Abstractive microblogs summarization

Nataliia Uvarova

Master's Thesis
Master of Science in Media Technology
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2015

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Abstract

Microblogging is a new electronic communication medium based on short status updates containing personal and instant information. Due to the popularity of microblogs, the volume of information is enormous and big portion of it is duplicative or irrelevant. The effective way to summarize information can be used by scientists, journalists and marketing analysts to get cleverer insights about people's reactions and opinions on different topics: political debates, sport events or product presentations.

Existing summarization algorithms can be enhanced in several ways. The first way is to add sentiment analysis. As information in microblogs is very opinionated, analyzing tweets polarity can improve machine summaries by selecting more sentiment tweets than pure topical. Another enhancement is to use different summary length for different topics. Previous studies often limit summaries to be particular length. Relaxing this restriction can present summaries that are more optimal for a particular topic.

The goal of this research is to perform qualitative study of these enhancements and to provide insights and suggestions for conducting bigger qualitative research. In total ten topics are selected, for which human summaries are compared to state-of-the-art non-sentiment and sentiment summarizers.

Resulting observations are the following: there is more topical than sentiment content in summaries generated by humans, however individual biases could be against the trend; the length of the summary is an important feature that influences both generation of human summaries and interpretation of evaluation results, different topics require summaries of different length; sentiment summarization doesn't produce better results for any evaluation metric used, but there could be possibility for its application in proper settings with specific topics.

Preface

I want to thank all those people who support me through this thesis. First of all, my supervisor Rune Hjelsvold for ideas and suggestions during whole process and especially for proposition to make a qualitative research and without whom my writing would not be nearly as scientific or meaningful. Next to all those, who participated and created a golden summary — finding useful information in modern social media is not easy task, I know. Those people are: Dima Koval, Nikita Uvarov, Kateryna Koval and her brother, Sergii Nuzhdin, Konstantin Hantsov, Roman Rader, Vladislav Magdin, Angelina from Belgium and Aleksandr Yakushev for testing the prototypes.

And finally, special thanks to my parents and husband, who believed in me much more than I did.

Contents

List of Figures

List of Tables

Acronyms

DUC Document Understanding Conference. 12

FoTW Frequency of Topical Words. 13, 15, 18, 24, 25, 27, 40

JSD Jensen Shannon divergence. 14, 15

JST Joint Sentiment Topic. 10

LCS Longest Common Subsequence. 13

LSA Latent Semantic Analysis. 9

NIST National Institute of Standards and Technology. 12

NLP Natural Language Processing. 5, 7, 12, 17, 42

PR Phrase Reinforcement. 8

ROUGE Recall-Oriented Understudy for Gisting Evaluation. 12, 17, 25, 27, 40

SCU Summary Content Unit. 13, 14

1 Introduction

1.1 Topic covered by the project

Modern Web is huge. Amount of information increases each year and most of it is unstructured, containing duplicates and errors. Search algorithms for usual static HTML web-pages are developed and optimized and can satisfy the requirements that users and companies impose on them. But modern Web consists not only from the usual HTML pages: it also includes other content's type: dynamic pages and social portals with streams of user-generated content. One special type of the social content is a microblog.

Microblogging service is a special social network that allows *users* to post very brief *posts* or *messages*. Users could *follow* others — subscribe to their updates. Some authors have few or no followers, whereas others are read by a lot of people. The main feature of the microblogs is short messages — Usually single message has limit on the amount of characters it could contain. For example, Twitter[1] limits each *tweet* to be 140 characters.

In contrast to information found in the web, represented by articles and blog posts (which are mostly generated by single author and represent either factual data or reflections), microblogs contain a lot of very relevant, personal and instant information about the topics. The effective algorithm for search, extraction and summarization of this information could create coherent and comprehensive overview of the topic presented from several points of view.

1.2 Keywords

Information Search and Retrieval, Web-based services, Multi-document summarization, Microblog

1.3 Problem description

Although there is a lot of useful information in the microblogs, its extraction is not a trivial task. Several aspects limit usage of existing algorithms: limited content of a single post (low textual diversity of the posts); big amount of posts (about 500 million posts per day in Twitter[2]); a lot of posts contain opinions and sentiments; people search information related to named entities such as people, events, places, organizations etc; many posts don't provide useful information at all.

There are two main improvements to summarization this research is concentrated on. First is variable length of the summary. When designing and benchmarking summarization algorithms the common length is usually used for all summaries. Especially it is true for multi-document summarization[1]. Xu et al. [2] and Inouye et al. [3] proposed to solve this problem by identifying the number of sub-events or clusters in the topic and to use this number as a base for summary length. But this approach is not general enough and more research can be conducted in the area.

Another improvement to microblog summarization is incorporating a sentiment fea-

[1] https://twitter.com/

[2] `https://about.twitter.com/company`

ture to the summaries. The microblogs contain many personal tweets and opinions and thus the usage of sentiment feature might enrich the summaries with reviews of different polarity. It was suggested in several researches [4], [5] in 2011, but there is still no successful system, so the current research aims to investigate possibilities and problems.

1.4 Justification, motivation and benefits

The need for development of modified or new versions of algorithms for information retrieval from the microblogs is a result of two aspects: users have different usage patterns and the structure of information is different compared to usual web pages.

The research [6] of user search behaviour shows that users search for more personal and social information, whereas in the web the information is mostly navigational and factual. Users tend to repeat the same query to get the updates.

On the other hand, a microblog service could be distinguished from blogs and web by the role of the content in the overall usefulness of the post. In microblogs the content itself is rarely unique; the context is what makes it works: the location, the time and the author determine the usefulness of the particular post.

Thus, development of more suitable algorithms allows using microblogs to better extent. The one could also retrieve not only individual posts (as they are usually very short), but also clustered collection of them, which gives broader perspective.

It is important to do opinion summarization considering both text and sentiment rather than simple textual consideration, since it will provide more comprehensive overview. Negative and positive tweets could be very textually similar (due to limited length and same topic) and thus only one will be presented in simple text summaries, but both carry important information.

With such system marketing people could analyze people's reaction to companies events, such as participation in conferences, product presentations. Sociologist could analyze such reactions for public events, such as political speeches, economy state reports etc.

1.5 Research questions

how human summaries for a topic are different from each other? Human summaries are basis for evaluation, but even for the same topic different people create different summaries. The goal is to study what is common and what is distinct in human summaries for the same topic. By answering this question we obtain important information about sentiment and length of human summaries, that is required for later stages of this research project.

how to decide on proper length of the summary? Generating a summary of proper length is a hard problem and we do not assume it is possible to find a simple solution. Anyway, analysis of human summaries and topics' properties could give hints or even some reasonable upper/lower bounds.

how do sentiment-based summaries compare to non-sentiment ones? Adding sentiment feature to analysis might improve quality of the summaries, but also result in undesired consequences: longer or less informative summaries. By analyzing different summary properties, like similarity to human summaries, amount of topical and sentiment content in them, we expect to come up with hypotheses on advantages

and disadvantages of sentiment-rich summaries.

1.6 Covered scope and contribution

The scope of the research is extractive summarization of microblogs generated to be read by humans. After answering the specified research questions new knowledge will be obtained:

1. in-depth study of summaries produced by humans in terms of sentiment and length
2. the hypothesis about optimal length of the summary for different topics and purposes
3. results of comparing the sentiment-based summarizers to traditional textual ones and suggestions about how to setup bigger experiment for studying them

1.7 Ethical and legal considerations

The datasets that are planned to be used are based on Twitter microblog data, thus Twitter Terms of Service[7] should be met. The Terms of Services of Twitter requires not to distribute downloaded content further. Therefore, the project report should not contain dumps of the whole corpus or big parts of the it: only results in form of summaries and small extracts as examples should be present in the work. In case bigger chunks need to be presented, the best way is to give only IDs, allowing the one who needs it to download the content by themselves directly from Twitter. In addition, there is a requirement to regularly delete tweets, that are removed from the Twitter.

The crawling of the content should be done either via TwitterAPI or with respect to `robots.txt`, presented on the site.

Although special permission is not required to use the publicly available content, it should not be used to harm users or violate their rights. The obtained data should be used only for academic purposes.

1.8 Thesis structure

The thesis has the following structure. Chapter 1 contains Introduction stating the problem, the research questions and limitation of the scope of the research. Chapter 2 provides all required information about text summarization in general, its evaluation and applications to microblogs. Chapter 3 describes which experiments were conducted, which tools were used and how. In the Chapter 4 overview of obtained results is given in form of extracts, plots and tables to justify the discussion in the Chapter 5. Finally, Chapter 6 condenses conclusions and contains some suggestions about future work.

2 Background

Both summarization and sentiment analysis are old and well-studied fields of Natural Language Processing (NLP). In contrast, microblogging is a reasonably new application area with its features and caveats. The chapter is organized in a following way: first we discuss the history and state-of-the-art algorithms in each of these areas, then transition to sentiment summarization of microblogs and finish with survey on evaluating quality of summarization.

2.1 Text summarization

Text summarization is a process of condensation available information creating a *summary*[8]. The definition is broad enough to cover all variety of different summarization types. In order to have meaningful conversation about this variety Spärck Jones proposed a taxonomy scheme in 1998, that is still the most well-known and utilized approach of describing summarization algorithms [9]. The taxonomy is based on analyzing three groups of context factors: input, purpose and output factors. Combination of all these factors defines the application area and how well each individual algorithm can be applied to it.

Input factors include such parameters as number of source documents, their form and language. For instance, the input document could be a long novel written with very figurative natural language or an automatically generated report with pre-defined structure and lexicon. The input factors defines a fundamental division of the algorithms on single document summarization versus multi-document summarization. The former works with one source document from which information is extracted, in the latter several input sources are combined into a single summary.

Next set of factors are purpose factors. This group is the most important — it determines the consumer of the summary, reasons for producing it. Although most summaries are produced to be read by humans, they also can be used to reduce input content for clustering or classification performing as a noise filter[10].

From the output point of view researchers distinguish two main approaches to text summarization: extractive and abstractive. Extractive algorithms select and order chunks of text without changing them. The unit of selection is in most cases a sentence, but can also be a phrase, a paragraph or even a single word depending on the purpose of the summary. The main task of the extractive summarization is content selection. In contrast, abstractive summarization builds a summary from scratch using information in the source only as a reference. This approach is more complex and requires deeper understanding of the text and the domain area, but provides more flexibility. For instance, you could adapt the summary language to specific audience, omit long words for children. In addition, use of abstractive approach can lead to higher rates of compression. Where extractive approach is limited to content density of the original document, abstractive summarization have ability to generate new sentences with much higher content density. Often ultra-condense extractive summarization is discussed as a separate type — headline summarization. Its aim is to produce a single sentence for the whole input[11].

The terminology in this field is rather stable, but have synonyms.

golden or human summary is a summary created by a human. As most summaries are meant to be read by humans, golden summary represents a ground truth of what humans want to see in it.

reference or model summary is a summary used by evaluation tools to compare to. In most cases, the human summaries are used as reference, but it is possible to use automatically generated summary, that is known to have a good quality, as a reference one.

machine or system summary is a summary generated by a particular algorithm.

In our case we will use human summaries as reference ones. Also there are different types of content we will talk about. The terms here do not have strong consensus and their meaning depends on the purpose of the research. In our case, we distinguish between topical and sentiment content of the text. The topical content is about facts, entities and new information present in the text. Topical information is more or less objective, where sentiment content covers opinions and feelings. In the sentence *I am happy the McDonalds exists*, the *McDonalds exists* is a topical content accompanied with the sentiment *I am happy*.

2.2 Microblogs

Microblogging is a relatively new electronic communication medium based on the messages with fixed short length [12].

Microblogging service is a special social network that allows *users* to post very brief *posts* or *messages*. Users could *follow* others — receive their updates in reverse-chronological order. Some authors have few or no followers, whereas others are read by thousands or even millions. The single message has limit on the length — the amount of characters it could contain. For example, Twitter[1] — the most well-known microblog service — limits each *tweet* to be no more than 140 characters.

Most research projects on microblogs are concentrated mainly on Twitter as the most prominent example of microblog service, but other services also have parts similar to microblogs. For example, status updates in Facebook[2] could be also seen as microblog [5]. Throughout the thesis terms *microblogging service* and Twitter; *post message* and *tweet* will be used interchangeably.

Special markup culture evolved in Twitter ecosystem:

retweet usually is marked with `RT` on the beginning of the tweet. When you make a retweet, you duplicate a tweet to your stream in order to share it with your followers. The retweets are the most common way of information spread in Twitter[12].

mention is marked with `@` sign followed by the nickname of the user. Mentions are a way to organize discussions, reply to a tweet of other user or notify someone about it.

[1] `https://twitter.com/`
[2] `http://facebook.com`

hashtag is marked with # followed by a sequence of characters and used to label or tag tweets in order to make it easier for other users to find messages with a specific theme or content.

2.2.1 Comparison to other domains

Most of the unique aspects of microblogs are direct consequences of the tweet length constraint. Intuitively, microblogs could be distinguished from blogs and web articles by the role of the content in the overall usefulness of the post. In microblogs the content itself is rarely unique; the context is what makes it work: the location, the time and the author determine the usefulness of the particular post. “Good morning” from the astronaut in space is not the same “Good morning” your friend posts every day.

Content of each tweet is limited so users apply creativity to put more in it. Heavy use of emoticons, reductions, jargon and slang words is an essence of each tweet. It was reported that the performance of traditional well-established approaches to NLP drops when applied to modern Web2.0 lexicon [13]. Consider example tweets:

> Oh noooes. The Whites Stripes. . . They break up. Such a good rock band. . . *sighs*
>
> The White Stripes have split up? YAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAY! Soz.

Actually, in terms of language microblogs are just expansion of the problems existed earlier in another communication medium — SMS. Therefore some ideas previously developed for SMS are applied to parsing tweets[14].

Another set of unique features is a result of combination of two factors: Twitter’s popularity and retweet capability. Enormous amount of posts is produced every second — about 500 million posts per day [3]. Around 15% of this amount is either spam tweets or retweet [15].

And the last feature is presence of opinions and other forms of sentiment content. In this aspect tweets are very similar to comments and reviews. The difference though that in comments and reviews the target of the sentiment is usually known (the article or product) and very often is useful. It is hardly possible that review of the laptop will include negative sentiment from the fact that the laptop was stolen, which is not true for the Twitter. In the study comparing use of reviews and Twitter to rate movies it was discovered that about 51% of tweets about movies were irrelevant to actual opinions about them[16].

As a result, although microblogs contain some useful information, its extraction is not a trivial task. The main aspects of microblogs that prevent from usage of general-purpose algorithms are following:

1. limited content of single post (low textual diversity of posts);
2. big amount of posts;
3. a lot of posts contain opinions and sentiments;
4. people search information related to named entities such as people, events, places, organizations etc.;
5. many posts don’t provide useful information at all.

[3] `https://about.twitter.com/company`

2.3 Microblogs summarization

We start by describing microblogs summarization based on the taxonomy described in the Section 2.1 and then provide examples of state-of-the-art algorithms. The main difference from other types of summarization is in input factors — features of microblogs including short messages, unique language features etc. But more importantly microblog summarization occupies a unique position on the range of single- vs multi-document summarization.

A collection of tweets can be viewed as a single document from which information is extracted. From another point of view, combined document is nothing more than a collection of tweets. Single-document summarization is considered easier due to the structure of the document that could be used, low percentage of content duplication and usage of similar language across the whole document. But collection of tweets doesn't have these advantages. Each tweet is inherently a separate entity and it is legitimate to see microblog summarization as very specific multi-document summarization where each document is very short, rarely containing more than one sentence. But it should be noted that although a tweet does not always consist of one sentence, it always represents one information unit.

Due to this duality both types of algorithms — those designed for single-document and the ones for multi-document summarization — are used for microblogs.

First summarization algorithms designed specifically for microblogs were Hybrid TF-IDF [17] and Phrase Reinforcement (PR) algorithm [18]. Both of them are from the same group of researchers, the former is based on multi-document summarization and the latter — on single-document summarization ideas.

Hybrid TF-IDF is an extension of an old idea to use Term Frequency (TF) and Inverse Document Frequency (IDF) to score sentences for summarization. In single-document setup it is a very straightforward approach: weight of the sentence is a summation of its terms' TF-IDF scores. TF-IDF score is sensitive to the document length, since longer documents tend to have higher term frequencies. It is not a problem for single-document summarization as there is only one document to work with, but when source consists of multiple documents the use of such scoring system could lead to undesired results. Hybrid TF-IDF changes how score is computed: TF component works with collection of posts as with a single document, thus making term frequencies equal for all documents-tweets. While when computing IDF each post is treated as single document, making sense to use IDF at all.

Another proposed algorithm is **Phrase Reinforcement**. Unlike Hybrid TF-IDF, it is a single-document summarization and is based on the observation, that users often use same words or even phrases describing a particular topic. The algorithm builds a graph of word adjacency weighting how often each pair occurs. The graph is built around the topical phrase: the couple of words by which tweets were filtered and thus present in all of them. Node is also assigned a weight based on the distance from the root and the uniqueness. Summary is built based on the most overlapping weighted sequence.

But summarization algorithms developed for general use, rather than for microblogs specifically, also can be applied successfully. SumBasic, LexRank, and LSA are examples of such algorithms[3].

SumBasic [19] is a simple and effective algorithm based on utilizing word frequencies. Original article didn't consider the algorithm a standalone and production-ready

version. It was rather developed to confirm observation that users tend to include to their summaries words and phrases that often appears in the source. By relying of word frequencies the approach is similar to the one from Phrase Reinforcement, but has a major distinction. The component re-weighting step allows not to include identical sentences and to cover more aspects of the problem. On each step sentence is selected by the average TF of the words it contains, but after selection weights of words from the winner sentence are reduced. Although algorithm was developed as an experiment it shows good results and nowadays is used as one of the baseline for assessing more modern algorithms.

The idea to use Latent Semantic Analysis (LSA) in text summarization was first introduced in 2001 [20] and was later improved in 2004 [21]. **LSA** is a fundamental text processing tool used in areas like Information retrieval [22]. It produces a set of statistical relationships between documents and terms they contain. Starting from the sentence by term matrix, algorithm divides it to linearly-independent parts and assigns weights. The sentences then can be ranked using these term weights for each word combined. Particular form of weighting and ranking functions differs between implementations and could be looked at in corresponding articles.

Another general-purpose summarization algorithm that works well for microblogs is **LexRank**[23]. This algorithm unlike two previous is from the multi-document group. Its goal is to select the most central sentences (so called Centroid-based summarization). The central sentences provide sufficient and necessary amount of information to cover the topic. To find such central sentences a graph with nodes representing sentences and edges weights equal to cosine similarity between sentences in term vector space, is built. Resulting graph is processed with the algorithm similar to PageRank. The sentences corresponding to most central nodes are included in the summary.

Later, experiments were conducted to independently compare different algorithms to each other in microblog settings [24]. Researchers obtained that SumBasic and Centroid algorithms outperform both TF-IDF based (including Hybrid TF-IDF) and Phrase Reinforcement algorithms, and for more topics produce results that are statistically better than baseline Random summarizer.

2.4 Sentiment analysis

Sentiment is a complex entity, consisting of different aspects. The most commonly identified aspects are: polarity, intensity, subjectivity, target entity. Polarity divides all sentiment values to three categories: positive (such as happiness, joy, approval), negative (sadness, anger etc.) and neutral. Intensity presents the numerical value of the strength of expressed sentiment. The target entity shows which particular object the sentiment is expressed about[25].

Adding sentiment features to summarization process can be seen as a purpose factor. It reflects the need of the consumer to have a full range of opinions about the topic covered in the summary.

2.4.1 Sentiment analysis of microblogs

One of the first successful implementations of sentiment analysis over microblogs was a research project by OConnor et al. aiming to study how Twitter reflects public political opinions[26]. Researchers incorporated basic post retrieval by filtering tweets using

keywords and simple sentiment analysis using subjectivity lexicon. Anyway a correlation between the aggregated sentiment and public polls reaches 70% and even more when smoothing and time gap correction were applied. Smoothing is required to soften random peaks, and time gap correction accounts for the difference in information appearing in two sources: almost instantly in Twitter and several days later in polls.

Further we discuss some sentiment analysis algorithms designed specifically for microblogs. It should be noted that it is not comprehensive survey of such algorithms since it is not the goal of the research, only several algorithms are presented to show current state-of-the-art situation in the field.

To create more advanced sentiment analysis algorithm tailored to Twitter specifically the machine learning approach can be used. To train a good classifier a big dataset is required. One of the ways to build such dataset is to transform Twitter corpus by analyzing emoticons present in the tweets[27]. Each emoticon has a particular sentiment attached to its meaning. By expanding the sentiment of emoticon to be the sentiment of the whole tweet containing it the good and big train dataset could be built. The resulting corpus does contain some percent of noise entries since users tend to use text and emoticon opposite by polarity to express sarcasm and irony.

One of the systems built with such dataset is Sentiment140[28]. The main difference from other machine-based systems is use of distant supervision as obtained dataset is automatically labelled and contains noisy data. Authors compare different classifiers such as SVM, Naive Bayes and Maximum Entropy and different set of features: unigrams, bigrams and POS-labeled unigrams. Researchers concluded that combination of unigrams and bigrams can be used with any of the discussed classifiers to achieve high accuracy.

