label' to prevent the same entity appearing many times in the cloud.

Our term clouds are annotated, to assist in browsing larger numbers of terms. When a term is clicked, the user sees live feedback on how many tweets contain terms from the current selection. Annotators are expected to select enough terms such that at least one

term appears in 50 or more tweets before continuing. This is necesary, in order to ensure that there are a sufficient number of tweets that need to be summarised.

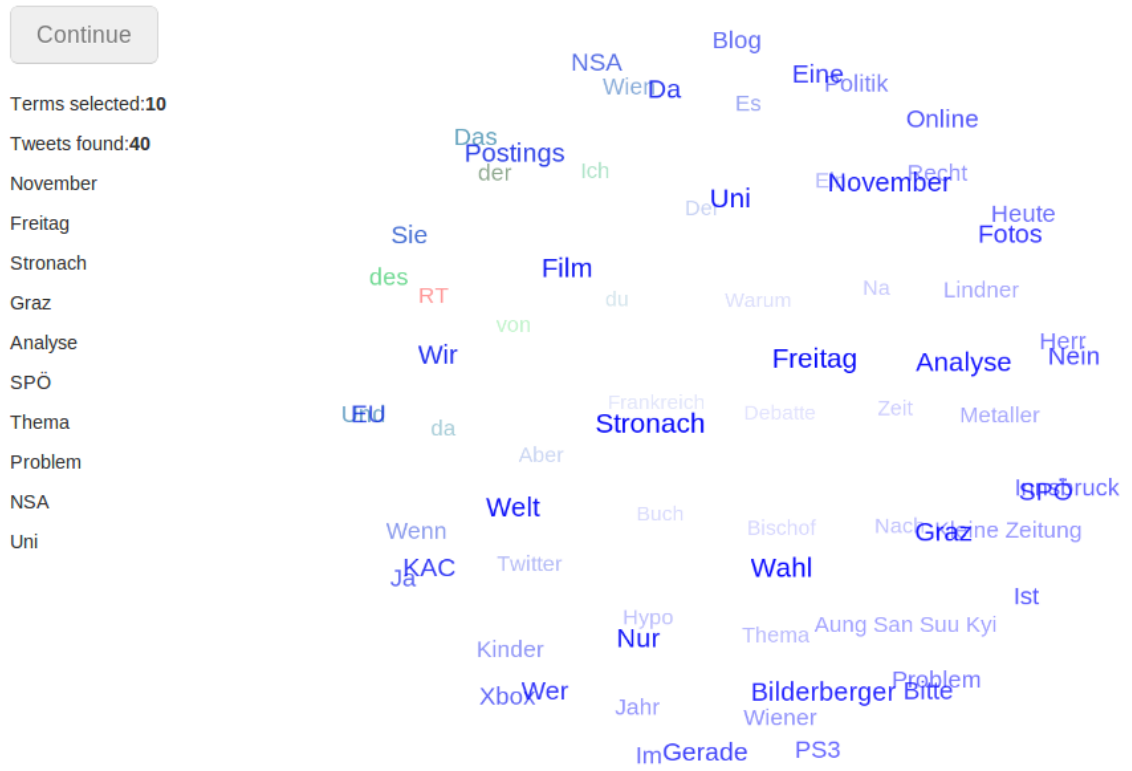

Figure 2.2: Selecting terms to indicate a broad topic

2.3 Selecting tweets

In the third and main stage of the annotation process, the user is asked to emulate the decisions of an ideal content ranking algorithm. By clicking eight of the most relevant tweets to their information need, they create a gold standard (see Figure 2.3). The interface is designed to make it easy to revise decisions, which is often necessary as the user browses through the displayed 50 tweets.

Due to the nature of microblog streams, many posts are duplicates or near duplicates of one another, due reposting. A preliminary experiment showed that users only mark as relevant at most one instance of the same post. Therefore, a post-processing step was introduced, which marks all tweets that are extremely similar to a relevant tweet as also being relevant.

