Sentiment Analysis of Data from Online Forums on the Newborn Genome Sequencing

By

Hamid Poursepanj

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For Master of Computer Science (M.C.S)

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

In this thesis, we classified user comments posted on online forums related to "Newborn Genome Sequencing" (NGS). User comments were annotated as *irrelevant, positive, negative,* or *mixed* by two annotators. The objective was to create a classification model that could predict the sentiment of each user comment with a high accuracy. To compare classifiers, a baseline classifier (Accuracy 52%) was created. We created a single classifier (called flat comment-level classifier with accuracy of 65.14%) to classify comments into *irrelevant, positive, negative,* or *mixed*. A more sophisticated classifier, named two-level comment classifier, consisting of two classifiers, was created (Accuracy 69.81%):

- The first classifier that classified each comment into *relevant* or *irrelevant* ones.
- The second classifier that classified each *relevant* comment (predicted by the first classifier) as *positive, negative,* or *mixed*.

18 extra features were generated to improve the accuracy of the flat classification compared to baseline classifier (from 52% to 65.14% for flat comment classification, and 69.48% to 69.81% for two-level comment classification). Attempts were made to enhance the result of the two-level comment classifier by using the discourse structure of each sentence in a comment. The accuracy achieved by this enhanced two-level classifier was 64.24%. Therefore, removing irrelevant EDUs did not improve the accuracy.

To achieve the above-mentioned enhancement, all comments were segmented into their consisting *elementary discourse units (EDUs)*. We removed *irrelevant* EDUs from the *relevant* comments before running the second classifier. Furthermore, we performed EDU-level classification by creating two classifiers:

- A flat classifier: classified all EDUs into irrelevant, positive, negative, or neutral
- A two-level EDU: classified EDUs, first, into *relevant* or *irrelevant* and then classified the *relevant* EDUs (predicted by the first classifier) into *positive, negative,* or *neutral* ones.

The accuracy achieved for the flat EDU-level classifier was 81.84%. However, due to the highly imbalanced nature of the EDU dataset, the F-measure for *positive, negative,* and *neutral* class was very low. Under-sampling was performed to improve the F-measure for *positive, negative,* and *neutral* class.

Another topic investigated was to know why forum users supported or rejected NGS. To extract the arguments, the comments were segmented into EDUs. Following segmenting, each EDU was annotated as *relevant* or *irrelevant* to NGS. Each *relevant* EDU was annotated as for or against NGS. Topic related EDUs were selected as well as two EDUs before and after the topic-related EDUs. Bigrams, trigrams, four-grams and five-grams were created from extracted EDUs. Five-grams were more meaningful for human annotators, and were therefore favoured and ranked based on frequency in the dataset. Following ranking of the five grams, the top five were selected as the possible arguments.

Acknowledgement

First and foremost I offer my sincere gratitude to my supervisors Dr. Diana Inkpen and Dr. Marina Sokolova for their knowledge and support. Thanks to their dedication and effort I was able to complete my Masters of Computer Science requirements.

I would like to thank Dr. Julian Little, Canada Research Chair in Human Genome Epidemiology, Faculty of Medicine at University of Ottawa, for suggesting the study of parents' opinion on newborn genome screening. I also would like to thank my supervisors at work that allowed me to work flexible hours while attending university.

In addition, I would like to delightedly thank Marcia Hooper for her support throughout my education and gratefully acknowledge the support of Amer Adas and Nasren Elsageyer for the data annotation task.

And finally, I would like to dedicate this thesis to my parents and thank them for their restless support in all phases of life.

Table of Content

List of Tables

List of Figures

List of Acronyms

ANNIC	Annotations in Context
BOW	Bag of Words
BOW-EF	Bag of Words with Extra Features
CF	Cystic Fibrosis
CH	Hypothyroidism
CRF	Conditional Random Field
DNA	Deoxyribonucleic Acid
EDU	Elementary Discourse Unit
FN	False Negative
FP	False Positive
GATE	General Architecture for Text Engineering
GI	General Inquirer
HTML	Hyper Text Markup Language
IDF	Inverse Document Frequency
JAPE Java	Annotation Patterns Engine
KNN	K-nearest Neighbours
LSA	Latent Semantic Analysis
LIWC	Linguistic Inquiry Word Count tool
MMDH	Maximal Marginal Decision Hyper-plane
MPQA	Multi-perspective Question Answering
N	Negative
NB	Naïve Bayes
NGS	Newborn Genome Screening
NLP	Natural language Processing
NSO	Newborn Screening Ontario
P	Positive
PAC	Probably approximately correct
PHI	Personal Health Information
PKU	Phenylketonuria
PMI	Point-wise Mutual Information
RST	Rhetorical Structure Theory
RST-DT	RST Discourse Treebank
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency
TN	True Negative
TP	True Positive
URL	Uniform Resource Locator
WEKA	Waikato Environment for Knowledge Analys

Chapter 1 Introduction

1.1 Introduction

Approximately 140,000 babies are born every year in Ontario. Some appear healthy, but have the potential to develop life-threatening diseases later in life. It has been shown that preventive medicine and early treatment have a great potential in fighting intellectual and physical diseases and sudden infant deaths. In Ontario, provincial program Newborn Screening Ontario (NSO) ensures that every newborn can be screened for at least 29 life-threatening diseases[1]. Some conditions commonly tested for include cystic fibrosis[2] (CF), the enzyme deficiency phenylketonuria[3] (PKU), and hypothyroidism[4] (CH), a thyroid hormone deficiency.

Newborn Genome Screening (NGS) is done through a blood test that is administered shortly after birth to test for diseases that are not apparent in the neonatal period. Guthrie test, or a heelprick, allows health care providers to collect sample blood from newborns and send it to NSO where it is tested for possible diseases.

The Government of Ontario mandates NSO to provide every Ontario newborn a high quality neonatal screening. NSO also provides analysis and follow-up education for parents to help prevent possible diseases to develop in newborns. However, according to statistics[5] provided by NSO, since 2008 there has been a 1.3% decline in the number of newborns tested in Ontario.

