Mining and Visualising Contrastive Opinion from Text

Christopher David Coe

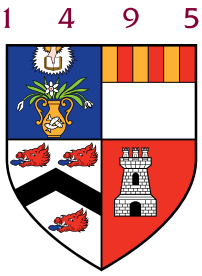
A dissertation submitted in partial fulfilment

of the requirements for the degree of

Bachelor of Science

of the

University of Aberdeen.



Department of Computing Science

2016

Declaration

No portion of the work contained in this document has been submitted in support of an application for a degree or qualification of this or any other university or other institution of learning. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.

Signed: Christopher Coe

Date: 29/04/2016

Abstract

The ability to automatically analyse texts to detect the topics and sentiment within has been a longstanding goal in computer science. In addition, finding a way to adequately summarise and visualise the data from such analysis is important if it is to be useful. This project sought to contribute a solution that dealt with both of these problems, with the development of a new model capable of contrastive opinion analysis via classification and related summary and visualisation.

The project was broken up into three constituent parts: first, a new corpus of text, drawn from iTunes reviews of the OSX El Capitan operating system update, was collected to be annotated with both sentiment and topic labels at the per-review and per-sentence level, to be used as a gold-standard for the classification parts of the project. As there is no current equivalent for such a jointly labelled dataset of this size, this in itself represents a significant contribution to the body of work in this area. Multinominal Naïve Bayes and Support Vector Machine classifiers were run against this dataset to produce a baseline accuracy for evaluation. Second, a new joint class classification model, named the Supervised Joint Class Model, was developed by adapting a previously published model, the weakly supervised Joint Sentiment Topic model. This model was capable of classifying sentiment and topic at the same time, so was well suited to the task at hand. Finally, a prototype visualization was coded to display the results of the classifier in an easy-to-understand at a glance format.

The results showed that the project was a success: contrastive opinion mining was able to be performed successfully, though accuracy did not quite reach state-of-the-art levels. The dataset collection and visualisation show much promise for future work.

Acknowledgements

I would like to sincerely thank all those who have provided me with their time, assistance, and guidance during the course of this project, and throughout my degree, for which I am enormously grateful.

Firstly, I would like to thank my project supervisor Dr. Chenghua Lin, who has offered a huge amount of support and expertise, starting from my summer internship, through the project planning stages, and right through to the end of this project. There is no doubt that without his continued support, and kindly making his previous work available to build upon, this project would not have been possible.

I would also like to thank Ebuka Ibeke, whose assistance and collaboration on the dataset were invaluable.

I would like to acknowledge the efforts of the staff in the computing department, and all those who have kindly imparted their knowledge and contributed to my education over the course of my degree.

Finally, I would like to thank Kat for always making sure I'm looked after, for getting me through it all, and for making it all worthwhile.

Publications

Section 4.1 is based on work contributed to:

> Ibeke E, Lin C, Coe C, Wyner A, Liu D, Barawi MH, et al. A Curated Corpus for Sentiment-Topic Analysis. In *Proceedings of the 10th Language Resources and Evaluation Conference*, Portorož, Slovenia. Forthcoming, 2016.

Contents

Table of Figures

Table of Tables

List of Abbreviations

JST	Joint Sentiment-Topic model
LDA	Latent Dirichlet Allocation
LSI	Latent Semantic Indexing
ML	Machine Learning
NB	Naïve Bayes
NLP	Natural Language Processing
OM	Opinion mining
R-JST	Reverse Joint Sentiment-Topic model
SA	Sentiment analysis
SJCM	Supervised Joint Class Model

1 Introduction

Opinion mining – the ability to automatically analyse texts and detect the topics and sentiment within – has been a longstanding goal in the field of data mining. In addition, finding a way to adequately summarise and visualise the data from such analysis is important if it is to be useful.

This project will seek to produce a piece of software that can perform the task of **contrastive opinion mining** – analyse a large corpus of documents (e.g. a set of reviews for a product), automatically extract common topics and sentiment information from these, display contrasting opinions for each topic, and visualise the output in a way that makes interpretation of the results easier.

This will be useful to, for example, people who need to make sense of a large number of individual documents that would be too laborious to read and classify individually. For example, an employee at a company sifting through hundreds of feedback forms or reviews, and having to keep track of recurring topics.

1.1 Motivation

The current state-of-the art classification algorithms that are suitable for classifying sentiment and opinion have some drawbacks – they are too domain-dependent, producing the best results only when trained and tested on texts from the same domain; and they are generally only capable of classifying on one dimension at a time.

Additionally, finding suitable datasets of labelled data for training purposes can be a difficult task, and producing such sets can be time-consuming. Also, the datasets that do exist with annotated opinions or sentiments tend to be classified on one dimension only, e.g. either sentiment or topic. For the purposes of contrastive opinion mining, a dataset that includes sentiment and topic dimensions will be required to properly train and test the model. As a suitable dataset of this type does not seem to exist, one will be created for this project, which in turn will make a significant contribution to the current body of work in this area.

Ambiguity of language is a common problem in the field of Natural Language Processing (NLP), and one that is difficult to solve without somehow encoding real-world knowledge into a classifier so that it may understand context. A topic/sentiment model would be able to provide somewhat of a workaround for this situation, as the topic classification of a piece of text could be used to inform which meaning of words to use. Taking for example the domain of laptop reviews, the word "longer" may be positive when related to the topic of battery life, but negative when part of the topic of loading times. Being able to handle such ambiguities would allow such a model to be much more context-sensitive than the current state of the art.

The ability to automatically analyse and categorise text according to topic and text would be extremely useful in a world where data in increasing at an exponential rate, and the requirement for tools to understand and summarise large volumes of data will only increase as time goes on. For example, companies want to be able to sift through thousands of tweets to find out attitudes towards specific products or services; software developers may want to analyse reviews to work out how particular aspects of the software are being received; or researchers may want to analyse the entire output of a news organisation in order to detect bias. A tool that can perform such analysis automatically would be of huge value, not only research value to natural language

researchers, but also business value to companies interested in the opinions of their customers.

1.2 Objectives

There are three objectives to be achieved in this project.

The main objective will be the development of a topic and sentiment classification model that will be able to perform the contrastive opinion mining task previously mentioned. As development from scratch may be very time-consuming, and therefore outside the scope of this project, a suitable candidate model will be identified to be adapted and developed to provide the necessary topic/sentiment joint model.

Before the development of the model can begin, a suitable corpus will need to be collected for the model to operate on and be tested against during the development process, as none of the available datasets of this type are suitable. This objective will consist of the identification, collection, structuring, and annotation of an opinion-based corpus, which can provide opinion information along two dimensions, sentiment and topic. This stage is important as it lays the groundwork for the model to be developed later – if the training and testing is to be successful, the dataset itself will have to be of a high enough quality to allow for meaningful results.

In addition to the construction of this dataset, the current state-of-the art classification algorithms will be trained and tested on the dataset to find the optimal strategy for classification according to current best practise. This step will provide the baseline that the classification model to be developed will be compared to in order to gauge its success. As these classifiers are only designed to work on a single dimension at once, performance will be tested against individual opinion dimensions, and then on a combination of the two.

Finally, a prototype visualisation will be designed and developed that can provide a simple visual overview of the dataset, broken down by topic and sentiment, so that a user may easily see the relations between each at a glance.

1.2.1 Objective summary

The objectives can be summarised in the following:

- Create a corpus of text, annotated for topic and sentiment, to be used for the classification tasks.
- Adapt an existing classification model to create a new contrastive opinion mining model that will be able to operate on both topic and sentiment simultaneously.
- Create a visualisation that uses the data produced by this model to summarise the results.

1.2.2 Project contributions

If the objectives for this project are met, each represents a significant contribution to the respective field of knowledge.

First, the dataset represents a significant contribution because currently there are no available opinion-based corpora of comparative size and quality that are annotated at the document and sentence level for both topic and sentiment. Producing such a corpus, and making it available to the text/opinion mining research community, provides an excellent opportunity for much more research to be produced based on this dataset.

Second, the classification model will also make a significant contribution to the fields of sentiment analysis and topic modelling, as it successfully combines the two into a single combined topic-sentiment classification model. Though such models already exist, most are based on either the LDA topic generation model, or require pre-separation; as such, none have the capability to specify the expected topics beforehand

when classifying a single dataset. The the supervised model implemented here, named the Supervised Joint Class Model (SJCM) will allow that to happen. Also, in contrast to most of the existing supervised topic models, SJCM can account for the correspondence between class labels and data in a hierarchy.

Third, the visualisation represents the third contribution from this report, as relatively little work has been carried out on visualising data produced by similar classification model. The prototype developed demonstrates a novel representation style of topic/sentiment.

1.3 Report structure

This project report will follow the following structure:

Section 1: This **Introduction** section has introduced the relevant concepts, explained the problem that this project aims to solve, and lists what should be achieved by the end.

Section 2: The **Background and Related Works** section discusses previous and current work in this field, and how this work fits in to the general body of research.

Section 3: The **Requirements** section lays out exactly what will be required of the different parts of the project.

Section 4: The **Methodology** section details exactly how the project was undertaken, and what was done in order to achieve the objectives.

Section 5: In the **Results & Evaluation** section, the end results of the programming part of the project will be detailed, and the results of testing the new model against the baseline levels will be discussed.

Section 6: The **Conclusion, Discussion, and Future Work** section will summarise the project as a whole, how successful it has been, and discuss where further work should take the project.

Then follows the references and the appendices, which contain the software manuals and source code listing.

2 Background and Related Works

This section discusses the background to the problems this project hopes to provide a solution for, and discusses previous and current work in the field that is relevant to this project.

2.1 Background

This project is mostly based around the area of contrastive opinion mining, which is rooted in the fields of opinion mining/sentiment analysis and topic modelling, and so work in these areas is most relevant to the tasks at hand.

2.1.1 Opinion mining

Opinion mining (OM), also commonly referred to as Sentiment Analysis (SA), refers to the general problem of extracting subjective information from natural language. It is a major topic in the field of natural language processing, and important because opinions are fundamental to most things humans do, and the choices we make [11].