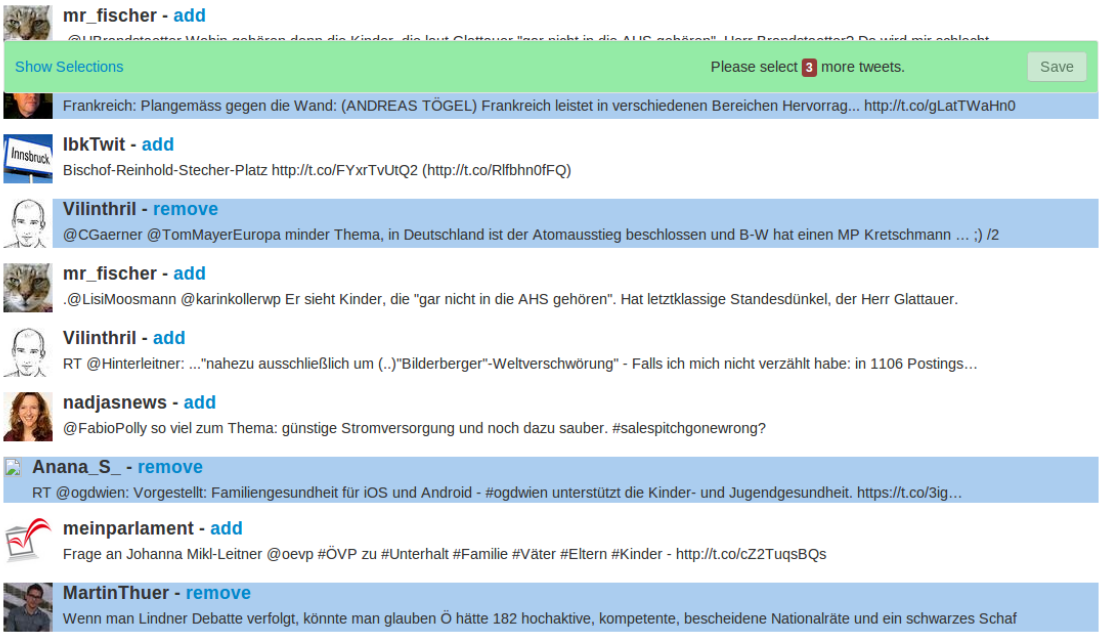

Figure 2.3: Selecting the most relevant tweets in a collection

2.4 Collected Data

Over the period of April 2013 to October 2013, the partners at SORA annotated a total of 62 sets of tweets using the process we have described. Each set contains 50 tweets, of which 8 were marked relevant. In total, our data set consisted of 3100 tweets, with 496 positive and 2604 negative examples. The following table lists some created filters.

Filter used
Alexander, Marko, Christine, Sek, Sen, wJ, fck, gt, amp, de, pic, 2M, Muc, who is, Gita, I, Now, new, Gregor, georg, Top, lol
ZiB, Donau, ZIB, Groe Koalition, Faymann, Stegersbach, Wien, EU, orf, Zib, deported, Ungarn, NR
Michael Douglas, Pamela Anderson, mayer, Smiths, Faymann, Graz, Rudi, Schieder, Alexander, RT
Karl, karl, ZIB, Beatrix, RT, MEP, Mursi, Jane, ha!
Haider, Petzner
Sozialstaat, Premier, France, Luxembourg, Interview, Knig, Debatte, Ende, Und, des
JUSOS, Salzburg, Austria, Zukunft, Online, Bank, System, Euro, Frag, Nur, Monika, Ist, Sie
Jovanka Broz, Sozialstaat, Knig, Sofiensle, Altaussee, Interview, Syrien, Luxembourg, Kirche, Deutschland, Vienna, Mnner, Ordnung, Austria

Chapter 3

Related work

When identifying and interesting content from microtext, it is important to consider that different posts can be relevant for different reasons. As such, work has arisen addressing the problem in different ways.