This thesis presents Text Data Mining study of online forums on which parents and other involved parties discuss merits of NGS (NGS forums). We empirically explore whether parents support newborn genome sequencing or they disagree with it. An Internet forum is an online discussion site where people participate in a discussion about a specific topic. During recent years, due to rapid development of the Internet and mobile devices, more people are referring to forums to find useful information about their topic of interest. Approximately 19%-28% of Internet users participate in online health discussions (Balicco and Paganelli, 2011). Therefore, health forums discussions can be a useful source of information for organizations specialized health care. By mining data collected from such forms health care organization can understand attitude and opinions of the general public and make educated decision about future health care policies.

In our study, we build a Sentiment Analysis system that automatically classifies sentiments in the posts collected from NGS forums. Furthermore, we develop and

[1] Retrieved 2015-01-02, from http://www.newbornscreening.on.ca/bins/content_page.asp?cid=7-21&lang=1
[2] Retrieved 2015-01-02, from http://www.newbornscreening.on.ca/data/1/rec_docs/669_MOH_Fact_Sheet_CF.pdf
[3] Retrieved 2015-01-02, from http://www.newbornscreening.on.ca/data/1/rec_docs/542_fs_pku.pdf
[4] Retrieved 2015-01-02, from http://www.newbornscreening.on.ca/data/1/rec_docs/529_fs_ch.pdf
[5] Retrieved 2015-01-02, from http://www.newbornscreening.on.ca/data/1/rec_docs/700_2012_NSO_Annual_Public_Report.pdf

implement an Information Extraction method to identify and extract the reasons that parents articulate to support or reject newborn genome screening.

1.2 Motivation

Sentiment analysis of online medical forums has been made necessary by the demands of policy-makers, the emergence of infoveillance and by the emergence of infodemiology (Eysenbach, 2009). Infodemiology and infoveillance are frameworks to study the health information on the Internet with the goal to improve public health (Eysenbach, 2009).

Our motivations for this thesis were as follows:

1. Develop and implement a system that analyzes health-related discussions, which may have a long-lasting impact on policy development.

2. Discover whether online medical forum users (mostly parents) supported "Newborn Genome Sequencing" or disagreed with it.

1.3 Study Goals

The following goals are identified for this thesis:

1. Identify online forums and blogs that are relevant to newborn genome sequencing (NGS) and build a corpus by extracting relevant user comments from the relevant forums and blogs.

2. Develop classifiers that accurately predict the sentiment of each comment from possible *positive*, *negative*, or *mixed* classes.

3. Develop a simple argumentation method that can identify and extract the reasons that parents (forum users) expressed to either support newborn genome sequencing (NGS) or disagree with it.

4. Analyze the obtained results from the above mentioned goals.

1.4 Hypothesis

The hypothesis for this thesis claims that:

1. Text Data Mining methods can be successfully applied in predicting the sentiment analysis of messages harvested from NGS forums. The prediction can be done with a better accuracy than a simple baseline (majority choice) despite a large volume of *irrelevant* text.

2. The reasons that parents supported or disagreed with newborn genome sequencing can be extracted by developing an argumentation method that extracts topic related Elementary Discourse Units (EDUs) from *relevant* comments.

1.5 Intended Contribution

This thesis intended to bring the following contributions:

1. We collected a dataset on "Newborn Genome Screening" (NGS) from wide range of medical forums and blogs. Furthermore, this dataset is available to research community for further research.

2. We developed a simple argumentation technique that extracted reasons expressed by forum users to support or reject NGS. Topic-related EDUs as well as two EDUs before and after each topic-related EDU were extracted. Five-grams were created and ranked based on their frequencies in the corpus. Finally top 5 five grams were chosen as the five most important reasons to accept or reject NGS.

3. We engineered 18 extra features to improve the accuracy of the comment classification, and 17 extra features to improve the accuracy of EDU-level classification.

4. We developed a two-level comment classification method. In the first level, comments were classified into *relevant* or *irrelevant* ones. In the second level, relevant comments were classified into *positive, negative*, or *mixed*. t-test with p-value equals to 0.05 showed that the two-level classification had the better accuracy than flat classification. In the flat classification, comments were classified into *irrelevant*, *positive, negative*, or *mixed*. Figure 1 shows the two-level comment classification method.

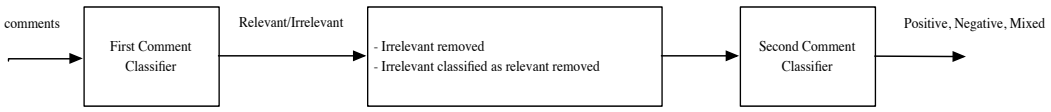

Figure 1 - Two-Level comment classification Method

5. We developed a two-level EDU classification method. In the first level, EDUs were classified into *relevant* or *irrelevant* ones. In the second level, relevant EDUs were classified into *positive, negative*, or *neutral*. . t-test with p-value equals to 0.05 showed that the two-level classification had the better accuracy than flat classification. In the flat classification, comments were classified into *irrelevant*, *positive, negative*, or *neutral*. Figure 2 shows the two-level EDU classification method.

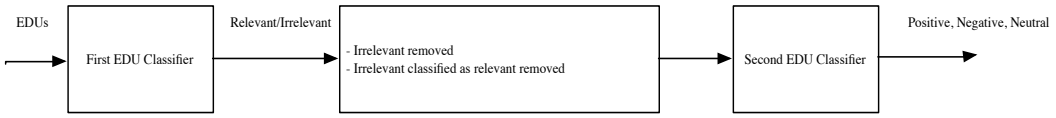

Figure 2 - Two-Level EDU Classification Method

6. Furthermore, we tried to improve the accuracy of the two-level comment classification by removing the irrelevant EDUs from the relevant comments in the second level. Figure 3 shows the enhanced two-level comment classification method.