Some Twitter specific sentiment[29]

The more advanced way to analyse sentiment is to detect the sentiment-topic features. The most common model adapting such approach is Joint Sentiment Topic (JST)[30], that was adopted to microblog sentiment analysis in [31]. The training of JST model assigns each word two labels: the sentiment (positive or negative) and topical. In other words, all words are clustered into groups members of which share the same topic and sentiment. The model produces topics consisting of numerical ID and a set of corresponding words, so corresponding semantical meaning and thus title should be identified by human. For instance, in discussed research one of the positive groups contains words like *eat, food, coffee, tasty* and looks like topic about good food. But produced groups can be used for classifying tweets sentiment as is, without human labelling of topics.

A study was performed to benchmark different sentiment analysis systems with each other in different settings and for different topics[32]. In total 15 systems were analyzed on 5 groups of topics having different themes (like Pharma, Retail, Tech etc.). Important result of the research is that not all systems perform equally well for different themes. Most of the systems have a peak performance for one or two themes with accuracy around 60–70% and for other topics produce results lower than 45%. Four systems including Sentiment140 shows stable performance for all topics with average accuracy equal to 66%. Second important contribution of this research is actual case study of the errors. Most prevalent reason occupying 10–15% of the errors were due to misinterpreting semantic/sentiment pair resulting in misclassifying sarcasm, jokes. In contrast, errors due to mixed sentiment in the tweet, that are a major problem in other domains, were not frequent. Probable reason is in the length of the tweet, that limits ability to put two

sentiment in one post.

2.4.2 Sentiment summarization

Sentiment summarization, also known as opinion summarization, can take many forms. Hyun Duk Kim et al. made the most comprehensive review of available options [33]. They divide everything in two big groups: aspect-based and non-aspect based.

Aspect-based summarization is the most common type of opinion summarization. The goal is to extract sentiment for set of aspects, also known as subtopics or features. Various methods have been proposed for each step: identifying aspects, aggregating corresponding sentiment for each aspect, visualization of the results. This approach is widely used in reviews, where set of features are easily distinguishable and overall sentiment is useless [34].

Non-aspect based summarization has several types as well: basic sentiment classification, text summarization, entity-based summary and all types of visualization. Basic sentiment classification is about summarizing only sentiment and presenting it in aggregated form: percentage, polarity or polarity plus intensity. It doesn't suit our need to improve textual summarization. Another popular way is contrastive summarization, where summary consists of two parts with positive and negative opinions respectively. It suits well for contradictory topics with two possible points of view both having meaning to the summary consumer.

From all variety of opinion summarization only textual opinion summarization is suited for general-purpose microblogs. The aspects can rarely be identified, there are more than two points of view and sentiment can target both the topic directly and the circumstances around it.

Surprisingly, there are not so many textual opinion summarization algorithms. Both surveys ([33] and [35]) which we rely on and our independent search pointed only to experiments performed in [36] for summarizing reviews.

Authors described sentiment summarization as an optimization problem, that is formulated like this:

$$\arg\max_{S \subseteq D} \mathcal{L}(S) s.t : \text{LENGTH}(S) <= K$$

In this definition D stands for all sentences in the document, S is sentences forming a summary, $\mathcal{L}(S)$ — an optimization function, that can look like this:

$$\mathcal{L}(S) = -\alpha \text{MISMATCH}(\text{SENT}(S), \text{SENT}(D)) = -|\frac{\sum_{T in S} \text{SENT}(T)}{\text{LENGTH}(S)} - \frac{\sum_{T in D} \text{SENT}(T)}{\text{LENGTH}(D)}|$$

This optimization function matches the average sentiment content of the summary to the average sentiment content of the source document.

Authors studied three optimization functions: Sentiment Match, Sentiment Match + Aspect Coverage and Sentiment-Aspect Match. The first function is the simplest one and targets the summary where average sentiment of the summary reflects the average sentiment of the topic. It will not work well for topics with mixed sentiment where average sentiment could be near zero. Optimization can just select any neutral tweets to satisfy such function. So more complex function is actually used:

$$\mathcal{L}(S) = -\alpha \text{MISMATCH}(\text{SENT}(S), \text{SENT}(D)) - \beta \text{INTENSITY}(S)$$

In this optimization function not only the average polarity should match, but also intensity, which can be computed from the same sentiment values ignoring its polarity sign.

Two other functions in addition to the optimization of the sentiment ensure that all aspects of the topics are covered (SMAC) or covered with corresponding sentiment (SAM). The optimization function for SMAC is presented below:

$$\mathcal{L}(S) = -\alpha \text{MISMATCH}(\text{SENT}(S), \text{SENT}(D)) - \beta \text{INTENSITY}(S) + \gamma \text{DIVERSITY}(S)$$

The function for SMAC has one downside — it produces good results even if different aspects are covered with wrong sentiment. The only requirement is that all aspects are covered and the average sentiment corresponds to the one from the source. To fix this most advanced SAM optimization function is used. It can hardly be used for general-purpose microblogs summarization as aspects are rarely clearly defined. Interested reader can find all omitted details in the article.

2.5 Summarization evaluation

To analyse and compare different approaches and algorithms good and reliable evaluation is required. First and main entity in research community that performed large evaluation in the field of summarization is Document Understanding Conference (DUC). It was annual event organized by National Institute of Standards and Technology (NIST) in 2001 — 2007[4]. Since 2008 DUC is incorporated as summarization track in Text Analysis Conference run by NIST[5]. Originally only manual evaluation was used. It involved many people and was costly and impossible to reproduce. Many automatic or semi-automatic algorithms were introduced on this conference to ease reproducible evaluation and direct comparison of the algorithms.

In general there are two broad categories of evaluation metrics both for text summarization and for general NLP tasks: intrinsic and extrinsic[37]. Intrinsic evaluation tests summarization on its own. The examples are completeness, informativeness of the summary etc. Extrinsic, in contrast, evaluates the summary with regard to some external task, how it can fulfill its purpose, usefulness to the consumers. Example metrics are relevance and reading comprehension.

Following subsections cover most widely-adopted evaluation metrics used for summarization. However automatic objective metrics is not the only method used. Most researches still incorporate human judgment on preferences of one algorithm over another or relevance, informativeness, and other aspects.

2.5.1 ROUGE

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is one of the most used evaluation metrics for summarization[38]. It is suitable for both single document and multi document summarization. Basically, ROUGE is an n-gram recall between a summary and set of reference summaries. It is computed by dividing number of n-grams that match by the total number of n-grams in the references.

ROUGE is a set of metrics each using different type of n-grams or computing the match of n-grams differently. ROUGE-1 and ROUGE-2 are simple recall metrics based

[4] `http://duc.nist.gov/`
[5] `http://www.nist.gov/tac/about/index.html`

on unigrams and bigrams respectively. ROUGE-S counts matches of skip-bigrams — pair of words with any gap between them but in the same order in both documents. One problem with ROUGE-S is that it does not give any score to a sentence if the latter does not contain any word pair co-occurring with its references. ROUGE-SU adds unigram match to ROUGE-S to fix this problem. ROUGE-L and ROUGE-W use different approach from other metrics: instead of relying on n-grams they search for Longest Common Subsequence (LCS) in both sentences. The intuition is that the longer LCS is — the more similar summaries are.

Authors report next properties of ROUGE metric after comparing its results to human judgments:

1. ROUGE-2, ROUGE-L, ROUGE-W and ROUGE-S work well in single document summarization,
2. ROUGE-1, ROUGE-L, ROUGE-W, ROUGE-SU4, and ROUGE-SU9 performs good in evaluating headline summaries,
3. exclusion of stopwords improves correlation with human judgments,
4. high correlation is hard to achieve for multi-document summarization tasks but ROUGE-1, ROUGE-2, ROUGE-S4, ROUGE-S9, ROUGE-SU4, and ROUGE-SU9 work reasonably well when stopwords are excluded from matching,
5. correlation to human judgments is increased by using multiple references.

ROUGE is by definition a recall based metric, and thus sensible to the length of the reference summary. The best results are produced when the length of reference and system summary are approximately equal. The reason is total number of n-gramms of the reference summary in the denominator. Thus if compared summary is shorter than reference one, the ROUGE value can never reach 1.0, since the number of common n-gramms in best case will be equal to number of n-gramms in shorter summary. In the opposite situation, it is possible to get the value of 1.0, but its meaning will be vague. It basically means that all content of the reference summary is covered in the analyzed one, but in addition the analyzed summary could contain hundreds of totally irrelevant sentences. Proper interpretation of such results becomes a hard task.

2.5.2 Pyramid method

There are problems with the simple evaluation model offered by ROUGE. They include human variations, analysis granularity, semantic equivalence and problem of evaluation abstracts. To deal with these drawbacks new evaluation method was proposed[39]. The key idea of Pyramid Evaluation is in utilization of several human summaries that together form a single golden-summary. The method uses two stages of human involvement — annotations for Summary Content Units and generation of golden summaries. Summary Content Unit (SCU) is a single fact that can be included in the summary. Its length can vary from couple of words to sentence, but can not be bigger than sentential clause. SCU that appears in more model summaries is considered more important. Thus, the Pyramid evaluation has several advantages over ROUGE: it uses annotated SCU rather than all content, that allows not accounting the inclusion or exclusion of additional and supporting text to the summary; it assigns weights to each SCU depending on how important it is. By weighting SCUs it is somewhat similar to intrinsic Frequency of Topical Words

(FoTW) but still remains human-based extrinsic metric. The SCU are arranged in levels with respect to their weights. According to the metric the best generated summary will include all SCU from top layer, then if length allows SCU from lower layer, and so on until the summary is long enough.

This metric is current state-of-the-art, although its main limitations are lack of reference implementation and need for two stages of people involvement.

2.5.3 Automatic evaluation

Introduction of ROUGE was a huge step to the reproducibility of the evaluation results. Previously human judges were required to read each summary and give it a score. With ROUGE humans are required only to generate golden summaries, that can be reused for next projects and algorithms. But sometimes even such golden summaries are not possible to obtain. So possibility to create fully automatic metrics, that don't require model summaries at all was investigated[40]. Automatic metrics can be divided into several types, based on source information for comparison: input-summary similarity, pseudomodels, consensus-based.

Input-summary similarity uses only source documents as a based for evaluation. The core idea is that a good summary is similar to the input in terms of content. Metrics that utilize such approach differ by the similarity function used. Similarity functions could be based on computing probability distribution. Examples are Kullback Leibler divergence and Jensen Shannon divergence. Other way of comparing summary to an input is based on identifying topical words in the source and computing what percent of them appear also in the summary. Presence of more topical words signals about higher quality of the summary. The most simple way to identify topical words is to analyze which words have higher probability in the source than in reference text.

Pseudomodels are about adding system summaries to available human summaries. This approach works best when human summaries are present, but are not enough. On first step all system summaries are ranked using only human summaries, and best of them are selected to be pseudomodels. The final scoring is based on the extended set of models: human summaries and pseudomodels. The scoring function could be ROUGE or Pyramid or any other that requires golden summaries.

Last type of full-automatic evaluation is consensus based, that uses only system summaries. It is an extension of previous idea with pseudomodels, but that doesn't require human summaries at all. The entire collection of produced summaries is aggregated and word probabilities are computed. Good summary has a distribution similar to the aggregated one — the distribution similarity can be computed by use of Jensen Shannon divergence (JSD), for instance.

2.5.4 Microblogs summarization evaluation

Choosing proper methods of evaluation is important for understanding the results of the research and it substantially influences how research is conducted. Different methods are suitable for different application areas. Evaluation of evaluation metric itself is usually performed using correlation with human judgments. Even complex metrics are based on simple textual similarity, but manual judgments assess the semantic, grammar, readability and similar advanced features. If the metric conclusions correlate with conclusions made by judges for wide range of summaries of particular type then this metric can be used to evaluate this particular type of summarization.

Such type of analysis was performed for microblog summarization in the [41]. Authors studied how different metrics agree with each other in assessing algorithms, what metrics produce results close to the humans' opinions. Three metrics (ROUGE-1, FoTW and JSD) were used to evaluate three summarization algorithms: SumBasic, Hybrid and Cluster-based. The results of the research were the following:

1. ROUGE, FoTW and JSD often do not agree on the best algorithm and thus measure different aspects of the summaries.
2. FoTW has the highest correlation with human judgements and thus is recommended metric to use.

2.6 Background wrap-up

The most important conclusions of this Chapter with respect to the goal of the research are presented below:

1. Current research is concentrated on extractive summarization with the humans as a consumers.
2. Sentiment summarization will be based on the algorithm for reviews described in the Section 2.4.2.
3. LexRank, Sumbasic and LSA will be used as baseline algorithms as they are current state-of-the-art approaches.
4. One of the current state-of-the-art sentiment analysis tool will be used (like Sentiment140).
5. ROUGE will be used as extrinsic and FoTW as intrinsic metric.

3 Methodology

The text summarization as a part of much broader NLP area works with processing of textual information with the specific purpose in mind. Different methods and algorithms achieve the target with different performance. The most commonly used method in summarization evaluation is developing a proof-of-concept and assessing it using either objective metrics through experiments or subjective through surveys. The most typical setup is direct comparison of the proposed algorithm or modifications to current state-of-art and baseline algorithms.

This research is based on applying qualitative methodology. The goal is to gather insights for further analysis about problems and possibilities of the area. The small set of topics, algorithms and human summaries are analyzed in great depth. Such setup doesn't allow us to make significantly strong conclusions but has other advantages: the small amount of data could be analyzed for reasons and motivations, like instead of reporting that human agreement on average is 30%, we can look on the produced summaries and see what are particular reasons for disagreement.

Developing proof-of-concept is an important part of the project. Reference implementation can be measured to obtain information needed for answering the research questions. Implementing proof-of-concept is not the same as developing real world implementation of the algorithm. The attention should be given to the parts and aspects required for answering the research questions. For instance, in case of this project the software should implement summarization algorithms correctly, but not necessarily efficiently, since the applicability of a particular algorithm in online or real-time settings is out of scope of the thesis. Reference implementation of the baseline algorithms should be used where possible.

The process of obtaining information to answer the research questions consists of the following stages:

1. choice of methodology
2. search or creation of a dataset and data for evaluation
3. implementation of the algorithms
4. collection and analysis of the results

3.1 Evaluation metrics

As described in Section 2.5 there are several widely-accepted evaluation metrics used for text summarization. The ROUGE is one of the most popular from the extrinsic group that measures how well generated summaries reflect the golden summaries. Although it has been shown to have some pitfalls, it is still commonly used, as alternatives like Pyramid evaluation requires much more work and data.

ROUGE requires golden-standard summaries, generated by human, as input, and compares how generated ones are similar to them. To properly use this metric special conditions should be met[42]. The required number of samples and references was cal-

culated for results to be statistically significant. Microblog summarization falls in the category of multi-document summarization in terms of ROUGE. The reference summary is a human generated summary for particular topic. For one topic there could be several reference summaries. There are several types of ROUGE metrics. ROUGE-1 use only unigrams for calculating similarity. It is the simplest form of it. More advanced versions like ROUGE-2 and ROUGE-S use bigrams or so called skip bigrams, allowing several tokens between each part of the bigram. ROUGE-SU is a combination of both: it uses unigrams as well as skip-bigrams.

In current research the ROUGE-2 was used as a compromise between too simple unigrams and too complex use of skip bigrams. In general the more reference summaries are available the more accurate and truthful the results are, especially in terms of evaluation opinions and sentiment-rich summaries. Due to the resource and time constraints limited number of golden summaries could be generated. Three human-generated summaries for each topics leave enough resources to cover enough topics. For instance, for ROUGE-1 metric in case of 3 reference summary, 9 topics should be investigated, for ROUGE-SU4 13 topics could be used. For ROUGE-2 the value is between them and is 10 topics.

As a result three reference summaries are required for 10 topics — that gives 30 summaries in total. The produced summary should be in approximate range of 5–10 sentences and it is estimated to take no more than 30 minutes for a summary. It gives 15 human-hours. For single person not to spend more then 1.5 hours on the task, it should be around 10 persons, which is a reasonable amount that could be found among the students of the university.

Several intrinsic metrics exist, but based on the comparison of them to the evaluation of microblog summarization 2.5 we decided to stick with the FoTW. It measures topical content and answers a question how introduction of an additional (sentiment) feature influence the informativeness of the summary if the length stays the same. In addition, it can provide data for assessing summaries of different length and which factors lead to optimal length of the summary.

FoTW compares how many topical words are present in the summary compare to amount of topical words in the source documents. Better summaries tends to contain more topical words. It is fully automatic metric that doesn't need human-generated ground-truth. Using it we can calculate the dependency of informativeness with respect to the length of the summary with and without the sentiment. Thus we can estimate how big is drop in the topical content when sentiment feature is added with the same length of the summary and whether increasing length could improve it.

3.2 The dataset

There are only few datasets available for Twitter. Mostly it is due to the latest changes in the Twitter Terms of Service that don't allow redistributing and publishing tweets, only their identifiers [7]. The other reason is the amount of work needed to create such dataset. Often researches create their own small targeted datasets by collecting the tweets through Twitter API. This approach is easy and convenient but it hurts reproducibility of the research results since rarely the datasets are made available for the community.

For dataset we used Tweets2011 [43] collected by NIST for microblog track. It is one of the most widely adopted datasets for Twitter. This dataset consists of identifiers provided by Twitter for approximately 16 million tweets sampled between January 23rd

and February 8th 2011. The corpus is designed to be a reusable, representative sample of the Twitter — i.e. both important and spam tweets are included. Also 50 samples queries and relevant tweets are provided.

The dataset contains 50 queries with marked relevant tweets from which 10 were selected. Queries were selected to represent different topics.

They differ in several ways:

1. by the specificity of the topic. Several queries represent very specific events, like Oscars nominations of the particular movie; several are more broad — opinion gathering about McDonald's food or about advertisement videos shown during sport event. It was demonstrated that specificity of the topic affects, for instance, performance of classification. More texts are specific and coherent — more accurate the classification is[44].
2. by absence/presence of the polarity tweets. Some topics contain more negative messages, other more positive. Important fact to understand is that all topics have a big chunk of neutral tweets as a base of the dataset.
3. by the amount of relevant tweets for the query. Several topics have up to 100 relevant tweets, other — around 20. The queries that had more than 100 relevant tweets available were artificially cut to contain only 100 tweets as test run of human summary generation showed that people don't want to process long lists of tweets and stop paying attention by selecting random items.

Selected queries are presented in the table 1. Later in the discussion often only topic id will be used, the topic id is original query id from the Tweets2011 dataset, that is widely used in all researches that incorporate this dataset.

3.3 Creating golden summaries

As we discussed earlier for ROUGE evaluation metric the human-generated golden summaries are required. The fellow students were recruited to do it.

There are different ways researchers generate golden-standard summaries. First is to use public sources of information about event (like news and Wikipedia) to write a bunch of simple sentences that mention its all important aspects. Next option is to present the most relevant tweets about the topic found in the corpus and allow volunteers to select what they consider the most appropriate for the summary.

The first approach is threated to the vocabulary mismatch and guidelines for the participants should be written with huge attention. In addition, it could take more time for the participant to make the summary since they should do the research by themselves. The second approach requires clever ranking of the tweets. Improper ranking could lead to loss of important piece of information. Since the Tweets2011 corpus contains 50 queries with ids of the relevant tweets, some queries could be used as topics and corresponding relevant tweets — as input for participants. Participants select tweets to form the summaries from each set of relevant tweets that also works as input for summarization algorithms.

The work was divided so each student spent around half an hour on the task. As a result, there were 10 students in total. Each one generated three summaries, producing in total three summaries for each of ten topics. The topics were divided and assigned

ID	Topic name	Topic description	#tweets	Positive	Negative
14	release of The Rite	The Rite is a drama/horror movie released in 2011	57	0.25	0.12
17	White Stripes breakup	White Stripes were american rock group	27	0.07	0.19
24	Super Bowl seats	SuperBowl is annual championship american football game. in 2012 there was scandal with not enough and bad seats on the stadium	50	0.34	0.12
29	global warming and weather	news/opinions about global warming during February 2011	67	0.03	0.33
36	Moscow airport bombing	In 2011 there was terrorist attack on Moscow airport	100	0.03	0.52
78	McDonalds food	The opinions about McDonalds during February 2011	100	0.18	0.20
79	Saleh Yemen overthrow	Ali Saleh was a President of Yemen in 1990-2012. In 2012 there were masive protests against his rule	100	0.06	0.08
88	Kings' Speech awards	The King's Speech is a movie about King George VI	100	0.38	0.04
99	Superbowl commercials	Superbowl is annual championship american football game. The topic is about commercial shown during Superbowl match in 2011	100	0.34	0.08
101	Natalie Portman in Black Swan	Black Swan is a movie about ballet where Natalie Portman played leading role	26	0.46	0.04

Table 1: Selected topics

randomly, and there was no assumption about previous knowledge of the topic. The test

run shows that it could be hard to understand essence of some topics just by its title (the Tweets2011's query). For instance "White Stripes breakup" is a topic about the breakup of the American rock band "White Stripes", but if you don't know about such band it will be hard to guess. To eliminate need for users to search or guess details of the topic, simple description was presented. The description was carefully formulated in a way to provide information about possibly unknown terms, names and events, and not to create a summary of tweets.