Topical relevance is among the most widely studied and is typically specific to a Twitter user user. Topically relevant tweets are often personal and of interest due to a user's specific location, hobbies or work. Research that captures this kind of relevance has built bag-of-word models for users and used these to score tweets, typically using techniques such as TextRank, LexRank and TF.IDF [IK11, WZO10].

These bag-of-word models can be sparse, so attempts have been more to enrich user interest models with semantic information. [AGHT11, KOSP11, MM10] create user profiles using semantic annotations grounded in ontologies (e.g. DBpedia) and, in the former case, also expanding URLs mentioned in the user's own posts. None of these approaches, however, have been evaluated formally on timeline summarisation.

Some tweets can also be considered universally relevant, regardless of the interests of the specific user. Such tweets may include those that are humorous, or those that originate from trustworthy or influential authors. [HYZ11] give prominence to tweets of higher quality, judged both linguistically and in terms of the reputation of the author. Likewise, [DJQ$^+$10] use measures of user authority in order to rank search results; their work also demonstrates the use of RankSVM in order to learn rankings from features attached to tweets.

The social relationship between an author and the reader can also affect relevance. [YLL12] present personalised recommendations for tweets, incorporating both textual and social features using a graphical model. Similarly, [CNN$^+$10] recommend URLs to Twitter users based on both previous posts and on social connections.

We consider largely topical relevance in this work, as social relevance and humour are unlikely to be useful for those who read Twitter for professional reasons, or who wish to carry out analysis on tweets.

Chapter 4

Features

We describe and implement a number of different features to use as inputs to content ranking. Many of the ranking algorithms described operate not on text directly but on numerical features, which we will describe once and refer to many times.

4.1 Textual features

We generate counts of tokens that appear in tweets themselves. Tokens are numbered by the order in which they first appear, and this mapping is stored and re-used. We compare vectors of tokens when executing our algorithms.

Since the upper and lower-case variants of words can often be the same referent, but at different points in a sentence or typed differently by a hurried user, we also ignore case. This may have different results for German, wherein case changes for all nouns, than for English, in which case is used to mark only proper nouns.

When counting only single words, the order in which they appear is completely ignored. Counts of unigrams (individual tokens) can only help to compare the vocabulary, rather than the phrases used. To this end, we also count pairs of terms (bigrams) and triples of terms (trigrams). In this part of the work we combine n-grams in a simplistic way.

4.2 Term Weighting

To deal with uninformative words in text, we weight our term vectors be Inverse Document Frequency. Document frequency estimates are gathered from a large background corpus, where each hour is considered to be a single document, rather than each tweet. This allows us to weight more heavily very topical terms (which don't appear in every single hour) than general background vocabulary.

Chapter 5

Algorithms

For the purpose of multi-document summarisation of social media streams, we have implemented a number of different algorithms. Their relative performance is evaluated using the gold-standard data set (see Chapter 2).

The approaches described in this section are homogeneous in that they produce the same form of output as one another - a better performing alogirithm can be used interchangeably with a worse one.

5.1 Vector Space Model

In selecting terms by which to filter their stream, the user is apparently expressing an interest in those terms; the tweets that contain the selected terms can then be ranked according to how well they address the query. For the input set "Salzburg, Faymann, Mai, online, Svejk", one might expect to see a number of tweets matching 'online' but only a few that also discuss Salzburg.

We tested a number of measures to discover the tweets that best match the query. These include cosine similarity, the binary independence model and BM25.

5.2 Centroid and Textrank

Though the words that are selected for the set give some indication of the interests of the reader, they do not take into account the general textual content, including other entities which may be mentioned.

Methods from classical text summarisation incorporate also the rest of the document content, indicating those documents which are most central to the collection. We implement ranking by Centroid [RJST04] and by TextRank [MT04].

In Centroid summarisation, the terms in a collection are counted and 'averaged'. In this way, a central proto-document is obtained (the centroid), and compared to all other documents, using cosine similarity.

TextRank, meanwhile, draws upon similarity between all pairs of documents. A graph is created, with nodes to represent the tweets themselves and edges weighted by the cosine similarity between them. Many documents have no words in common whatsoever, in which case the similarity is zero and the edge is ommitted.