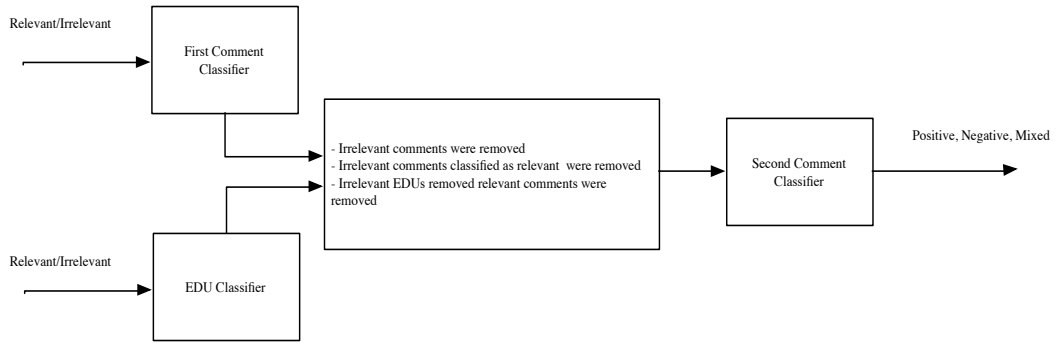

Figure 3 - Enhanced Two-Level Comment Classification Method

1.6 Outline

The thesis is organized as follows:

Chapter 2 covers related work; we reviews methods used in sentiment classification of online data, especially methods applied to data from Twitter and online forums.

Chapter 3 provides description of concepts important in our research such as applied Machine Learning classification algorithms, evaluation measures of classifiers, measures used to assess agreement between annotators; the chapter also discusses the notion of argumentation.

Chapter 4 introduces automated and manual methods applied in data collection, detailed information about the annotation process and definition of argumentation units, and finally, statistical analysis of the dataset.

Chapter 5 presents classification methods in details.

Chapter 6 explores the empirical results of the experiments.

Chapter 7 presents the conclusions drawn from the work performed for this thesis. In addition, future work section of chapter 7 discusses further methods than can be performed to improve the result of comment classification.

Chapter 2 Related Work

2.1 Sentiment Analysis

Sentiment analysis is the combination of information retrieval[6] and computational linguistics[7], which studies sentiments found in a set documents (e.g., *positive* and *negative*). There are potential applications for sentiment classification such as automatically identifying synonyms and antonyms (Hatzivassiloglou & McKeown, 1997), filtering newsgroup messages for verification by human moderators (Spertus, 1997), and sentiment classification of movie reviews (Turney & Littman, 2003).

In one of the early efforts to determine the polarity of adjectives, Hatzivassiloglou & McKeown (1997) used a four-step algorithm to classify conjoined adjectives into positive and negative. The approach relied on analysis of the correlation between conjunctions and semantic orientations (i.e., *same* or *different*) of conjoined adjectives. At first stage, all conjunctions of adjectives were extracted. At the second stage, a log-linear regression model determined if each two conjoined adjective is of the same or different orientation (accuracy of 82.44%). At the third stage, a graph was constructed from adjectives extracted in the previous stages with adjectives as nodes and orientations (i.e., *same* or *different*) as edges. Adjectives were divided into two groups by applying a clustering algorithm to the above-mentioned graph. Adjectives inside a group were of the same semantic orientation whereas ones between two groups were of different semantic orientation. Finally, at stage four, the average frequency of words in each group was calculated and the group with higher average frequency was expected to contain *positive* adjectives. Adjectives related in form (e.g., *adequate-inadequate*) always had different semantic orientations; therefore, a morphological analyzer implemented to match adjectives in this manner. The classification algorithm proposed by Hatzivassiloglou & McKeown (1997) was limited to conjoined adjectives and requires labeled data.

In contrast to Hatzivassiloglou & McKeown (1997), Turney & Littman, (2003) proposed a method for computing the polarity of any word. Turney & Littman showed that the polarity (*i.e., positive or negative*) of a word could be calculated from the strength of its association with seven positive words (*good, nice, excellent, positive, fortunate, correct, and superior)*, minus the strength of its association with seven negative words (*bad, nasty, poor, negative, unfortunate, wrong, and inferior*). The fourteen words were chosen based on intuition. Turney& Littman, (2003) used point-wise mutual information (PMI) and latent semantic analysis (LSA) to calculate the strength of the positive and the negative association of a word. The algorithms were evaluated using 3,596 words (1,614 positive and 1,982 negative) from the General Inquirer Lexicon (Stone et al., 1966). The PMI-based algorithm for detection of positive and negative words achieved the accuracy of

[6] Information retrieval is the Activity of retrieving domain related information from sources of information such as The Internet.

[7] Computational linguistic is mainly statistical modeling of natural language from a computational perspective.

80%, 77.47%, and 62.32% with a one-hundred-billion-word, a two-billion-word, and a ten-million-word corpus respectively. The LSA algorithm achieved the accuracy of 65.24% with a ten-million-word corpus. Turney & Littman (2003) reported that PMI required a large corpus for accurate classification. However, the easy implementation of the PMI algorithm was an advantage. To extrapolate from words to document, the sentiment of a document was the sum of the sentiment scores of its words. Contrary to Hatzivassiloglou & McKeown (1997), Turney& Littman's (2003) approach was not limited to adjectives and did not require labeled data.

Beineke, Hastie & Vaithyanathan (2004) suggested using the Naïve Bayes (NB)[8] algorithm for binary sentiment classification while each document was represented as bag of words. Beineke, Hastie & Vaithyanathan (2004) achieved accuracy of 63%. Support vector machine (SVM)[9] algorithm is also very popular in sentiment classification due to its ability to handle a high number of features (Panget al., 2002).

The majority of efforts for sentiment classification are focused on using the bag-of-words model and classifiers such as SVM, NB (Panget al., 2002) or Conditional Random Fields (CRF) (Yang et al. (2007).

