Enhanced Topic-Based Modeling for Twitter Sentiment Analysis

by

Swetha Baskaran

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved May 2016 by the
Graduate Supervisory Committee:

Hasan Davulcu, Chair
Arunabha Sen
Ihan Hsiao

ARIZONA STATE UNIVERSITY

August 2016

ABSTRACT

In this thesis multiple approaches are explored to enhance sentiment analysis of tweets. A standard sentiment analysis model with customized features is first trained and tested to establish a baseline. This is compared to an existing topic based mixture model and a new proposed topic based vector model both of which use Latent Dirichlet Allocation (LDA) for topic modeling. The proposed topic based vector model has higher accuracies in terms of averaged F scores than the other two models.

ACKNOWLEDGMENTS

I take this opportunity to thank my chair Professor Hasan Davulcu for his consistent and whole-hearted support throughout the course of my masters program and for hiring me to be part of his CySIS Research Lab. The experience I gained working as a research assistant under him is what motivated and enabled my to embark on this thesis.

T want to I also would like to thank Dr.Sharon Hsiao for providing me useful insight in visualization which helped shape that part of my project.

I'd like to thank Dr.Arunabha Sen and Dr.Mohamed Sarwat for taking time of their busy schedules to be part of my thesis panel.

I also take this opportunity to thank the PhDs in my lab, Nyunsu Kim and Mert Ozer for always helping me and giving me spontaneous feedback on different phases of my Thesis.

And I finally thank my parents and friends who were a great support system and took time off to help me generate ground truth.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Oxford Dictionary defines Sentiment analysis as the process of computationally identifying and categorising opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral. The reason this term which was relatively unknown even a decade ago is cemented in literature today is because of its ramifications in diverse fields.

Previously, the sentiments of people were gauged manually in the form of surveys and focus groups. There was an increasing need to automate the process to keep up with the growth of the market. This was also the time when the popularity of social media was on a meteoric rise. Initially Machine learning algorithms were used to classify texts as positive or negative. When this information became insufficient, research was directed at exploring the semantics of people's social media content. Today, Sentiment analysis is used to determine marketing strategy, improve campaign success, improve customer service, in recommendation systems and to detect radical groups.

1.1 Motivation

Micro-Blogging platforms, especially Twitter have fundamentally changed the way we consume news, interact with organisations and people, from relationships and dialog with like minded individuals. This makes twitter an indispensable resource to understand the sentiments of people on a particular topic and the nature of the sentiments. This thesis focuses on determining the best model to analyse sentiments

of tweets and pits regular statical models and concept-based models against topic-based models.

1.2 Proposed Work

The idea is to compare the main routes taken with sentiment analysis, statistical methods and concept level techniques along with a topic based model. Concept-based approaches to sentiment analysis focus on a semantic analysis of text through the use of web ontologies or semantic networks, which allow the aggregation of conceptual and affective information associated with natural language opinions.(Cambria, 2013) Statistical methods, such as Bayesian inference and support vector machines, have been popular for affect classification of texts. By feeding a machine learning algorithm a large training corpus of affectively annotated texts, it is possible for the system to not only learn the affective valence of affect keywords (as in the keyword spotting approach), but also to take into account the valence of other arbitrary keywords (like lexical affinity), punctuation, and word co-occurrence frequencies. (Cambria, 2013)

I plan to also use other features such as abbreviations, popular lingo, hashtags and emoticons and see if the addition of these features can positively affect the accuracy. Latent Dirichlet Allocation (LDA) (Blei *et al.*, 2003) is to be used for topic modeling. Each topic which will be a document may be viewed as a mixture of various topics. The data is split into multiple subsets based on topic distributions using clustering. For each subset, a separate sentiment model can be trained to predict the probability of the sentiment class of the tweets.

Chapter 2

RELATED WORK

Millennials have been responsible for numerous changes in the world and the ramifications have resonated in e-Commerce industry as well. The growing popularity of online shopping environments has garnered a lot of attention to recommendation systems. Some of the earliest work in sentiment analysis was triggered by this domain, which was detecting the polarity of a given text at feature, sentence or document levels as positive, negative or neutral. Pioneers (Pang *et al.*, 2002) at a document level captured the polarity of product and movie reviews. (Pang and Lee, 2005) extended this work to depict the polarity on scale.

This research is focused on sentiment analysis of tweets. The rise in the popularity of the micro blogging technique, where use of informal language and emoticons is common place and its innate nature of expressing opinions and emotions in a scaffolded number of characters makes it a challenge to translate the conventional methods used in sentiment analysis to fit it. This led to developing techniques unique to this type of text. There has research on applying topic sentiment analysis by (Mei *et al.*, 2007), (Branavan *et al.*, 2008), (Jo and Oh, 2011) and (He *et al.*, 2012). (Kouloumpis *et al.*, 2011) looks at how including emoticons and hashtags can impact the detection of sentiments. There have also been some recently proposed semi-supervised learning methods (Xiang and Zhou, 2014).

Chapter 3

DATA COLLECTION

The following figure gives a detailed description of the proposed work.