A random-walk algorithm is used over that graph to find the prominent documents, or those that are supported by the greatest similarity to other documents in the collection.

TextRank differs from Centroid in that it allows for multiple 'themes', rather than forcing us to consider the collection as one coherent collection.

5.3 Machine Learning

We experimented with machine learning techniques for discovering the kinds of documents that are generally interesting. Features such as unigrams and bigrams are derived from the text of the posts (Chapter 4), and additional, social media-specific features (e.g. number of retweets and user location) can be included. In general, however, when using a large feature space it may be unfeasible to perform useful supervised learning, given the relatively small size of the data sets that are available to use.

We generate training examples for each tweet in each set. We will use regression, where the objective is 1 for the top 8 tweets and -1 for the remainder. We rank more highly the tweets in a set which are given higher scores by the trained model.

5.4 Dimensionality

All of our methods can be weakened by vocabulary mismatch. While there may be a single, prominent topic in a collection, it is possible that different terms are used to refer to the same entities. This is especially plausible in tweets, where hashtags, long names or abbreviations can be used to refer to the same organisations or people or events.

One way to address the problem of dimensionality is to train topic models, such as latent semantic indexes (LSI) [DDF$^+$90] on a background corpus of suitable tweets, before transforming our documents into a reduced space and continuing with our methods as before. This approach is particularly compatible with vector space retrieval models, but as both algorithms are totally unsupervised, they can be imprecise. The correct number of topics to be used depends on the application, and should be discovered empirically.

Chapter 6

Experiments

In order to test and evaluate various approaches to ranking tweets, we conducted a series of experiments, in which the algorithms are compared to one another using a gold standard data set.

6.1 Evaluation

Our gold standard consists of annotations produced by use-case partners. The resulting judgements are binary, of the form 'this post is interesting, but this other post is not'. Our rankers, meanwhile, produce an ordering or a series of scores; a better ranking should ideally place the interesting media towards the top, and the uninteresting posts very low.

In order to compare continuous ranks to a binary gold standard, we use Mean Average Precision (MAP) [MRS08]. MAP captures the intuition that a user is looking for all of the relevant documents. When our hypothetical user reaches an interesting post, we evaluate the portion of uninteresting posts they would have had to read to get there (precision). The same calculation is carried out for all relevant posts in a set, and averaged to give average precision. The mean of these averages across all sets is the MAP.

MAP has some disadvantages:

- It tests only inclusion of posts, not textual similarity to the gold standard.
- Since users are unlikely to be 'interested' in several very similar posts, we may penalise systems that select differently to users.
- It can be difficult to achieve acceptable variance in results.

In future work, we plan to implement use, a common evaluation metric for summarisation. We also plan to evaluate the summaries interactively, using qualitative feedback from a use case partner.

6.2 Results

6.2.1 Social Features

Similar to previous work [DJQ+10, UC11] retweet counts (ready available from Twitter) are used as a baseline for relevance; one would expect these to be a very impersonal indicator, as for a tweet to be retweeted it must be relevant to a large number of users, not just the reader that will consume the summary. We also utilise counts of number of users that have marked the tweet as their 'favourite', an approach that has not to our knowledge been used in previous work.

Indeed, we find that although retweets and favourites alone outperform random ordering, they are among the worst approaches. It may be beneficial to attempt to combine these features with other, more successful kinds of ranking.

Feature	MAP
Retweet count	25.44%
Favourites count	27.61%

6.2.2 Information Retrieval

We report the performance of cosine similarity, binary independence model and Okapiw BM25 where used to compare queries to documents[CMS09]. All of the information retrieval methods were implemented using unigrams, and the cosine method uses TF.IDF.

Algorithm	MAP
Cosine	30.9%
Binary independence model	21.37%
BM25	26.5%

As expected, these methods do not perform very well, since they are generally developed for the task of retrieving longer documents, from very large collections. As can be seen from the table, the best performing of these algorithms is cosine similarity, in which the key terms which were used to perform the data collection are compared, using cosine, to the available tweets.