In sentiment classification, along with a classifier of choice, feature engineering becomes very important. A common practice is to use available semantic/sentiment lexicons to generate extra features such as the number of domain specific words, the number of subjective versus objective words, and the number of positive versus negative words. The practice helps to improve the classification accuracy (Wilson et al., 2005; Pak et al., 2010; Kouloumpis et al., 2011; Bobicev et al., 2012; Mohammad et al., 2013).

As an early effort to generate a domain specific (semantic) lexicon, Riloff & Jones (1999) presented a two-level bootstrapping algorithm to generate reliable domain-specific semantic lexicon. At the first level, a mutual bootstrapping technique is used to extract the best extraction patterns from an initial semantic lexicon (a set of domain related seed words). Then, more related terms are identified by these learned patterns and added into the initial semantic lexicon. At the second level, a meta-bootstrapping technique was used to keep only the best lexicon entries. The process was restarted with the enhanced semantic lexicon. Riloff & Jones (1999) used the accuracy rate as an evaluation measure for the two-level bootstrapping algorithm. The accuracy rate is defined as the number of true semantic category members over the total number of extracted words. The accuracy rate achieved by the two-level bootstrapping algorithm for generating semantic lexicon for terrorism weapon category was 51% in 50 iterations whereas the previous work done to generate the same semantic lexicon for the same category achieved an accuracy of 17% (Riloff &Shepherd, 1997).

An unsupervised bootstrapping algorithm was used by Riloff et al. (2003) to create a learning system that can automatically annotate sentences to subjective and objective

[8] Explained in chapter 3

[9] Explained in chapter 3

ones automatically. First, a high precision subjectivity classifier was developed capable of labeling sentences into subjective and objective. The high precision classifier achieved accuracy of 91.5% for subjective sentences and accuracy of 82.6% for objective ones. Once sentences were labelled (*subjective* or *objective*), an extraction pattern learner (precision between 71% and 85%) used the sentences and learned subjective patterns that were statistically correlated with subjective expressions. Finally, a pattern-based classifier labelled un-annotated sentences and fed the pattern-based classifier with more labeled sentences and this process was repeated. The learned extraction patterns were added to the high precision objectivity classifier to improve their accuracy. Riloff et al. (2003) reported in bootstrapping process, adding the learned extraction patterns learned in the first iteration improved the performance of the high-precision subjective classifier in the second iteration by increasing the recall number by 2%-4%.

Riloff et al. (2003) built the lexicon of subjective items by collecting subjectivity clues (mostly unigrams and some n-grams) from manually developed resources. They argued that using extraction patterns to identify subjective expressions is more useful than n-grams since the meaning of an expression is different from the meaning of individual words and an expression is not a fixed sequence of words that can be captured by n-grams.

2.2 Sentiment Classification of Social Network Data

With the emergence of online social media such as Twitter, Facebook, various blogs, and forums, data collected from the social media became a rich source for opinion mining and sentiment analysis (Yang et al., 2007; Pak & Paroubek, 2010; Osman et al., 2010; Kouloumpis et al., 2011; Bobicev et al., 2012; Mohammad, Kiritchenko, & Zhu, 2013). Since this study focused on sentiment analysis of blogs and forums, first the sentiment classification of microblogs' (i.e., Twitter) data will be discussed. Furthermore, we present the specific challenges of online forum data.

In supervised sentiment analysis, the first step is to manually annotate training data. Wiebe et al. (2005) suggested that in order to produce a reliable annotated training data to more than one annotator should be trained with an annotation scheme. An annotation scheme is a guideline to systematically distinguish between subjective and objective sentences. Subjective sentences carry sentiment, whereas objective sentences are factual and do not convey opinions. In contrast, Strapparava & Mihalcea (2008) suggested that to avoid biasing annotators towards text categorization approaches, it is necessary not to provide any training for annotators; therefore, annotators should be free to use any resources they want to label data based on their own perception of any sentiment.

In order to measure the reliability of the annotation scheme, the kappa statistics was used (Wiebe et al., 2005; Osman et. al., 2010). The authors measured the degree of agreement between raters using kappa statistics. A kappa score is the indicator of agreement between assessors and ranges from 0 (agreement by chance) to 1 (total agreement).

Cohen's kappa was used as a measure of agreement between two annotators and Fleiss's kappa is used to determine the measure of agreement between multiple annotators. Osman et al. (2010) and Wiebe et al. (2005) recommended that removing the most dissenting annotators from the pool of annotators would result in a higher level of consistency in the assessments, and a higher level of consistency in the training data; therefore, better polarity detection system with higher accuracy.

Support Vector Machine (SVM), Conditional Random Fields (CRF), and Naïve Bayes (NB) learning algorithms were often used for sentiment classification of Twitter data (Pak et al., 2010; Bobicev et al., 2012; Mohammad et al., 2013). Sentiment Analysis of informal text collected from the Twitter account of popular newspapers such as "New York Times" and "Washington Post" was done by Park et al., (2010). Each document was a Twitter message and each twitter message had up to 140 characters. The author experimented with SVM, CRF, and NB in classifying of documents into positive, negative, and neutral class labels. The authors used unigram, bigram, and trigram as main features plus extra features such as emoticons, and part-of-speech distribution information. The best accuracy resulted from NB using bigrams plus the extra features with feature selection performed on all features (63%). The authors reported that attaching negation words when constructing bigram and trigram improved classification accuracy by almost 1% (from 63% to 64%).

Furthermore, Mohammad et al. (2013) performed sentiment classification at term and message (standard Twitter message including 140 characters) level for a corpus of Twitter data using SVM. They used a comprehensive set of features namely, N-grams, character n-grams, the number of words with all letters capitalized, part-of-speech distribution information, various lexicons, punctuation, emoticons, elongated words, and negations. This set of features helped Mohammad et al. (2013) to achieve the F-score of 0.69 in message-level sentiment classification compared to the F-score of 0.292 reported for the message-level baseline classifier using SVM. The message-level baseline classifier is a one that classifies all messages to the majority class. Furthermore, Mohammad et al. (2013) achieved the F-score of 0.89 in term-level sentiment classification using SVM classifier and all sets of above-mentioned features, whereas, they reported the F-score of 0.38 for the term-level baseline classifier.