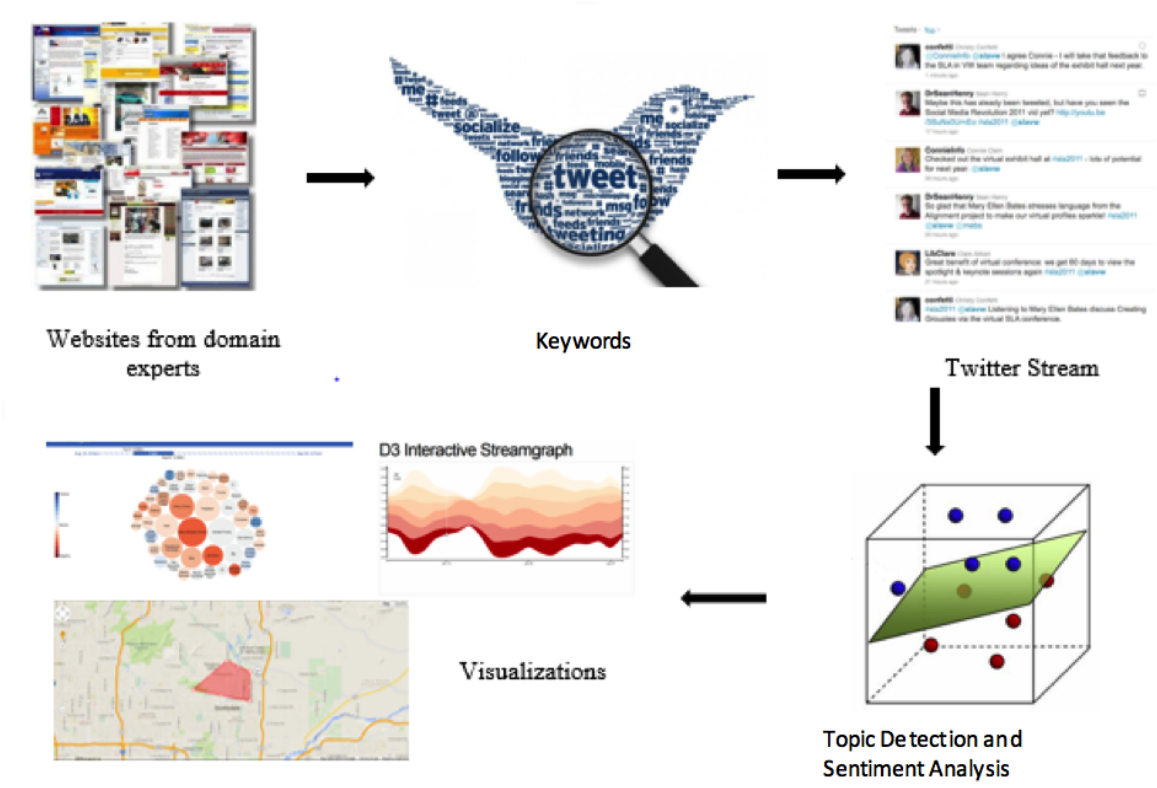

Figure 3.1: System Architecture

Area experts with field and domain expertise identified different political ideologies prevalent in UK and the major political parties affiliated with them. They also compiled a list of major players in the current political scene which included NGOs, politicians, journalists and potential separatists. This resulted in large amounts of text collected from a wide variety of organizations media outlets (e.g. web sites, blogs, news, RSS feeds, tweets, leaders speeches etc.) to discover hotly debated topics.

Discriminating keywords from these are queried on the Twitter public streaming API to get tweets that are topic rich and getter suited to be classified.

A human analysis component is required in sentiment analysis as humans are an authoritative source to judge sentiment. Volunteers manually labelled tweets as positive, neutral or negative. A total of 2998 tweets were manually classified into positive, negative or neutral. The labelled data was used to train the models and a portion of it was later used as test data.

Positive	1140
Negative	1502
Neutral	356
Total	1140

Table 3.1: Labelled Dataset

Chapter 4

STANDARD SENTIMENT MODEL

The standard method that I propose to use here is a non-topic based sentiment approach that can act as a baseline method for comparison.

Support vector machines (SVM) are universal learners. SVM works well with sentiment analysis for a number of reasons. It learns independent of dimensionality of feature space and can give good accuracies in high dimensional feature space. It is an approximation to a bound on the test error rate and there is a substantial body of theory behind it which suggests it should be a good idea. Also SVMs provide a good out-of-sample generalization, if the parameters C and r (in the case of a Gaussian kernel) are appropriately chosen. This means that, by choosing an appropriate generalization grade, SVMs can be robust, even when the training sample has some bias (Auria and Moro, 2008).

4.1 SVM

A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

I have used the libSVM (Chang and Lin, 2011), a popular SVM toolkit for JAVA. Probability estimation is used to calculate P(s/tw), where s is the sentiment class which can be positive, negative or neutral and tw is a tweet and P(s/tw) is probability

of sentiment class s given tweet tw.

libSVM performs only binary classification. In order to achieve multiclass classification libsvm performs one vs all Classification.One vs All or One-against-all (OAA) SVMs were first introduced by (Cortes and Vapnik, 1995). The initial formulation of the one-against-all method required unanimity among all SVMs: a data point would be classified under a certain class if and only if that class's SVM accepted it and all other classes' SVMs rejected it.

4.2 Tweet Specific Tokenization

Tweet specific tokenization is used to tokenize the tweets to include valuable information like emoticons, punctuations, hashtags, etc. while taking into account common spacing errors in tweets. I have used the CMU Twitter NLP tool (Gimpel *et al.*, 2011) as a reference.

4.3 Features

Listed below are the features I have used for classification. They have been derived using various web ontologies and libraries. They are similar to the features used in the univeral sentiment classifier in (Xiang and Zhou, 2014).

4.3.1 N-gram Tokens

If some highly occurring informative N-grams, here I have taken bigrams, trigrams and 4-grams, appear in a tweet, then the feature is set as 1 otherwise 0. The tokenization for this feature is done using Apache lucene and it's ShingleFilter. Using the tfidf scores, a cutoff is set to get the top tokens.

4.3.2 Emoticons

Emoticons can be a rich source of sentiment indication and is often ignored. It's an integral part of tweeting and so two features are allocated for occurrence of positive and negative emoticons

4.3.3 Punctuations

If there is a punctuation like a exclamation point or a question mark, for every such punctuation a count is incremented. In the absence of any, the feature is set to 0.

4.3.4 Hashtags

The number of hashtags is added as a feature because intuitively a passionate tweeter with a polarized view will tweet with multiple hashtags and this reflects either a positive of negative sentiment.