SA has its origins around the turn of the century. Some of the first work by Turney [32] and Pang, et al [17] involved classifying film reviews into "thumbs up" or "thumbs down", i.e. favourable or unfavourable reviews, respectively. Though this binary model of sentiment cannot capture the range of human opinion possible, it lends itself well to machine learning techniques, and so remains one of the most commonly used approaches today [26]. Turney began with a lexicon-based approach, while Pang used a machine learning (ML) approach, utilising three different algorithms – Naïve Bayes

(NB), maximum entropy classification, and support vector machines (SVM). Lexicon-based approaches typically use a "sentiment lexicon" of words pre-classified with a particular sentiment label, while the ML approach must be trained on a dataset before it can be used on a target dataset. As noted in Pang [17], the ML approach was not as successful at the time, compared with performance on other classification tasks, such as topic classification.

Since then, there has been a huge expansion in the body of work available. A recent survey of approaches by Medhat, et al. [13] details the range of methodologies that have been tested. By far the two most popular approaches remain the ML-based, in particular NB and SVM.

Opinion mining has has remained one of the most important tasks in the field of data mining for the aforementioned reasons, but in general work does not tend to tie in relation of sentiment to other factors, such as topical information.

2.1.2 Contrastive opinion mining

While opinion mining operates at the word, sentence, or document level, contrastive opinion mining seeks to analyse opinion contained in a collection of text, and attempts to present the opinions of viewpoints within these collections, and if possible, quantify their differences [4]. This approach is more useful on very large corpora of text, where manual classification is infeasible. Given the number of huge datasets becoming available – for example, the set of 2.2 million business reviews made publically available by Yelp as part of their Dataset Challenge[1], or the approximately 35 million Amazon product reviews collected by McAuley and Leskovec [12] – tools that can better analyse these sets of data are becoming all the more important to make sense of these ever-growing resources.

[1] https://www.yelp.co.uk/dataset_challenge

2.1.3 Topic Modelling

Topic modelling is a rather different, though related, concept to opinion mining. Though it also attempts to ascertain subjective information from text, its purpose is to identify patterns in a textual corpus [3], in order to discover the hidden thematic structure within [1]. The applications of topic modelling are diverse: the most common being to automatically categorise large collections of documents, but there has been success applying the techniques to varied data types such as social networks, images, and genetic information [1]. A prime example of topic modelling in use is Google News automatically consolidating multiple news articles from different sources into a single story, and those stories further into news categories such as politics, sport, etc.

Among the first approaches to topic modelling were Latent Semantic Indexing (LSI) [18] and Probabilistic LSI (pLSI) [5] models, originally proposed to deal with two problems in information retrieval known as *synonymy* (e.g. searches for "car" do not find "automobile") and *polysemy* (e.g. searches for "surfing" include results about the internet) [18]. The reasoning behind this model was that if documents could be represented by their underlying topics, rather than the words themselves, then these problems in information retrieval could be eliminated. These early topic models assigned one topic to each document. The Latent Dirichlet Allocation (LDA) [2] model, which stemmed from LSI, sought to assign topics to individual words, so that each document could become a mixture of topics. LDA is statistical model that essentially makes the assumption that the words in each document are the products of a known number of unknown topics, and tries to determine which word was generated by which topic. In doing so, the model can discover the topics that each sentence relates to. LDA has since become the de facto base for topic models, and has been modified and extended in many ways to adapt it to particular tasks; those relevant to this project are discussed in the related work section.

2.2 Related works

As the focus of this project is on a mix of contrastive opinion mining and topic modelling, of particular interest are work that include a significant aspect of these.

2.2.1 Topic and sentiment analysis

There is a large body of work to draw from in the domain of SA. In early works, the focus was on classifying documents with sentiment labels at various structural levels; Pang et al. [15,17] and Turney [32] produced models for the overall document level, Täckström and McDonald [28] and Yang and Cardie's [36] models aimed to classify the sentence level, and models for word-level sentiment were proposed by Turney and Littman [33] and Wilson et al. [35].

Though sentiment classification is an important part of this project, the focus is the relation between topic and sentiment in particular. This area has seen a lot of active research. Much of this related work falls under the category of sentiment-topic models, whose purpose is to model sentiments in conjunction with topics, and mainly vary on how the sentiment-topic relation is modelled [6]. These are probabilistic models that generally model topics and sentiments as probability distributions over the dataset, and use these to discover the most likely pairings of sentiment and topic, as in the JST family of models proposed by Lin, et al. [8,9], which extend LDA.

Some work has been done on analysing different opinions and viewpoints on the same topic. A model called the Cross-Collection Mixture model (ccMix) was proposed by Zhai, et al. [37], which makes use of the pLSI topic model. This model takes separate corpora of viewpoints towards a topic, and seeks to find the similarities and differences towards a particular theme or topic. For example, reports on a war from several different news agencies might be compared, each agency's articles being collected and represented as one dataset for each agency. Paul and Girju [19] improved on this model by replacing the pLSI topic model with LDA, leading to the Cross-Collection LSA

model (ccLSA). Fang, et al. [4] takes this concept further with the Cross-Perspective Topic model (CPT), which is able to discover not only similar topics, but corresponding opinions from these viewpoints or perspectives. It is able to do by separating the generation of topic words from the generation of opinion words, which gives distributions well-suited to the task of contrastive OM. However, this separation is based on the assumption that topics are expressed by nouns, and opinions are expressed with adjectives, verbs, and adverbs – a simplification that could reduce effectiveness, as there are many nouns that provide sentimental information (e.g. *hope*, *concern*, *failure*) [16], or verbs that hint at a topic (*scored*, *played*, *graduated*).

A limitation of these cross-collection models is that their input datasets are already separated into different perspectives, in order to find the similarities and differences between them. They would be unlikely to perform well if this data was not pre-separated, as is the case with much of the opinionated data available "in the wild".

Other models focus on the summarising of contentious data and opposing viewpoints. Paul et al. use the Topic Aspect Model (TAM) [20,21] to generate summaries of multiple viewpoints. Trabelsi and Zaïane [31], proposed the Joint Topic Viewpoint (JTV) model to cluster arguing expressions based on the topic of discussion and the viewpoints expressed. The JTV model is very similar to work on the TAM, except that the JTV models the dependency between topics and viewpoints: each document is subdivided into one or more topics, and each topic is subdivided into one or more viewpoints.

2.2.2 Corpus resources

A significant portion of this project involves the collection of new corpus of opinionated data. Much work has been done on creating pure sentiment analysis models and lexica in order to identify opinion and sentiment in general text. Although sentiment is often

intrinsically linked to context and therefore topic, far less work has focussed on creating resources that include topical data as well.

Pang and Lee [15] produced Movie Review Data, a polarity dataset of 1000 positive and 1000 negative movie reviews. This was produced by combining standard machine learning models NB and SVM with a novel "subjectivity detector", which aimed to separate objective statements of fact from subjective opinions. Another popular opinion corpus is the Multi-Perspective Question Answering (MPQA), which provides 10,000 annotated sentences sourced from the world press [34], which was developed in order to detect many different subjective features from text, such as opinions, emotions, sentiments, and speculations.

Datasets have also been produced which aim to cover data from new media such as Twitter[2], such as Pak and Paroubek [14] and Sanders [24]. Others attempt to provide sentiment data for non-textual communication such as emoticons and emoji[3], which are extremely common on such social networks: Read [22] links common emoticons to sentiment labels, and Novak et al. [7] produced the Emoji Sentiment Ranking, a lexicon linking 751 of the most frequently used emoji to sentiment scores.

Though the aforementioned datasets have seen heavy use, they only consider sentiment information, rather than topical. Language resources that model both topic and sentiment are relatively rare. One such resource was created by Takala, et al. [29], who annotated over 9000 sentences over 297 documents with both topic and sentiment information, based on financial news reports collected from Thomson Reuters newswire. Stoyanov and Cardie [27] annotated a documents from the MPQA corpus at the phrase level, by first identifying the opinion in textual data, and then further annotating the topics that constitute the primary information goal of the opinion expressions. However, the resulting corpus is rather small, having only annotated 150

[2] https://twitter.com

[3] http://www.unicode.org/reports/tr51/

documents from the MPQA corpus, and performed their inter-annotator agreement study on 20 documents.

3 Requirements

For the model to be considered a success, it will need to be judged against a set of requirements specified beforehand, which detail exactly what it should be capable of doing. The function and non-functional requirements of the model are detailed in this section.

3.1 Functional requirements

The functional requirements of the system are as follows:

1. The model must be capable of taking in an input dataset consisting of documents labelled on two dimensions, topic and sentiment.
2. The model must produce informational output that allows its performance to be evaluated, e.g. accuracy measures
3. The model must produce output that allows its internal state to be loaded and used again, e.g. a serialised version of itself.
4. The model must be accurate to some degree; the minimum useful accuracy is one that is better than a random selection, but the ideal accuracy will be greater than the baseline provided by other classifiers.
5. The model must be efficient and scalable; some ML datasets have entries reaching into the millions (e.g. Yelp review dataset), and so the more efficient the model, the bigger the datasets that will be available for use with the model.
6. The model must produce output in a format that can be used by the visualisation component.

3.2 Non-functional requirements

1. The model must be able to be used easily from the command line, e.g. via command-line switches or configuration file.
2. If any external modules are used, they must be open source or otherwise freely available.
3. The model should be general enough to work on other datasets; it shouldn't be tied to the particular dataset used in this project.

4 Methodology

The method in this project is divided into three main areas of work. First, as detailed in the objectives (section 1.2), a sufficiently large dataset was collected, structured, and annotated along at least two subjective dimensions. State-of-the-art classification models were run on this dataset (and variations of it) to provide a baseline level of accuracy that could be expected from this dataset, and this was used as a target for the new model to be developed. The second stage was the development of a new classification model that would be able to perform the contrastive opinion mining objective on the collected data. To do this, an existing classification model was selected and adapted to the The model's performance was evaluated and compared to the baseline, and results reported. The final stage was the development of a simple visualisation for the output of the model.