The instructions given to the users are presented in the Appendix A. In the instructions the topic of the research was generalized to "Study the microblogs summarization". The particular research questions about sentiment and summary length were not mentioned, not to guide users about what tweets they should select. It was extremely important, so the produced summaries represent the actual users view of the summary.

Two options exist how to deal with the length of the summary: either force summaries to be one particular length or to allow humans to create summaries with different length. Former approach is widely used among the researches. But different topics could have different topical content and thus forcing the same length for all topics could have undesired consequences. If the limit is set too low users will be forced to select which of the content to include and which leave out. In case the limit is too high participants will include not relevant or not informative tweets. Both variations lead to summaries not representing actual snapshot of the users' need. So no hard limit was set. Instead, users were advised to create summaries 5 to 10 tweets long, and include what they feel was more appropriate (with possibility to break the limits). But if lengths differ much the ROUGE could not be objective in comparing them. To simplify normalization to the same length they were asked to order tweets. The most important and topical tweets should go before the more detailed and auxiliary ones.

3.4 Selected algorithms

3.4.1 Random

Random summarizer is the most simple of all baselines. It randomly selects sentences (in our particular case tweets) until it reaches the required length. When analyzing the results of metrics for this summarizer, three random summaries for each topic were generated and scores were averaged. Although it might look too simple to include it in the research, as we will see in the chapters 4.3 and 4.4, for some topics many summarizers behave no better than random one.

3.4.2 LexRank and LSA

The reference implementations were used for both LexRank and LSA from `sumy` package for Python[45]. This implementation uses stop-list filtering and allows to specify desired number of sentences to include. Since it works with sentences as unit of selection, in the preprocessing step internal punctuation in the tweet was removed.

3.4.3 SumBasic

As was discussed in Chapter 2.3 SumBasic is a simple frequency-based summarization algorithm. There is no good and easy to use implementation of this algorithm, so we developed it by ourselves. The algorithm was developed as an extension to the `sumy` library to use the same interface for all summarizers. The source code is it the Appendix E. For detailed algorithm consult the original article [19].

3.4.4 Sentiment-based algorithms

Sentiment summarization algorithms are heavily based on the ideas of sentiment summarization for the reviews [36] described in the Chapter 2.4.2.

In the original article the constraint on the summary length was weak, it should be no more than some predefined upper-bound. To answer the research questions of this research it is required to generate summary of specific length, so the restriction was changed to stricter:

$$\arg\max_{S \subseteq D} \mathcal{L}(S) s.t : \text{LENGTH}(S) = K$$

In this case K represents desired length of the summary.

The notion of aspect is very meaningful in the consumer reviews of the products, but less suitable for most topics in microblogs. So the optimization functions were changed and simplified. The most simple one, Sentiment Match remains the same, but instead of working with aspects, second optimization works similar to SumBasic and identifies high frequency words.

Sentiment Match optimization functions look like this:

$$\mathcal{L}(S) = -\text{MISMATCH}(\text{SENT}(S), \text{SENT}(D)) = -|\frac{\sum_{T in S} \text{SENT}(T)}{\text{LENGTH}(S)} - \frac{\sum_{T in D} \text{SENT}(T)}{\text{LENGTH}(D)}|$$

This algorithm will be referred as Sentiment summariser.

The one with word frequencies looks similar (and is used in Sentiment+Frequency summarization algorithm):

$$\mathcal{L}(S) = -\text{MISMATCH}(\text{SENT}(S), \text{SENT}(D)) + \sum_{T in S} \frac{\sum_{W in T} \frac{\text{FREQ}(W)}{\text{LENGTH}(T)}}{\text{LENGTH}(S)}$$

The optimization of such kind usually is a NP-hard problem, so the greedy linear search (other known as hill climbing) is used[46]. Since it is prone to getting stuck in local maxima version of the method with random restarts is used. The source code could be found in the Appendix E.

3.5 Implementation details

3.5.1 Tweets preprocessing

All tweets in the dataset relevant to the queries are in English. Thus, there is no need for additional filtering for the language as it is done in other researches. Tweet preprocessing consists of two main stages.

During first stage tweets are prepared to be used in human summaries. The text is left untouched, except URLs and mentions are removed. The mentions serve for conversations and tagging people and rarely contain meaningful content. Thus, there is no difference when they are removed — during first or second stage. Removing mentions earlier in the pipeline results in total de-personification of the tweets, transforming them into bunch of texts and protecting anonymity of Twitter users involved in the conversation about selected topics. In contrast, cleaning for URLs is an important step. Many tweets do not include the information in the content directly, but rather consist of the headline and the URL to the webpage with additional information. This research focuses

on the summarization of the content presented directly in the tweets, not on the content they have links to. Automatic summarization algorithms use content from the body of the tweet exclusively and removal of the URLs ensures that humans generate their summaries using the same information.

The second preprocessing stage prepares tweets as input documents to the summarization algorithms. Following steps were performed:

1. lowercasing;
2. punctuation removal inside the tweet. The step clusters tweet's body to a sentence, single semantic unit. Summarization algorithms often work with sentences and this step allows usage of such implementations without change.
3. removal of hashtag sign. Hashtags are words specially marked, so tweet can be searched by them. This step converts them to ordinary words. It is not wise to remove hashtags completely since they often convey the most important information in the sentence and could be even built into the sentence structure:

   Natalie Portman receives #oscar

   Such simple processing doesn't always produce good results, hashtags could also contain several words glued together, like

   Natalie Portman receives #TheBestActress

   Then this hashtag will create separate term. In more advanced setup more tailored processing of Twitter body could produce better results.

3.5.2 Tool for human summaries generation

The simple tool was developed[1] to help participants create the summaries easier and faster. It is a web application, deployed on the remote server that can be accessed through the internet. Use of web technology provides the easiest way to build tool accessible by the user with any operating system. The topics were randomly assigned between users. Each user received unique URL identifying his session linked to the list of selected topics.

The screenshot of the interface is presented on the figure 1, additional screenshots can be found in the Appendix B. The UI contains two pages: the page with instructions and the page where actual tweets selection is taking place. The latter page is based on the twin column selector, where on one side all available tweets are presented, on the other — tweets selected by the user. By pressing buttons "select"/"un-select" user moves tweets between lists. From this page user can switch to the page with instructions at any time. As the number of tweets per topic reaches up to 100 tweets it could be hard to navigate, search and rank them. To facilitate participants' concentration on relevant tweets, a special function to hide tweets was added. The source code for the tool could be found in the Appendix D.

3.5.3 Sentiment analysis

For sentiment analysis the Textblob library is used. It has two modes of sentiment analysis: using Pattern library[47] or using trained classifier. Both modes was compared to Sentiment140 sentiment analysis system[28], the reason to compare to Sentiment140 was described in the Chapter 2.4 — Sentiment140 has one of the most stable accuracy among different topics for microblogs. Pattern mode is selected as Sentiment140 and Pattern produced results very coherent with each other: around 80–90% of tweets receives

[1] `https://github.com/AAzza/react-twincolumn-selection-app`

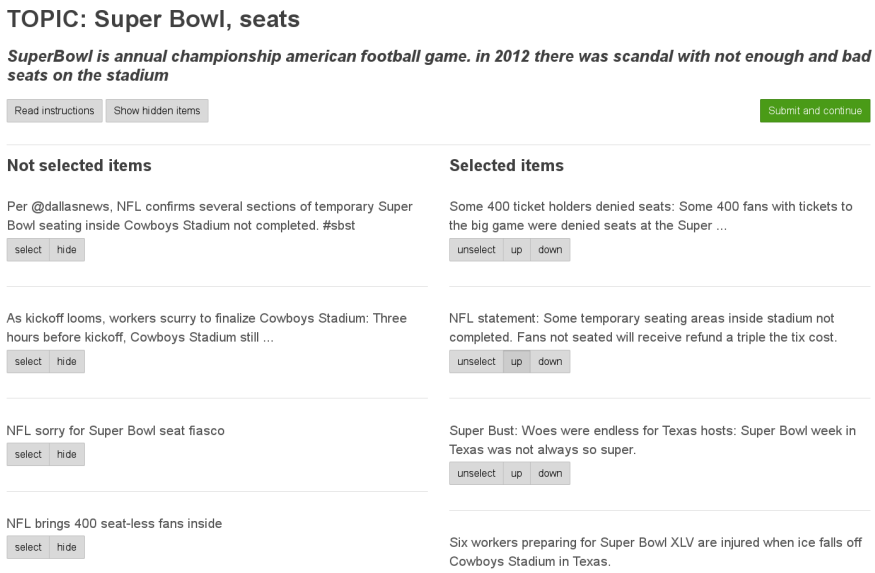


Figure 1: User interface of the tool

the same sentiment. The reason to use Pattern library instead of directly Sentiment140 is a requirement to have sentiment intensity rather than simple polarity, that Sentiment140 provides. Pattern Python library is Natural Language Library implementing several state-of-art algorithms. The method it uses for sentiment analysis is based on the subjectivity lexicon and subjectivity detectors that was introduced in this article [48].

3.5.4 Libraries

Implementing software correctly is a hard task and use of proper tools ease the process much. The Python language is popular in the scientific community due to the big and consistent set of libraries for data manipulation, text processing and machine learning. All Python scientific framework is based on the two libraries: `numpy` for providing data manipulation capabilities and `nltk`[49] — a library for natural language processing. More libraries from the framework were also used: the `scikit-learn`[50], well known Python library for machine learning tasks, `pandas`[51], that provide powerful features to work with tabular data and `matplotlib`[52] a de-facto standard visualization tool.

Both evaluation metrics (ROUGE and FoTW) have official implementation. ROUGE originally was distributed as a package written in Perl [53] and was not trivial to use — both input and output files as well as file with the settings used their own format. In 2015 the second version of the package was released. ROUGE 2.0[54] is developed in Java and uses simple text files accompanied with directory structure conventions that simplifies the usage.

The implementation for FoTW can be found in Simetrix evaluation suite[55]. It is a Java package that also uses a text file with settings to map summaries to its corresponding source files. Simetrix contains implementations of several intrinsic metrics, like Kullback Leibler divergence or Jensen Shannon divergence, but only FoTW was used from available metrics. In all metrics and summarizers the stop-words were filtered out.

3.5.5 The pipeline

An approximate processing pipeline is presented on the figure 2. It transforms the input data to the output CSV files, containing all available information about topics, human and automatic summaries.

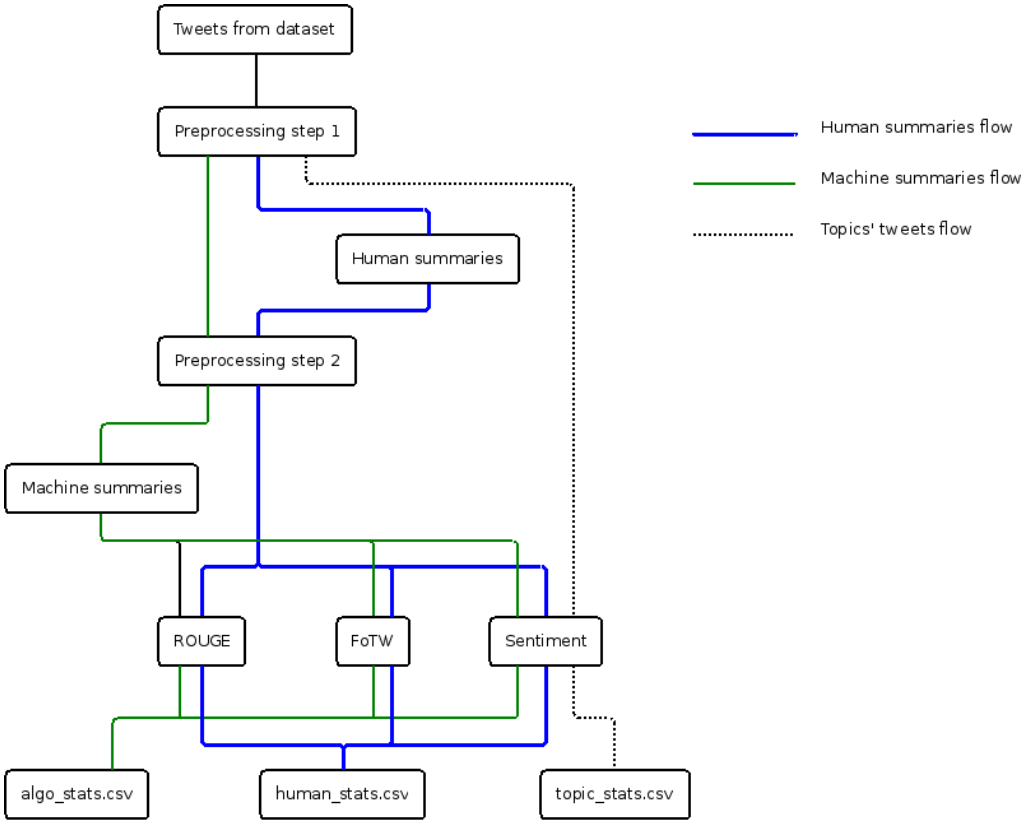


Figure 2: Processing pipeline

1. On the input of the processing pipeline there are source documents and summaries generated by the participants.
2. Both source documents and human summaries are going through second step of preprocessing (lowercasing, punctuation signs removal etc.)
3. For all available summarizing algorithms automatic summaries are generated. The summaries are generated for a range of lengths (mostly the range is 1–30).
4. The human agreement is calculated using ROUGE applied to human summaries themselves.
5. FoTW for human summaries is calculated.
6. Both ROUGE and FoTW are calculated for all automatic summaries.
7. Sentiment content of all available summaries (both human and automatic) is analyzed.
8. CSV files with all available information are saved for further processing.

   The output files are next:

topic stats containing topic ID, number of available tweets, percent of positive and negative tweets.

human stats containing topic ID, human ID, length of the produced summary, percent of positive and negative tweets.

metrics for human containing topic ID, human ID, ROUGE results (precision, recall and f_score) and FoTW value.

metrics for algorithms containing topic ID, algorithm, length, ROUGE results and FoTW value, accompanied with percent of sentiment content.

Extracts and aggregations of this information are presented in the Chapter 4.

4 Results

In this chapter the main results of experiments are presented. We start with human summaries alone, then compare them with automatically generated summaries and finish with evaluating machine summaries using ROUGE and FoTW. In the chapter results are be presented in form of plots and tables accompanied by the explanation who to read them. More in-depth discussion can be found in Chapter 5.

4.1 Human summaries

In the Table 2 the basic statistics about human summaries are present: the mean and standard deviation of summaries lengths.

Topic	Summary mean length	Summary length's STD	Length of topic	Ratio summary length to topic length
17	4.00	1.00	27	0.15
101	5.33	1.15	26	0.21
78	6.33	1.53	100	0.06
88	6.33	2.89	100	0.06
24	7.33	1.15	50	0.15
14	7.67	2.89	57	0.13
29	10.00	5.57	67	0.15
36	10.67	6.43	100	0.11
79	11.00	5.29	100	0.11
99	14.33	4.51	100	0.14

Table 2: Mean and standard deviation of the length of human generated summaries

We can see that summaries are generated with different length and often the deviation in the lengths is big (around 5 tweets for topics #79, #99 etc.)

The human agreement is a commonly adopted metric that shows how similar to each other humans summaries are. Human agreement is often used to see how good the automatic summary could be in principle. If humans don't agree on a particular topic, then no machine summary could score high for this topic if evaluation metric is based on these human summaries. The user agreement for each topic is shown in the table 3.

The common way to calculate human agreement is to use the same human-based evaluation metric (in our case it is ROUGE) and apply it to human summaries themselves. The ROUGE is calculated for each summary using two remaining summaries as references. Results for each summary are averaged producing the final value for each topic. From precision, recall and f_score, produced by ROUGE, the f_score is used. The

Topic ID	Human 1	Human 2	Human 3	Average
14	0.204	0.268	0.230	0.234
17	0.552	0.393	0.411	0.452
24	0.453	0.459	0.362	0.425
29	0.484	0.354	0.383	0.407
36	0.327	0.449	0.476	0.417
78	0.187	0.154	0.143	0.161
79	0.274	0.286	0.273	0.278
88	0.329	0.321	0.465	0.372
99	0.386	0.482	0.475	0.448
101	0.551	0.511	0.488	0.516

Table 3: Human agreement for each topic

values are ranged from 0 to 1, where 0 is the worst value showing no similarity at all, and 1 is the best, showing identical content in the summaries (in case the length is the same).

ROUGE-based human agreement values usually fall in range 0.3–0.5. Higher level of similarity is possible, but in real setup is rarely seen. Since ROUGE-2 works with bi-gramms, single exchange of two words affects three bi-gramms, one missed word affects two bi-gramms.

Consider examples from topic #101:

> oscar's best actress nominee, go natalie portman.
>
> Natalie Portman is nominated for Best Actress

Both tweets convey the same information, but use different word ordering. As a result they have only two bi-gramms in common: *Natalie Portman* and *Best actress* from 6 and 4 containing in first and second tweet respectively.

As we can see some topics have high level of human agreement, like topics #17 or #88 with average ROUGE values being more than 0.3. But some topics show very low level of agreement. For instance, topics #14 and #78 have values less than 0.1.

Taking into the consideration, that ROUGE metric is sensible to the differences in lengths of the summaries, let's examine if this could be a reason for low level of agreement for some topics. As was pointed in the Chapter 3 humans were asked to place tweets in the summary in the order of increasing importance, so it is possible to make shorter versions of it, by selecting first N items. Using this fact we can estimate how much difference in lengths influence level of human agreement. We can see results on the figure 3.

The different "modes" of length normalization are presented:

full All summaries are left untouched.

middle The longest summary is cut to the length of the second one: so there is one short summary and two with same length.

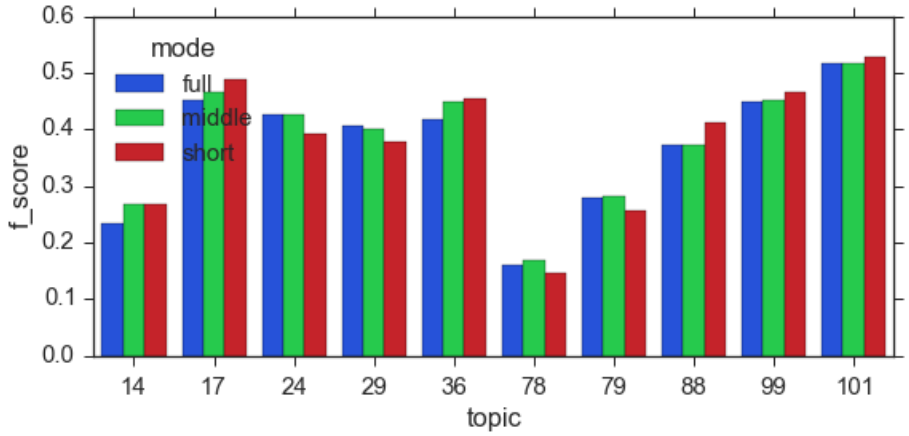


Figure 3: Human agreement with different summary lengths

short All three summaries are cut to the shortest length.

As we can see normalizing the length changes the level of agreement, but not only for few percent. For some topics it improves agreement, but for some it doesn't. So the reason for bad agreement is not the big difference in the lengths of human summaries.

To have insights about what can influence the level of agreement, we can look at the correlation matrix on the figure 4 It presents correlation between human agreement and different aspects of the topics and summaries.

Different features presented on the figure are:

human_length_mean Mean length of human generated summaries

human_length_std Standard deviation of the length of human generated summaries

total The length of the topic — how many tweets are to select from

positive Percent of positive tweets in the topic

negative Percent of negative tweets in the topic

polar Percent of positive and negative tweets in the topic

f_score Human correlation

The figure shows Spearman correlation between each two features. To get the correlation between length of the topic and length of the summary, we need to check the intersection of former's row and latter's column, that gives us 0.62 showing some level of correlation. The opposite use of columns and rows gives us a color representation of this value. As we can see, there is no strong correlation between any of the features and human agreement. Weak negative correlation exists for number of tweets in the topic (the longer the topic is — less chances that people will agree on the summary). Also there is weak positive correlation with percent of sentiment content in the topic. These findings will be investigated more deeply in next chapter.

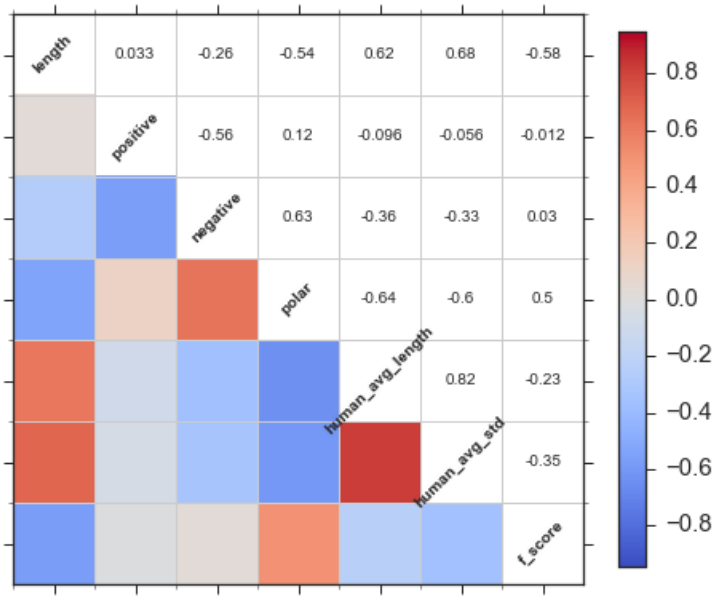


Figure 4: Correlation between human agreement and topic features

4.2 Comparing human and automatic summaries

First lets compare algorithms and human summaries by simple metrics: average length of the tweet and sentiment content of it. On the figure 5 we can see how many words on average are in tweet. The average value for all tweets in the dataset is 14. Human summaries are a little bit longer. Most of the algorithms have similar value, but algorithms relying on average frequency (like SumBasic and Sentiment+Frequency) tend to have shorter tweets. The reason is that such algorithms prefer tweets containing only frequent words and not containing additional not-so-frequent words.