4.3.5 Pointwise Mutual Information Unigrams

(Mohammad *et al.*, 2013) in his paper had two lexicons based on PMI (pointwise mutual information). They are the NRC Hashtag Sentiment Lexicon with 54K unigrams, and the Sentiment140 Lexicon with 62K unigrams. Each unigram in the lexicon has a positive and negative score which is depending on the number of occurrences corresponding to the respective sentiments. The tweets are tokenized with the CMU Twitter NLP tool abd compared to the lexicon. The following features are computed here:

- sum positive sentiment score
- sum negative sentiment score

- total number of positive words
- total number of negative words
- maximum positive score
- maximum negative score

4.3.6 Pointwise Mutual Information Bigrams

There are 316K bigrams in the NRC Hashtag Sentiment Lexicon. We derive the following features here:

- total number of positive words
- total number of negative words
- maximum positive score
- maximum negative score

4.3.7 SentiwordNet Scores

In Sentiment Score, a phrase that corresponds to a feature, is scored to reflect its sentiment. To do this, the first step is to identify the different tokens or sub-phrases which exude a sentiment. Here, we consider verbs, adjectives and adjectival phrases. Adjectival phrases have an adjective at their head, which are usually preceded by an adverb. To achieve this we use the Stanford Core-NLP tool.

Example: very catchy and inspired

Semantic Tree:

(ROOT

(ADJP (RB very) (JJ catchy)

(CC and)

(JJ inspired)))

This is the tree structure of the phrase which has been tokenized and tagged with respective part-of-speech label. We use the Stanford part-of-speech (POS) tagger to filter out the verbs, adjectives and adjectival phrases. We consider the tokens with the following labels.

TAG	**PART-OF-SPEECH**
JJ	Adjective
JJR	Comparative Adjective
JJS	Superlative Adjective
RB	Adverb
RBR	Comparative Adverb
RBS	Superlative Adverb
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non3rd person singular present
VBZ	Verb, 3rd person singular present

Table 4.1: Table to Test Captions and Labels

To score these token, we use a tool called SentiWordNet. It is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet three sentiment

scores: positivity, negativity, objectivity. For adjectives and verbs, we directly add the scores returned by SentiWordNet to the phrase score.

difficult	-0.6875	left	-0.09375	recent	0	ready	0.1
bad	-0.642857143	serious	-0.08333	current	0	close	0.1
wrong	-0.597222222	common	-0.07567	private	0	easy	0.104167
poor	-0.479166667	natural	-0.075	central	0	high	0.107143
black	-0.410714286	public	Chart Title	itional	0	possible	0.125
low	-0.3875	likely	-0.0625	cultural	0	available	0.125
international	-0.3125	environmental	-0.0625	real	0.013889	significant	0.125
foreign	-0.3125	various	-0.0625	new	0.022727	entire	0.125
dead	-0.308823529	long	-0.05556	sure	0.029222	economic	0.15
cold	-0.307692308	early	-0.04167	full	0.03125	whole	0.15
blue	-0.296875	late	-0.03571	certain	0.035714	able	0.15625
free	-0.291666667	simple	-0.03571	medical	0.041667	democratic	0.166667
small	-0.25	religious	-0.03125	different	0.05	large	0.178571
short	-0.25	social	-0.02083	legal	0.05	financial	0.25
green	-0.25	general	-0.02083	hot	0.059524	TRUE	0.260417
other	-0.21875	physical	-0.01786	popular	0.0625	great	0.291667
red	-0.208333333	American	0	special	0.071429	fine	0.291667
final	-0.208333333	national	0	white	0.072917	important	0.3
little	-0.203125	young	0	strong	0.075	right	0.303571
single	-0.196428571	political	0	major	0.078125	clear	0.345588
hard	-0.1875	only	0	main	0.083333	happy	0.5625
dark	-0.170454545	human	0	old	0.09375	nice	0.575
open	-0.148809524	local	0	big	0.096154	good	0.613095
past	-0.125	military	0	personal	0.1	better	0.625
huge	-0.125	federal	0	similar	0.1	best	0.75

Figure 4.1: Top 100 Common Adjectives

In case of the adjectival phrases, we make use of the following Variable Scoring algorithm (Benamara *et al.*, 2007).

Suppose adj is an adjective and adv is an adverb. The variable scoring method (VS) works as follows.

We manually annotates frequently occurring adverb to represent strong, weak, affirmation and doubt adverbs.

For Example:

TYPE OF ADVERB	EXAMPLES
Affirmation	absolutely, certainly, exactly, totally
Doubt	possibly, roughly, apparently, seemingly
Weak	barely, scarcely, weakly, slightly
Strong	astronomically, exceedingly, extremely, immensely

Table 4.2: Types of Adverbs

The following features are computed here:

- sum positive sentiment score
- sum negative sentiment score
- total number of positive phrases
- total number of negative phrases
- maximum positive score
- maximum negative score