4.1 El Capitan Dataset

This section describes how the dataset was identified, collected, and structured.

There are several reasons why the decision was made to create a new dataset, rather than re-use an existing one. Firstly, in order to be able to train, test and evaluate the performance of a contrastive opinion mining model, a dataset that includes both topic and sentiment information would be required. As discussed briefly in the related works section (2.2.2), datasets that include this level of information are rare, and those that do exist are either too small, do not contain detailed enough labels (e.g. document-

level only), or are not reliable enough (e.g. automatically classified by sum of sentiment values of words, or low inter-annotator agreement levels).

The dataset collection part of this project has been included as a contribution to a work being published separately, as a more in-depth analysis and discussion of the creation and make-up of the dataset, in Ibeke, Lin, Coe, et al. [6]. The work that follows in this section makes up my contribution to the above paper.

4.1.1 Corpus selection and collection

The first task to be completed was the identification of a suitable corpus from which a usable dataset could be produced. The data would have to contain enough individual documents for training, and testing of a ML classifier to adequately take place, so ideally would need to be on the scale of several thousand documents. The documents would also need to be able to be categorised along two dimensions, sentiment and topic – a set of reviews would be an ideal candidate.

Such a prospective dataset was identified in customer reviews of Apple's OSX El Capitan update, released in September 2015, which could be found on the iTunes Store page. This source was selected as there were many thousands of reviews available at the time, were all opinionated in some way, and had potential for each review/sentence to be categorised into a well-defined topic (e.g. performance, compatibility, etc.), and importantly had potential to provide an insight into how public opinion is created and evolves around the release of a new software product.

A data acquisition script was created to collate the contents of the reviews and related metadata from the relevant iTunes store review page, in order to provide the raw data to work with. This raw data was then processed, and assembled into a structured format, including breakdown by sentence, so as to allow annotation of both review-level and sentence-level sentiment and topic. The raw data available from this stage is listed in Table 2.

The script was written in PHP[4] as it is well suited to parsing web data. The script captured all reviews in raw HTML form from the requested iTunes store fronts (e.g. UK, US, Australia, etc.), parsed the relevant metadata from each one, and output the full set in CSV format. No additional libraries were required, though the script made use of the PHP cURL[5] extension to retrieve the web pages. The data, once downloaded, only needed minor adjustments – the dates and times, and many country names, were all localised to that of the home territory for that iTunes store, and had to be normalised to a standard format.

A summary of the make-up of the dataset is shown in Table 1, giving totals of reviews and individual sentences found in the data. The individual data points available from the raw data are shown in Table 2.

Table 1: Summary statistics of dataset

# Reviews	# Sentences	Avg. Review length	Avg. Sentence length	Total word count
2,232	10,248	77.7	16.7	173,264

Table 2: Available dataset data points

Data point	Purpose
Date	Date the review was posted
Version	Software version review is for
Rating	Rating from 1 to 5
Review title	Title of review
Review	Full text of review
Helpful Votes	Number of people who voted this review "helpful"
Total Votes	Total number of people who voted helpful or unhelpful
Username	Username of reviewer
User page	Store profile page of reviewer
Review ID	Unique ID for this review
Country	The iTunes country storefront the review was posted on

[4] https://secure.php.net

[5] http://php.net/manual/en/book.curl.php

4.1.2 Annotation

The dataset was then filtered for suitable reviews – non-English reviews were removed, leaving 2,232 reviews, consisting of 10,348 sentences, in the dataset. It should be noted that due to the sheer size of the dataset, it would not be possible for me to annotate over 10,000 sentences within the timescale of the project, so the actual annotation task was undertaken by a PhD student who was also interested in studying the dataset for his research, after agreeing on an annotation strategy. The sentences in the set were then annotated for sentiment and topic. The review-level entries essentially aggregate the topics of the constituent sentences, and the review-level sentiment reflects the general attitude towards this aggregated topic.

It should be noted that while each review in the dataset did include a user-provided 5-point star rating at the overall review level, this was not used to inform the sentiment annotation. While most studies that work on 5-point ratings tend to translate 1- and 2-star ratings into negative sentiment, and 4- and 5-star ratings into positive, 3-star ratings are usually ignored as they tend to be rather ambiguous, or inconsistently used by different reviewers for either positive or negative reviews. It may be interesting to at some point study the correlation between 3-star ratings and different sentiment ratings.

4.1.3 Baseline classification standard

Once the dataset was collected and finalised, two classifiers were chosen to conduct automatic sentiment classification on both the topic and sentiment dimensions. The classifiers were chosen to represent the current state-of-the-art in opinion mining.

The classifiers chosen for this part of the task were Multinominal Naïve Bayes (NB) and Support Vector Machine (SVM). These classifiers are two of the most popular currently methods currently in use [13]. They have been consistently shown in studies

to be capable of providing good results when operating on good quality input data [23,30]

A further classification model, the Recursive Neural Tensor Network (RNTN), had some promising results being reported on text classification tasks [25], and so was evaluated for inclusion to provide a further spread of techniques. However, its performance on the El Capitan dataset was much poorer than the others – giving an accuracy of around 40% – possibly because it expects training input in the form of a tree with labelled nodes, which the EC dataset does not have. For these reasons, the RNTN was dropped from the baseline classifiers.

To get a good variation of approaches for the baseline, the dataset was separated into two separate sets; one with each review (containing multiple sentences) labelled with the overall review sentiment, and another where each sentence was taken separately, and labelled individually. This gave four separate training and testing pairs, as represented in Table 3 to Table 6.

The training and testing was performed using Weka 3, a software package that contains many machine learning algorithm implementations and dataset processing tools. To process the dataset into a format that the statistical classifiers could use, each review or sentence was converted into a word vector of 1-, 2-, and 3-grams, with word occurrences represented using TF-IDF frequencies. All tests used a 5-fold cross-validation with average scores for each reported in order to provide reliable results. In order to fully see the differences between variants of the above sets, the classifications were repeated as above, but with all neutral sentiment reviews/sentences removed (leaving only positive and negative), and again with all "n/a" topic labelled reviews/sentences removed. Each of these variations markedly improved performance of each classifier, as seen in the aforementioned tables, with the the best performer (NB, trained on sentences and tested on reviews) increasing from around 84% to 92% accuracy.

On the overall classifier level, the NB model consistently performed better than the SVM in the sentiment classification tasks, outperforming it in all but one of the tests by anywhere from 0.4% to 5.5% accuracy, and generally performing best when trained on sentences and tested on reviews. However, SVM generally outperformed NB on the topic classification tasks, and typically performed better on these tests when both trained and tested at the sentence level. This could suggest that SVM handles large numbers of classes better than NB. NB was also much quicker at both training and testing, though this was not explicitly measured; NB typically trained on the full dataset in sub-second times, with tests completing in less than 10 seconds. SVM, on the other hand, typically had to train on each cross-fold for up to several minutes, and up to two minutes again on testing.

Training on sentence-level sentiment and testing on full reviews provided the best scores for both classifiers (accuracy from 84.5%–92.5%). Testing on individual sentences produced relatively poor results for both classifiers, giving an accuracy level approximately 20% lower than when trained on sentences and tested on full reviews. The poor performance in this case was not seemingly affected by whether the model was trained on sentences or full reviews. Using reviews as both training and testing set gave results that fell in between those two scenarios. Using the 2-valued sentiment set rather than 3-valued produced a marked rise in measurements across the board for all tests. Overall, the best performing setup for classification was a NB classifier trained on sentences and tested on full reviews, using the 2-valued sentiment set. This arrangement gave an average accuracy of 92.5%.

In order to test the NB and SVM classifiers on the combined sentiment and topic set, a slight modification was needed: as these classifiers are only capable of classifying a single attribute at a time, the two labels were merged into one, e.g. a review tagged as positive sentiment relating to the "office" topic was given a single "positive-office" label. This enabled the combined testing to take place. Given that these classifiers are not

capable of modelling any kind of link between the two labels, it might be expected that the results would be simply a combination of the results from the previous round. For example, if a classifier scored 50% accuracy on both topic and sentiment, it might be expected that it would score 25% when combined (50% × 50%).

After carrying out the tests on the combined set, the results of which are in Table 7, which will be used to define the baseline standard, it can be seen that the best performer is the NB classifier, getting on average 47.0% accuracy when trained on sentences and tested on reviews. The accuracy, however, dropped to 28.4% when trained on reviews, regardless of test set. These results are just slightly better than what would be expected from a simple product of the two rating dimensions.

Given the above results, in order to provide the closest comparison, the baseline will have to be selected depending on the training and testing method of the new classifier: when trained and tested on sentences, the baseline accuracy will be **32%**. When trained on sentence and tested on reviews, the baseline will be **47%**. When trained on reviews, the baseline will be **28%**.