Feature selection and feature engineering proved to improve the accuracy of SVM, NB, and CRF classifiers for Twitter data (Pak and Paroubek 2010; Kouloumpis, Wilson, & Moore, 2011; Bobicev et al., 2012). Pak and Paroubek (2010) checked whether removing common features that do not strongly indicate any sentiment would affect the classification accuracy. To identify common bi-grams, entropy of probability distribution of bi-grams in different datasets (different sentiment) was calculated. According the authors "The high value of the entropy indicates that a distribution of a bigram in different sentiment datasets is close to uniform. Therefore, such a bigram does not contribute much in the classification." Bobicev et al. (2012) showed that performing feature selection on bag-of-words will improve the performance of sentiment classification tweets that disclose information about person's health. The best reported accuracy for binary classification using bag-of-words with frequency greater than 2 was

75.5%. The authors also performed classification on a set of features that were highly correlated to class labels and achieved accuracy of 82.4%. It can be seen that feature construction based on the class label improved the accuracy of classification.

In some cases, part-of-speech features can improve the sentiment classification accuracy by a small margin (almost 1%) (Mohammad, Kiritchenko, & Zhu, 2013). However, some researchers prefer not to use part-of-speech features due to the minor contribution to the classification accuracy (Kouloumpis, Wilson, & Moore, 2011). Kouloumpis et al., (2011) reported part of speech features might not be useful in sentiment analysis of micro-blogging domain. The authors trained an AdaBoost classifier and achieved F-measure of 0.65 using N-grams, lexicon features (i.e., *positive, negative,* and *neutral*) features. In addition they achieved the same F-measure was achieved using N-grams and POS features. However, the authors, reported F-measure of 0.68 using N-grams, lexicon and twit specific features. Kouloumpis et al. decided to drop POS features and keep N-grams, lexicon and Twitter specific features.

Adding domain specific features such as the number of emoticons can be useful in sentiment analysis of Twitter data. Kouloumpis et al., (2011) performed classification experiments to evaluate the effectiveness of various features for sentiment classification. N-grams, lexicon features (i.e., *positive, negative,* and *neutral*), part-of-speech features, and micro-blogging features were used in this experiment. An AdaBoost classifier was trained on four sets of features: n-gram and lexicon features, n-gram and part-of-speech features, n-gram and lexicon features and micro-blogging features, and finally all features. The best performance on the evaluation data came from n-gram, lexicon features and micro-blogging features (F-measure of 0.68) indicating the part-of-speech features give a drop in performance (F-measure drop from 0.68 to 0.63), while including emoticons improved the classification accuracy: once the micro-blogging features was included, the improvement dropped.

Few papers focused on the personal health information (PHI) aspect of sentiment analysis of twitter data (Bobicev et al., 2012; Chew & Eysabach, 2010; Salathe & Khandelwal, 2011).

The presence of terms discussing personal health (e.g., headache, diabetic, flu, injury), and personal references such as *I, he, and she* in a tweet usually emphasize that the tweet message refers to personal health (Bobicev et al., 2012). The authors performed binary (i.e., *positive* or *negative*) and three-class (i.e., *positive, negative,* or *neutral*) classification by training SVM, KNN, and NB classifiers on bag-of-words with word frequency of greater than two (i.e., *BOW2*), bag-of-words with word frequency of greater than 5 (i.e. *BOW5*). The authors reported accuracies of 80.5%, 80%, and 82.4% for binary classification using SVM, KNN, and NB classifiers and consistSub set of features. Alternatively, the reported three-class classification accuracies using consistSub set of features and SVM, KNN, and NB classifiers were 75.3%, 72.8%, and 71.7% respectively. NB achieved the best binary classification performance whereas SVM achieved the best three-class classification performance.

Capturing the personal health aspect of sentiment analysis through feature engineering was experimented by Biyani et al. (2014). They analyzed interactions between members of Cancer Survivor Network (CSN) forum to identify emotional support as well as informational support members provide to each other. The authors built a Naive Bayes classifier to classify comments into emotional-support or informational-support classes. Emotional supports appear in messages that have one or more of the following supports: caring/concern, understanding, empathy, sympathy, encouragement, affirmation and validation. Informational supports are defined as providing advice, knowledge, and referral. The authors used 5 lexicon-based features namely, the number of weak subjective words, the number of strong subjective words, cancer drugs, the effects of cancer medication, and cancer procedures. In addition, the authors used linguistic features. By analyzing the dataset, Biyani et al. (2014) figured out that users use certain word patterns to express emotional and informational supports. The authors extracted these patterns and build 4 features that represented the presence and absence of these patterns in a comment including, the presence of emotional pattern/s the absence of emotional pattern/s, the presence of informational pattern/s, and the absence of informational pattern/s.

In contrast to the Twitter text, comments collected from online forums consist of more sentences since the number of characters usually does not limit user comments; therefore, forum users can argue about the discussion topic and other users' point of views. This will lead to a relatively high number of sentences (*relevant* and *irrelevant* ones) and a more complex argumentation structure in a comment.

One way to handle the above-mentioned difference in online forum data is to remove objective sentences from a comment to reduce feature space dimension (Pang & Lee, 2004). Subjectivity detection could help compressing comments into much shorter extracts with polarity information at a comparable level to the whole comment. Building an undirected weighted graph from sentences in a document, Pang & Lee (2004) applied a maximum flow minimum cut algorithm to classify sentences into subjective and objective ones. The authors showed that using subjectivity extract improved the accuracy of SVM from 82% to 86.4%.