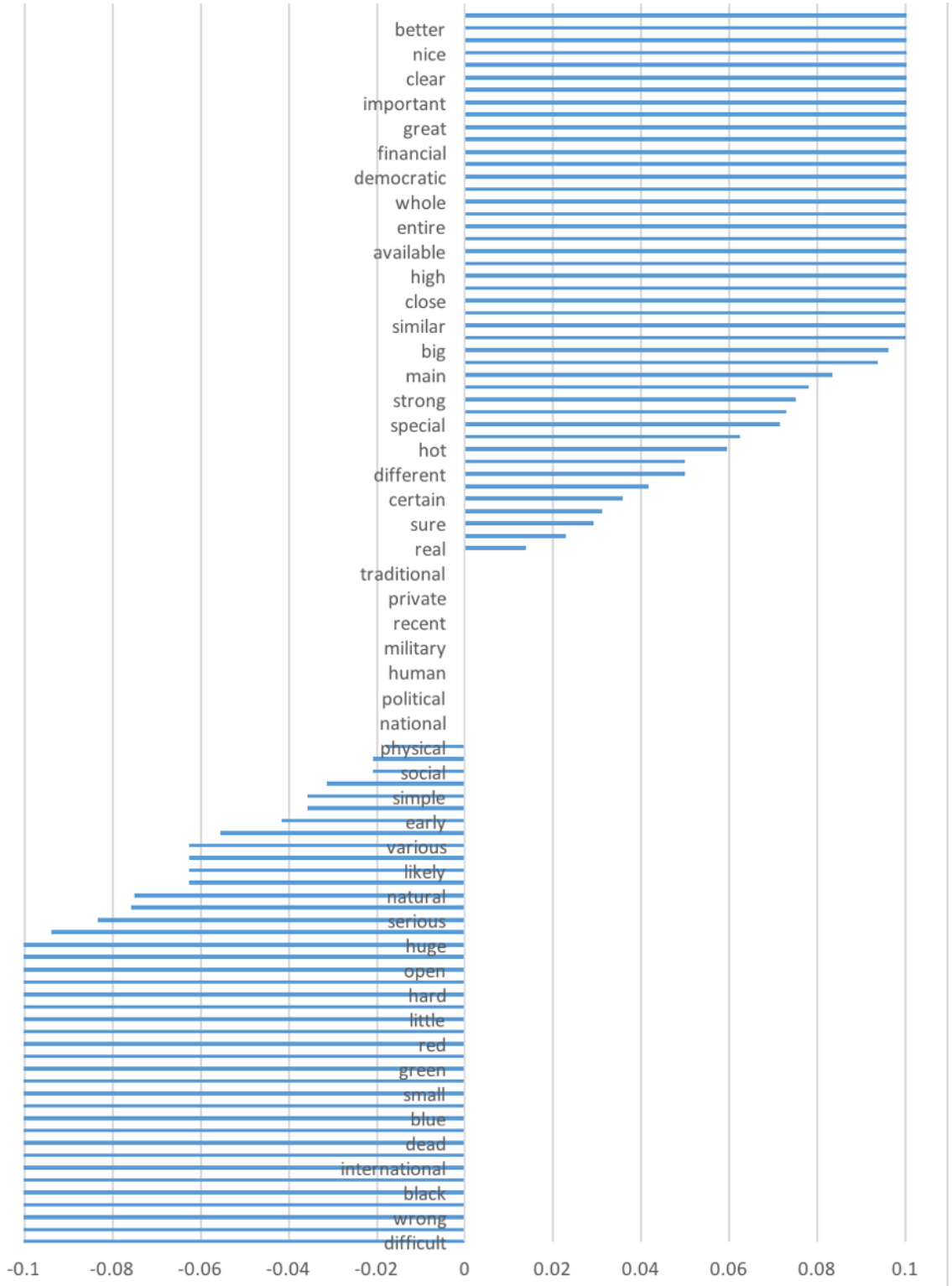

Figure 4.2: Common Adjectives and their Scores

- If adv ∈ AFF ∪ STRONG,
  If sc(adj) > 0,

$$fVS(adv,adj) = sc(adj) + (1-sc(adj)) \times sc(adv)$$

If sc(adj) < 0,

$$fVS(adv,adj) = sc(adj) - (1-sc(adj)) \times sc(adv)$$

- If adv ∈ WEAK ∪ DOUBT,
  If sc(adj) > 0,

$$fVS(adv,adj) = sc(adj) - (1-sc(adj)) \times sc(adv)$$

If sc(adj) < 0,

$$fVS(adv,adj) = sc(adj) + (1-sc(adj)) \times sc(adv)$$

Figure 4.3: Variable Scoring Algorithm

	Bad	Inferior	Ordinary	Average	Nice	Good	Pleasant	Charming
(Unmodified)	-0.642857143	-0.29167	0.0625	0	0.575	0.613095	0.625	0.3125
Slightly	-0.4375	-0.13021	-0.0546875	0.070313	0.521875	0.564732	0.578125	0.226563
Somewhat	-0.705357143	-0.35417	0	-0.125	0.5125	0.550595	0.5625	0.25
Rather	-0.767857143	-0.41667	-0.0625	-0.1875	0.45	0.488095	0.5	0.1875
Pretty	-0.767857143	-0.41667	-0.0625	-0.1875	0.45	0.488095	0.5	0.1875
Quite	-0.642857143	-0.29167	0.0625	-0.0625	0.575	0.613095	0.625	0.3125
Decidedly	-0.392857143	-0.04167	0.3125	0.1875	0.825	0.863095	0.875	0.5625
Usually	-0.642857143	-0.29167	0.0625	-0.0625	0.575	0.613095	0.625	0.3125
Very	-0.848214286	-0.45313	0.1796875	-0.19531	0.628125	0.661458	0.671875	0.398438
Extremely	-1.053571429	-0.61458	0.296875	-0.32813	0.68125	0.709821	0.71875	0.484375