*Table 3: Baseline sentiment classification results considering **positive**, **negative**, and **neutral** classes*

Train on	Test on	Naïve Bayes				SVM			
		Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Sentences	Sentences	66.4	67.0	66.4	66.6	63.2	63.8	63.2	63.5
	Reviews	84.5	90.2	84.5	86.9	**87.5**	89.5	87.5	88.3
Reviews	Sentences	66.0	62.3	66.0	62.5	64.2	61.4	64.2	57.4
	Reviews	82.4	84.5	82.4	83.4	79.3	77.9	79.3	78.6

*Table 4: Baseline sentiment classification results considering only **positive**, **negative** classes*

Train on	Test on	Naïve Bayes				SVM			
		Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Sentences	Sentences	82.0	82.5	82.0	82.2	79.6	79.8	79.6	79.7
	Reviews	**92.5**	92.5	92.5	92.5	92.1	92.0	92.1	92.0
Reviews	Sentences	83.3	83.9	83.3	83.5	79.3	78.6	79.3	78.3
	Reviews	88.3	88.3	88.3	88.3	82.8	82.5	82.8	82.6

Table 5: Baseline topic classification results for all topics

Train on	Test on	Naïve Bayes				SVM			
		Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Sentences	Sentences	52.3	60.0	52.3	53.1	**68.3**	68.1	68.3	67.8
	Reviews	51.7	63.5	51.7	53.4	50.3	64.6	50.3	51.7
Reviews	Sentences	35.8	44.3	35.8	33.0	37.4	58.5	37.4	37.3
	Reviews	34.0	36.2	34.0	34.3	46.8	45.8	46.8	45.3

Table 6: Baseline topic classification with n/a and low-occurring topics removed

Train on	Test on	Naïve Bayes				SVM			
		Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Sentences	Sentences	61.1	62.6	61.1	61.4	**69.8**	70.9	69.8	69.9
	Reviews	55.7	67.1	55.7	58.0	51.9	69.0	51.9	55.0
Reviews	Sentences	47.6	46.0	47.6	44.3	47.0	60.6	47.0	43.3
	Reviews	37.2	40.4	37.2	37.9	52.1	52.2	52.1	50.8

Table 7: Baseline combined sentiment and topic; full testing sets used

Train on	Test on	Naïve Bayes				SVM			
		Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Sentences	Sentences	32.8	35.7	32.8	32.9	40.6	41.4	40.6	40.4
	Reviews	**47.0**	65.7	47.0	50.0	45.9	64.1	45.9	49.8
Reviews	Sentences	28.4	34.3	28.2	24.1	22.6	43.7	22.6	20.7
	Reviews	28.4	29.0	28.4	28.2	35.8	35.0	35.8	33.6

4.2 Supervised Joint Class Model

The main task of the project is the development of a new classification model that is able to perform the task of contrastive opinion mining, and also conform to the requirements specified in section 3. In order to do this, it must be able to classify

documents on both topic and sentiment aspects at the same time, and it must be capable of using these classifications to automatically extract contrasting opinions from the input text. Developing an entirely new model from scratch would be outside the scope of this project, so a suitable candidate would have to be found to adapt to the new style of classification.

4.2.1 Selection of model for adaption

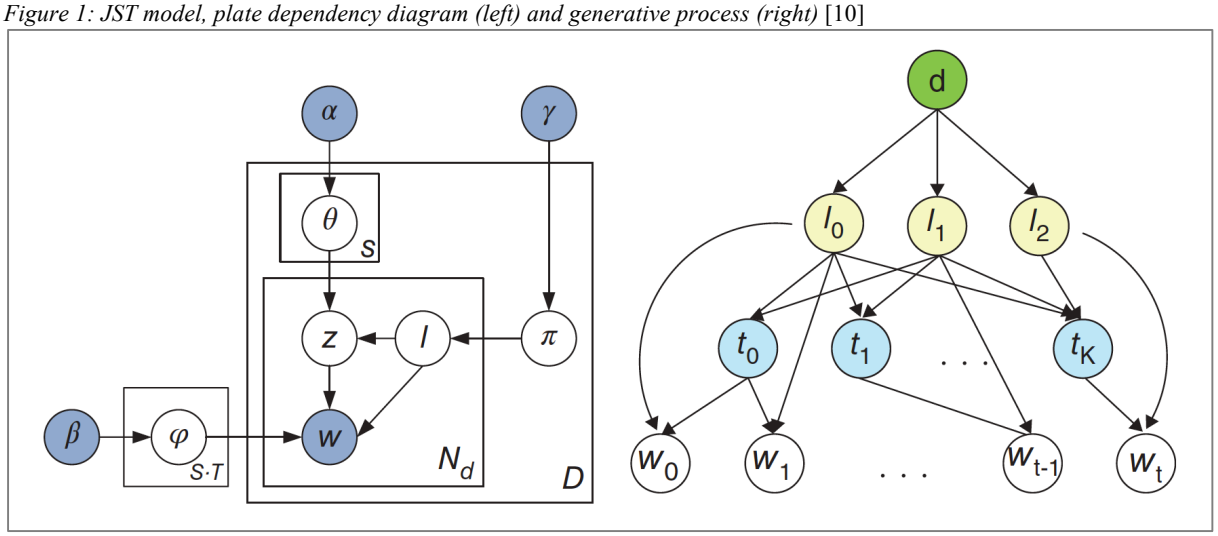
There are several classification models with open source implementations available which focus on the general task of topic and sentiment. The related work on sentiment-topic classifiers, which might provide the is detailed in section 2.2.1. Potential models that could be adapted include JST, ccMix, ccLSA, CPT, JTV, and TAM. Most of the classification models of interest to this project base their topic modelling component on the LDA model.

Out of all the options, JST was selected for several reasons. The primary reason was that the JST model is already capable of classifying on both topic and sentiment at the same time, and producing top words and sentences for sentiment-topic pairs, so the work required to fully meet the requirement would not be too heavy. Its source code was also readily available. Furthermore, the paper on JST [9] suggests a variation, Reverse-JST, which would provide an even closer match to what could be required to implement the requirements.

In JST, an extra sentiment layer is added to the three already covered in LDA, as can be seen in Figure 1. LDA uses three hierarchical layers, where topics are associated with documents, and words are associated with topics. JST adds a sentiment layer between documents and topics, so that each sentiment value has its own topical information contained within it. JST also adds a weak supervision of sentiment method, whereby a sentiment word list is used to inform the sentiment predictions. The advantage of this model is that it can achieve higher than average accuracy when

applied to datasets from different domains [9], which is a useful attribute for when classifying on multiple topics.

Figure 1: JST model, plate dependency diagram (left) and generative process (right) [10]



4.2.2 The need for adaption

The JST model, however, does not quite fit the requirements to be used directly for this project. Firstly, as JST is weakly supervised on the sentiment level only, there is no method available to train the model on a dataset on the topic level. Weakly supervised models cannot exploit the class label information directly from the training dataset. Rather, labelled features are used (in this case, a lexicon containing sentiment probabilities for common words) a form of supervision for guiding the classification for each training instance.

As a further hurdle, the LDA method of topic generation used by JST is unsupervised and purely generative, and so has no inherent way of providing prior information on which words relate to which classes, which is a requirement of the model to be created. There are positives and negatives associated with this approach: the generative method allows one to specify the number of topics required, and the data will be processed into exactly that many groupings. However, these groupings will not be named, and could change between runs, so the only way to tell what they refer to is by manual inspection of the results. This makes it difficult to automatically visualise the topics if visual

inspection by a human is required to identify them. With a supervised approach, the topics are fixed, but it should be far easier to see which words and labels relate to which topics.

Removing the dependency on the sentiment lexicon would also allow the model to make use of datasets with with different number of sentiment levels – for example, if instead of positive and negative, the dataset modelled weak and strong versions of these also.

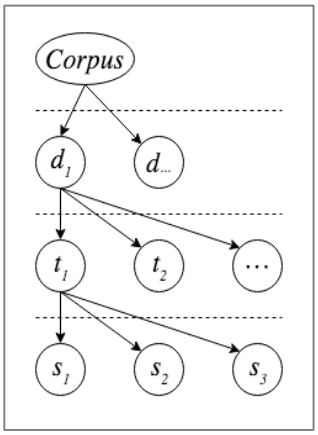
Finally, as a problem specific to JST, due to topics being a sub-layer of sentiment, there is no guarantee that *<sentiment 1 → topic 1>* is directly comparable to *<sentiment 2 → topic 1>*, as each topic model is generated for each sentiment label. The solution to this is mentioned briefly in the JST paper, and is described as Reverse-JST (R-JST) [9]. In this re-parameterisation of the JST model, the topic layer exists directly under the document layer, under which each topic has a sentiment layer. The arrangement allows direct comparison of topics and the sentiment makeup of each.

In order to address the above problems, the new classifier model will be fully supervised at both the sentiment and class levels. This means the model will learn the sentiment labels and topics to use directly from the training data, and in doing so allows the output to make more sense, and be more easily interpreted. The problem with the ordering of the layers will be resolved by basing the new model on the Reverse-JST variation, rather than JST.

4.2.3 Model conceptual design

Therefore, it was decided that the model to be created would be an adaption of the concept of R-JST, transformed into a fully-supervised model, taking all of its training information directly from the input dataset. As this change meant that the model no longer fit within the definition of the R-JST model, the new model was given the name Supervised Joint Class Model (SJCM). It was given a more generic name that didn't relate specifically to topic or sentiment, as with this change to full supervision, it is technically agnostic as to what the classes actually are, as long as they can be represented hierarchically – the pairing of sentiment and topic is just one possible input to this new model. Further work may be able to identify how well it performs on datasets with other hierarchical classification information.

Figure 2: The Corpus, document, topic, and sentiment layers of the R-JST and SJCM models

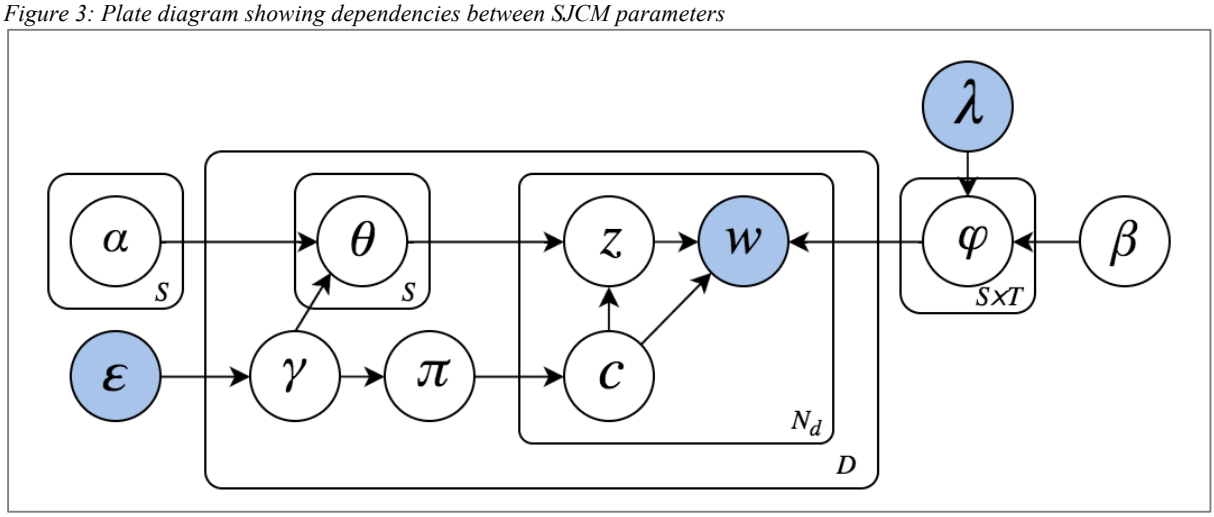


As the model is based on (Reverse-)JST, the conceptual design is based on the workings and theory of JST, so the diagrams and formulae that describe JST have been updated and are presented in the following [9,10].