Another way is to develop a rule-based classifier that can capture the sentiment of a comment based on the sentiment of the sentences composition (Yang et al., 2007). They compared sentence level classification with comment (post) level classification of blog post data. The average length of data collected was 272 characters for posts containing emotions. The average length was 465 characters for posts without emoticons. The authors generated an emotion lexicon to choose a set of emotion words from the whole corpus (consisting of 17887 posts) as features for classification. For binary classification (i.e., *positive* and *negative*) at the sentence level, the authors reported 74.02% and 79.8% as classification accuracy for SVM and CRF respectively. However, at the comment level three criteria were used to classify a document into *positive* or *negative* classes and took the document emotion as the same emotion that appears in:

1. The most classified sentences.
2. The longest series of sentences that have the identical emotions.
3. The last classified sentence.

In those three cases, the highest accuracy was achieved under the third rule (59.23%). Therefore, Yang et al. (2007) showed in the context of blog post reviews the emotion of last sentence of a post is usually representative of the emotion of the whole post.

2.3 Lexical resources

A common approach for sentiment analysis is to use a lexicon of words tagged with prior polarity. Two types of lexicons are used for sentiment analysis namely sentiment lexicons and emotion lexicons. Sentiment lexica mostly cover binary labels (*positive* and *negative*) whereas emotion lexicons cover more labels (e.g., *joy, sadness, anger, fear, enthusiasm, surprise*).

2.3.1 Sentiment lexicons

In this section, we review the General Inquirer, the Subjectivity Lexicon, and SentiWordNet, which are examples of the major available sentiment lexicons.

The General Inquirer (Stone et al. 1966) is a mapping tool capable of mapping each document with counts on dictionary-supplied categories. The USA National Science Foundation and Research Grant Councils of Great Britain and Australia have supported the development of the General Inquirer (GI). GI has 182 categories including positive and negative. 4000 words are covered in binary classification of positive and negative. Wilson et al. (2005) extracted the positive and negative words from GI to build a custom lexicon for sentiment analysis. The lexicon covers binary classification (positive/negative) of 6000 words. Esuli et al. (2006) reported using GI as one of the available resources to create a publicly available lexicon for opinion mining (SentiWordNet).

A subjectivity lexicon was created through a two-step process for automatically distinguishing the contextual polarity (Wilson et al, 2005). At the first stage each phrase was classified as neutral or polar. At the second stage, all polar phrases were classified as positive, negative, both, or neutral. Both classifiers used the bag-of-words model and a boosting machine-learning algorithm for classification.

Wilson et al. (2005) used Multi-perspective Question Answering (MPQA) as training corpus. First, the authors started with the list of subjectivity clues from (Riloff & Wiebe, 2003) and expanded the subjectivity clues list using a dictionary and thesaurus. Words were added from General Inquirer Lexicon (Stone et al., 1966). A manual annotation schema was developed to tag the clues in the lexicon with prior polarities (positive,

negative, mixed, or neutral). The authors constructed 28 extra features. 10 features were constructed by using prior polarity from Subjectivity Lexicon. The best reported accuracy of 75.9% for the first classifier with 28 extra features (2.3% higher than the baseline classifier). The second classifier with 10 extra features achieved the accuracy of 65.7%, which performed 4.3% better than the baseline classifier. As observed from the results, the combination helped to improve accuracy over the baseline.

SentiWordNet (Esuli et al., 2006) is the first polarity/subjectivity lexicon with distinction between different senses of a word. SentiWordNet is built upon WordNet (Fellbaum, 2005), a dictionary of English words grouped to sets of synonyms (synsets), which are linked by semantic and lexical relations. Words in a synset are grouped together based on meanings and are interchangeable in many contexts. Two of the most common relationships between synsets are ISA (hyperonymy and hyponym) and part-whole (Meronymy). Due to the structure, WordNet is a useful tool for computational linguistic and natural language processing.

Esuli et al. (2006) associated three numerical scores namely Obj(s), Pos(s), and Neg(s) to each WordNet synsets. Each of the three scores range from 0.0 to 1.0 and their sum is 1.0 for each synset. Furthermore, each score denotes the confidence in the correctness of labeling. Esuli et al. (2006) developed a semi-supervised learning process consisting of a committee of seven ternary classifiers including k-nearest neighbour (KNN), Naïve Bayes (NB), and Support Vector Machine (SVM) algorithms. Each classifier was trained on a different subset of the original training data. Then, the portion of the classifiers that have assigned the positive, negative, and objective label to a word sense determined the normalized corresponding scores of Pos(s), Neg(s), and Obj(s) for that word sense. For instance, the synset *estimable* that corresponds to the sense of adjective *estimable* has Obj score of 1.0 and Neg and Pos score of 0.0. This means that all the seven classifiers agreed on the sense of *estimable* to be an objective term. Each entry takes the form of lemma#pos#score. The reliability of SentiWordNet is not clear since evaluation of it requires a full manual annotation of WordNet synsets according to positive, negative, and objective labels, which does not exist. Therefore, lower precision is compensated by high coverage. SentiWordNet is used in various lexicon-based methods in sentiment analysis (Taboada et al., 2011; Thelwall et al., 2012)

2.3.2 Emotion lexicons

WordNetAffect, Depechemode, and the NRC emotion lexicon are the three examples of the major emotion lexicons.

WordNet-Affect (Strapparava and Valitutti, 2004) is a subset of WordNet in which each synset is annotated with an affective domain label, namely: *anger, fear, joy, sadness, surprise, and disgust*. Affective labels are categorized into affective categories namely emotion, cognitive state, trait, behavior, attitude, and feeling. For instance, *skepticism* is an attitude, *pleasure* is a feeling, *fear* is an emotion and *cry* is a behavior. According to Strapparava et al. (2004), the WordNet-Affect lexicon was developed in two steps. First,

the affect lexical database containing 1903 terms related to emotional states was realized. Then, a frame (a set of attributes) including POSR and ORTONY attributes assigned to each term. POSR related terms having different part-of-speech but pointing to the same category. ORTONY was used to indicate the affective category of each term in Affect database. Then, the ORTONY attribute was projected from the affect database onto the corresponding senses of WordNet-Affect. In the second step, WordNet lexical and semantic relations were exploited to extend the core of WordNet-Affect. WordNet-Affect has 2874 synsets and 4787 words.