On the figure 6 we can compare how sentiment content of the topic influence sentiment content of the human summaries. As we can see the percent of the sentiment tweets is the same or slightly bigger in summaries compared to source topics. It means that humans have small preferences to sentiment tweets.

4.3 ROUGE for different summarizers

On the figure 7 values of ROUGE-2 presented. ROUGE is a set of metrics, that compare similarity of two summaries. It can work with multiple golden-summaries (in our case there are three golden human generated summaries). ROUGE is precision/recall metric and on the plots the value of combined precision and recall to single f_score is shown.

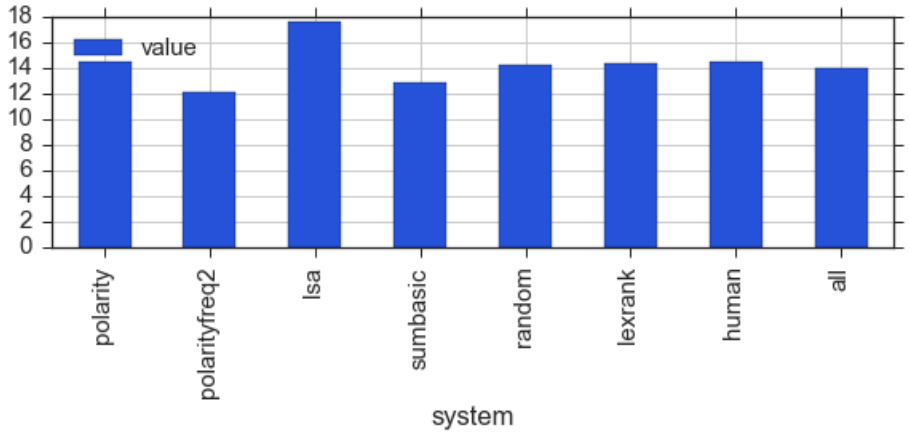


Figure 5: Words per tweet for different algorithm, in human summaries and in source documents

The values are in range 0 to 1 and the bigger the value — the better the summary (closer to golden ones). The ROUGE is calculated for each available length, the system summaries for generated for a particular length and human summaries were cut to the same length.

4.4 Topical content of the summaries

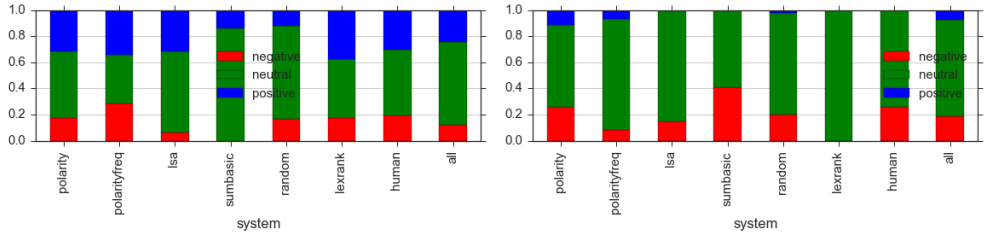
Fraction of Topical Words is a fully automatic metric, that shows the relation of content in the summary to the content in the input document. The values are ranged in [0, 1], with 0 being the worst, showing nothing in common in summary and the source document, and the 1 being the best value meaning everything in the source document is included in the summary. This metric is based on calculating the amount of topical words in both documents: source and summary and comparing them. The topical word is a word that is used more frequently in the dataset compared to big background dataset.

You can see the FoTW values for all topics on the figure 8.

On the X axis the length of the summary is shown, not in absolute values (number of tweets), but relative to the length of the topic. Like summary containing 5 tweets for topic having 100 tweets is shown as 0.05. Lines of different colors represent different algorithms. Three points on each plot represent human summaries. They have one particular length and topical content.

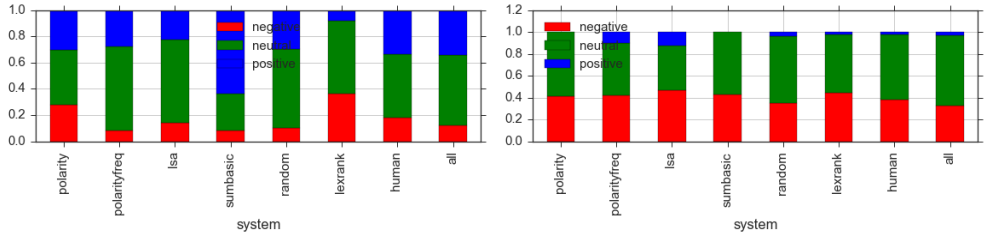
The main trend is the longer the summary is — the more topical content it contains. Longer summary has much more chances to include more different aspects of the topic, than shorter one. What is also noticeable, that this dependency is not linear on the whole range. There is a saturation point: before it the grows is nearly linear, but after reaching it adding more tweets doesn't increase topical content to the summary that much. We can clearly see the saturation point for the topic #24 (Superbowl seats) has a saturation point around 0.3 that is around 15 tweets.

Another feature of the plots is most of the topics reach the level of 80% topical content in at most 20% of the tweets.



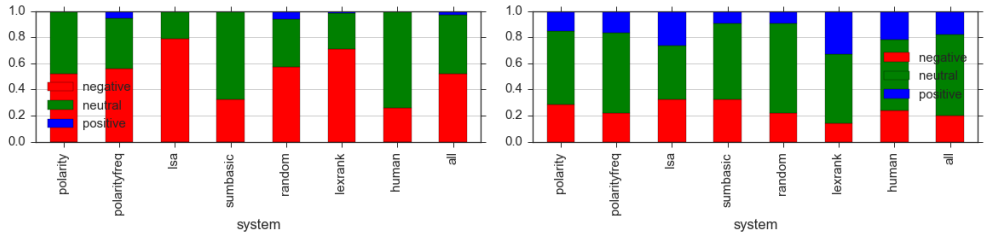
(a) Topic #14

(b) Topic #17



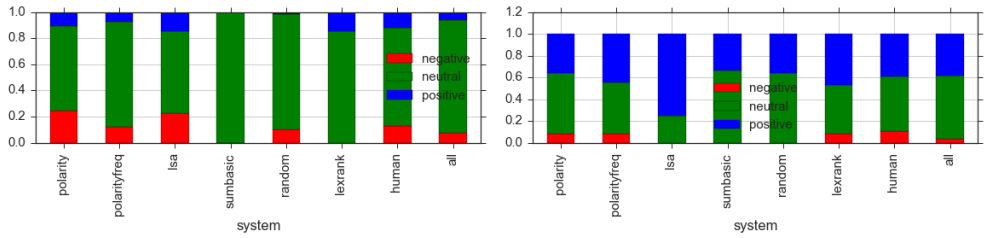
(c) Topic #24

(d) Topic #29



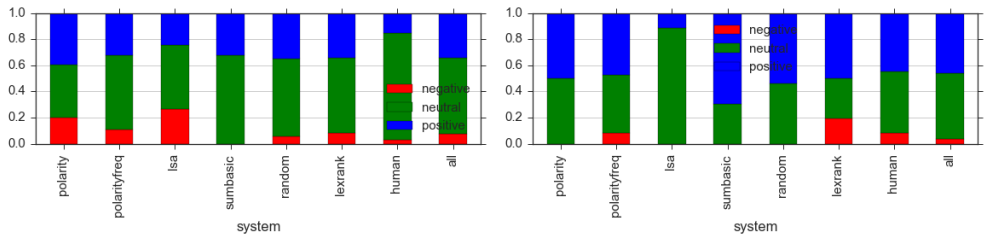
(e) Topic #36

(f) Topic #78



(g) Topic #79

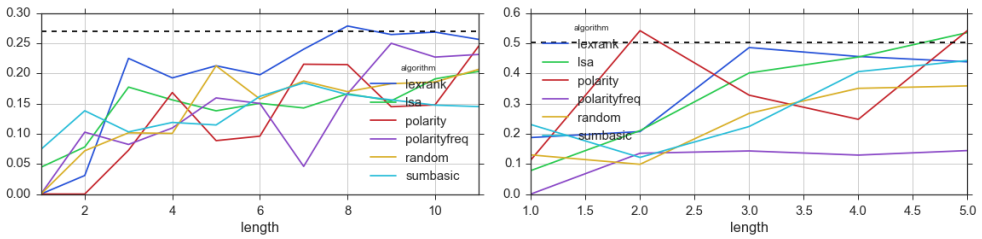
(h) Topic #88



(i) Topic #99

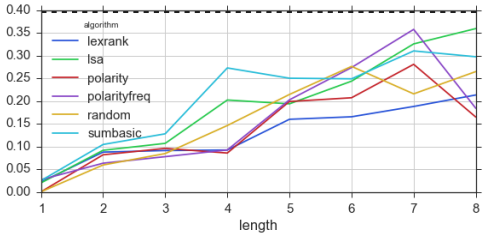
(j) Topic #101

Figure 6: sentiment values for all topics

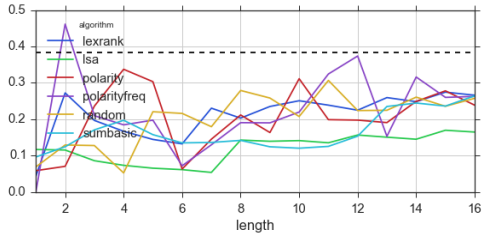


(a) Topic #14

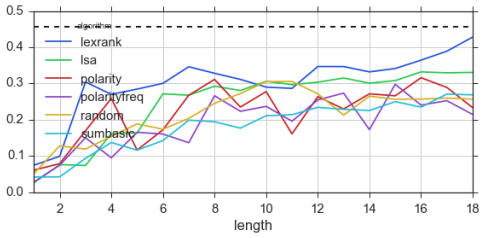
(b) Topic #17



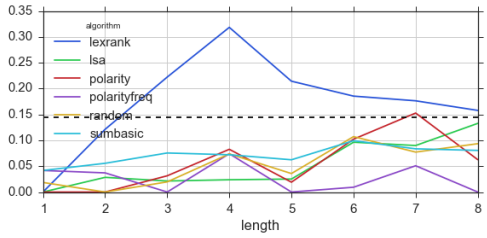
(c) Topic #24



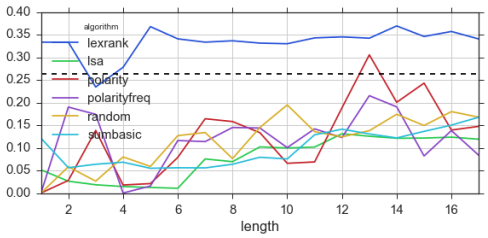
(d) Topic #29



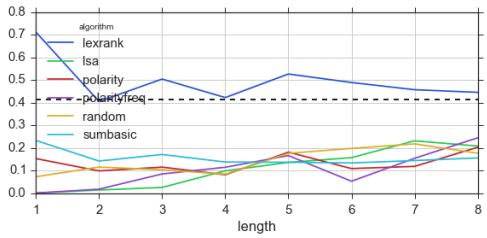
(e) Topic #36



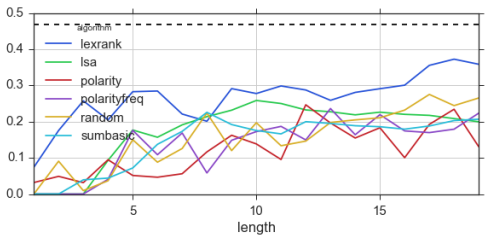
(f) Topic #78



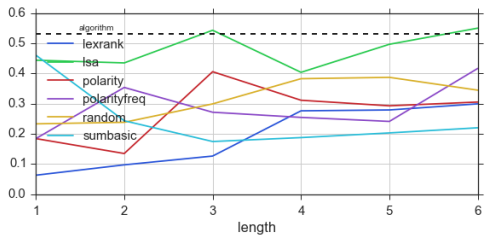
(g) Topic #79



(h) Topic #88

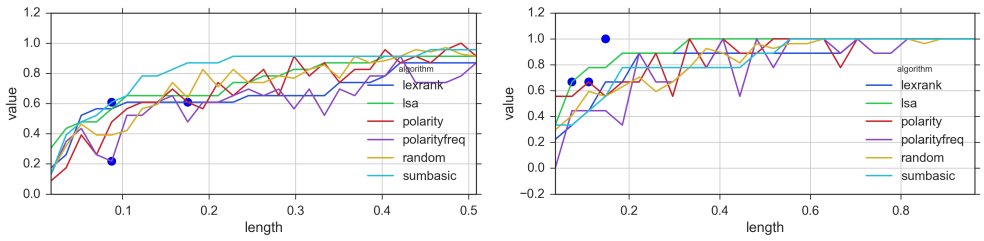


(i) Topic #99



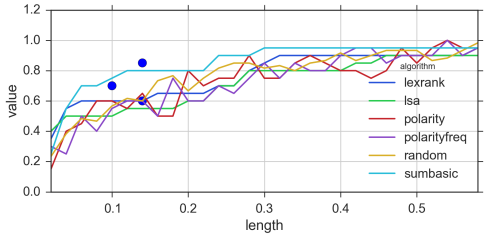
(j) Topic #101

Figure 7: ROUGE values for all topics

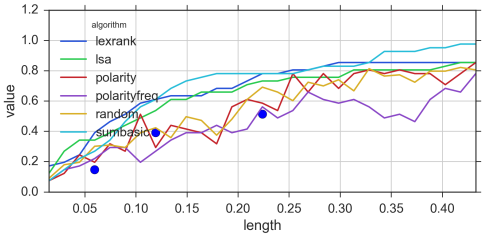


(a) Topic #14

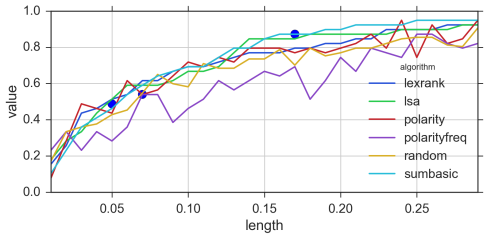
(b) Topic #17



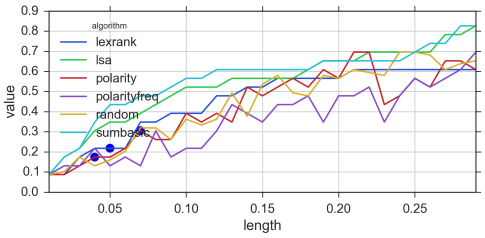
(c) Topic #24



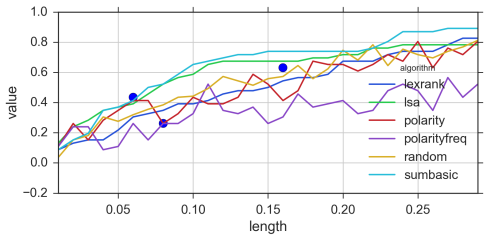
(d) Topic #29



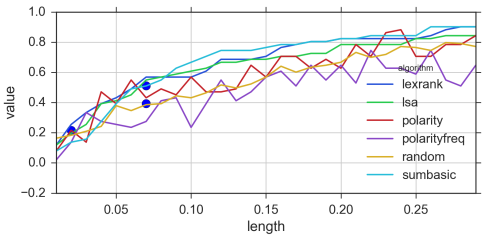
(e) Topic #36



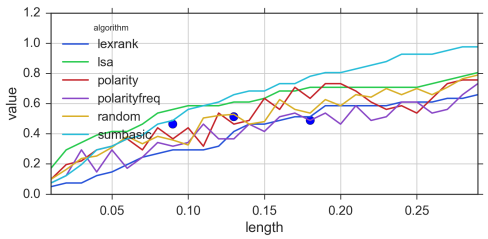
(f) Topic #78



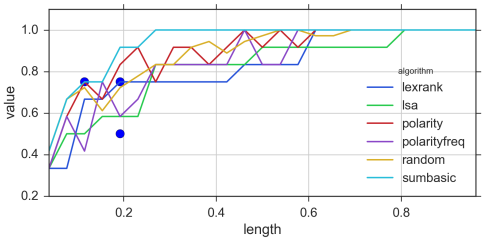
(g) Topic #79



(h) Topic #88



(i) Topic #99



(j) Topic #101

Figure 8: FoTW values for all topics

5 Discussion

Current chapter contains detailed discussion of the obtained results.

We start with the notes about setup. Due to resource limitations and the need to cover both different topics and have several golden summaries for each topic, the scheme *10 topics / 3 humans per topics* was selected. In the retrospect, it is clear that 10 topics are enough for qualitative research and in our case they provide good diversity of the topics by length, sentiment and specificity. Use of three golden summaries, in contrast, is not enough to make strong conclusions: the individual human preferences is visible and sometimes hides more general trends.

The decision not to constrain participants in term of summary length and their understanding about its purpose was required to answer the research questions, but it leads to some inevitable problems. One of them — for some topics difference in length was too big creating difficulties in use of such summaries as ground truth for automatic summarization evaluation. The participants were students having no or little knowledge of the summarization field. Because of this the instructions contained an attempt to achieve both goals: describe what the summary is and simultaneously not to restrict what can be included into it. The study performed on the target audience: journalists or marketing analysts could produce more relevant results, which are different from ours.

The remaining chapter is organized in a following way. First we look only at the human summaries and especially at how human agree for the different topics and try to explain possible outliers from the trends. In the next section the length of the summary is investigated in greater detail. Some observations and ideas about the optimal length of the summary is presented. After that, automatic summarization algorithms is studied, their performance in two kinds of evaluation: intrinsic and extrinsic. The chapter ends with remarks about research limitation.

5.1 Human agreement

Human agreement is an important value defining the upper-bound for the automatic algorithms. As we can see on the figure 4 different topics have different level of human agreement. We discuss some reasons for low level agreement later and now let's concentrate on the ROUGE's sensitivity to the length.

As we discussed in the Section 2.5 the ROUGE works best when the length of summaries is equal. Since in our case the generated summaries can have different length, we compute ROUGE values with shortened versions as well. One could expect the values to improve with all summaries cut to the same length. We can see such trend for many topics, but some shows opposite results (like topics #29 and #79). Actually there is no dependency of such kind: the improvement with shorted summaries is possible, when common n-gramms positioned in the beginning of the summary. Respectively, the drop in similarity could be explained that common n-gramms were in the end of the longer summary and were cut in shorter version. As humans were asked to order tweets in the summary by priority, the explanation of such drop could be that humans don't agree on priorities for topics #29 and #79.

For instance, comparing summaries 2 and 3 for topic #29 about global warming, we can see such differences. The summary 3 is the shortest is this group and its length was taken as base value. All summaries for this topic contain information about *storms*, *more snow in winter* and other visible evidences of global warming. The difference between summaries is in the priorities of less factual information. Both second and third summaries include tweets that global warming *is a religion* and *does not exist*, but in longer summary the positions of such tweets are much lower and they were cut during the normalization.

Looking on the level of agreement, we can see that for some topics it is extremely low, where for others it is around 30–50% what is equal to usual level of agreement reported by the researchers. The low level of agreement for topic on McDonalds food has a good reason: comparing the summaries it is easy to spot, that all summaries contain different tweets both semantically and textually. All summaries are just mix of opinions, like *New McDonalds outmeal is great!* and random facts *mcdonald's reports 5.3 percent jan sales growth*. You can find all human summaries in the Appendix C. The topic about McDonalds differs much from all others — it is too broad and doesn't contain specific idea to build summary around. Participants could select almost any tweet, and it would be about "McDonalds food". This broadness has more consequences: like fluctuation in a human summaries' length trend, that we discuss in the next section.

Continuing analyzing low human agreement lets look at topic #14. There are several unusual things with the topic about Release of the Rite. It has an average amount of tweets in the source document — 57, the theme is very specific and bounded, but yet the human agreement as one of the lowest. It is as low as for topic about McDonalds, which has 100 tweets and discusses all kind of opinions about McDonalds food. The question is whether it is a problem in evaluation itself or three participants managed to create totally different summaries from short bounded topic.

Another interesting feature of human summaries for this topic is the relation of topical content with respect to the length of the summary. The usual trend is the longer the summary is — the more topical content it includes. We can see on the figure 8 that most of the human summaries, as well as summaries generated automatically, respect the trend. But for topic #14 there are two summaries of the same length with dramatical difference in topical content (0.2 vs 0.6).

To solve these issues lets look closer on the summaries themselves.