Figure 4.4: Common Adverb-Adjective Combinations and their Scores

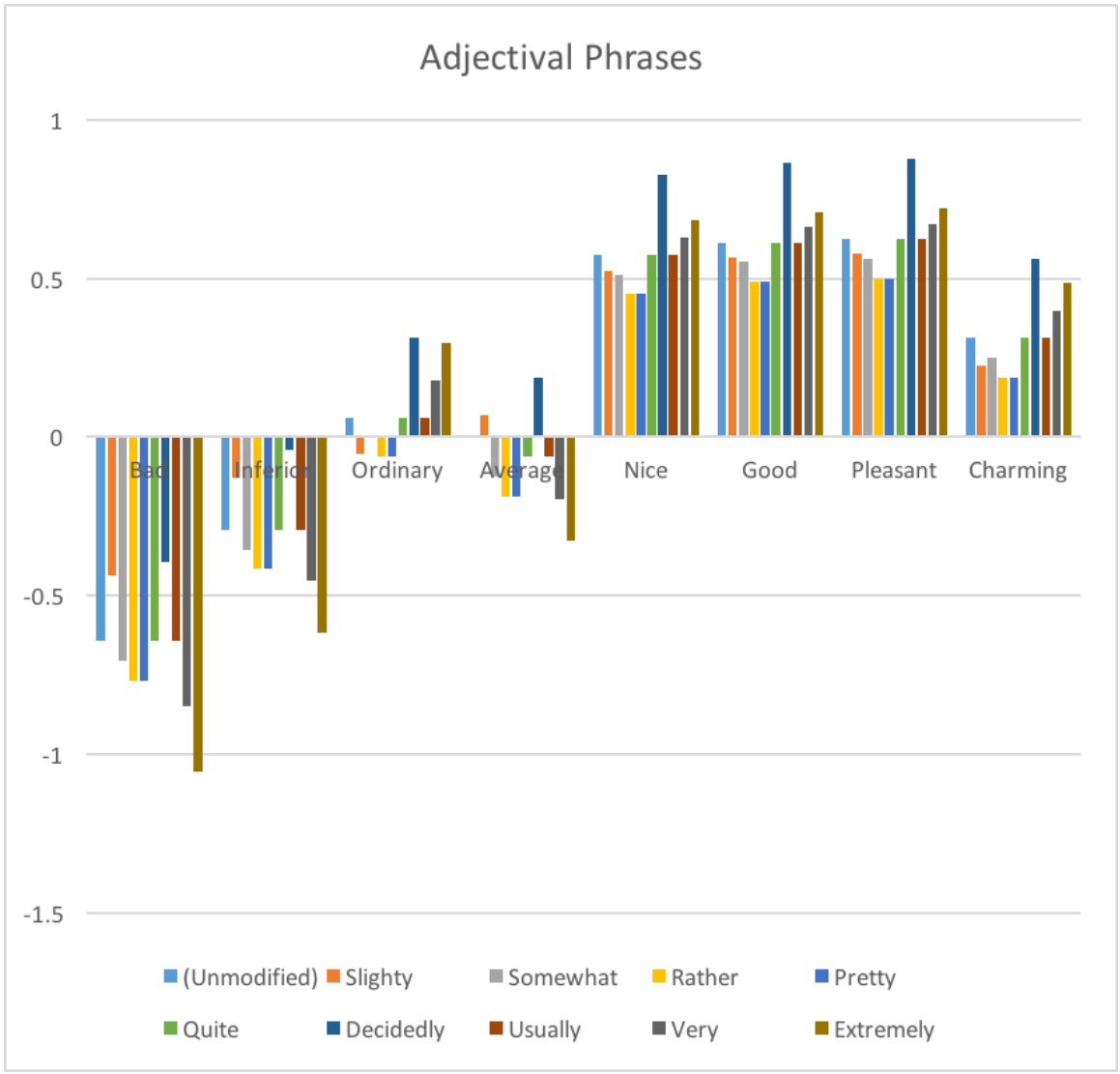

Figure 4.5: Common Adverb-Adjective Combinations Chart

Chapter 5

TOPIC BASED MIXTURE MODEL

5.1 Topic Modeling

Latent Dirichlet allocation (LDA) is used for topic modelling. LDA is a technique that automatically discovers topics that documents contain. Here each tweet is considered a document.

Dirichlet is a distribution specified by a vector parameter α containing some α_i corresponding to each topic i, which we write as Dir(α). The formula for computing the probability density function for each topic vector x is proportional to the product over all topics i of $x_i \alpha_i$. x_i is the probability that the topic is i, so the items in x must sum to 1. This prevents from getting arbitrarily large probabilities by giving arbitrarily large values of x.

Gibbs sampling is one Markov chain Monte Carlo (MCMC) technique suitable for the task. The idea in Gibbs sampling is to generate posterior samples by sweeping through each variable (or block of variables) to sample from its conditional distribution with the remaining variables fixed to their current values.

Here MALLET is used for topic modelling. MALLET is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text. The MALLET topic modeling toolkit contains efficient, sampling-based implementations of Latent Dirichlet Allocation. The MALLET topic model package includes an extremely fast and highly scalable implementation of Gibbs sampling, efficient methods for document-topic hyperparameter optimization, and tools for inferring topics

for new documents given trained models.

Table 5.1 has the topics and the top tokens associated with each of them along with their Dirichlet Parameter. The number of topics is set to 10.

We can make out some distinctive topics, like topic 1 is about al-queda and terrorism, topic 3 about muslim preachings, topic 6 about taliban in pakistan, topic 9 about terrorism in Africa.

Suppose that there are T topics in total in the training data, i.e. t_1, t_2, ..., t_T, the posterior probability of each topic given tweet xi is computed as in Eq.1, where C_{ij} is the number of times that topic t_j is assigned to some word in tweet x_i, usually averaged over multiple iterations of Gibbs sampling. a_j is the j_{th} dimension of the hyperparameter of Dirichlet distribution that can be optimized during model estimation. (Xiang and Zhou, 2014)

$$P_t(t_j|x_i) = \frac{C_{ij} + \alpha_j}{\sum_{k=1}^{T} C_{ik} + T\alpha_j} \qquad (1)$$

Figure 5.1: Equation 1

This can also be calculated using MALLET itself. It returns a doc-topic matrix which gives the probability of topic given document.

Take the following tweet for example:

"#BokoHaram terror is product of violence of #Nigeria's decades-long military rule @AfricaAtLse @saratu"

The probability distribution of the topics is given in table 5.2:

We notice that the probability of topic 9 is the highest at 0.994692160921789. This does hold true because the tweet keywords in the tweet are among the high

frequency words in topic 9.

5.2 Clustering

The tweets are clustered together based on the topics. Soft clustering is opted because the premise is based on the fact that a tweet can have different topics. If $P_t(t_j/x_i)$ is greater than a threshold value it is assigned to cluster j.

5.3 Training

For each topic a separate topic specific model is trained using the previously suggested standard method, with the various features.