The new SJCM model can therefore be described as a supervised generative topic model for text classification, which extends LDA [2] as shown in Figure 3. SJCM accounts for document labels during the generative process, where each document can associate with a single class label or multiple class labels. In contrast to most of the existing supervised topic models [1], SJCM can account for the correspondence between class labels and data in a hierarchy.

A full listing of the new model parameters, as adapted from JST, are given in Table 8.

Figure 3: Plate diagram showing dependencies between SJCM parameters



Gamma becomes a matrix to encode two different set of labels.

4.2.3.1 Incorporation of supervision information

SJCM incorporates the supervised information from document sentiment labels by constraining that a training document can only be generated from the topic set with sentiment labels corresponding to the document's observed label set. This is achieved by introducing a dependency link from the document label matrix ε to the Dirichlet prior γ. Suppose a corpus has 2 unique sentiment labels denoted by $S = \{s_1, s_2\}$ and for each label s_k there are 5 topics denoted by $\theta_{s_k} = \{z_1, s_k, \dots, z_5, s_k\}$. Given document d's observed label vector $\varepsilon_d = \{1,0\}$ which indicates that d is associated with sentiment label s_1,we can encode the label information into SJCM as $\gamma_d = \varepsilon_d^T \times \gamma$, where $\gamma = \{\gamma_1, \gamma_2\}$ is the Dirichlet prior for the per-document-per-topic sentiment proportion $\pi_{d,z}$ and $\gamma_d = \{\gamma_1, 0\}$ is the modified Dirichlet prior for document d after encoding the sentiment label information. This ensures that d can only be generated from topics associated with class label s_1 restricted by γ_1.

4.2.3.2 Model inference

From the SJCM graphical model depicted in Figure 3, the joint distribution of all observed and hidden variables can be factored into three terms:

$$P(w,z,c) = P(w|z,c)P(z,c) = P(w|z,c)P(z,c)P(c) = \quad (1)$$

$$\int P(w|z,c,\Phi)P(\Phi,\beta)d\Phi \cdot \int P(z|c,\Theta)P(\Theta|\alpha)d\Theta \cdot \int P(c|\Pi)P(\Pi|\gamma)d\Pi \quad (2)$$

By integrating out Φ, Θ, and Π in the first, second and third term of the above respectively, we can obtain

$$P(w|z,s) = \left(\frac{\Gamma\left(\sum_{i=1}^{V}\beta_{k,j,i}\right)}{\prod_{i=1}^{V}\Gamma\left(\beta_{k,j,i}\right)}\right)^{S\times T}\prod_{k}\prod_{j}\frac{\prod_{i}\Gamma\left(N_{k,j,i}+\beta_{k,j,i}\right)}{\Gamma\left(N_{k,j}+\sum_{i}\beta_{k,j,i}\right)} \quad (3)$$

$$P(z|s) = \left(\frac{\Gamma\left(\sum_{j=1}^{T}\alpha_{d,k,j}\right)}{\prod_{j=1}^{T}\Gamma\left(\alpha_{d,k,j}\right)}\right)^{D\times S}\prod_{d}\prod_{k}\frac{\prod_{j}\Gamma\left(N_{d,k,j}+\alpha_{d,k,j}\right)}{\Gamma\left(N_{d,k}+\sum_{j}\alpha_{d,k,j}\right)} \quad (4)$$

$$P(s) = \left(\frac{\Gamma\left(\sum_{k=1}^{S}\gamma_{d,k}\right)}{\prod_{k=1}^{S}\Gamma\left(\gamma_{d,k}\right)}\right)^{D}\prod_{d}\frac{\prod_{k}\Gamma\left(N_{d,k}+\gamma_{d,k}\right)}{\Gamma\left(N_{d}+\sum_{k}\gamma_{d,k}\right)} \quad (5)$$

where $N_{k,j,i}$ is the number of times word i appeared in topic j with class label k, $N_{k,j}$ is the number of times words are assigned to topic j and class label k, $N_{d,k,j}$ is the number of times a word from document d is associated with topic j and class label k, $N_{d,k}$ is the number of times class label k is assigned to some word tokens in document d, N_d is the total number of words in document d and Γ is the gamma function.

The main objective of inference in SJCM is then to find a set of model parameters that can best explain the observed data, namely, the per-document topic proportion π, the per-document topic-specific sentiment label proportion ϑ, and the per-corpus word distribution φ.

Table 8: SJCM model parameters and variables (adapted from JST [10]*)*

Variable	Description
D	The number of documents in the corpus
N_d	The number of words in document d
V	The number of unique words in the corpus
S	The number of sentiment labels

T	The number of topic labels
α	Asymmetric Dirichlet priors on the mixing topic proportions. $\boldsymbol{\alpha} = \left\{\left\{\left\{\alpha_{d,z,l}\right\}_{l=1}^{S}\right\}_{z=1}^{T}\right\}_{d=1}^{D}$ (a $D\times T\times S$ matrix)
β	Asymmetric Dirichlet prior on the sentiment label and topic specific word distribution. $\boldsymbol{\beta} = \left\{\left\{\left\{\beta_{z,l,i}\right\}_{z=1}^{T}\right\}_{l=1}^{S}\right\}_{i=1}^{V}$, (a $V\times S\times T$ matrix)
γ	Asymmetric Dirichlet prior on the mixing sentiment proportions $\boldsymbol{\gamma} = \left\{\left\{\gamma_{d,z}\right\}_{z=1}^{T}\right\}_{d=1}^{D}$ (a $D\times T$ matrix)
π_d	Parameter notation for the topic mixing proportions for document d and topic z (T– vector). For D documents, $\boldsymbol{\Pi} = \left\{\left\{\pi_{d,z}\right\}_{z=1}^{T}\right\}_{d=1}^{D}$ (a $D\times T$ matrix)
$\theta_{d,z}$	Parameter notation for the sentiment mixing proportions for document d, topic z, and sentiment l (S– vector). For D documents, T topics, and S sentiment labels, $\boldsymbol{\Theta} = \left\{\left\{\left\{\theta_{d,z,l}\right\}_{l=1}^{S}\right\}_{z=1}^{T}\right\}_{d=1}^{D}$ (a $D\times T\times S$ matrix)
$\varphi_{l,z}$	Parameter notation for the multinomial distribution over words for sentiment label l and topic z (V– vector). For S sentiment labels and T topics, $\boldsymbol{\Phi} = \left\{\left\{\left\{\varphi_{l,z,i}\right\}_{l=1}^{S}\right\}_{z=1}^{T}\right\}_{i=1}^{V}$ (an $S\times T\times V$ matrix)

4.2.4 Implementation of SJCM

Unfortunately, there was no implementation of Reverse-JST available to adapt directly, so the new SJCM model would need to be implemented by modifying the JST code. A number of modifications were made to the JST model in order to implement SJCM.

The source code for the JST was kindly made available to me for this project by Dr Lin. It is a command-line based program, written in C++. The project code is split into several classes (see Appendix C.3). The main bulk of changes were made to the Dataset and Model classes, and minor changes were made to the Document class to accommodate storing the topic label as well as sentiment. The same general program architecture and structure was still usable, so the class relationships and general method of execution were left as-is.

4.2.4.1 Dataset class modifications

The Dataset class is found in the source in `dataset.cpp` and `dataset.h`.

The first change to the model was to allow it to work on the El Capitan dataset that was collected in the first part of the project. Initially, the El Capitan dataset was structured into a format that matched the structure expected by JST as closely as possible. The initial dataset was structured rather simply into a document ID followed by each of the document's sentences on each line, with no metadata about the document included. This format was not suitable for the SJCM implementation, as both the prior sentiment and topic information for each sentence would need to be included in the input data. To accommodate this information, the input format was modified to treat one sentence as one document – partly because training at the document level was shown in the baseline to provide poor results. The new structure uses two lines for each document – the first line containing the metadata consisting of document ID, sentiment label, topic label, and training or testing set designation.

The document IDs are set to be of the form "D_S", where D is the original review ID, and S refers to the sentence number from that review, so the required information is still present should the need to group sentences into their original full reviews be needed. Both the old format and the new include the number of documents contained within the set as the first line.

The train/test set designation was added to allow testing of the model in the same way as the baseline was tested, as a 5-fold cross-validation. It became apparent that this method of designating data was unmanageable if it needed to be changed, so the dataset reading code was further modified to include an "auto" designation, which would automatically split the training and testing data into a specified cross-fold, enabling much faster cross-validation results. Further work could adjust these folds to be stratified to improve reliability.

4.2.4.2 Model class modifications

The model class is found in `model.cpp` and `model.h`. It required much more extensive changes to be adapted into the new model.

The main fundamental change is that where previously counts and distributions were held in layers ordered by documents → sentiment → topic, these have been reversed to become documents → topics → sentiment. The parameters and hyper-parameters of the model were updated to match those expected by and similar to R-JST, in order to model the new hierarchy. The changes of hyper-parameter dimensions are detailed in Table 9.