../data/summaries/golden/1/14.txt

```
why are people giving shitty reviews for "the rite" i think it
    was well written directed and the actors were stunning simply
     amazing .
i saw the rite last night and it scared me so bad i had
    nightmares :( but it was a really good movie .
that movie therite is crazy as hell it got me shook its a good
    movie.
just watched the rite spoiler alert: jesus exists .
for the record: "the rite" was long and dry .
have yall seen the commercials for that movie the rite that movie
     looks crazy
```

../data/summaries/golden/2/14.txt

```
that movie therite is crazy as hell it got me shook its a good
```

```
    movie.
went to the midnight showing last night @ 12:01 am and saw
    anthony hopkins movie "rite" most awesome yes indeed a super
    - must see .
the rite - movie review: i have always been a fan of any movie
    that deals with the battle between good and .
sir anthony hopkins tops the box office with the rite: what a
    slow weekend at the box office not much came out .
the rite was good shoulda seen little fockers though.
anthony hopkins the rite tops friday box office with $5 3 million
     (hollywood reporter) .
saw the rite today with stephen excellent movie .
holy anthony hopkins just saw the rite awesome movie reminded me
    of good ol' hannibal lecter days .
 the rite was quite good anthony hopkins can pull off "creepy"
     very well here i would refer to him as "father hannibal" :-)
     .
anthony hopkins takes 'the rite' to top of box office - .
'the rite' captures friday box office: thriller makes $5 3
    million by mawuse ziegbe anthony hopkins and mart
```

../data/summaries/golden/3/14.txt

```
hopkins takes 'the rite' to top of box office (ap) - ap - the
    anthony hopkins horror film the rite topped the b .
new film the rite is based on the training of a real priest who
    says the film is about faith catholic exorcist.
chat w/ director stars of "the rite" anthony hopkins is like the
    english grandfather i never had .
the rite ; good movie .
bishop exorcist praise new exorcism movie - catholic culture:
    washington postbishop exorcist praise new exorci .
the movie news channel: box office: anthony hopkins has 'the rite
    ' stuff for no 1 friday movies
```

Comparing first summary that has topical content of 0.21 to third with topical content 0.61 for the same 5 tweets, we can see that indeed there is difference in included content. First summary includes almost only sentiment tweets showing expressions about the movie. Phrases like *excellent movie*, *good movie* or *long and dry* are present almost in all sentences. But in order to be sentiment-rich some content ought to be excluded: like there is even no single mention of actor playing leading part — Anthony Hopkins. Third summary is almost totally opposite — it includes many facts about the movie: who played leading role, that it is about exorcist or that Anthony Hopkins has British accent.

The second summary is a combination of both worlds: it contains many facts about the movie, but also has some tweets that doesn't add topical content, only sentiment one. To achieve this more tweets are required so the length is 10 tweets covering the same 60% of topical content as summary #3. As we can see the low human agreement is due to clearly different visions of humans created summary 1 and 2 about what should be covered.

These two topics (#78 and #14) are examples of two main reasons for human disagreement in our study: the broadness of the topic and different relation to sentiment content in the summary. Very broad topics have too much content that can be included in the summary without clear idea which one is more important than other. Another reason is different percent of sentiment content in human summaries. For the same length a

limited amount of content could be included so more topical results leads to drop in less sentiment.

5.2 Length of the summary

The defining feature of this research project is the study of summary length. Obtained data shows that usual approach to summarization and to evaluation algorithm, when "one size fits all" doesn't work well.

When people are permitted to generate summaries of any length, they indeed generate them with different length. The same person prefers different lengths of the summary for different topics and different persons prefer different length for the same topic. The range of the lengths varies dramatically: there are only 3 tweets in the shortest summary, and 19 — in the longest.

First we discuss the deviation of summaries' lengths for topics and after that the individual preferences of the people. Mean lengths of human summaries for each topic are shown in the table 2, on the figure 9 you can see the plot of dependency of the length of the summary to the length of the topic.

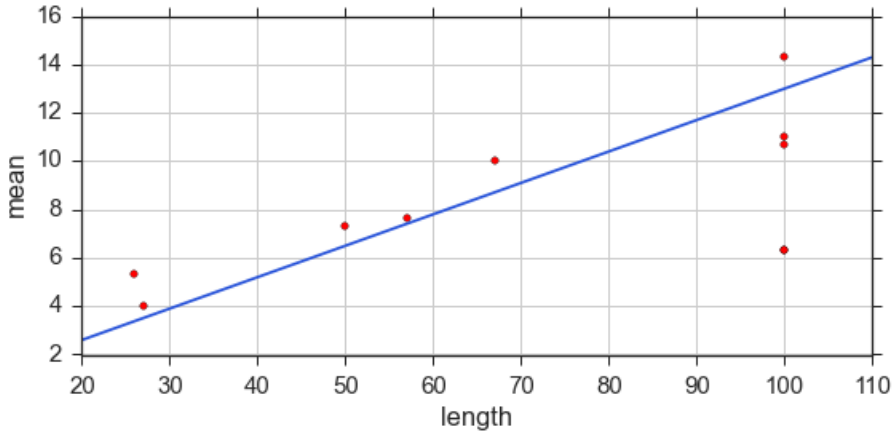


Figure 9: Dependency of summary mean length from # tweets in the topic

The trend "the longer the source document is — the longer the summary" is clearly visible. Only two topics #78 and #88 fall out of the trend. They are shorter than expected. It looks like the topic #78 about McDonalds is too long and broad, there was no central idea to build summary around, so humans consider most of the information irrelevant. Second topic #88 about awards of the movie The King's Speech is strange for another reason: there was a duplication of the content, so the length of topic doesn't reflect the amount of topical content in it.

But it is much better to talk about the length of the summary not in the absolute values but rather in percentage of the length of the topic. From the table we can see that average length summary is around 10–15% of the length of the topic (not counting two strange topics we already discussed and the topic #101 covering almost 20%). These results are consistent with the results reported by the researchers [56] studying length of the general-purpose summary. Their results concluded that optimal summary length is between 10 and 20%.

Similar to discussed in previous section for topic #14 situation with topical content

is for the topic #101. On the figure 8 three summaries form a noticeable figure: first summary with length 6 has 0.5 topic coverage, second with the same length has topic coverage of 0.75 and third with only four tweets covers the same 75% of the topic.

../data/summaries/golden/1/101.txt

```
just finished watching black swan so twisted as it shows the
    mental strain it takes to embrace a role natalie portman was
    incredible .
 black swan is such a great psychological movie natalie portman
     performance was fab a lot of sexual content .
finally saw the black swan loved it natalie portman is very
    believable as an adult with lots of stuffed animals in her
    bedroom .
the black swan: artistically perfect portman's acting is
    brilliant the plot is shallow but it doesn't affect the
    overwhelming experience .
black swan good movie but very melodramatic natalie portman
    phenominal shoe in for best actress.
yes rt : natalie portman is nominated for best actress for
    blackswan teamblackswan
```

../data/summaries/golden/2/101.txt

```
black swan is amazing natalie portman is unbelievable.
natalie portman better win an oscar for best actress for "black
    swan" that movie is amazing just saw it for the 3rd time .
just finished watching black swan so twisted as it shows the
    mental strain it takes to embrace a role natalie portman was
    incredible .
yes rt : natalie portman is nominated for best actress for
    blackswan teamblackswan.
i was spellbound by black swan last night - budget $13m gross
    $84m & growing natalie was amazing and hope she sweeps up
    every nomination .
i don't know much about ballet but from the first scene of the
    black swan alone i'm already convinced natalie portman isn't
    all good at it
```

../data/summaries/golden/3/101.txt

```
the black swan: artistically perfect portman's acting is
    brilliant the plot is shallow but it doesn't affect the
    overwhelming experience .
wowsa black swan quite a movie very intense and stirring i hope
    natalie portman wins an oscar for that role.
natalie portman better win an oscar for best actress for "black
    swan" that movie is amazing just saw it for the 3rd time .
my sinchew: oscar nominee natalie portman thrills in 'black swan
    ': afptv caught up with oscar nominee natalie po
```

By analyzing the summaries we can see that basically all summaries have the same topical content, mentioning Natalie Portman, the movie "Black Swan" about the ballet and her Oscar nominations for best actress. The difference in topical content for the first summary can be explained by failure of the evaluation tool to identify that topic about Oscar nomination was actually covered (just without using the word *oscar*). The difference in length is explained by the proportion of sentiment content users was willing

to include. First two summaries include extra tweets not adding topical content, just emotions using words like *amazing, incredible*.

Analyzing one more topic #99, where we can see that all three summaries have the same topical content for the different length, we can see that the additional length is because of the sentiment tweets included. For instance, the first summary contains much more tweets about which advertisements were the *funniest* or *best*.

Important fact to notice is that for most topics 15-20% of the tweets could cover 80% percent of the topical content of the topic. It gives a reasonable upper-bound for length, since the longest summary produced by a human and no bigger than 20% and it covers most of the content.

It is noticeable that there are differences in the people's preferences to the summary length. Three longest summaries were produced by the same person. Similarly, the shortest summaries for three topics were generated by the same person. This fact adds challenges to finding optimal length for the summary as people do not agree between themselves on good length.

5.3 Comparing the algorithms

First it needs to be mentioned that it is impossible to select "the ultimate" summarization algorithm. Different metrics, like ROUGE and FoTW, measures different aspects of summaries. Usually algorithm performing well in one metric performs worse in other. We can see, that SumBasic while showing good results for FoTW, is not that good in similarity with human summaries.

Another problem with identifying "the best" summarization algorithm is the performance on short or narrow topics. All algorithms perform almost identically. It is true especially for the FoTW metric, where for some topics no algorithm is significantly better than random algorithm (topics #17, #36) and for many only SumBasic and LSA show small advantage over random algorithm (topics #79, #88). The possible explanation could be in good filtering of tweets in the source document. If all of them contain relevant information and there is not so much duplication, even the random algorithm performs reasonably well.

Similar situation is with ROUGE: short topics don't have clearly visible winner, where for long topics with many input tweets (topics 78, 79, 88, 99) the LexRank clearly works best (see figure 7). The f_score for LexRank could be up to several times bigger than for others algorithms.

Investigating the figures 7 and 8 with ROUGE and FoTW results, we can see that the use of the sentiment feature exclusively does not create good summaries. The results for Sentiment summarizer are not statistically different from using random algorithm for nearly all topics. It contrasts to the findings from the original article that report satisfactory results in case of product reviews. As we discussed in previous section, different users tend to include different amount of sentiment tweets, but still most of them concentrate on the topical content more. From the figure 6 we can see that in humans summaries percentage of sentiment tweets is very close to such percentage in the source documents. For some topics humans emphasise primarily emotion, but not much. Thus, summarizer that ignores content and concentrates on the sentiment is neither close to human summaries, nor is rich with topical content. Often the FoTW results are even worse than random because adding sentiment content dilutes relevant information. As we saw in

human summaries, the sentiment-rich summary often contains less topical content for the same length.

Sentiment algorithm that optimizes both sentiment and topical words produces much better results, but is still not better than any other algorithm. By topical content it is close to the SumBasic (but not as good) and works much better for ROUGE than SumBasic.

Consider the four-tweets summaries produced for topic #99 (SuperBowl commercials): LexRank, Sentiment, Sentiment+Frequency and Sumbasic:

../data/summaries/lexrank/4/99.txt

```
so far doritos commercial with dog best .
brasky says the doritos commercial wins the superbowl .
the superbowl commercials have launched: as almost everyone in
    north america is aware the .
super bowl commercials .
```

../data/summaries/polarity/4/99.txt

```
5 best superbowl commercials 2011: good game last night some good
     ads too here are the top 5 superbow
that last doritos commercial got my vote so far funniest
    superbowl commercial
rofl over vw passat commercial superbowl
eminems super bowl commercial: behind the scenes:
```

../data/summaries/polarityfreq/4/99.txt

```
doritos with the commercial redemption superbowl
i estimate that doritos has to sell 1 5 million large bags to
    recoup what they spend on superbowl commercials
had some inproductive minutes with this brilliant volkswagen
    commercial aired during the superbowl trekkie in
ok the superbowl audi commercial gets a b+ the others f--
```

../data/summaries/sumbasic/4/99.txt

```
super bowl commercials .
the sauna - 2011 doritos superbowl commercial ad via .
latest news: volkswagen darth vader super bowl xlv commercial .
like that volkswagon commercial superbowl.
```

Here we see summaries with different sentiment/topical balance. SumBasic contains no sentiment at all, just some names of the companies produced advertisements. Sentiment summaries contain much more opinions about the advertisements and actually look more useful than SumBasic for this topic. Here we see, mentioned in previous chapter, the tendency of SumBasic to select shorter tweets, containing only topical words, without additional low-frequent words, lowering average frequency of the tweet.

ROUGE ranked them in the following order (from best to worst): LexRank, Sentiment, Sentiment+Frequency, SumBasic. The FoTW ranks are in opposite order: SumBasic is the best algorithm and LexRank is the worst. In our personal opinion, the sentiment based algorithms behave relatively well, so the question of proper evaluation could be raised and should be investigated in future research.

Another noticeable drawback of particular sentiment summarization is low stability of results compared to other algorithms. Even small change in length of the summary

sometimes results in big change in performance. The reason is not sentiment summarization itself, but rather that the discussed algorithms are formulated as an optimization problem. This process is not deterministic and can produce different summaries between runs. The summaries are equal in terms of optimization function: sentiment or sentiment+frequency pair, but could have dramatic difference in performance by other aspects.

5.4 Limitations

The proposed approach has some limitations due to the lack of time and resources. The most important are following:

5.4.1 Use of simple NLP techniques for Twitter

As we discussed in the Chapter 2 the language used in the microblogs is unique. In current research no advanced techniques were applied when parsing and processing tweets. Use of state-of-the-art algorithms could improve results. In particular, clever processing of hashtags, especially ones that are composed from several words, can improve the quality of both summary generation and its evaluation. As hashtags are meant to highlight important part of the tweet, proper parsing will add more topical words to the tweet. For instance, the tweet

> Premiere (via GetGlue) #TheRite

contains hashtag *TheRite*. Without hashtag processing this tweet is considered not containing the main phrase of the topic *The Rite*.

5.4.2 Use of simple sentiment feature

As we discussed in the Chapter 2.4 the sentiment analysis of the text is rapidly developing area of NLP. State of art sentiment analysis systems go beyond simple polarity or polarity-intensity and polarity-subjectivity pairs. Complex models like Joint Sentiment Topic are built. Such models when applied identify sentiment not just for sentences, but for topics, entities etc.

Use of advanced sentiment systems is beneficial in several ways. First, they tend to have lower error rate in predicting sentiment of the sentence and as a result improve quality of systems incorporating them. Next, they could provide capability to distinguish between sentiment in the sentence and topic targeted sentiment. Consider the example:

> Oh no, bad day! It is raining, just when I am in Disneyland

Although the overall polarity of the tweet is clearly negative, the sentiment addresses the particular day, rather than Disneyland. So for topic about Disneyland, use of JST-based systems allows marking such sentence as neutral in respect to the topic.

5.4.3 Golden summaries of non-fixed length

We have already discussed available options of dealing with the length of the summary: restrict humans to particular length or allow them to use the number of tweets they believe is optimal for particular topic. Only second approach provides required information for answering research questions but it leads to a problem. The ROUGE metric is sensitive to the length and dramatical difference in lengths weakens produced results and corresponding discussion.

The most accurate approach would have been to ask participants to generate all versions of the summary for a range of lengths, but it is very time and resource consuming

method. To overcome this restriction, participants were asked to prioritize tweets with regard to their importance for the summary.

The idea is to make shorter versions of the human summaries by cutting to the required number of tweets. As last included tweets cover additional details and rather than essence of the topic, the approach looks reasonable. But it does not necessarily mean that short-cut summary will be identical to the one which participants would have generated for particular length in the first place.

6 Conclusions and future work

In this chapter the conclusions and answers to research questions are given. The goal of the project was to conduct the qualitative research, so answers are given in form of hypotheses and suggestions. To prove or disprove a particular result a proper quantitative research should be performed.

6.1 How human summaries for a topic are different from each other?

The human summaries are a basis for developing and analyzing any summarization algorithm. Three summaries for ten topics were deeply investigated and compared. For most topics human agreement is high. It appears that two main reasons for human disagreement in our setup are personal preferences for sentiment content and broadness of the topic. Although in average, percent of polar tweets is approximately the same as the one in the source document, there are clearly visible biases between participants. In extreme cases, there are summaries without the sentiment content at all or consisting with only polar tweets. In similar way, if the topic is too broad (not necessarily long), there are several versions of legitimate summaries, containing totally different tweets.

6.2 How to decide on proper length of the summary?

The length of the summary is an important factor and different topics require different summary length. Length of human summaries for some topics vary dramatically. When studying length much more productive is to talk about percents rather than precise number of words, sentences or tweets. On average humans create summaries around 15% of the length of the topic (the shortest is 6%, the longest is 20%). In addition, for most topics 15–20%-long summary covers 80% percent of the topical content of the source, so there should be a good reason to create summaries longer than that, as it is rather long summary. To determine the saturation point in topical content even Random summarizer can be used. It allows estimating how fast the topical content grows with respect to increase in summary length.

If purpose of summarization also requires the sentiment content, additional length should be reserved. As a result, 10–20% percent of the length of the topic looks like optimal starting point for evaluating other criteria. It is important that this result is coherent with results of previous researches for non-microblog summarization. The pure topical summaries might require less tweets to cover all desired content, but additional use of sentiment feature requires summaries to be longer.

6.3 How do sentiment-based summaries compare to non-sentiment ones?

Across Future Work section of many scientific articles about microblogs or Twitter the idea to add sentiment feature is very popular. It has some successful usage in different NLP domains (such as ranking of microblogs), but the questions is if there is a reason to think about sentiment in the summarization of microblogs. The answer is as usual — it depends. In particular it depends on the purpose of the summary, what need it should

solve for the consumers. As we saw in case of general-purpose summary without a specific goal, adding sentiment content makes it behave worse in both areas: topical content and similarity to human summaries. But the need for sentiment summaries exists: some golden summaries were full of sentiment rather than topical content.

This research is not meant to answer what particular sentiment integration method works best, it doesn't have enough data for that, but we can suggest that for studying sentiment summarization in microblogs, it should be conducted with the proper purpose factors in mind.

6.4 Future work

The research can be continued in several directions.

Suggestions for qualitative research

The first is to plan another qualitative research based on current results and aiming to propose more targeted hypothesis. The first change is to use more participants to generate more golden summaries for a topic. It will provide more sustainable information about trends and personal biases. The background of participants also can be considered: target consumers of the summary could produce more reliable results. We also propose to add an additional step and make a survey of participants after summary generation about reasons for particular length and sentiment content. The intent of the summary also can be investigated that way. Completely another idea for qualitative research of sentiment summarization is to study evaluation metrics, recruiting people to perform manual evaluation of summary. Given this information we can exclude possibility that existing metrics are not capable of ranking sentiment summaries correctly.

Suggestions for quantitative research

Another direction of future work is to perform quantitative research to check conclusions. The first we can suggest is to perform a study devoted to sentiment summarization. It means to find specific topics (like movie or product reviews, political opinions) and more importantly have sentiment in the purpose factors of the summarization. Important information that should be obtained is relative importance of topical content to sentiment content. There are different ways they could be included and their contribution doesn't necessarily should be equal. And the last suggestion for such kind of study is to use long enough summaries to capture all variety of topical and sentiment content and long enough topics to see the real difference in algorithms' performance.

Bibliography

[1] Mcdonald, R. 2007. A Study of Global Inference Algorithms in Multi-Document Summarization. *Proceedings of the European Conference on Information Retrieval*, 557–564.

[2] Xu, W., Grishman, R., Meyers, A., & Ritter, A. 2013. A Preliminary Study of Tweet Summarization using Information Extraction. *Naacl 2013*, (Lasm), 20–29.

[3] Inouye, D. & Kalita, J. K. 2011. Comparing twitter summarization algorithms for multiple post summaries. *Proceedings - 2011 IEEE International Conference on Privacy, Security, Risk and Trust and IEEE International Conference on Social Computing, PASSAT/SocialCom 2011*, 298–306.

[4] Rosa, K. D., Shah, R., Lin, B., Gershman, A., & Frederking, R. 2011. Topical Clustering of Tweets. *SIGIR 3rd Workshop on Social Web Search and Mining*, cited 2.

[5] Miles, E. 2011. Information Search and Retrieval in Microblogs. *Journal of the American Society for Information Science and Technology*, 62(6), 996–1008.

[6] Teevan, J., Ramage, D., & Morris, M. R. 2011. #TwitterSearch: a comparison of microblog search and web search. *Proceedings of the fourth ACM international conference on Web search and data mining - WSDM '11*, 35.

[7] Twitter Terms of Sevice. `https://dev.twitter.com/overview/terms`. [Online; accessed 6-May-2015].

[8] Lloret, E. & Palomar, M. 2012. Text summarisation in progress: A literature review. *Artificial Intelligence Review*, 37(1), 1–41.

[9] Jones, K. S. 1998. Automatic summarising: factors and directions. 1–21.

[10] Kolcz, A., Prabakarmurthi, V., & Kalita, J. 2001. Summarization as feature selection for text categorization. *Proceedings of the tenth international conference on Information and knowledge management - CIKM'01*, 365.