5.4 Testing

The test data is run through the previously saved topic model of training data. Then it is run through the specific topic specific sentiment models and the probability estimates are obtained for each class. The final probability of a sentiment class c for a tweet t is given by equation 2.

$$P(c|x_i) = \sum_{j=1}^{T} P_m(c|t_j, x_i) P_t(t_j|x_i) \qquad (2)$$

Figure 5.2: Equation 2

Topic	**Dirichlet Parameter**	**Tokens**
1	0.00662	terrorist killed muslim make people taliban police al-qaeda uberfacts bomb
2	0.00759	muslims halal islam muslim meat country ukip people kill religion
3	0.00793	allah muhammad people muslim love heart prophet man pray ali
4	0.0055	good idea messenger end social message god work lives prayers earth
5	0.00932	ukip party farage candidate racist councillor people labour vote bnp london
6	0.00542	life live rest minute training suffer hated quit champion sins terror
7	0.00643	bin laden muslims terror taliban hijab pakistan osama muslim education
8	0.00597	terrorism jewish shia war terror terrorist video islamic radical camp
9	0.0069	palestinian boko haram israeli terrorist israel terror gaza human girls military sell children settlements nigeria
10	0.00929	allah muslim good quran islam knowledge worship heart man forgiveness

Table 5.1: Topic Modelling

Topic	$\mathbf{P_t(t_j/x_1)}$	**Topic**	$\mathbf{P_t(t_j/x_1)}$
1	5.486778279143347E-4	2	6.286327010130056E-4
3	6.567913474000851E-4	4	4.5544807136891735E-4
5	7.720601426802395E-4	6	4.4933284921234323E-4
7	5.324834125637067E-4	8	4.949326554877456E-4
9	0.9946921609217894	10	7.694800705703655E-4

Table 5.2: Topic Distribution

Chapter 6

TOPIC BASED VECTOR MODEL

6.1 Topic Modeling

Here also Latent Dirichlet allocation (LDA) is used for topic modelling. The same procedure as the previous modelling method is used for the first half, to get $P_t(t_j/x_i)$

$$P_t(t_j|x_i) = \frac{C_{ij} + \alpha_j}{\sum_{k=1}^{T} C_{ik} + T\alpha_j} \qquad (1)$$

Figure 6.1: Equation 1

6.2 Keyword-Topic Matrix

Mallet also outputs very word in the corpus of materials and the topic it belongs to. We can see in 6.2 that every file, tweets in this case is broken down into tokens and the topic each token belongs to is also given.

Here rather than training separate models, for each topic, vectors of tokens are obtained for each topic. We have a matrix of topics vectors for each sentiment class.

Using this we can compute a keyword-topic adjacency matrix where 1 is assigned when a keyword belongs to a topic, otherwise 0 is assigned. However this is done after the data is split into three datasets by the class. We have three matrices, one for each class.

FILE INDEX	FILE NAME	KEYWORD INDEX	KEYWORD	TOPIC NUMBER
0	tweet1.txt	0	libyanproud	6
0	tweet1.txt	1	countries	6
0	tweet1.txt	2	state	6
0	tweet1.txt	3	declares	6
0	tweet1.txt	4	war	6
0	tweet1.txt	5	terrorism	6
0	tweet1.txt	6	libya	6
0	tweet1.txt	7	terrorists	6
0	tweet1.txt	8	declare	6
0	tweet1.txt	4	war	6
0	tweet1.txt	2	state	6
1	tweet10.txt	9	ukip	4
1	tweet10.txt	10	press	4
1	tweet10.txt	11	adviser	4
1	tweet10.txt	12	janice	4
1	tweet10.txt	13	atkinson	4
1	tweet10.txt	14	kent	4
1	tweet10.txt	15	today	4
1	tweet10.txt	16	stay	4
1	tweet10.txt	17	classy	4

Figure 6.2: Mallet Output

6.3 Testing

The test data is also put though the same topic model and $P_t(t_j/x_i)$ is obtained. In a similar fashion a keyword-tweet matrix is also obtained, for the whole test set.

The matrix can be viewed as T topic vectors, here T, the number of topics is 10. Let TOV_{ct_i} be the topic vector for topic j in the topic-keyword matrix for class c. Let TWV_{t_i} be the tweet vector for tweet i. The similarity between the two is found using cosine similarity and multiplied with the weight $P_t(t_j/x_i)$ and summed over all topics to get $Wt(c/x_i)$. $Wt(c/x_i)$ is the weight for each class c given a tweet which determines the class of the tweet. The weight is a value between 0 and 1. The class with the highest weight is the predicted class.

The intuition behind this method is that it will help underrepresented classes which cannot be corrected even with sampling. When the class distribution is very biased, if the classifier doesn't recognise a small class, it wouldn't be reflected in

$$\mathbf{Wt}(c|x_i) = \sum_{j=1}^{T} P_m(c|t_j, x_i) P_t(t_j|x_i) \qquad (2)$$

Figure 6.3: Equation 3

the accuracy. This method helps overcome that problem and the results have better representation of the smaller classes.

6.3.1 Reasons for using Cosine Similarity

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. Using binary vector data works perfectly for doing cosine similarity studies. Actually, it makes the arithmetic much simpler because the magnitude of each vector is simply equal to the square root of the sum of its entries. The other similarity measures that could be used instead of cosine similarity is Hamming Distance. This is because they are the ones often used for binary vectors. However Hamming distance or Hamming similarity in this case is not suitable because it takes into account the 0s too, which denote the keywords not present in both in the topic vector and the tweet vector. Tweets being short, have only few keywords and this throws off the weight being computed. So Cosine Similarity is the better choice in this case.

Chapter 7

RESULTS

7.1 Standard Sentiment Model

MODEL	**Avg. F Score**
Standard Model with N-gram Tokens	63.2
+Emoticons	63.5j
+Punctuations	63.5
+Hashtags	64.8
+Pointwise Mutual Information Unigrams	67.4
+Pointwise Mutual Information Bigrams	68.4
+SentiwordNet Scores	71.9

Table 7.1: Results for Standard Sentiment Model

7.2 Topic Based Sentiment Models

MODEL	**Avg. F Score**
Standard Sentiment Model	71.9
Topic Based Mixture Model	72.5
Topic Based Vector Models	74.6

Table 7.2: Results for Topic-Based Sentiment Model

Chapter 8

VISUALIZATION COMPONENT

The twitter stream data and analysed data are stored in PostgreSQL database. The data is indexed using Apache Solr and AJAX-Solr Framework is used to facilitate AJAX calls to query Solr from the front end. This makes interactive visualization widgets possible.

8.1 Apache Solr

Apache Solr is an open source search platform built upon a Java library called Lucene.

Solr is a popular search platform for Web sites because it can index and search multiple sites and return recommendations for related content based on the search querys taxonomy. Solr is also a popular search platform for enterprise search because it can be used to index and search documents and email attachments.