Given that the terms used in the cosine method were used to filter the initial set, they are also the words that are likely to appear most often in the source document set. This indicates that the centroid summarisation method could be a more promising approach.

6.2.3 Centroid and TextRank

We implement centroid and TextRank for a variety of n-gram features. IDF weighting was used for unigrams. For bigrams and trigrams, however, there was no large enough background corpus available, in order to generate usable IDF values.

Features	Centroid	TextRank
Unigram	28.98%	29.89%
Unigram (case preserved)	28.33%	29.19%
Unigram with IDF	34.83%	33.62%
Unigram with IDF (case preserved)	34.39%	32.63%
Bigram only	33.24%	29.66%
Unigram with IDF & bigram	35.61%	35.27%
Trigram	33.38%	30.69%
Unigram with IDF, bigram & trigram	35.89%	35.61%

The TextRanking approaches do not appear to outperform those using centroid alone. The scores for TextRank and Centroid for unweighted unigrams did not differ significantly (p=0.14). The approaches that use unigrams with IDF are apparently more effective those that use unweighted terms, though the differences were not significant (p=0.32 for Centroid unigram, p=0.20 for TextRank unigram).

6.2.4 Dimensionality

Methods such as Centroid or TextRank can perform poorly when entities are mentioned in different ways, or they are mispelled. This is a common occurence in our data set, where political parties can be abbreviated, used within a hashtag or shortened. Additionally, though we are interested in the prominent entities in a set of documents, we are also seeking to discover the important themes, and the tweets which best describe those themes.

We train a Latent Semantic Index from historical SORA data. One year of tweets was used for training, accounting for 1,799,924 tweets overall. Using a method similar to that in WP3, we assume little topical variance within a single hour and thus treat each hour as a seperate document.

Our index is used to map our higher dimensioned feature space of n-grams onto a lower dimensional space of topics. We then proceed to use Centroid as before, in order to rank tweets. Using topic models in this way significantly outperforms using Centroid ranking alone (p=0.0015). All figures reported below are for unigrams with IDF.

Topics	Centroid
10	34.84%
50	35.25%
100	40.03%
200	42.03%
400	41.01%
600	40.34%

6.2.5 Machine learning

We evaluate two kinds of machine learning algorithms on our data: support vector regression (SVR)[Joa98] and Naive Bayes classification [MS99]. Support vector regression works well with balanced sets, with limited noise, though they can take a long time to train on many examples. Our data has relatively few samples, but the classes are unbalanced and there is a great deal of noise.

We also consider Naive Bayes classification. Naive Bayes makes the assumption that all features are independant - this is clearly not true of written language. Nonetheless, Naive Bayes can be trained very quickly on our data set, and currently outperforms support vector regression.

For this problem we are using linear or probabilistic models when we are in fact attempting to optimise a ranking system. There exist structured versions of SVM such as rankSVM [Joa02] and mapSVM [YFRJ07], which are evaluated specifically for learning rankings. We intend to evaluate the use of such models in future work. An effective learning approach may allow us to incorporate different kinds of information, such as retweet counts with the text of the tweets themselves.

Features	Naive Bayes	SVR
Unigram with IDF	21.11%	23.49%
200 topics & Retweet count	30.40%	24.07%
Centroid score & Retweet count	27.12%	26.40%

Chapter 7

Usage

According to our current evaluation, the best approach we have tested for summarisation of data from the SORA timeline is centroid, with topics as features. We have trained this method, and made a version of the summariser available as an online api at the following address:

`http://demos.gate.ac.uk/trendminer/summarization/rank_api`

Requests can be made to an endpoint with a collection of tweets, represented in JSON in the format given by the Twitter API and encoded as a multipart file. A response is given, including the original JSON, re-ordered and augmented with the rank of the individual tweets.

A simple interface to allow uploading of JSON files and the display of results has also been provided. An example of the output for this system is shown in Figure 7.1.