[11] Witbrock, M. J. & Mittal, V. O. 1999. Ultra-summarization (poster abstract): a statistical approach to generating highly condensed non-extractive summaries. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 315–316.

[12] Kwak, H., Lee, C., Park, H., & Moon, S. 2010. What is Twitter , a Social Network or a News Media? *The International World Wide Web Conference Committee (IW3C2)*, 1–10.

[13] Foster, J., Cetinoglu, O., Wagner, J., Le Roux, J., Nivre, J., Hogan, D., & van Genabith, J. 2011. From News to Comment: Resources and Benchmarks for Parsing

the Language of Web 2.0. *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011): November 08-13 2011 Chiang Mai, Thailand*, Chiang Mai, 893–901.

[14] Bo, H. & Baldwin, T. 2011. Lexical Normalisation of Short Text Messages: Makn Sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, 368–378.

[15] Gao, Q., Abel, F., Houben, G. J., & Yu, Y. 2012. A comparative study of users' microblogging behavior on Sina Weibo and Twitter. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7379 LNCS, 88–101.

[16] Wong, F., Sen, S., & Chiang, M. 2012. Why watching movie tweets won't tell the whole story? *Proceedings of the 2012 ACM workshop . . .* , 61–66.

[17] Sharifi, B., Hutton, M. A., & Kalita, J. K. 2010. Experiments in microblog summarization. *Proceedings - SocialCom 2010: 2nd IEEE International Conference on Social Computing, PASSAT 2010: 2nd IEEE International Conference on Privacy, Security, Risk and Trust*, 49–56.

[18] Sharifi, B., Hutton, M.-a., & Kalita, J. 2010. Summarizing Microblogs Automatically. *Computational Linguistics*, 15(June), 685–688.

[19] Nenkova, a. & Vanderwende, L. 2005. The impact of frequency on summarization. *Microsoft Research Redmond Washington Tech Rep MSRTR2005101*, (MSR-TR-2005-101).

[20] Gong, Y. & Liu, X. 2001. Generic Text Summarization Using Relevance Measure and Latent Semantic Snalysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01*, 19–25.

[21] Steinberger, J. & Ježek, K. 2004. Using Latent Semantic Analysis in Text Summarization. *In Proceedings of ISIM 2004*, 93—-100.

[22] Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. 2013. *Handbook of latent semantic analysis*. Psychology Press.

[23] Erkan, G. & Radev, D. R. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22, 457–479.

[24] Mackie, S., Mccreadie, R., Macdonald, C., & Ounis, I. 2014. Comparing Algorithms for Microblog Summarisation. *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, 8685, 153–159.

[25] Liu, B. 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1), 1–167.

[26] O'Connor, B., Balasubramanyan, R., Routledge, B. R., & Smith, N. a. 2010. From tweets to polls: Linking text sentiment to public opinion time series. *From tweets to polls: Linking text sentiment to public opinion time series*, 122–129.

[27] Pak, A. & Paroubek, P. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Lrec*, 1320–1326.

[28] Go, A., Bhayani, R., & Huang, L. 2009. Twitter Sentiment Classification using Distant Supervision. *Processing*, 150(12), 1–6.

[29] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. 2011. Sentiment analysis of Twitter data. *Association for Computational Linguistics*, (June), 30–38.

[30] Lin, C., Road, N. P., & Ex, E. 2009. Joint Sentiment / Topic Model for Sentiment Analysis. *Cikm*, 375–384.

[31] Saif, H., He, Y., & Alani, H. 2012. Alleviating data sparsity for twitter sentiment analysis. *CEUR Workshop Proceedings*, 838, 2–9.

[32] Abbasi, A., Hassan, A., & Dhar, M. 2014. Benchmarking Twitter Sentiment Analysis Tools. *Lrec-Conf.Org*, 823–829.

[33] Kim, H. & Ganesan, K. 2011. Comprehensive review of opinion summarization. *Illinois Environment for . . .* , 1–30.

[34] Lu, B., Ott, M., Cardie, C., & Tsou, B. K. 2011. Multi-aspect sentiment analysis with topic models. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 81–88.

[35] Gupta, V. & Lehal, G. S. 2010. A Survey of Text Summarization Extractive techniques. In *Journal of Emerging Technologies in Web Intelligence*, volume 2, 258–268.

[36] Lerman, K., Lerman, K., Blair-Goldensohn, S., Blair-Goldensohn, S., McDonald, R., & McDonald, R. 2009. Sentiment summarization: Evaluating and learning user preferences. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 514–522.

[37] Mani, I. 2001. Summarization Evaluation: An Overview. In *Proceedings of the NTCIR Workshop*.

[38] Lin, C. Y. 2004. Rouge: A package for automatic evaluation of summaries. *Proceedings of the workshop on text summarization branches out (WAS 2004)*, (1), 25–26.

[39] Nenkova, A., Passonneau, R., & McKeown, K. 2007. The Pyramid Method. *ACM Transactions on Speech and Language Processing*, 4(2), 4–es.

[40] Louis, A. & Nenkova, A. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2), 267–300.

[41] Mackie, S., Mccreadie, R., Macdonald, C., & Ounis, I. 2014. On Choosing an Effective Automatic Evaluation Metric for Microblog Summarisation. In *Proceedings of the 5th Information Interaction in Context Symposium*, 115–124. ACM.

[42] Lin, C.-Y. 2004. Looking for a few good metrics: Automatic summarization evaluation - how many samples are enough. *Proceedings of the NTCIR Workshop*, (April 2003), 1765–1776.

[43] McCreadie, R., Soboroff, I., Lin, J., Macdonald, C., Ounis, I., & McCullough, D. 2012. On building a reusable Twitter corpus. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '12*, 1113.

[44] Bouma, L. & De Rijke, M. 2006. Specificity helps text classification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3936 LNCS, 539–542.

[45] sumy 0.3.0: Python package index. `https://pypi.python.org/pypi/sumy`. [Online; accessed 18-May-2015].

[46] Rios, L. M. & Sahinidis, N. V. 2013. Derivative-free optimization: A review of algorithms and comparison of software implementations. In *Journal of Global Optimization*, volume 56, 1247–1293.

[47] Smedt, T. D. & Daelemans, W. 2012. Pattern for Python. *Journal of Machine Learning Research*, 13, 2063–2067.

[48] Pang, B. & Lee, L. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts.

[49] Bird, S. 2006. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, number COLING-ACL '06, 69–72. Association for Computational Linguistics.

[50] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. 2012. Scikit-learn: Machine Learning in Python. *... of Machine Learning ...*, 12, 2825–2830.

[51] McKinney, W. 2011. pandas: a Foundational Python Library for Data Analysis and Statistics. In *Python for High Performance and Scientific Computing*, 1–9.

[52] Tosi, S. 2009. Matplotlib for Python Developers.

[53] ROUGE: Recall-Oriented Understudy of Gisting Evaluation. `http://www.berouge.com/Pages/default.aspx`. [Online; accessed 8-May-2015].

[54] ROUGE 2.0 - Java Package For Evaluation Of Summarization Tasks With Updated ROUGE Measures. `http://kavita-ganesan.com/content/rouge-2.0`. [Online; accessed 8-May-2015].

[55] SIMetrix: Summary Input similarity Metrics (Version 1). `http://homepages.inf.ed.ac.uk/alouis/IEval2.html`. [Online; accessed 8-May-2015].

[56] Jing, H., Barzilay, R., McKeown, K. R., & Elhadad, M. 1998. Summarization evaluation methods: Experiments and analysis. *AAAI Symposium on Intelligent Summarization*, 51–59.

A User instructions for creating golden summaries

Instructions

I am Nataliia Uvarova, Master student at Gjovik University College, writing Master thesis in the field of Information Retrieval.

The topic of the research is Summarization of Microblog data. You are hereby asked to help the research by producing a golden human-generated summaries for the given topics. It should take from 30 minutes to 1 hour of work, depending on your speed and what topics you will get. Each person produces summaries for three topic from nine available. The topics are assigned randomly.

The summary of the topic — is a subset of all tweets posted on the topic. In an ideal situation, the summary covers every aspect of the topic, as presented in the complete set of tweets. Each tweet included in the summary should therefore add some piece of information about the topic that is not already covered in the summary. The final summary may consist of a few core tweets, that cover the generics of the topic, and some that adds some relevant detail to this core summary.

What you should do:

1. After you have read these Instructions and know what is expected from you, press the green button on the top right corner. You will then be redirected to the page with the first topic.
2. On the top of the page you will see the topic name. The presented tweets and the produced summary will be about this topic. There are different types of topics. The topic can be specific — for instance about event like Oscar nominations or be broad — like opinions about fast-food chains in February 2011.
3. You will see the list of tweets on the left — it is a list of all available tweets about particular event. Empty list on the right — it is where you should produce your summary.
4. You can move tweets between lists pressing ‘‘`select`’’/‘‘`unselect`’’ buttons on the desired tweets.
5. There are quite a few tweets available, and your task is to select a group of them which *together* form a summary of all content. There is no right or wrong answers — select tweets that you believe represent the topic the best.
6. The length of the summary should be around 5 to 10 tweets. Please try not to make it too short or to long, except if you really fill that you cannot add/delete tweets.
7. Please organize the tweets in the summary in order of increasing importance: the first tweet represent the essence of the topic, next tweets add more information and details. You can use ‘‘`up`’’/‘‘`down`’’ buttons for this.
8. To simplify work with long list of tweets there is a ‘‘`hide`’’ button on each unselected tweet. You can hide tweets that you sure, should not be present in the summary. If you accidentally hide tweet, use “Show hidden items” button to show all hidden tweets.

Then you can restore any of them.

9. You can press “Read instructions” button at any time, don’t worry, your progress will be saved.
10. When you are satisfied with the summary, press the ‘‘`Submit and next`’’ button, your summary will be saved on the server and new topic will be presented.
11. Repeat the same procedure for new topic. There will be around three topics.

Thank you for participating!

If you are interested in knowing more about results of this research project, please contact me at nataliia.uvarova@gmail.com

B The screenshots of the tool for human summaries generation

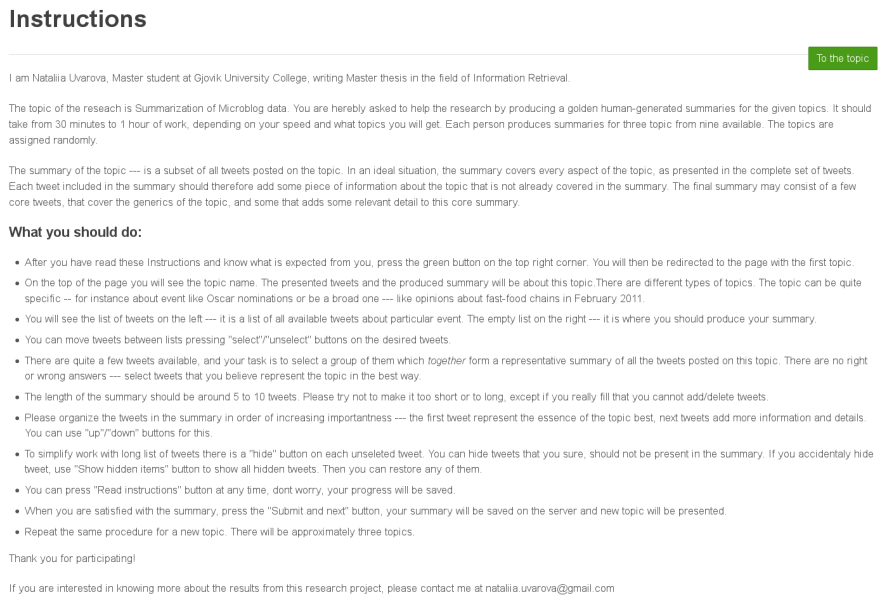


Figure 10: User interface of the tool: Instructions page

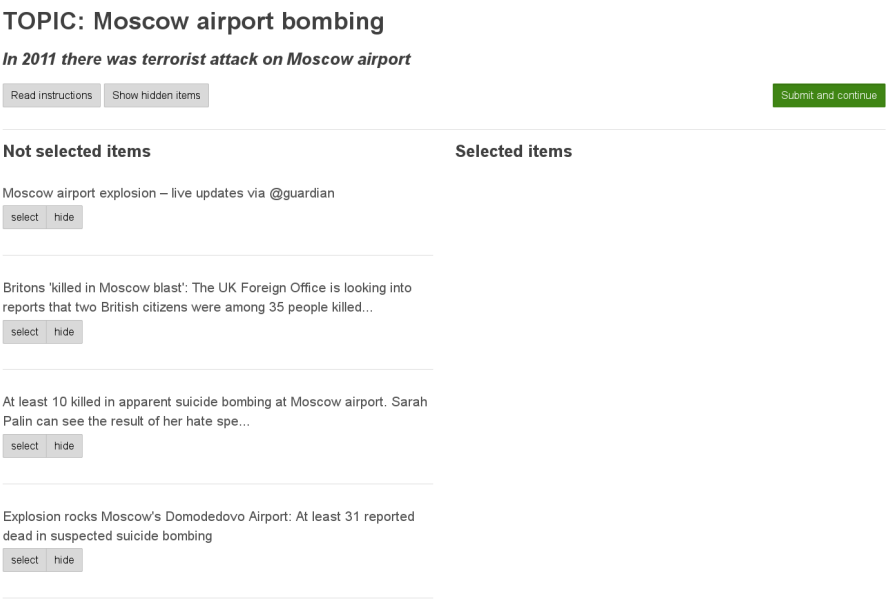


Figure 11: User interface of the tool: Initial state with no tweets selected

C Human summaries

C.1 Topic 14: Release of the Rite

../data/summaries/golden/1/14.txt

```
why are people giving shitty reviews for "the rite" i think it
    was well written directed and the actors were stunning simply
     amazing .
i saw the rite last night and it scared me so bad i had
    nightmares :( but it was a really good movie .
that movie therite is crazy as hell it got me shook its a good
    movie.
just watched the rite spoiler alert: jesus exists .
for the record: "the rite" was long and dry .
have yall seen the commercials for that movie the rite that movie
     looks crazy
```

../data/summaries/golden/2/14.txt

```
that movie therite is crazy as hell it got me shook its a good
    movie.
went to the midnight showing last night @ 12:01 am and saw
    anthony hopkins movie "rite" most awesome yes indeed a super
    - must see .
the rite - movie review: i have always been a fan of any movie
    that deals with the battle between good and .
sir anthony hopkins tops the box office with the rite: what a
    slow weekend at the box office not much came out .
the rite was good shoulda seen little fockers though.
anthony hopkins the rite tops friday box office with $5 3 million
     (hollywood reporter) .
saw the rite today with stephen excellent movie .
holy anthony hopkins just saw the rite awesome movie reminded me
    of good ol' hannibal lecter days .
 the rite was quite good anthony hopkins can pull off "creepy"
     very well here i would refer to him as "father hannibal" :-)
     .
anthony hopkins takes 'the rite' to top of box office - .
'the rite' captures friday box office: thriller makes $5 3
    million by mawuse ziegbe anthony hopkins and mart
```

../data/summaries/golden/3/14.txt

```
hopkins takes 'the rite' to top of box office (ap) - ap - the
    anthony hopkins horror film the rite topped the b .
new film the rite is based on the training of a real priest who
    says the film is about faith catholic exorcist.
chat w/ director stars of "the rite" anthony hopkins is like the
    english grandfather i never had .
the rite ; good movie .
bishop exorcist praise new exorcism movie - catholic culture:
    washington postbishop exorcist praise new exorci .
the movie news channel: box office: anthony hopkins has 'the rite
    ' stuff for no 1 friday movies
```

C.2 Topic 17: White Stripes breakup

../data/summaries/golden/1/17.txt

```
sad 2 see the white stripes split they did run out of steam with
    2 many side projects their best moment .
i guess jack finally realized he's a better drummer than meg rt :
     what a bummer rip white stripes .
it's official foresight doesn't make this news easier to take rt
    : the white stripes break up .
awful news for rock n' roll rt confirmed white stripes have
    broken up.
 i'll be honest i thought the white stripes broke up 2-3 years
     ago
```

../data/summaries/golden/2/17.txt

```
the white stripes broke up oh well .
sad 2 see the white stripes split they did run out of steam with
    2 many side projects their best moment .
and their site is down rt : rip the white stripes - .
white stripes documentary is certified gold: the white stripes
    documentary under the great white northern light
```

../data/summaries/golden/3/17.txt

```
news+videos the white stripes call it quits (via ) read breakup
    rip bummer.
it's official foresight doesn't make this news easier to take rt
    : the white stripes break up .
awful news for rock n' roll rt confirmed white stripes have
    broken up
```

C.3 Topic 24: Super Bowl seats

../data/summaries/golden/1/24.txt

```
nfl statement: some temporary seating areas inside stadium not
    completed fans not seated will receive refund a triple the
    tix cost .
super bust: woes were endless for texas hosts: super bowl week in
     texas was not always so super .
~400 fans with tickets who were shuttled to a basement area in
    the cowboys stadium will get refunded 3x face value
    superbowlxlv.
six workers preparing for super bowl xlv are injured when ice
    falls off cowboys stadium in texas .
some 400 ticket holders denied seats: some 400 fans with tickets
    to the big game were denied seats at the super .
the nfl offered $200 tickets to fans to stand outside of cowboys
    stadium and watch super bowl xlv on big screen .
super bowl fans denied dallas seats get offer from nascar track
    superbowl.
hoosier forced to watch super bowl from basement
```

../data/summaries/golden/2/24.txt

```
nfl says 400 of the 850 fans affected by the ticket / seat
    scofflaw weren't able to be given new seats tickets were $900
     each superbowl.
```

```
six workers preparing for super bowl xlv are injured when ice
    falls off cowboys stadium in texas .
dallas fire marshall deems some temporary seats unsafe .
the nfl offered $200 tickets to fans to stand outside of cowboys
    stadium and watch super bowl xlv on big screen .
displaced nfl super bowl fans offered free nascar tickets (
    examiner com) .
nfl sorry for super bowl seat fiasco
```

../data/summaries/golden/3/24.txt

```
omg did know this terrible jerry jones should give them his seats
     fb.
some super bowl fans left without seats to get triple refunds .
dallas fumbles: super bowl fans go seat-less - .
some seats not ready for super bowl fans at cowboys stadium:
    arlington texas -- the nfl has announced that a .
i wish joe buck's seat was one of the 400 that couldn't be
    accommodated at the super bowl .
who were all thes people in the parking lot super bowl -people
    that got kicked out bad seats .
displaced fans will receive free super bowl tickets next year:
    what no coupon book i get that the nfl wants to .
super bowl blog: crews busy clearing snow and ice off stadium
    roof
```

C.4 Topic 29: global warming and weather

../data/summaries/golden/1/29.txt

```
warming leads to increased evaporation & precipitation which
    falls as increased snow in winter (via ).
climate change will bring more monster winter storms; global
    warming news today provides complete coverage: (pr- .
how extreme weather could create a global food crisis joseph romm
     via .
yes it has been cold but read this extreme weather report from
    santa fe - climate change is real .
melting sea ice forces polar bear to swim for nine days - climate
     change environment - .
i guess the world cycle of global warming started a little
    earlier than scientist thought it would .
there's a storm ranging from texas to maine can we get serious
    about global warming and climate change now .
tonight a cyclone is supposed to hit australia were i live and
    than a catergory 5 cyclone on thursday damn global warming
    sucks .
fish threatened by global warming to be moved north rgk lbp
```

../data/summaries/golden/2/29.txt

```
climate change will bring more monster winter storms; global
    warming news today provides complete coverage: (pr- .
polar bear's record 9-day continuous swim blamed on global
    warming "bear swam in 2-6 degrees c for 232 hrs and 687 km".
this is global warming : virginia burkett a senior scientist with
     the us geological survey said global warming .
how extreme weather could create a global food crisis joseph romm
     via .
```

```
expert now warns global warming will lead to brittain getting
    colder climategate how inconvenint lol.
more evidence "global warming" does not exist .
global warming 101 -&gt; globalwarming capandtrade
    inconvenienttruth climatechange environment algore gogreen.
the global warming conspiracy global warming you've got to be
    kidding me i'm freezing .
global warming update: bizarre weather destroyed crops and no
    more right whales .
krauthammer: global warming is a religion .
fish threatened by global warming to be moved north rgk lbp.
proof of global warming - it is supposed to be -40 right now
    global warming makes it -30 celcius .
global warming or global governance (full length) global : www
    facebook com if you were to ask ten people .
global warming  scientists say climate change to bring more .
uk climate change adaption: endangered fish will be moved from
    warming waters to colder northern lakes .
news from our neighbor to the east on global warming and harrison
     schmitt
```

../data/summaries/golden/3/29.txt

```
there's a storm ranging from texas to maine can we get serious
    about global warming and climate change now .
proof of global warming - it is supposed to be -40 right now
    global warming makes it -30 celcius .
krauthammer: global warming is a religion .
warming leads to increased evaporation & precipitation which
    falls as increased snow in winter (via ).
more evidence "global warming" does not exist
```