Table 9: Model parameter dimensions in JST vs. SJCM

Parameters	**JST Dimension**	**SJCM Dimension**
Alpha α	S × T	D × T × S
Beta β	S × T × V	T × S × V
Gamma γ	D × S	D × T
Pi π	D × S	D × T
Phi φ	S × T × V	T × S × V
Theta ϑ	D × S × T	D × T × S
Lambda λ	S × V	Unused

Where D = no. of documents, T = no. of topics, S = no. of sentiment labels, V = no. of unique words

The next major change was to remove the weak supervision by means of the sentiment lexicon and replace this with full supervision based on the input dataset. To make this change, α, in which the new model stores the sentiment distribution per topic per document, has its topic distributions set directly from what is observed in the training set values. The β hyper-parameter is kept of the same size, but the order is reversed so that sentiment counts are now kept under topics. For β, this is a purely structural change, and has no effect on the numbers that would be contained within it (the number of times a word appears with a particular sentiment and topic is the same as the number of times it appears with a particular topic and sentiment). A similar change is made to φ, which again does not affect its function.

The λ parameter, which stores the prior vocabulary sentiment information from the sentiment lexicon, is removed in SJCM, for two reasons. The first is that as the sentiment information is now directly observed from the input dataset, and so it is not strictly necessary. The second in the as the model now does not make assumptions about the number of either sentiment or topic classes, using a sentiment lexicon of fixed size no longer makes sense, even to provide an additional level of sentiment information.

The Gibbs sampling procedure of JST is modified in a few ways. The updated algorithm and sampling equation are shown in Figure 4 and Equation (6). The main changes, aside from being updated to use the new parameter dimensions, are that instances designated to be in the training set are skipped from the sampling, and have their topic and sentiment labels set directly. This leaves the sampling to work only on the test set where the actual labels are hidden from the model.

4.2.4.3 Inference class modifications

The inference class is the class of the model that performs inferencing on new data after the dataset had been trained upon. The changes to this model were very much equivalent to those from the model class, as they have very similar functions. The inference class parameters were simply updated to the new dimensions as mentioned above.

Figure 4: Gibbs sampling algorithm for SJCM, based on that of R-JST [9]

Algorithm 1. Gibbs sampling procedure of SJCM.
Require: α; β; γ, Corpus
Ensure: sentiment and topic label assignment for all word tokens in the corpus
1: **Initialize** T × S × V matrix Φ, D × T × S matrix Θ, D × T matrix Π.
2: **for** $i = 1$ to *max* Gibbs sampling iterations **do**
3: **for** all documents $d \in [1, M \mid test\ set]$ **do**
4: **for** all words $t \in [1, N_d]$ **do**
5: Exclude word t associated with sentiment label l and topic label z from variables N_{jki}, N_{dj}, and N_d
6: Sample a new sentiment-topic pair l' and z' using Equation (6)
7: Update variables $N_{jki}, N_{jk}, N_{djk}, N_{dj}, N_d$ using the new sentiment label l' and topic label z'
8: **end for**
9: **end for**
10: **for** every 25 iterations **do**
11: Update hyperparameter α with the maximum-likelihood estimation;
12: **end for**
13: **for** every 100 iterations **do**
14: Update the matrix Φ, Θ, and Π with new sampling results;
15: **end for**
16: **end for**

$$P(z_t = j, l_t = k | w, z^{-t}, l^{-t}, \alpha, \beta, \gamma) \propto \frac{N_{jkw_t}^{-t} + \beta}{N_{jk}^{-t} + V\beta} \cdot \frac{N_{djk}^{-t} + \gamma}{N_{dj}^{-t} + S\gamma} \cdot \frac{N_{dj}^{-t} + \alpha_j}{N_{d}^{-t} + \sum_j \alpha_j} \quad (6)$$

4.2.4.4 Output modifications

The output of the model was also updated in several ways. The main output of model parameters was updated to reflect the new parameters mentioned above. Additional output files were also added. First, a form of the word-topic-sentiment assignments for each document was output in JSON[6] format, so that the data may be easily imported into a standard format for the visualisation part of the project. Secondly, an evaluation method was written that would compare the predicted classes to those estimated by the model, and output some simple statistics on how accurate the estimation was. These statistics would allow the model to be compared against the baseline values taken from the reference classifiers.

Once all the relevant modification had been made to the model, its performance was tested, results of which are reported in Results & Evaluation section.

[6] http://www.json.org

Figure 5: Screenshot of the prototype visualisation component

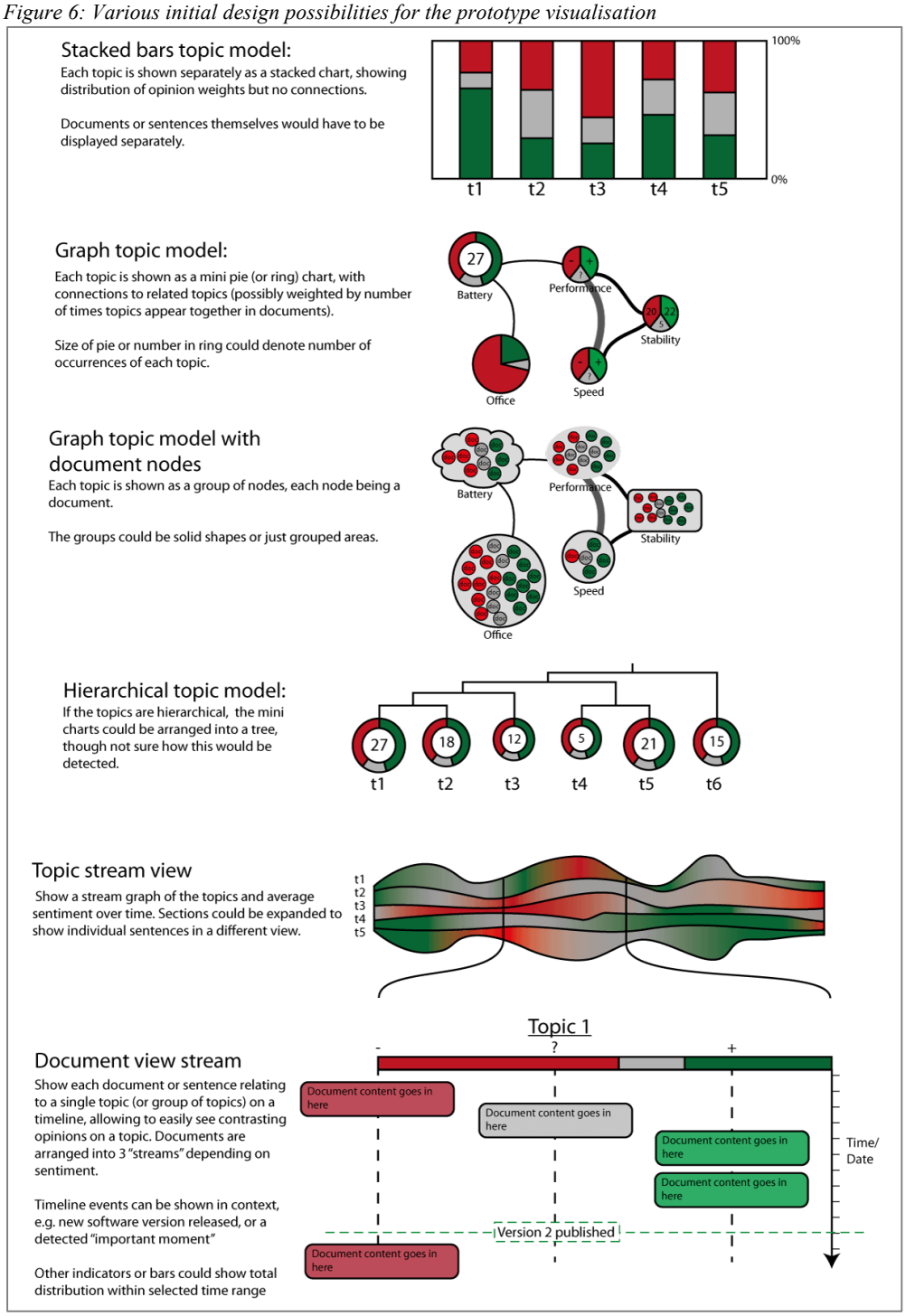
4.3 Visualisation

In order to best show the results, a simple visualisation was produced that produces a visual summary of the output of the classifier.

The main aspects of the output visualisation are that it should be possible to see at a glance are the identified topics, the relationships between them, the size of each topic (i.e. how many documents are in it), and the balance of positive, negative, and neutral labels within that topic. A number of different styles were sketched, as can be seen in the design options document shown in Figure 6, and a design was settled on to prototype that consisted of a "doughnut" chart for each topic, scaled by number of appearances of that topic, with each chart showing the sentiment distribution within

each topic. This design, was chosen as it provided a simple overview that was easy to understand. A screenshot of the working version of this is shown in Figure 5.

Figure 6: Various initial design possibilities for the prototype visualisation



The visualisation itself was written in Processing 3[7], a data visualisation framework based in Java. The toolkit provides a number of methods that allow for custom graphics to be built up. Processing 3 was chosen because it provides good tools and methods for displaying any kind of graph or chart, though its usage is not limited to such. It can also easily export cross-platform ready executables, which greatly simplifies the process of running the visualisation – all that is needed to run the visual is that it is running in the same directory as the classifier output.

In order to get the data from the classifier into the visualizer, a custom JSON export function was added into the many outputs from the classifier, which could then be easily and natively imported by the Processing code.

Due to the time constraints of this project, further visualization patterns that were planned were not able to be completed. Future work on this classifier would be good to improve the visual offering from the visualizer.

4.4 Development process & tools

Due to having several different stages of the project, the development process was conducted using several different pieces of software. The data collection used a PHP script to collect the data, and the resulting data was stored in a MySQL[8] database for ease of processing and exporting to other formats.

The baseline calculation and analysis stage used Weka 3[9], a collection of machine learning algorithms and models, in order to pre-process that dataset and produce the baseline results.

[7] https://processing.org

[8] https://www.mysql.com

[9] http://www.cs.waikato.ac.nz/~ml/weka/

The model itself was programmed in C++, using the CLion IDE produced by Jetbrains[10], which provided extensive debugging capabilities, and additionally provided source control based on Git[11]. All major changes to the model were tracked in source control to ensure that differences to the original model could be monitored, and any bugs that were introduced could be identified.