7.1 Future Usage

We hope to eventually adapt the annotation process, described in Chapter 2, to display output from the ranking API, and thus to assist in evaluating the summaries. Once integrated, judgements made by the algorithm will determine which tweets are to be shown, and in what order. These judgements can then be evaluated by use case partners, both for correctness and also qualitatively, in terms of usability.

maralkon: Collins: "I'm a 34-year-old NBA center. I'm black. And I'm gay." 1st athlete in major US sports comes out. #LGBT http://t.co/XFCXKhvPqe *(0.17)*
kl1lercher: Router-Linux OpenWRT 12.09 ist fertig http://t.co/mN4hFOY9wX *(0.16)*
andrangl: Wenn das der strittigste Punkt ist, gute Nacht! @ipoStandardat: Promi-Einbürgerung entzweite Rot und Schwarz http://t.co/7AZ60RJmRa *(0.16)*
BayrPetra: Discussion w Gita Sen about overcoming critical barriers to gender equality in development at the diplomatic academy http://t.co/NRxq7RoQyE *(0.16)*
INFOGRAZ: Bilder von der Buchpräsentation #Jörg #Martin #Willnauer – Verlag #keiper – Gedichte - Frauen und Männer http://t.co/B1fHKwokyu *(0.16)*
DiePresse_Eco: Kodak: Filmgeschäft geht an Pensionsfonds http://t.co/eypuQ7hsDg *(0.16)*
derstandard_at: Fremdenrecht - Promi-Einbürgerung entzweite Rot und Schwarz http://t.co/Yno4b2umuw *(0.15)*
mic_ung: Oha. "Das Ende der Arbeiterbewegung" http://t.co/tcEupex6QY *(0.15)*
Heute_at: Unverantwortlicher Vater in Indien ließ Sohn im Volksschulalter mit #Ferrari fahren: http://t.co/WoL0WBvJiP *(0.14)*
krone_at: Geschworene urteilen über "Canadian Psycho" http://t.co/2M8OTa6nJ4 *(0.14)*
orfbrk: "Gegen EU-Trend: Steuerlast für Topverdiener unverändert" – http://t.co/9bV5L2SGop *(0.14)*
monibratic: hat einen Runtastic Lauf über 3,48 km in 26m 53s mit der #Runtastic iPhone App absolviert: http://t.co/bDvXNFJMJ9 *(0.14)*

Figure 7.1: Ranking generated for example tweetset

Chapter 8

Conclusion

This deliverable described the latest work in summarisation of social media as part of WP4, and presented some preliminary evalution of the various approaches.

8.1 Relation to other Work Packages

The tweet collection process is driven by entity annotations produced by the algorithms developed in WP2. Entities, identified by LODIE [AGBP13], are are used to drive the interactive summaries, where users select topics of interest. In future work, this manual clustering may be replaced by automated work from WP3.

Our most successful content ranking approach to date uses topic models rather than direct word features. These topic models are derived using a method similar to that which was first proposed in WP3, although we use LSI rather than LDA. Further experimentation is needed to verify that LSI is not less appropriate than LDA.

In future work, we will evaluate the use of spectral clustering models from WP3 in content ranking.

8.2 Ongoing Work

Future work will integrate the best content ranking approach into the time- and topic-based user interface, that was used for collecting the gold standard data. The next step will then be to evaluate the system qualitatively, alongside dimensions such as ease of use, as well as quantitively, assessing how many of the predictions made are considered to be incorrect.

With respect to method development, higher order n-gram models will be integrated with lower order counts using a smoothing or back-off method, allowing their use without

sacrificing the performance already gained by low-order n-grams. Moreover, there are some ranking-oriented learning models which are worth investigating further, since at present the best performing machine learning method is still worse than simple centroid.

Lastly, we will also investigate the further use of disambiguated named entities for summarisation, perhaps including tweets that mention synonyms or terms that are closely related to frequently discussed entities.