C.5 Topic 36: Moscow airport bombing

../data/summaries/golden/1/36.txt

```
britons 'killed in moscow blast': the uk foreign office is
    looking into reports that two british citizens were among 35
    people killed .
at least 10 killed in apparent suicide bombing at moscow airport
    sarah palin can see the result of her hate spe .
explosion rocks moscow's domodedovo airport: at least 31 reported
     dead in suspected suicide bombing .
breaking news: terrorist attack kills 31 injures at least 100 at
    moscow airport watch live news on our tv channels .
fatal blast reported at moscow airport terminal: russian
    authorities monday reported a fatal explosion at .
moscow airport bomb toll is up to 35 dead and 130 injured
    islamist website is praising the bomber time to get serious
    on security here .
blast at moscow airport kills 30 injures 130 .
moscow airport explosion - live updates: at least 31 people
    killed and more than 130 injured in suicide bombing .
moscow airport blast: dozens killed after explosion hits
    domodedovo airport: early unconfirmed reports suggest .
suicide bomber kills 35 at russia's biggest airport: moscow (
    reuters) - a suicide bomber killed at least 35 people at
    russia's news.
v : domodedovo airport cameras catch the moment of the blast (via
     ).
```

```
from ap: bomb at moscow airport kills 31 injures 130 story @ .
cnn 31 killed 130 injured in terror blast at moscow airport: a
    terrorist suicide bomber is blamed for the mosco .
explosion kills 31 at moscow airport: an explosion ripped through
     the international arrivals hall at moscow's bu .
latest: at least 20 killed in explosion at moscow's busiest
    airport .
suicide bomber blamed for moscow airport attack that killed 31
    russian state tv reports .
blast rocks moscow's main airport: moscow's domodedovo airport -
    the busiest in the russian capital - is h .
at least 31 killed in moscow airport suicide bomb (afp): afp - a
    suspected suicide bombing monday killed at least 31 people
    and wound
```

../data/summaries/golden/2/36.txt

```
moscow airport bomb toll is up to 35 dead and 130 injured
    islamist website is praising the bomber time to get serious
    on security here .
v : domodedovo airport cameras catch the moment of the blast (via
     ).
explosion kills 31 at moscow airport: an explosion ripped through
     the international arrivals hall at moscow's bu .
search for moscow blast britons: british airways and bmi flights
    had arrived there shortly before the blast a t .
russian president: apparent 'terrorist attack' witnesses say
    carried out by two suicide bombers according to ria: .
russians find head of suicide bomber - arab in appearance
```

../data/summaries/golden/3/36.txt

```
bbc news - 16:13 gmt - moscow bombing: carnage at russia's
    domodedovo airport .
russian media now reporting that at least 31 people were killed
    and 130 injured in bombing at moscow's domodedovo airport
    from afp via bbc.
russian president: apparent 'terrorist attack' witnesses say
    carried out by two suicide bombers according to ria: .
v : domodedovo airport cameras catch the moment of the blast (via
     ).
russians find head of suicide bomber - arab in appearance .
white light thoughts and prayers go out to the families and
    victims of the moscow airport bombing .
moscow airport bomb kills dozens including two britons world news
     the guardian .
just in: president obama sends condolences to victims of today's
    terror attack at domodedovo airport in moscow
```

C.6 Topic 78: McDonalds food

../data/summaries/golden/1/78.txt

```
of course they did it's mcdonald's rt : mcdonalds oatmeal is too
    good they had to lace it with something .
[mcdonalds] where your food is made of plastic and all other
    sorts of unedible shit .
dear mcdonalds - why you so awesome .
i had that fruit and raisin oatmeal from mcdonalds yesterday for
    the first time and omgggg that oatmeal was sent from heaven.
```

```
hell is a place like mcdonalds so if saying what the hell is bad
    then saying what the mcdonalds is bad too.
the only way i'd ever eat at mcdonald's is if the girls dressed
    like they do in the fifth element: .
mcdonalds is fuggin nasty now im not eating that for awhile
    except maybe the mcnuggets.
i wish mcdonalds would open up in my backyard
```

../data/summaries/golden/2/78.txt

```
mcdonalds sounds soooo good right now .
i nominate for a shorty award in food because i love the food i'
    ve been eating at mcds for 40+years yea .
drive-thru fraud fast food worker helps steal $50k from customers
    ' credit cards .
mcdonalds so fattening but who cares .
[mcdonalds] where your food is made of plastic and all other
    sorts of unedible shit
```

../data/summaries/golden/3/78.txt

```
jus heard a mcdonalds commercial dude said u can take my plasma
    tv but u can't have my nuggets hesadamnfool.
want a mcdonald's.
profit edges up at mcdonald's and its prices will too .
mcdonald's franchise: the ultimate fast-food business - .
mcdonald's reports 5 3 percent jan sales growth business news.
the guys who started mcdonalds had 4 or 5 food ventures fail
    before the great arches is it bad that at times thats a
    comforting thought
```

C.7 Topic 79: Saleh Yemen overthrow

../data/summaries/golden/1/79.txt

```
thousands protest against government in yemen: the protests which
     organizers said were inspired by events in t .
anti-government rallies in yemen stay calm: .
yemen protests: thousands call on president to leave .
saba: saleh announces raise in salaries of armed and security
    forces .
no more saleh in yemen will be fascinating but very uncertain
    country may go from ungovernable to whatever is worse than
    that .
20 000 march in yemen against president in 'day of rage' .
correcting link yemen is latest arab state to join unrest at
    least 10 000 activists demonstrating .
there were demos today in yemen several cities although president
     already gave up plan to run again for presidency yemen.
over 20 000 take to streets in yemen "day of rage" .
after egypt - yemen jordan on the brink of uprising many protests
     underway .
yemen president not to extend term aljazeera.
students activists stage rival demonstrations at yemeni
    university .
yemen president facing protests says he will not seek to extend
    his term when it expires in 2013 .
now anti-government protests are spilling into yemen - business .
yemen president signals won&amp;#8217;t stay beyond 2013 freedomwar
    egypt jan25 syria feb4 .
```

```
yemeni president says he won't seek another term: sanaa yemen (ap
    ) - the yemeni president told parliament .
teepeecreek com yemen arrests female activist in student protest
    teepeecreek com
```

../data/summaries/golden/2/79.txt

```
20 000 march in yemen against president in 'day of rage' .
 yemen pres salih will not seek new term but not step down
     immediately follows egypt pres mubarak.
experts estimate the number of opposition protesters in sanaa one
     hundred and fifty thousand yemen feb3 (cont) .
yemen arrests female activist in student protest: yemen has
    arrested a woman activist who led student prot .
protesters shout slogans during a protest against the arrest of
    rights activist karman outside t yemen photo.
pak thousands of yemenis demand govt change: thousands of yemenis
     took to streets of the capital sanaa -- inspir .
news media paints revolts in yemen tunisia & egypt as popular
    unrest citing use of fb&twitter to make the arrangements4 the
     demonstrations
```

../data/summaries/golden/3/79.txt

```
20 000 yemenis protesters urged president ali abdullah saleh to
    step down - - muhammadjusuf.
thousands of yemenis call on president to quit: thousands of
    yemenis demonstrated in the capital on thursday ca .
thousands march against yemen president - middle east world - the
     independent .
bbc news - yemen protests: thousands call on president to leave .
ticker: thousands of yemenis call on president to quit .
yemen protests: thousands call on president to leave .
scores protest against yemen president (usa today): share with
    friends: world news - top stories news .
yemenis rally against president .
20 000 march in yemen against president in 'day of rage'
```

C.8 Topic 88: The King's Speech awards

../data/summaries/golden/1/88.txt

```
saw it yesterday & it was brilliant it deserves oscar gold hooper
     gets top dga prize for 'king's speech' abc7 com .
so all the critics groups chose social network and all the guilds
     so far chose king's speech and i'm ambivalent about both
    sagawards.
yes  : and cast of the king's speech won 'outstanding performance
     by a cast in a motion picture ' sagawards
```

../data/summaries/golden/2/88.txt

```
'the king's speech' gets 12 oscar nominations that's ridiculous
    the state of the union isn't till tonight .
the kings speech has been nominated for baftas but yet it still
    is at the bottom at the uk chart only making 9.2 million .
the royal me: the king's speech is worthy of the best actor award
     colin firth will likely garner at this year's .
"the king's speech" wins art directors guild award .
```

```
king's speech true grit lead oscar race: the kings speech a
    british drama about the stammering monarch k .
the kings speech was superb colin firth a shoe in for a bafta and
     oscar i'd say omniplex in larne was better than i expected
    too .
the king's speech for best ensemble sagawards.
not the worst recent film to receive a gluttony of academy award
    nominations but quite likely the most boring lolz
```

../data/summaries/golden/3/88.txt

```
saw it yesterday & it was brilliant it deserves oscar gold hooper
     gets top dga prize for 'king's speech' abc7 com .
'the king's speech' gets 12 oscar nominations that's ridiculous
    the state of the union isn't till tonight .
natalie portman colin firth and the kings speech are all winners
    at the sags melissa teo and christian bale also .
yes : and cast of the king's speech won 'outstanding performance
    by a cast in a motion picture ' sagawards.
the king's speech leads oscar nominations 12 nominations .
bbc news - the king's speech leads oscars field .
thrilled to see that colin firth has won best actor sag award as
    well as the king's speech for best ensemble woo hoo .
the king's speech got 12 oscar nominations
```

C.9 Topic 99: SuperBowl commercials

../data/summaries/golden/1/99.txt

```
brasky says the doritos commercial wins the superbowl .
the superbowl commercials have launched: as almost everyone in
    north america is aware the .
doritos commercials are by far the funniest .
loved budlight branding superbowl commercial lets go packers .
complete list of 2011 super bowl commercials & advertisers on
    youtube .
volkswagen has put out their star wars themed commercial for the
    2011 superbowl .
r funny super bowl commercial kids always need shoes and a
    chicken heaven bless you funjoy .
in case you haven't seen the brilliant motorola xoom ad from the
    super bowl .
lololol so far doritos and pepsi max are killing it superbowl
    commercials.
physically rt i'd like to congratulate pepsi no women were
    physically harmed in making of that commercial brandbowl
    superbowl.
volkswagen star wars commercial dominates super bowl twitter buzz
     .
had some inproductive minutes with this brilliant volkswagen
    commercial aired during the superbowl trekkie in.
 superbowl are all the commercials going to be bud light doritos
     and pepsi .
why are car commercials so boring except for volkswagon i enjoyed
     both of their superbowl spots.
 superbowl so far the pepsi commercials are the best.
must be the daddy in me that prefers the vw commercial of messing
     with the mini-darth superbowl dadstalking.
2011 super bowl commercial: i love doritos .
```

```
best commercial so far "hold me closer tiny dancer " budlight
    superbowl.
 superbowl that was the lamest penalty after the green bay
     interception-touchdown love that eminem commercial
```

../data/summaries/golden/2/99.txt

```
volkswagen has put out their star wars themed commercial for the
    2011 superbowl .
commercials that are winning so far: audi with the old-time
    luxury prison pepsi max with the dieting wife and husband
    superbowl.
motorola attacks apple in superbowl commercial motorola apple .
kia has a commercial advertising a superbowl commercial like a
    snake consuming itself .
audi super bowl commercial 2011 is online [video] audi.
budwiser just got off with a "to be continued" commercial to air
    on superbowl sunday swag.
wow awesome bold superbowl commercial seriously going to consider
     xoom tablet now before ipad2 .
superbowl commercials are ok seems like doritos vs pepsi i was
    stoked for and your commercial sucked .
doritos commercial where guy licks the cheese off another guys
    finger rt : what was your favorite superbowl commercial .
are these doritos super bowl commercials offensive [video]:
    dorito .
dragon looks so cute drinking a cola superbowl cocacola
    commercial.
that kid wearing a darth vader costume is the best commercial for
     this yr's superbowl .
diddy stars in superbowl ad for mercedes diddy featured in a new
    superbowl ad for mercedes .
so far doritos commercial with dog best
```

../data/summaries/golden/3/99.txt

```
superbowl commercials winners: vw vader xoom (before i heard the
    price) and bridgestone beaver .
volkswagen has put out their star wars themed commercial for the
    2011 superbowl .
motorola attacks apple in superbowl commercial motorola apple .
physically rt i'd like to congratulate pepsi no women were
    physically harmed in making of that commercial brandbowl
    superbowl.
commercials that are winning so far: audi with the old-time
    luxury prison pepsi max with the dieting wife and husband
    superbowl.
thi doritos superbowl commerical is a hot mess: .
dragon looks so cute drinking a cola superbowl cocacola
    commercial.
budwiser just got off with a "to be continued" commercial to air
    on superbowl sunday swag.
hashtags used in tv ad for superbowl rt audi to launch r8
    commercial with twitter hashtag .
i estimate that doritos has to sell 1 5 million large bags to
    recoup what they spend on superbowl commercials
```

D The source code of the tool

/home/nata/projects/react–twincolumn–selection–app/index.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Items selector</title>
    <link rel="stylesheet" href="static/css/app.css">
    <link rel="stylesheet" href="static/ink.min.css">
  </head>
  <body>
    <section id="items"></section>
    <script src="js/bundle.js"></script>
  </body>
</html>
```

/home/nata/projects/react–twincolumn–selection–app/js/app.js

```
var React = require('react');
var Router = require('react-router');

var AppRoutes = require('./components/SelectorApp.react');

Router.run(AppRoutes, function (Handler) {
  React.render(<Handler />, document.getElementById('items'));
});
```

/home/nata/projects/react–twincolumn–selection–app/js/stores/ItemStore.js

```
var EventEmitter = require('events').EventEmitter;
var assign = require('object-assign');
var Dispatcher = require('../dispatcher');
var Constants = require('../constants');

var CHANGE_EVENT = 'change';

var state = {
  session_info: {
    session: null,
    topics: []
  },
  items: {
    selected: [],
    notselected: []
  },
  current: {
    topic: "",
    topic_id: null,
    loaded: false,
    show_hidden: false,
  },
};

function init(info) {
  state.session_info.session = info.session_id;
  state.session_info.topics = info.topics;
  state.items = {selected: [], notselected: []};
```

```
}

function load_topic(topic_id, topic_name, desc, items) {
  state.items.notselected = items;
  state.items.selected = [];
  state.current.topic = topic_name;
  state.current.topic_id = topic_id;
  state.current.loaded = true;
  state.current.desc = desc;
}

function submit_current_topic() {
  state.session_info.topics.splice(0, 1);
  state.items = {selected: [], notselected: []};
  state.current = {
    topic: "",
    topic_id: null,
    loaded: false,
  };
}

function _move_between_lists(id, from, to) {
  var i, item;
  for (i = 0; i < from.length; i++) {
    item = from[i];
    if (item.id === id) {
      to.push(item);
      from.splice(i, 1);
      return;
    }
  }
}

function select(id) {
  _move_between_lists(id, state.items.notselected, state.items.selected);
}

function unselect(id) {
  _move_between_lists(id, state.items.selected, state.items.notselected);
}

function move_up_down(id, list, direction) {
  var direction_step =  (direction === Constants.DIRECTION_DOWN) ? 1 : -1;
  var i, item;
  for (i = 0; i < list.length; i++) {
    item = list[i];
    if (item.id === id) {
      if ((direction === Constants.DIRECTION_DOWN && i === list.length) ||
          (direction === Constants.DIRECTION_UP && i === 0)) {
        return;
      }
      list.splice(i, 1); // remove from old position
      list.splice(i + direction_step, 0, item); // insert into new position
      return;
    }
  }
}

function set_hide(id, hide_status) {
  var i, item;
  for (i = 0; i < state.items.notselected.length; i++) {
    item = state.items.notselected[i];
    if (item.id === id) {
      item.hidden = hide_status;
      return;
    }
  }
```

```
}

function toggle_show_hidden() {
  state.current.show_hidden = !state.current.show_hidden;
}

var ItemStore = assign({}, EventEmitter.prototype, {
  getItems: function() {
    return state.items;
  },
  getTopic: function() {
    return state.current.topic;
  },
  getDescription: function() {
    return state.current.desc;
  },
  isLoaded: function() {
    return state.current.loaded;
  },
  getAllTopics: function() {
    return state.session_info.topics;
  },
  getSessionId: function() {
    return state.session_info.session;
  },
  getTopicId: function() {
    return state.current.topic_id;
  },
  showHidden: function() {
    return state.current.show_hidden;
  },
  emitChange: function() {
    this.emit(CHANGE_EVENT);
  },
  addChangeListener: function(callback) {
    this.on(CHANGE_EVENT, callback);
  },
  removeChangeListener: function(callback) {
    this.removeListener(CHANGE_EVENT, callback);
  }
});

Dispatcher.register(function(action) {
  switch(action.actionType) {
    case Constants.APP_INIT:
      init(action.session_info);
      break;

    case Constants.TOPIC_LOAD:
      load_topic(action.topic_id, action.topic, action.desc, action.items);
      ItemStore.emitChange();
      break;

    case Constants.TOPIC_SUBMIT:
      submit_current_topic();
      break;

    case Constants.ITEM_SELECT:
      select(action.id);
      ItemStore.emitChange();
      break;

    case Constants.ITEM_UNSELECT:
      unselect(action.id);
      ItemStore.emitChange();
      break;
```

```
    case Constants.ITEM_MOVE:
      move_up_down(action.id, state.items.selected, action.direction);
      move_up_down(action.id, state.items.notselected, action.direction);
      ItemStore.emitChange();
      break;

    case Constants.ITEM_HIDE:
      set_hide(action.id, true);
      ItemStore.emitChange();
      break;

    case Constants.ITEM_UNHIDE:
      set_hide(action.id, false);
      ItemStore.emitChange();
      break;

    case Constants.TOGGLE_SHOW_HIDDEN:
      toggle_show_hidden();
      ItemStore.emitChange();
      break;

    default:
      console.log(action);
  }
});

module.exports = ItemStore;
```

/home/nata/projects/react–twincolumn–selection–app/js/actions.js

```
var Dispatcher = require('./dispatcher');
var Constants = require('./constants');
var ItemStore = require('./stores/ItemStore');
var $ = require('jquery-browserify');

var Actions = {
  loadFromServer: function(id) {
    $.ajax({
      url: '/api/topic/' + id,
      dataType: 'json',
    }).then(function(data) {
      Dispatcher.dispatch({
        actionType: Constants.TOPIC_LOAD,
        items: data.tweets,
        desc: data.desc,
        topic: data.topic,
        topic_id: data.topic_id
      });
    });
  },

  loadNextTopic: function() {
    this.loadFromServer(ItemStore.getAllTopics()[0]);
  },

  submit: function() {
    var data = {
      session_id: ItemStore.getSessionId(),
      topic: ItemStore.getTopic(),
      tweets: ItemStore.getItems().selected,
      topic_id: ItemStore.getTopicId()
    };
    Dispatcher.dispatch({
      actionType: Constants.TOPIC_SUBMIT
    });
    $.ajax({
      type: 'POST',
      url: '/api/summary',
```

```
      data: JSON.stringify(data),
      dataType: 'json',
      contentType: 'application/json',
    });
  },

  init: function(session) {
    $.ajax({
      url: '/api/' + session,
      dataType: 'json'
    }).then(function(data) {
      Dispatcher.dispatch({
        actionType: Constants.APP_INIT,
        session_info: data,
      });
    });
  },

  select: function(id) {
    Dispatcher.dispatch({
      actionType: Constants.ITEM_SELECT,
      id: id
    });
  },

  unselect: function(id) {
    Dispatcher.dispatch({
      actionType: Constants.ITEM_UNSELECT,
      id: id
    });
  },

  move: function(id, direction) {
    Dispatcher.dispatch({
      actionType: Constants.ITEM_MOVE,
      id: id,
      direction: direction
    });
  },

  hide: function(id) {
    Dispatcher.dispatch({
      actionType: Constants.ITEM_HIDE,
      id: id
    });
  },

  unhide: function(id) {
    Dispatcher.dispatch({
      actionType: Constants.ITEM_UNHIDE,
      id: id
    });
  },

  toggleShowHidden: function() {
    Dispatcher.dispatch({
      actionType: Constants.TOGGLE_SHOW_HIDDEN
    });
  }
};

module.exports = Actions;
```

E Main source code

/home/nata/projects/sumy/sumy/summarizers/sumbasic.py

```
from __future__ import absolute_import, division

from collections import defaultdict
from warnings import warn

from ._summarizer import AbstractSummarizer


class SumBasicSummarizer(AbstractSummarizer):
    _stop_words = frozenset()

    @property
    def stop_words(self):
        return self._stop_words

    @stop_words.setter
    def stop_words(self, words):
        self._stop_words = frozenset(map(self.normalize_word, words))

    def __call__(self, document, sentences_count):
        distribution = self._get_distribution(document)
        sentences = list(document.sentences)

        if len(distribution) < len(sentences):
            message = (
                "Number of words (%d) is lower than number of sentences (%d). "
                "SumBasic algorithm may not work properly."
            )
            warn(message % (len(distribution), len(sentences)))

        ranks = defaultdict(int)
        step = 0

        while sentences:
            word = sorted(distribution, key=distribution.get, reverse=True)[0]
            ith_sentence = self._get_best_sentence(word, sentences, distribution
                )
            if not ith_sentence:
                # this word is not present in any of remaining sentences
                # we can safely remove it
                del distribution[word]
                continue
            ranks[ith_sentence] = 1 / (step + 1)
            sentences.remove(ith_sentence)
            for word in ith_sentence.words:
                distribution[self.stem_word(word)] **= 2
            step += 1
        return self._get_best_sentences(document.sentences, sentences_count,
            ranks)

    def _get_distribution(self, document):
        counts = defaultdict(int)
        for word in document.words:
            if word not in self.stop_words:
                counts[self.stem_word(word)] += 1

        for word in counts:
```

```
            counts[word] /= len(counts)

        return counts

    def _get_best_sentence(self, main_word, sentences, distribution):
        averages = {}
        for sentence in sentences:
            weight = 0
            is_candidate = False
            for word in sentence.words:
                stemmed = self.stem_word(word)
                weight += distribution[stemmed]
                if stemmed == main_word:
                    is_candidate = True
            if is_candidate:
                averages[sentence] = weight / len(sentence.words)
        if averages:
            return sorted(averages, key=averages.get, reverse=True)[0]
        return None
```

./data/Implementation.py