Solr works with Hypertext Transfer Protocol (HTTP) Extensible Markup Language (XML). It offers application program interfaces (APIs) for Javascript Object Notation (JSON), Python, and Ruby. According to the Apache Lucene Project, Solr offers capabilities that have made it popular with administrators including Indexing in near real time, Automated index replication, Server statistics logging, Automated failover and recovery, Rich document parsing and indexing, Multiple search indexes, User-extensible caching, Design for high-volume traffic, Scalability, flexibility and extensibility, Advanced full-text searching, Geospatial searching and Load-balanced querying.(Rouse and Gibilisco, 2013)

8.2 AJAX-Solr Framework

AJAX Solr is a JavaScript library for creating user interfaces to Apache Solr.

It is JavaScript framework-agnostic, but requires an AJAX implementation to communication with Solr. As such, you may use the library whether you develop using jQuery, MooTools, Prototype, Dojo, or any other framework. You need only define a Manager object that extends the provided AbstractManager object, and define the function executeRequest() on that object. A jQuery-compatible Manager is provided at managers/Manager.jquery.js.

AJAX Solr loosely follows the Model-view-controller pattern. The ParameterStore is the model, storing the Solr parameters and, thus, the state of the application. The Manager is the controller; it talks to the ParameterStore, sends requests to Solr, and delegates the response to the widgets for rendering. The widgets are the views, each rendering a part of the interface.evolvingweb (2009)

8.3 D3 Visualization

D3.js is an open source JavaScript framework written by (Bostock, 2011) helping you to manipulate documents based on data. It is hugely popular because of its flexibility. Since it works seamlessly with existing web technologies, and can manipulate any part of the document object model, it is as flexible as the client side web technology stack (HTML, CSS, SVG). This gives it huge advantages over other tools because it can look like anything you want, and it isnt limited to small regions of a webpage like Processing.js, Paper.js, Raphael.js, or other canvas or SVG-only based libraries. It also takes advantage of built in functionality that the browser has, simplifying the developers job, especially for mouse interaction. (Skau, 2013)

8.4 Widgets in the Visualization

8.4.1 Volume Chart

This is the Google Annotated Timeline that displays the volumes of tweets on the different topics. At the bottom of the timeline is a zoom range selection which allows the user to zoom in on the volume trends for a particular range of dates.

8.4.2 Chord Diagram

User-Group mappings are rendered using d3 chord diagram. On a weekly basis, it shows shifts and flows of users among groups. All user and group information is indexed by Apache Solr server, supporting keyword and parametric search.

8.4.3 Map Widget

Google map API was used to display user geographical footprint. The users locations are ascertained using a three-part logic and rendered using a heat map. The users location can be determined if the user has enabled location services while tweeting. Another way is to check the users profile for any information on current address. If neither information is available, then we resort to mining the tweets for any locations mentioned. Another feature of the map widget is that a polygon can be drawn on the map and selected to focus on the users whose geo locations fall inside the boundary.

8.4.4 Event Timeline

TimelineJS, an open-source tool to build interactive timelines was used to showcase the popular events for each day. They are presented as trending hashtags, news articles and YouTube videos.

8.4.5 Network Widget

This widget is a flexible D3 force-directed graph layout implementation to represent the network of users in twitter. Based on retweet between users, a centrality score is calculated to ascertain the central users who are influential in the community.

8.4.6 Sentiment analysis visualization

A D3 interactive streamgraph is used to visualize the distribution of sentiments over a period of time. Pie charts are used to display the daily sentiments.