Depechemode (Staiano et al., 2014) is an emotion lexicon containing 37,000 entries. Each row-wise entry takes the form of lemma#pos followed by seven scores for *afraid, amused, angry, annoyed, don't-care, happy, inspired, and sad* emotion states respectively. In order to develop Depechemode, Staiano et al. (2014) took advantage of massive crowd-sourced affective annotations associated with news articles, obtained by crawling the rappler.com social news network. Depechemode was generated in two steps. First, a documents-by-emotion (M_{DE}) matrix providing the voting percentage for each document in the eight affective dimensions available in Rappler. In the second step, first, an entry in the form of lemma#pos is created for each term (word) in a document. Then all entries are filtered to keep the ones that are also available in WordNetAffect, A term-by-document (M_{WD}) matrix was created using word frequencies, normalized word frequencies, and tf-idf. Finally, document-by-emotion matrix was multiplied by term-by-document matrix to result in term-by-emotion matrix ($M_{DE} \times M_{WD} = M_{WE}$). The resulting term-by-emotion matrix is normalized column-wise and scaled row-wise to sum up to one.

The NRC emotion lexicon is a list of words and their associations with eight emotions (*anger, fear, anticipation, trust, surprise, sadness, joy,* and *disgust*) and two sentiments (*negative* and *positive*). The annotations were completed manually through Amazon's Mechanical Turk (5 annotators) (Mohammad & Turney, 2013). Eight annotations were chosen from Plutchik's (1984) eight emotions because these emotions are more fundamental than other categorizations. The calculated Fleiss's kappa for all 8-emotion categories shows fair agreement amongst annotators (0.14-0.39)

2.4 Argumentation and Sentiment Analysis

Most of the works in sentiment analysis of micro-blogs such as Twitter try to determine the structure of the argumentation and how discourse relations such as contrast and generalization can affect the local sentiment flow in a comment. Argumentation and discourse relations are important contributors for sentiment analysis. Argumentation is composed of a set of arguments used to attack or support a hypothesis. Each argument can be a set of statements supported with some evidence. There are two types of argumentation styles namely, monologic and dialogic argumentation. In monologic argumentation, a person starts a debate by posting a topic. Debate participants can then post comments for or against the topic; therefore, there is no exchange of comments

between participants. In contrast, in dialogic argumentation, participants attack or support other participants' point of views. Most of the online forums have dialogic argumentation style, since participants usually comment on each other comments in addition to the topic of conversation.

The dialogic nature of forum data and the fact that the number of characters in forum comments is not restricted as they are for tweets, pose challenges to sentiment classification of comments. An author can express a positive opinion of the discussion topic but negative opinion of another author's comment. Furthermore, an author can express different opinions about topics that may not be related to discussion topic. These challenges can be addressed though sentence-based and discourse-based valence shifters (Polanyi & Zaenen, 2006).

Discourse is a component of language comprehension that consists of connected phrases and clauses with meaningful relations (Wolf et al., 2004). A relation reflects the quality of interaction between discourse segments. Discourse segments (or units) are non-overlapping spans of text. The relation between discourse segments, usually introduced by a discourse marker, can take various forms such as *elaboration, generalization, contrast, condition*, etc. Discourse relations that are important for sentiment analysis are *condition, contrast, violated-expectation, similarity, cause-effect, temporal-sequence, attribution, elaboration,* and *generalization* (Polyani et al., 2004; Mukherjee et al., 2012).

Some of the works in sentiment analysis using the bag-of-words model (Hatzivassiloglou et al., 1997; Turney et al., 2003; Beineke et al., 2004; Wiebe et al., 2005) ignore discourse particles such as *although, but* etc. as stop-words during feature vector creation. Indicators of discourse relations such as connectives and conditionals can be used to incorporate discourse information into the bag-of-words model to improve sentiment classification accuracy. Improvement can be achieved either by adding extra features which capture the contextual structure of the text (Wachsmuth et al., 2014) or by adjusting the weight of each individual feature in feature vector based on the discourse structure of the text (Mukherjee et al., 2012).

A weight-adjusting algorithm was used to capture the contextual structure of text through discourse relation analysis (Mukherjee et al., 2012). Discourse relations such as *elaboration, generalization, attribution, temporal sequence, condition, similarity, cause-effect, contrast,* and *violated expectation* were captured. The following discourse relation operators were constructed:

- Conj_Fol: *but, however, nevertheless, otherwise, yet, still, nonetheless*
- Conj_Prev: *till, until, despite, in spite, though, although*
- Conj_Infer: *therefore, furthermore, consequently, thus, as a result, subsequently, eventually, hence*
- Conditionals: *If*
- Neg: *not, neither, never, no, nor.*

The weight of each term in feature vector was adjusted as follows:

- The weight of terms coming before Conj_Prev is incremented by one.
- The weight of terms coming after Conj_Fol or Conj_Infer is incremented by one.

The sentiment of all words in the negation window (a window that come after Neg operator) of size 5 was reversed till the window size was exceeded or either Conj_Fol or Conj_Prev was encountered. An extra feature that captured the number of conditional discourse relations in a tweet was also added for Conditionals operator.

Mukherjee et al. (2012) performed binary (i.e., *positive* and *negative*) and three-class (i.e., *positive, negative,* and *neutral*) sentiment classification using SVM and rule-based classifiers on a dataset consisting of 8507 manually annotated tweets, as well as 15204 automatically annotated tweets. There was 2% improvement (against the baseline) in accuracy if discourse relations are taken into account for classification.