The visualisation component was written in Processing 3, a data visualisation-oriented package in Java. Though this introduced yet another language to the many already in use for the project, the trade-off for simplicity of use and development time saved more than makes up for it. Processing 3 additionally makes it very easy to export the visualisation component to all major systems (Windows, OSX, Linux, each 32- and 64-bit).

[10] https://www.jetbrains.com/clion/

[11] https://git-scm.com

5 Results & Evaluation

This section discusses the outcome of the tasks, and the success of each.

5.1 Dataset results

The results of the dataset collection were a success; a large dataset was able to be produced which, in addition to providing a good basis for this project's experiments, will have uses applicable to many other experiments, due to its forthcoming publication. All of the original metadata has also been retained in the dataset, making a much richer resource than if it had been used exclusively for this project, and as such provides an excellent basis for many more projects either using the data to develop new classification models, or for investigations on the dataset itself, to provide insight into the subject matter.

5.2 SJCM testing results

The programming on the SJCM classifier was successfully completed, and a functional classifier according to the proposed model was produced. The process took much longer than originally expected, due to some difficulties in the transition from the JST model to the SJCM – for example, the π model parameter was originally given the wrong dimension, and it took a long time to track down this as the cause of the problems. The evaluation component also ran into some problems, when it was reading the predicted sentiment incorrectly, thus making the results look a lot worse than they

were. This, too, took a long time to identify and address. Eventually, however, all of these difficulties were overcome, and a novel sentiment-topic classification model was produced.

In order to evaluate the performance of the new SJCM classifier, the results of training and testing the model against the El Capitan dataset were measured and compared to the baseline standard described in section 4.1.3 above. The SJCM model in its current form both trains and tests on sentences, using the full dataset, rather than any filtered for "n/a" or low-occurring classes. Therefore, the baseline combined labels accuracy will be used, which will be 32%. Further modifications to the operation of the model may make creating these filtered sets more feasible to train and test on.

The first test for the model is that is accurate in its predictions of sentiment and topic labels. As can be seen in the test results in Table 10, the model performed well on the classification task, with accuracy levels on 61.7% for sentiment labels, 44.4% for topic labels, and 28.3% for both combined, when averaged over the 5-fold cross-validation. This is a good result that shows that the model is working well, as the results are far greater than would be expected of a "random" classifier.

Table 10: Testing evaluation results of the SJCM classification model

	SJCM Performance (%)			
	Accuracy	Precision	Recall	F1
Sentiment	61.7	64.1	61.7	62.7
Topic	44.4	47.0	44.4	45.0
Both combined	28.3	30.5	28.3	28.8

Despite the results being better than random, the results are unfortunately not quite as good as the baseline set by the NB model, though they are fairly close, coming within 4% to 6% of the baseline measure for each test. It is difficult to tell whether this discrepancy is down to running the model for too many or too few iterations, poor choice of model parameters or hyper-parameters, poor implementation in code, or better pre-processing techniques for NB and SVM (e.g. TF/IDF scores and included

bi- and tri-grams). Further work should be carried out on the SJCM model to tune the accuracy, with the goal of matching or surpassing the baseline classifiers.

Aside from the accuracy levels, the main objective for the model is the ability to perform contrastive opinion mining. Its ability to do this well cannot be quantified as easily as label accuracy, but inspection of some of the outputs of the model show that it is capable of producing a good set of contrastive words and sentences for each topic. Table 11 shows five topics extracted from the 19 classified, and the positive and negative words groups underneath them. From these groupings, it is clear that certain themes and topics emerge, and that they are informative and coherent; for example, in the **Safari** topic, it is easy to see words in the topic jumping out as specific to the theme of Safari. Positive words mention *quicker*, and *loading*, which would indicate that the new update is running faster. However, the negative topic words include *slow* and *freez*, among others, which could indicate stability issues.

Table 11: Top 10 words of positive and negative topics extracted by SJCM

Performance		**Office**		**Safari**		**Mail**		**Outlook**	
+	−	+	−	+	−	+	−	+	−
work	crash	offic	offic	safari	safari	mail	mail	outlook	outlook
run	work	microsoft	use	great	use	folder	email	offic	work
perform	time	compat	work	improv	chrome	famili	account	initi	crash
faster	app	crash	microsoft	font	freez	messag	work	microsoft	use
app	use	fine	ms	new	time	work	appl	mbp	microsoft
use	slow	work	crash	load	open	email	lost	latest	open
new	mac	issu	updat	better	screen	mailbox	send	ms	el
pro	open	updat	word	littl	work	select	problem	work	capitan
macbook	freez	new	excel	quicker	slow	use	mac	good	mail
better	just	didn	appl	run	download	new	folder	vista	problem

The SJCM model is also able to output top-rated sentences for each topic-sentiment pair. This can be used to provide a useful insight into each topic, and perhaps provides a fuller picture than the topic words alone. Table 12 lists several topics and the sentences from the testing set identified for each one. Although there are some ambiguous sentences that might not refer to the topic directly, the sentences generally seem to match up well with the associated topics and sentiments. This is a great

success in terms of the contrastive opinion mining aspect – these extracts prove that the model is indeed capable of contrastive opinion mining, and presenting the opinions it has mined from the text in a contrastive manner.

Table 12: Top sentences extracted by SJCM

Topic	Sentiment	Extracted sentence
Safari	**Positive**	run great issu
		wait fix mail
		safari faster respons
		safari fix quicker
		safari vastli improve
		long await split view fantast safari satisfi use
		safari cool new trick sleev avail updat yosemit make somewhat moot point
	Negative	safari worst
		safari slower load
		constantli restart laptop
		safari keep glich
		safari slow constantli freez
		safari ridicul slow awesom
		hope soon updat fix safari
		run safari hour repeatedli freez
Office	**Positive**	work offic
		that offic updat instal new font compat new os
		screen freez app freez just say unhappi
		biggest problem occur updat upgrad softwar suit new
		microsoft slow ensur compat offic suit hang crash occur frequent address
		download new microsoft offic glitch smooth sail
	Negative	use offic
		ms offic work
		new updat excel
		offic good useless
		microsoft offic app open
		ms offic longer work
		offic disast offic better
		microsoft offic complet unstabl
Speed	**Positive**	macbook faster
		imac fast
		definit faster
		faster respons
		fast smooth
		love fast
	Negative	slow work
		mac slow
		mac slow
		run slow
		just slow
		imac slow
		slower start

5.3 Visualisation results

The visualisation component was also a success, though the prototype itself is rather simple. It is capable of reading the output from the SJCM model, and producing a visualisation that provides a quick summary of the distributions of the topics and sentiments within. This is a highly useful addition to the output of the model, as although the contrastive opinions, in the form of both words and sentences, are being successfully, they do not give an indication of the balance of the range of opinions.

6 Conclusion, Discussion, and Future Work

6.1 Conclusion

At the end of the project, examining the goals that were initially, set, it can be seen that overall, the project had excellent outcomes, and were a great success. The project succeeded in attaining the objectives set out in section 1.2, and meeting all the requirements set out in section 3.

A large corpus of opinionated text was created, that was suitable for the classification tasks at hand, and will also be ideal for using in many studies in the future. A new classifier was proposed and implemented that is capable of performing contrastive opinion mining, by classifying a dataset on two hierarchical classes at once, ideal for usage on topic and sentiment-based datasets. The lessons learned from its initial implementation have laid the groundwork for work on this area in the future, and will be useful and applicable to many areas of research once performance is tuned.

Finally, a working prototype visualisation was produced that provides a foundation for more advanced visualisations of topic-sentiment data in the future.

6.2 Discussion

Overall the project and development went extremely well. The development of the classifier hit some problems along the way, detailed in the Results section (5.2) in the

adaption from JST to SJCM, and dealing with these problems took much longer than expected to solve. The problems were eventually dealt with however, and the project was able to continue as planned. This was thankfully down to good initial planning that made time for such complications. that allowed the project to get back on track.

The visualisation, though implemented only as a simple prototype in its current form, has proven useful, as it provides an easy-to-understand visualisation of the output from the classifier, and gives insights into the classified data that was not directly available from the model itself.

A lot was learned from this project, and in a wide variety of areas; diverse areas of study, such as data mining, topic modelling, sentiment analysis, text processing, and data visualisation were touched on during this project.

6.3 Future Work

Each of the three main tasks the made up this project are ripe for further development opportunities. The El Capitan dataset has provided many interesting statistics, and there is opportunity for both further mining of information from the dataset itself, and as a basis for use in the training of future classifiers.

The SJCM model has provided a novel and interesting approach to the contrastive opinion mining objective, tackling the problem from a new joint topic-sentiment angle. A lot has been learned from the initial implementation, and future work to tune the performance of this classifier, or adapt it to more general usages would be highly useful.

The visualisation provides a high-level overview of the data output by the SJCM classifier, and future work would certainly be called for to expand on this work to provide more visualisations at a more detailed level, to allow interactivity with the data, and to enable switching between many different visualisations with ease.

7 References

[1] D. Blei, L. Carin, D. Dunson, Probabilistic topic models, IEEE Signal Processing Magazine. 27 (2010) 55–65.

[2] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, The Journal of Machine Learning Research. 3 (2003) 993–1022.

[3] M.R. Brett, Topic Modeling: A Basic Introduction, Journal of Digital Humanities. 2 (2012) 12–16.

[4] Y. Fang, L. Si, N. Somasundaram, Z. Yu, Mining contrastive opinions on political texts using cross-perspective topic model, in: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining - WSDM '12, ACM Press, New York, New York, USA, 2012: p. 63.

[5] T. Hofmann, Probabilistic latent semantic analysis, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 1999: pp. 289–296.

[6] E. Ibeke, C. Lin, C. Coe, A. Wyner, D. Liu, M.H. Barawi, N. Fazilla, A. Yusof, A Curated Corpus for Sentiment-Topic Analysis, in: 10th Edition of the Language Resources and Evaluation Conference, Forthcoming., n.d.