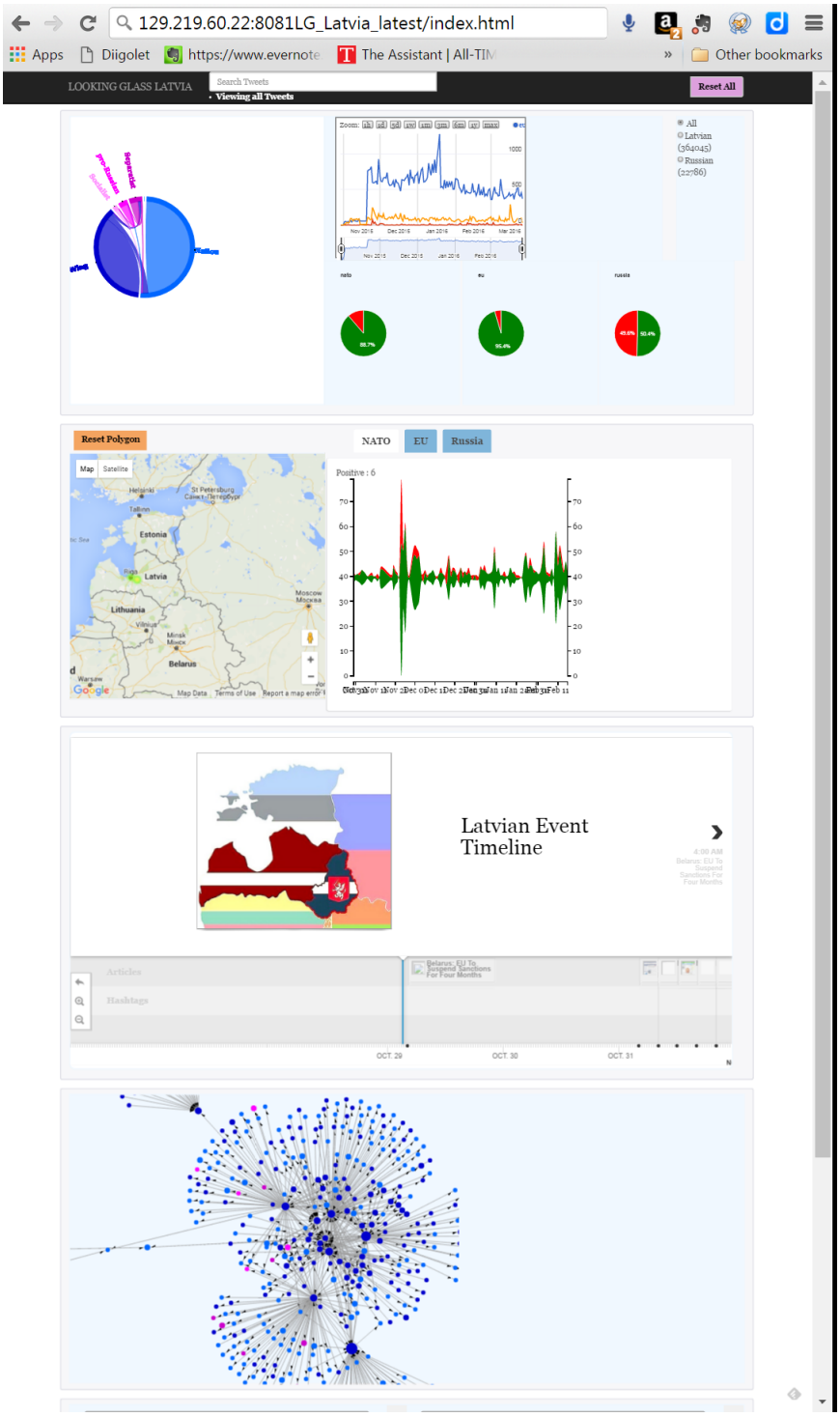

Figure 8.1: Looking Glass

Chapter 9

CONCLUSION

In this thesis, three different sentiment analysis models were implemented to compare and establish the best model. The standard sentiment model provided a good baseline and the topic based mixture model served as a standard for topic modeling. The proposed topic based vector method outperformed the former two. This method has scope to be improved and can be explored further for a better performance.

REFERENCES

Auria, L. and R. A. Moro, "Support vector machines (svm) as a technique for solvency analysis", Discussion Papers of DIW Berlin 811, DIW Berlin, German Institute for Economic Research, URL http://EconPapers.repec.org/RePEc:diw:diwwpp:dp811 (2008).

Benamara, F., C. Cesarano, A. Picariello and D. Reforgiato, "Sentiment analysis: Adjectives and adverbs are better than adjectives alone", in "In Proceedings of ICWSM conference", (2007).

Blei, D. M., A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation", J. Mach. Learn. Res. **3**, 993–1022, URL http://dl.acm.org/citation.cfm?id=944919.944937 (2003).

Bostock, M., "Mike bostock", URL https://bost.ocks.org/mike/ (2011).

Branavan, S., H. Chen, J. Eisenstein and R. Barzilay, "Learning document-level semantic properties from free-text annotations", in "Proceedings of ACL-08: HLT", pp. 263–271 (Association for Computational Linguistics, Columbus, Ohio, 2008), URL http://www.aclweb.org/anthology/P08-1031.

Cambria, E., "An introduction to concept-level sentiment analysis", in "Advances in Soft Computing and Its Applications - 12th Mexican International Conference on Artificial Intelligence, MICAI 2013, Mexico City, Mexico, November 24-30, 2013, Proceedings, Part II", pp. 478–483 (2013), URL http://dx.doi.org/10.1007/978-3-642-45111-9_41.

Chang, C.-C. and C.-J. Lin, "Libsvm: A library for support vector machines", ACM Trans. Intell. Syst. Technol. **2**, 3, 27:1–27:27, URL http://doi.acm.org/10.1145/1961189.1961199 (2011).

Cortes, C. and V. Vapnik, "Support-vector networks", Mach. Learn. **20**, 3, 273–297 (1995).

evolvingweb, "Ajax solr", URL https://github.com/evolvingweb/ajax-solr (2009).

Gimpel, K., N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan and N. A. Smith, "Part-of-speech tagging for twitter: Annotation, features and experiments", in "Proc. of ACL", (2011).

He, Y., C. Lin, W. Gao and K.-F. Wong, "Tracking sentiment and topic dynamics from social media.", in "ICWSM", edited by J. G. Breslin, N. B. Ellison, J. G. Shanahan and Z. Tufekci (The AAAI Press, 2012), URL http://dblp.uni-trier.de/db/conf/icwsm/icwsm2012.html#HeLGW12.

Jo, Y. and A. H. Oh, "Aspect and sentiment unification model for online review analysis", in "Proceedings of the Fourth ACM International Conference on Web Search and Data Mining", WSDM '11, pp. 815–824 (ACM, New York, NY, USA, 2011), URL http://doi.acm.org/10.1145/1935826.1935932.

Kouloumpis, E., T. Wilson and J. Moore, *Twitter Sentiment Analysis: The Good the Bad and the OMG!*, pp. 538–541 (AAAI Press, 2011).

Mei, Q., X. Ling, M. Wondra, H. Su and C. Zhai, "Topic sentiment mixture: Modeling facets and opinions in weblogs", in "Proceedings of the 16th International Conference on World Wide Web", WWW '07, pp. 171–180 (ACM, New York, NY, USA, 2007), URL http://doi.acm.org/10.1145/1242572.1242596.

Mohammad, S. M., S. Kiritchenko and X. Zhu, "Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets", CoRR **abs/1308.6242**, URL http://arxiv.org/abs/1308.6242 (2013).

Pang, B. and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales", in "Proceedings of ACL", pp. 115–124 (2005).

Pang, B., L. Lee and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques", in "Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10", EMNLP '02, pp. 79–86 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2002), URL http://dx.doi.org/10.3115/1118693.1118704.

Rouse, M. and S. Gibilisco, "Apache solr", URL http://whatis.techtarget.com/definition/Apache-Solr (2013).

Skau, D., "Why d3.js is so great for data visualization", URL http://www.scribblelive.com/blog/2013/01/29/why-d3-js-is-so-great-for-data-visualization/ (2013).

Xiang, B. and L. Zhou, "Improving twitter sentiment analysis with topic-based mixture modeling and semi-supervised training", in "Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)", pp. 434–439 (Association for Computational Linguistics, Baltimore, Maryland, 2014), URL http://www.aclweb.org/anthology/P/P14/P14-2071.