```
# coding: utf-8


# imports
import json
import os
import re
import requests
import time
from collections import defaultdict

from sumy.nlp.stemmers import Stemmer
from sumy.utils import get_stop_words
from sumy.parsers.plaintext import PlaintextParser
from sumy.nlp.tokenizers import Tokenizer
from sumy.summarizers.lsa import LsaSummarizer
from sumy.summarizers.random import RandomSummarizer
from sumy.summarizers.lex_rank import LexRankSummarizer
from sumy.summarizers.sumbasic import SumBasicSummarizer

TOPICS = [101, 14, 17, 24, 29, 36, 78, 79,  88, 99]
HUMANS = range(1, 4)

# file paths
source_dir = '/home/nata/Study/hig/master/data/input/txt/'
summaries_dir = '/home/nata/Study/hig/master/data/summaries/'
humans_dir = '/home/nata/Study/hig/master/data/summaries/golden/'


def summary_path(algorithm, topic, length):
    return os.path.join(summaries_dir, algorithm, str(length), str(topic) + '.
        txt')


def source_path(topic_id):
    return os.path.join(source_dir, str(topic_id) + '.txt')


def human_path(topic_id, person):
    return os.path.join(humans_dir, str(person), str(topic_id) + '.txt')


def human_path_store(topic_id, session_id):
    return os.path.join(humans_dir, 'txt', '{}_{}.txt'.format(topic_id,
        session_id))
```

```
# ## Input data load


spaces = re.compile(r'@\w+|[\s.|,!?#]+')


def clean_tweet(text):
    text = text.lower()
    text = spaces.sub(' ', text)
    text = spaces.sub(' ', text)
    return text


def save_summary(summary, index):
    topic = summary['t_id']
    user = summary['s_id']
    store_path = human_path_store(topic, user[:3])
    with open(store_path, 'w') as f:
        f.write('.\n'.join(clean_tweet(t['text']) for t in summary['tweets']))
    sh.ln('-s', store_path, human_path(topic, index))


def summaries_from_json(fp):
    summaries = sorted(json.load(fp), key=lambda x: x['t_id'])
    for i, summary in enumerate(summaries):
        save_summary(summary, i % 3 + 1)


with open(humans_dir + 'summaries.json') as f:
    summaries_from_json(f)


def topic_from_json(topic):
    json_path = os.path.join(
        '/home/nata/Study/hig/master/data/input/tweets/', "{}.json".format(topic
            ))
    with open(json_path) as f:
        tweets = json.load(f)['tweets']
    txt_path = os.path.join(
        '/home/nata/Study/hig/master/data/input/txt/', "{}.txt".format(topic))
    with open(txt_path, 'w') as f:
        f.write('.\n'.join(clean_tweet(t['text']) for t in tweets))


for topic in TOPICS:
    topic_from_json(topic)


with open(humans_dir + 'fix_summaries.json') as fp:
    summaries = sorted(json.load(fp), key=lambda x: x['t_id'])
    for i, summary in zip([1, 2, 2], summaries):
        save_summary(summary, i)


# ### Normalize all saved data


import pandas as pd


def get_sentiment_counts(data):
    url = 'http://www.sentiment140.com/api/bulkClassifyJson'
    res = requests.post(url, json=dict(data=data)).json()
    counts = defaultdict(lambda: defaultdict(int))
    for item in res['data']:
```

```
        d = counts[(item['topic'], item['human'])]
        d['length'] += 1
        if item['polarity'] == 4:
            d['positive'] += 1
        elif item['polarity'] == 0:
            d['negative'] += 1
    for ((topic, human), values) in counts.items():
        yield {
            'topic': topic,
            'human': human,
            'length': values['length'],
            'positive': values['positive'] / values['length'],
            'negative': values['negative'] / values['length'],
        }


def get_human_summaries_info():
    rows = []
    for topic in TOPICS:
        for human in HUMANS:
            with open(human_path(topic, human)) as f:
                entries = f.read().split('\n')
            rows.extend(({'text': t, 'topic': topic, 'human': human}
                         for t in entries))
    counts = list(get_sentiment_counts(rows))
    return pd.DataFrame(counts)


df = get_human_summaries_info().set_index(['topic', 'human'])


df.to_csv('/home/nata/Study/hig/master/data/results/human_summaries_stats.json')


# ## Summary generation


LANGUAGE = 'english'

stemmer = Stemmer(LANGUAGE)
stop_words = get_stop_words(LANGUAGE)
tokenizer = Tokenizer(LANGUAGE)

ALL_SUMMARISERS = ['random1', 'random2',
                   'random3', 'lexrank', 'lsa', 'sumbasic']
SENTIMENT_SUMMARIZERS = ['polarity', 'polarity_subj', 'polarity_freq']


TOPIC_LENGTHS = {
    14: 30,
    24: 30,
    29: 30,
    36: 30,
    78: 30,
    79: 30,
    88: 30,
    99: 30,
    101: 26,
    17: 27,
}

HUMAN_LENGTHS = {
    14: [6, 6, 11],
    17: [3, 4, 5],
    24: [6, 8, 8],
    29: [5, 9, 16],
    36: [6, 8, 18],
```

```
    78: [5, 6, 8],
    79: [7, 9, 17],
    88: [3, 8, 8],
    99: [10, 14, 19],
    101: [4, 6, 6]}


def ensure_dir(path):
    if not os.path.exists(path):
        os.makedirs(path)


def save_summary(text, topic_id, algorithm, length):
    path = summary_path(algorithm, topic_id, length)
    ensure_dir(os.path.dirname(path))
    with open(path, 'w') as out_file:
        out_file.write(text)


def generate_summaries(topic_id, text, algorithms, lengths):
    for algorithm in algorithms:
        for length in lengths:
            summary = generate_summary(algorithm, text, length)
            save_summary(summary, topic_id, algorithm=algorithm, length=length)


def generate_summary(algorithm, text, length):
    Summarizer = {
        'random1': RandomSummarizer,
        'random2': RandomSummarizer,
        'random3': RandomSummarizer,
        'lexrank': LexRankSummarizer,
        'lsa': LsaSummarizer,
        'sumbasic': SumBasicSummarizer,

    }.get(algorithm)

    if not Summarizer:
        print("Unknown algorithm")
        return ""

    summarizer = Summarizer(stemmer)
    summarizer.stop_words = stop_words
    parser = PlaintextParser.from_string(text, tokenizer)
    return "\n".join(map(str, summarizer(parser.document, length)))


def generate_all_for_topic(topic_id, lengths=None, summarizers=None):
    lengths = lengths or TOPIC_LENGTHS
    summarizers = summarizers or ALL_SUMMARISERS
    with open(source_path(topic_id)) as file_in:
        text = file_in.read()
    length = lengths[topic_id]
    generate_summaries(topic_id, text, summarizers, range(1, length))


for topic in TOPICS:
    generate_all_for_topic(topic)


# ## Sentiment summarization


import random
import textblob
from collections import namedtuple
```

```
def possible_swaps(selected, other):
    for item1 in selected:
        for item2 in other:
            yield selected.symmetric_difference({item1, item2})


def select_best(function, items):
    best = None
    best_score = None
    for item in items:
        item_score = function(item)
        if best is None or best_score < item_score:
            best = item
            best_score = item_score
    return best, best_score


def optimize(function, all_tweets, length):
    # Using hill climbing optimization
    selected = set(random.sample(all_tweets, length))
    selected_score = None
    for i in range(1000):  # no more than n steps, safe from cycling
        # doing random swaps
        next_states = possible_swaps(selected, all_tweets - selected)
        best, best_score = select_best(function, next_states)
        if best is None or selected_score is not None and best_score <
            selected_score:
            return selected
        selected = best
        selected_score = best_score
    return selected


def optimize_with_restarts(function, tweets, length, restarts):
    return select_best(function, (optimize(function, tweets, length) for _ in
        range(restarts)))[0]


tweets = frozenset([1, 2, 4, 5, 5, 5, 3, 2, 1, 4, 3, 11, 2])
function = lambda s: -abs(23 - sum(s))
print(optimize_with_restarts(function, tweets, 4, 5))


def sentiment_match(tweets):
    pos_sentiment = sum(
        t.polarity for t in tweets if t.polarity > 0) / len(tweets)
    neg_sentiment = sum(
        t.polarity for t in tweets if t.polarity < 0) / len(tweets)

    def predicate(tweets):
        tweet_pos = sum(
            t.polarity for t in tweets if t.polarity > 0) / len(tweets)
        tweet_neg = sum(
            t.polarity for t in tweets if t.polarity < 0) / len(tweets)
        return -abs(pos_sentiment - tweet_pos) - abs(neg_sentiment - tweet_neg)
    return predicate


def sentiment_subj_match(tweets):
    overall_polarity = sum(t.polarity for t in tweets) / len(tweets)
    overall_subjectivity = sum(t.subjectivity for t in tweets) / len(tweets)

    def predicate(tweets):
        subjectivity = sum(t.subjectivity for t in tweets) / len(tweets)
        polarity = sum(t.polarity for t in tweets) / len(tweets)
        return -abs(overall_polarity - polarity) - abs(overall_subjectivity -
```

```
            subjectivity)
    return predicate


def get_word_frequencies(tweets):
    counts = defaultdict(int)
    for tweet in tweets:
        for word in tweet.text.split(' '):
            counts[word] += 1

    for word in counts:
        counts[word] /= len(counts)

    return counts


def count_words(frequencies, tweets):
    frequencies = frequencies.copy()
    res = 0
    for tweet in tweets:
        tweet_res = 0
        for word in tweet.text.split(' '):
            tweet_res += frequencies[word]
            frequencies[word] **= 2
        res += tweet_res / len(tweet)
    return res / len(tweets)


def sentiment_frequency_match(tweets):
    overall_polarity = sum(t.polarity for t in tweets) / len(tweets)
    overall_freqs = get_word_frequencies(tweets)

    def predicate(tweets):
        polarity = sum(t.polarity for t in tweets) / len(tweets)
        freqs = count_words(overall_freqs, tweets)
        return -abs(overall_polarity - polarity) * freqs
    return predicate


Tweet = namedtuple('Tweet', ['text', 'polarity', 'subjectivity', 'id'])


def tweets_with_sentiment(topic):
    with open('/home/nata/Study/hig/master/data/input/tweets/{}.json'.format(
        topic)) as f:
        data = json.load(f)
        tweets = data['tweets']
    result = []
    for tweet in tweets:
        text = clean_tweet(tweet['text'])
        sentiment = textblob.TextBlob(text).sentiment
        result.append(Tweet(text, sentiment.polarity,
                            sentiment.subjectivity, tweet['tweet_id']))
    return frozenset(result)


def summarize_sentiment_for_topic(summarizer, tweets, topic, length):
    predicate = {
        'polarity': sentiment_match,
        'polarity_subj': sentiment_subj_match,
        'polarity_freq': sentiment_frequency_match,
    }[summarizer](tweets)
    res = optimize_with_restarts(predicate, tweets, length, 5)
    text = "\n".join(t.text for t in res)
    save_summary(text, topic_id=topic, length=length, algorithm=summarizer)
```

```
# In[ ]:

for topic in TOPICS[:]:
    tweets = tweets_with_sentiment(topic)
    for length in range(1, TOPIC_LENGTHS[topic]):
        print(topic, length)
        for algorithm in ['polarity_freq']:
            summarize_sentiment_for_topic(algorithm, tweets, topic, length)


# ## Evaluation

# ### Fraction of Topical words


import csv
import sh
import pandas as pd


FOTW_PATH = '/home/nata/Study/hig/master/implementation/evaluation/fotw.sh'


def read_raw_fotw(fp):
    reader = csv.reader(fp, delimiter=' ')
    next(reader)  # skip header
    results = []
    for row in reader:
        var1, var2, cosine, percent_topic, fraction_topic, topic_overlap = row
        results.append((var1, var2, fraction_topic))
    return results


# #### For generated summaries


def generate_mappings(fp, topic_id, algorithms, lengths):
    writer = csv.writer(fp, delimiter=' ', quoting=csv.QUOTE_MINIMAL)
    for algorithm in algorithms:
        for length in lengths:
            summary = summary_path(algorithm, topic_id, length)
            source = source_path(topic_id)
            writer.writerow((algorithm, length, source, summary))


def calculate_fotw(topic_id, algorithms=ALL_SUMMARISERS + SENTIMENT_SUMMARIZERS,
     lengths=None):
    if lengths is None:
        lengths = TOPIC_LENGTHS
    # generate settings
    mappings_path = str(topic_id) + '_mapping.txt'
    length = lengths[topic_id]
    with open(mappings_path, 'w') as mappings_file:
        generate_mappings(mappings_file, topic_id,
                          algorithms, range(1, length))

    sh.sh(FOTW_PATH, mappings_path)

    # collect results
    with open(mappings_path + '.ieval.micro') as fp:
        results = read_raw_fotw(fp)

    for (algorithm, length, value) in results:
        yield {'topic': topic_id,
               'algorithm': algorithm,
               'length': length,
               'value': float(value)}
```

```
    sh.rm(mappings_path)
    sh.rm(mappings_path + '.ieval.micro')
    sh.rm(mappings_path + '.ieval.macro')


def fotw_for_all_topics():
    results = []
    for topic in TOPICS:
        results.extend(calculate_fotw(topic))
    return pd.DataFrame(results)


def transform_random(data):
    random = data[data['algorithm'].isin(['random1', 'random2', 'random3'])]
    other = data[~data['algorithm'].isin(['random1', 'random2', 'random3'])]
    random_values = random.groupby(['topic', 'length']).mean()
    random_values['algorithm'] = 'random'
    random_values = random_values.reset_index()
    return pd.concat([other, random_values]).set_index(['algorithm', 'topic', '
        length'])


transform_random(fotw_for_all_topics()).to_csv(
    "/home/nata/Study/hig/master/data/results/fotw-auto.csv")


# #### For human-generates summaries


def generate_mappings_human(fp, topics):
    writer = csv.writer(fp, delimiter=' ', quoting=csv.QUOTE_MINIMAL)
    for topic_id in topics:
        for person in HUMANS:
            summary = human_path(topic_id, person)
            source = source_path(topic_id)
            writer.writerow((topic_id, person, source, summary))


def calculate_fotw_human(topics):
    # generate settings
    mappings_path = 'human_mapping.txt'
    with open(mappings_path, 'w') as mappings_file:
        generate_mappings_human(mappings_file, topics)

    sh.sh(FOTW_PATH, mappings_path)

    lengths = {}
    for topic in topics:
        lengths[topic] = {}
        for person in HUMANS:
            path = human_path(topic, person)
            length, _ = sh.wc('-l', path).split()
            lengths[topic][person] = length

    # collect results
    with open(mappings_path + '.ieval.micro') as fp:
        results = read_raw_fotw(fp)

    for (topic, person, value) in results:
        length = lengths[int(topic)][int(person)]
        yield {'topic': topic,
               'person': person,
               'length': length,
               'value': value}

    sh.rm(mappings_path)
```

```
    sh.rm(mappings_path + '.ieval.micro')
    sh.rm(mappings_path + '.ieval.macro')


def fotw_for_all_topics_human():
    results = calculate_fotw_human(TOPICS)
    return pd.DataFrame(results).set_index(['topic', 'person', 'length'])


fotw_for_all_topics_human().to_csv(
    "/home/nata/Study/hig/master/data/results/fotw-human.csv")


# ### ROUGE


import csv
import sh


ROUGE_DATA_PATH = '/home/nata/Study/hig/master/data/rouge/'
ROUGE_PATH = '/home/nata/Study/hig/master/implementation/rouge-java/'


def get_length(topic, mode):
    pos = {'full': 2, 'middle': 1, 'short': 0}
    return HUMAN_LENGTHS[topic][pos[mode]]


def prepare_folders(algorithm, mode='middle'):
    # clean system folder
    system_path = os.path.join(ROUGE_DATA_PATH, 'system')
    sh.rm('-rf', system_path)
    sh.mkdir(system_path)

    for topic in TOPICS:
        for length in range(1, TOPIC_LENGTHS[topic]):
            rouge_path = os.path.join(
                system_path, '{}_{}'.format(topic, length))
            sh.ln('-s', summary_path(algorithm=algorithm,
                                     topic=topic, length=length), rouge_path)

    reference_path = os.path.join(ROUGE_DATA_PATH, 'reference')
    sh.rm('-rf', reference_path)
    sh.mkdir(reference_path)

    for topic in TOPICS:
        for human in HUMANS:
            rouge_path = os.path.join(
                reference_path, '{}_{}'.format(topic, human))
            length = get_length(topic, mode)
            sh.head('-n', length, human_path(topic, human), _out=rouge_path)
#             sh.ln('-s', human_path(topic, human), rouge_path)


def get_rouge(algorithm):
    prepare_folders(algorithm)
    sh.java('-jar', 'rouge2.0.jar', _cwd=ROUGE_PATH)
    with open(os.path.join(ROUGE_PATH, 'results.csv')) as f:
        yield from parse_results(f, algorithm)


def parse_results(fp, algorithm):
    reader = csv.reader(fp)
    next(reader)  # skip header
    for line in reader:
        _, topic, length, recall, prec, f_score, _ = line
```

```
        yield {'topic': int(topic),
               'algorithm': algorithm,
               'length': int(length),
               'recall': float(recall),
               'precision': float(prec),
               'f_score': float(f_score)}


def save_rouge_all():
    results = []
    for algorithm in ALL_SUMMARISERS + SENTIMENT_SUMMARIZERS:
        results.extend(get_rouge(algorithm))
    data = pd.DataFrame(results)
    random = data[data['algorithm'].isin(['random1', 'random2', 'random3'])]
    other = data[~data['algorithm'].isin(['random1', 'random2', 'random3'])]
    random_values = random.groupby(['topic', 'length']).mean().reset_index()
    random_values['algorithm'] = 'random'
    return pd.concat([other, random_values]).set_index(['topic', 'algorithm', '
        length'])


save_rouge_all().to_csv("/home/nata/Study/hig/master/data/results/rouge.csv")


# #### for human agains human rouge


def prepare_folders_human(human, mode):
    # clean system folder
    system_path = os.path.join(ROUGE_DATA_PATH, 'system')
    sh.rm('-rf', system_path)
    sh.mkdir(system_path)

    for topic in TOPICS:
        rouge_path = os.path.join(system_path, '{}_{}'.format(topic, human))
        length = get_length(topic, mode)
        sh.head('-n', length, human_path(topic, human), _out=rouge_path)

    reference_path = os.path.join(ROUGE_DATA_PATH, 'reference')
    sh.rm('-rf', reference_path)
    sh.mkdir(reference_path)

    for topic in TOPICS:
        length = get_length(topic, mode)
        for other_human in HUMANS:
            if other_human != human:
                rouge_path = os.path.join(
                    reference_path, '{}_{}'.format(topic, other_human))
                sh.head('-n', length, human_path(topic,
                                                 other_human), _out=rouge_path)


def parse_results_human(fp, mode):
    reader = csv.reader(fp)
    next(reader)  # skip header
    for line in reader:
        _, topic, human, recall, prec, f_score, _ = line
        yield mode, int(topic), int(human), float(recall), float(prec), float(
            f_score)


def get_rouge_human(human):
    for mode in ('full', 'middle', 'short'):
        prepare_folders_human(human, mode)
        sh.java('-jar', 'rouge2.0.jar', _cwd=ROUGE_PATH)
        with open(os.path.join(ROUGE_PATH, 'results.csv')) as f:
            yield from parse_results_human(f, mode)
```

```
def save_rouge_all_human(fp):
    writer = csv.writer(fp, delimiter=' ', quoting=csv.QUOTE_MINIMAL)
    writer.writerow(['mode', 'topic', 'human',
                     'recall', 'precision', 'f_score'])
    for human in HUMANS:
        writer.writerows(get_rouge_human(human))


with open("/home/nata/Study/hig/master/data/results/rouge_human.csv", 'w') as
    res_file:
    save_rouge_all_human(res_file)


# ## Visualization


human_stats_raw = pd.read_csv(
    '/home/nata/Study/hig/master/data/results/human_summaries_stats.json')
human_stats_raw[['topic', 'length']].groupby('topic').apply(
    lambda x:  sorted(list(x['length']))).to_dict()


plt.figure()


def filter_group(group):
    return group[group.length == human_lengths.ix[group.topic]]
rouge = rouge_raw.groupby('topic').apply(
    filter_group).set_index(['topic', 'algorithm'])


rouge.ix[78]


rouge.unstack(level='algorithm').recall.plot(kind='box')
```