[7] P. Kralj Novak, J. Smailović, B. Sluban, I. Mozetič, Sentiment of Emojis, PLOS ONE. 10 (2015) e0144296.

[8] C. Lin, Y. He, Joint sentiment/topic model for sentiment analysis, in: Proceeding

of the 18th ACM Conference on Information and Knowledge Management - CIKM '09, ACM Press, New York, New York, USA, 2009: p. 375.

[9] C. Lin, Y. He, R. Everson, S. Ruger, Weakly Supervised Joint Sentiment-Topic Detection from Text, IEEE Transactions on Knowledge and Data Engineering. 24 (2012) 1134–1145.

[10] C. Lin, E. Ibeke, A. Wyner, F. Guerin, Sentiment-topic modeling in text mining, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 5 (2015) 246–254.

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

[12] J. McAuley, J. Leskovec, Hidden factors and hidden topics, in: Proceedings of the 7th ACM Conference on Recommender Systems - RecSys '13, ACM Press, New York, New York, USA, 2013: pp. 165–172.

[13] W. Medhat, A. Hassan, H. Korashy, Sentiment analysis algorithms and applications: A survey, Ain Shams Engineering Journal. 5 (2014) 1093–1113.

[14] A. Pak, P. Paroubek, Twitter as a Corpus for Sentiment Analysis and Opinion Mining, LREc. 10 (2010) 1320–1326.

[15] B. Pang, L. Lee, A sentimental education, in: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04, Association for Computational Linguistics, Morristown, NJ, USA, 2004: p. 271.

[16] B. Pang, L. Lee, Opinion Mining and Sentiment Analysis, Foundations and Trends® in Information Retrieval. 2 (2008) 1–135.

[17] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP '02, Association for Computational Linguistics, Morristown, NJ, USA, 2002: pp. 79–86.

[18] C.H. Papadimitriou, H. Tamaki, P. Raghavan, S. Vempala, Latent semantic indexing, in: Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems - PODS '98, ACM Press, New York, New York, USA, 1998: pp. 159–168.

[19] M. Paul, R. Girju, Cross-cultural analysis of blogs and forums with mixed-collection topic models, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3, Association for Computational Linguistics, 2009: pp. 1408–1417.

[20] M. Paul, R. Girju, A two-dimensional Topic-Aspect Model for discovering multi-faceted topics, Proceedings of the National Conference on Artificial Intelligence. 1 (2010) 545–550.

[21] M.J. Paul, C. Zhai, R. Girju, Summarizing contrastive viewpoints in opinionated text, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2010: pp. 66–76.

[22] J. Read, Using emoticons to reduce dependency in machine learning techniques for sentiment classification, Proceedings of the ACL Student Research Workshop. (2005) 43–48.

[23] I. Rish, An empirical study of the naive Bayes classifier, IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence. 22230 (2001) 41–46.

[24] N.J. Sanders, Sanders-Twitter sentiment corpus., (2011).

[25] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2013: p. 1642.

[26] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for

classification tasks, Information Processing & Management. 45 (2009) 427–437.

[27] V. Stoyanov, C. Cardie, Topic identification for fine-grained opinion analysis, Association for Computational Linguistics, 2008.

[28] O. Täckström, R. McDonald, Semi-supervised latent variable models for sentence-level sentiment analysis, Association for Computational Linguistics, 2011.

[29] P. Takala, P. Malo, A. Sinha, O. Ahlgren, Gold-standard for Topic-specific Sentiment Analysis of Economic Texts, LREC. (2014) 2152–2157.

[30] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, The Journal of Machine Learning Research. 2 (2002) 45–66.

[31] A. Trabelsi, O.R. Zaïane, A Joint Topic Viewpoint Model for Contention Analysis, in: 19th International Conference on Applications of Natural Language to Information Systems, Proceedings, Springer International Publishing, 2014: pp. 114–125.

[32] P.D. Turney, Thumbs up or thumbs down?, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02, Association for Computational Linguistics, Morristown, NJ, USA, 2001: p. 417.

[33] P.D. Turney, M.L. Littman, Measuring praise and criticism: Inference of semantic orientation from association, ACM Transactions on Information Systems. 21 (2003) 315–346.

[34] J. Wiebe, T. Wilson, C. Cardie, Annotating Expressions of Opinions and Emotions in Language, Language Resources and Evaluation. 39 (2005) 165–210.

[35] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05, Association for Computational Linguistics, Morristown, NJ, USA, 2005: pp. 347–

354.

[36] B. Yang, C. Cardie, Context-aware Learning for Sentence-level Sentiment Analysis with Posterior Regularization, (n.d.) 325–335.

[37] C. Zhai, A. Velivelli, B. Yu, A cross-collection mixture model for comparative text mining, in: Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '04, ACM Press, New York, New York, USA, 2004: p. 743.

Appendix A: SJCM User manual

The following user manual is based on that provided with Lin's JST [9], adapted for the modifications made.

A.1 Usage

The model has two modes of operation: estimation and inference. Estimation mode will iterate through the training data and produce an output of the estimated model parameters. Inference mode applies an already estimated model to a new set of testing data.

Estimation is performed with the command

```
sjcm -est -config YOUR-PATH/training.properties
```

where `training.properties` is key=value formatted file specifying the model settings. This will produce the following output files in the designated output folder:

`<iter>.others`	contains model parameter settings
`<iter>.pi`	contains the per-document topic distributions
`<iter>.phi`	contains the sentiment specific topic-word distributions
`<iter>.theta`	contains the per-document-per-topic sentiment proportions
`<iter>.tassign`	contains the sentiment label and topic assignments for words in training data
`<iter>.tassign`	contains the tassign data in JSON form, for input into the visualizer
`<iter>.evaluation`	contains an evaluation report of accuracy of actual v. predicted labels

These output files can then be used to perform inference on new data. Inference is performed with the command

```
sjcm -est -config YOUR-PATH/training.properties
```

Which will produce updated versions of the previous files, with the following names:

```
<modelName_iter>.newpi
<modelName_iter>.newphi
<modelName_iter>.newtheta
<modelName_iter>.newtassign
```

The input data follows the following format:

```
[number of docs]
[doc 1 ID] [sentiment label] [topic label] [train/test designation]
[doc 1 text]
⋮
[doc n ID] …
```

A.2 Properties file

In order to run the model with different parameters, the properties file must be provided with the relevant data. The options found in the properties file are listed in Table 13.

Table 13: SJCM properties file options

Property	Default	Purpose
nsentiLabs	3	Number of sentient labels
ntopics	18	Number of topic labels
niters	800	Number of times to iterate the estimator
savestep	200	Save an intermediate model after this many iterations
updateParaStep	50	Update the model parameters after this many interation
twords	10	Number of top words to output for each class pair
data_dir	[No default]	Directory that the input dataset is stored in
datasetFile	[No default]	Filename for the input dataset
result_dir	[No default]	Directory to output the model output to
sentiFile	[No default]	Sentiment lexicon for weak supervision
beta	0.01	Default β hyperparameter
reverseJST	1	No longer used; used during implementation to switch between JST and Reverse-JST models for debugging
model	final	Used in the inference stage, to determine the name of the model to base the inference on

Appendix B: Maintenance Manual

B.1 Requirements

The SJCM classifier is provided in the form of C++ source code, and so should run on any system that is capable of compiling and running such code. There are no dependencies on any external systems or modules.

Processing of large datasets may involve holding a not insignificant amount of data in memory, and therefore it is recommended that the SJCM code be run on systems with at least 4GB or RAM available.

B.2 Compiling

In order to compile the source, a makefile is provided, so one simply needs to run the `make` command from within the same folder as that file.

B.3 Installation

As the code is provided only in source code form, no installation is required. The program, once compiled, can be run from any folder.

Appendix C: Source code listing

C.1 iTunes crawler tool

Single file: `itunes-crawl.php`. This file is used to extract the review data from the relevant iTunes store page. A single output file will be generated depending on the task, of the form [appID]-reviews-[store collection].csv, for example with the default settings, this would be 1018109117-reviews-english.csv, where 1018109117 is the App Store ID of the OSX El Capitan update, and the code is set to get reviews from only English-speaking countries. Usage is via PHP command line:

```
php itunes-crawl.php
```

As this is a single-use script that is not intended to be used often, there are no parameters to run the program with, and changes must be made by editing the script directly. The variables that can be changed are, however, made clear at the top of the file.

C.2 Dataset

Copies of the initial dataset have been included in the source so that the annotation process method can easily be seen.

C.3 SJCM

A listing of the source code files for SJCM, and the purposes of each class, is provided in Table 14.

Table 14: SJCM source files and purposes of each

Source file	Purpose
cokus.h	Mersenne Twister generator implementation
constants.h	General implementation constants
dataset.cpp dataset.h	Class to contain the dataset and related functions, e.g. reading in data from file
document.h	Defines a Document class, which represents a single document within the dataset
inference.cpp inference.h	Contains functions for performing inference on a model that was estimated previously
map_type.h	Defines some map types, e.g. for relating sentiment and topic labels to their respective internal integer representations
math_func.cpp math_func.h	Provides some mathematical functions used in the model
model.cpp model.h	This class is where the model estimation is performed
polya_fit_simple.cpp polya_fit_simple.h	Not used by SJCM, leftover from JST implementation
sjcm.cpp	The main entry point to the application. Calls either the estimation or inference function, depending on command line switches
strtokenizer.cpp strtokenizer.h	String tokenizer class, used for parsing the input data in text form
utils.cpp utils.h	Class that provides various functions useful to the operation of the model, e.g. parsing the model properties file

C.4 Visualisation

The prototype visualisation is provided as a single file, `sjcmViz.pde`, which contains the Processing 3 code required to produce the visualisation. The code can be loaded into Processing 3 and run from there. A pre-compiled version for OSX has been included in the source which could be run directly. To operate correctly, this program expects the `final.json` file output by the SJCM model to be in the same working directory.