Chapter 9

Acronyms

SORA	SORA Institute for social research and computing	Use case partner on TrendMiner project
LODIE	Linked Open Data Information Extraction	Named entity extraction and disambiguation pipeline developed in WP2
MAP	Mean Average Precision	An evaluation metric for information retrieval [MRS08]
SVR	Supper Vector Regression	A method for automatic regression [Joa98]
IDF	Inverse Document Frequency	A weighting scheme for counts of words
LDA	Latent Dirichlet Allocation	Technique similar to LSI, drawn from Dirichlet priors
LSI	Latent Semantic Indexing	Method for dimensionality reduction based on latent 'topics' [DDF+90]

Bibliography

[AGBP13] Niraj Aswani, Genevieve Gorrell, Kalina Bontcheva, and Johann Petrak. Multilingual, ontology-based information extraction from stream media. Technical Report D2.2.2, TrendMiner Project Deliverable, 2013.

[AGHT11] F. Abel, Q. Gao, G. J. Houben, and K. Tao. Semantic enrichment of Twitter posts for user profile construction on the social web. In *ESWC (2)*, pages 375–389, 2011.

[Bea08] C. Beaudoin. Explaining the relationship between internet use and interpersonal trust: Taking into account motivation and information overload. *Journal of Computer Mediated Communication*, 13:550—568, 2008.

[CMS09] W. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines – Information Retrieval in Practice*. Pearson Education, Boston, MA, 2009.

[CNN+10] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: Experiments on recommending content from information streams. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, CHI '10, pages 1185–1194, 2010.

[DDF+90] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[DJQ+10] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An empirical study on learning to rank of tweets. In *COLING*, pages 295–303, 2010.

[Dou10] Fred Douglis. Thanks for the fish - but i'm drowning! *IEEE Internet Computing*, 14:4–6, 2010.

[GB12] Mark Greenwood and Kalina Bontcheva. Multi-lingual summarisation of stream media software. Technical Report D4.1.1, TrendMiner Project Deliverable, 2012.

[HYZ11] Minlie Huang, Yi Yang, and Xiaoyan Zhu. Quality-biased Ranking of Short Texts in Microblogging Services. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 373–382, 2011.

[IK11] David Inouye and Jugal K. Kalita. Comparing Twitter summarization algorithms for multiple post summaries. In *SocialCom/PASSAT*, pages 298–306, 2011.

[Joa98] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398 in Lecture Notes in Computer Science, pages 137–142, Chemnitz, Germany, 1998. Springer Verlag, Heidelberg.

[Joa02] Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142. ACM, 2002.

[KOSP11] P. Kapanipathi, F. Orlandi, A. Sheth, and A. Passant. Personalized Filtering of the Twitter Stream. In *2nd workshop on Semantic Personalized Information Management at ISWC 2011*, 2011.

[MM10] M. Michelson and S. A. Macskassy. Discovering users' topics of interest on Twitter: A first look. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data*, AND '10, pages 73–80, 2010.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, New York, NY, 2008.

[MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. Cambridge, MA, MIT Press, 1999. chapter 10.

[MT04] R. Mihalcea and P. Tarau. TextRank: Bringing order into text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 404–411, 2004.

[NM11] Ani Nenkova and Kathleen McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2-3):103–233, 2011.

[RBH13] D. Rout, K. Bontcheva, and M. Hepple. Reliably evaluating summaries of twitter timelines. In *Proceedings of the AAAI Symposium on Analyzing Microtext*, 2013.

[RJST04] Dragomir R. Radev, Hongyan Jing, Malgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938, November 2004.

[UC11] Ibrahim Uysal and W. Bruce Croft. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 2261–2264, 2011.

[WZO10] W Wu, B Zhang, and M Ostendorf. Automatic generation of personalized annotation tags for twitter users. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 689–692, 2010.

[YFRJ07] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 271–278, New York, NY, USA, 2007. ACM.

[YLL12] Rui Yan, Mirella Lapata, and Xiaoming Li. Tweet recommendation with graph co-ranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 516–525, Jeju Island, Korea, 2012.