In contrast to Mukherjee et al. (2012), Wachsmuth et al. (2014) relied heavily on feature engineering rather than weight adjustment. The authors hypothesized that the sentiment of a text is often decided by the structure of its argumentation rather than the content. To prove the hypothesis, features were constructed which captured distributional argumentation aspects and features that modeled the structure of argumentation. As for the features related to distribution of argumentation aspects, three feature types were constructed:

- Local sentiment
    - To capture the frequency of local sentiment (i.e., *positive, negative,* and *neutral*)
    - To denote the location of local sentiment is specific positions such as the last phrase.

- Discourse relation
    - To capture the frequency of certain discourse relation such as contrast and elaboration
    - To model the connection between discourse relation and local sentiment by constructing features through a combination of the sentiment of elementary discourse unit that come right before and after a certain discourse marker and the discourse marker itself (e.g., *contrast(pos, neg)* or *contrast(neg, pos)*).

- Domain concept
    - To capture the frequency of common domain concepts as well as how often each concept occurs with each type of local sentiment.

To capture the impact of argumentation structure, Wachsmuth et al. (2014) first constructed a sentiment flow pattern for each post in training data, then, applied clustering to partition all patterns. Each cluster centroid became a sentiment pattern and

one numerical feature is added for it. The value of each feature was calculated as the similarity of local sentiment pattern to the centroid pattern related to this feature. Wachsmuth et al. (2014) reported an accuracy of 68.2% for the hotel review domain and accuracy of 72% for the movie review domain using an SVM classifier.

2.5 Argumentation Extraction

Anand et al. (2011) experimented classification data from online debate site, convienceme.com. The authors performed binary (i.e., *rebuttal* and *non-rebuttal*) classification on a dataset consisting of 4873 posts from 1113 two-sided debates. The data was collected from convienceme.com website. Debate-side classification was used over a range of 14 topics (ranging from *playful* to *ideological*) by using Naïve Bayes (NB) and rule-based (JRip) classifiers. For binary classification, however, a rule-base (JRip) classifier was used. For both tasks, Anand et al. (2011) constructed features including

- Unigrams, bigrams, trigrams, repeated punctuations, the number of words in a sentence, the number of positive and negative words.
    - Linguistic Inquiry Word Count tool (LIWC) (Pennebaker et al., 2001) was used to construct features such as the number of words in a sentence, the number of positive and negative emotion words.

- Part-of-speech dependencies
    - For a given sentence the part-of-speech dependencies are in the form of (rel_i, w_j, w_k), where rel_i is the dependency between w_j and w_k; w_j is the head of dependency relation.

- Opinion dependencies
    - Opinion dependencies are a subset of part-of-speech dependencies that contains opinion words. For a list of opinion words, arguing lexicons a sentiment lexicon (Somasundaran et al., 2007).

- Context features
    - Theses features represent the parent posts. For the above-mentioned features, the authors constructed two versions, namely: parent and post.

The authors achieved an accuracy of 63% for JRip classifier for binary classification (i.e., *rebuttal* and *non-rebuttal*). For debate-side classification (14 classes in total), Anand et al. (2011) achieved accuracies ranging from 50% to 67.95% for 14 different topics using Naïve Bayes. Amongst all the extra features, part-of-speech dependencies and opinion dependencies had almost no impact on the mean accuracy over all topics. Context features had a low impact (an overall improvement of 1.62% in mean accuracy). LIWC features were the only set of features with the potential to systematically show improvement on mean accuracy over the various ranges of topics. Individually adding

extra features for many topics can beat the unigram baseline, the baseline could not be beaten over all topics.

In an effort to capture the relationship between dialog structure information and opinion expression (arguments and sentiments), Somasundaran et al. (2007) analyzed the relationship in the AMI corpus (Carletta et al., 2005) and hypothesized that lexical and discourse information are needed to detect opinion in multi-party conversations. In order to incorporate discourse flows and participant interactions in improving sentiment and argument classification, the authors developed extra features as follows:

- The authors took advantage of already existing annotations in the AMI dataset which included 15 dialog act categories such as *Backchannel, Stall, Fragment, Inform, Elicit-Inform, Suggest, Offer, Elicit-Offer-Or-Suggestion, Assess, Elicit-Assessment, Comment-About-Understanding, Elicit-Comment-Understanding, Be-Positive, Be- Negative, Other*. Two Dialog Acts (DAs) can be linked via an Adjacency Pair (AP) relation. One of the DAs is the source and the other is the target in the AP. There are 5 AP types, namely: *Support/Positive Assessment, Objection/Negative Assessment, Uncertain, Partial agreement/support, Elaboration*. The authors built 99 extra features based on DA-AP-DA combinations (900 possible combinations) that appeared in the AMI corpus.

- The authors built an arguing lexicon by first, extracting all instances of the Dialog Acts, and extracting frequent n-grams ($1 \leq n \leq 4$) and ranking them based on tf-idf metric. The procedure resulted in 226 instances, sorted into 18 categorize such as necessity, conditional, emphasis, generalization, contrast, causation, etc. Extra features such as *positive Arguing("emphasis"), Arguing("necessity")* was developed.

- The authors also developed 6 extra features, namely: GIPositive, GINegative, StrongSubjective, weakSubjective, Intensifier, and valenceShifter. The first two features were extracted from General Inquire (Stone et al., 1966) and the rest were extracted from Subjectivity Clue List (Wilson et al., 2005).

Using all the above-mentioned extra features, Somasundaran et al. (2007) proved their hypothesis by building a binary sentence-level arguing classifier with the accuracy of 90.3%. They also build a binary sentence-level sentiment classifier with the accuracy of 82.17%. The sentiment classifier achieved an 8% improvement over the baseline majority classifier.

Discourse relations that indicate a conclusion, a premise, or a contrast can be analyzed to extract information that enables the reconstruction of arguments in the text. Wyner et al. (2012) developed a tool to find specific patterns of terminology in the text that can be used to instantiate consumer argumentation schemes both for and against purchase of the Canon PowerShot SX220 HS Digital Camera. They identified 5 tiers of analysis.

Each user justified whether to buy a camera and the justification is expressed in terms of these five tiers:

1. Consumer argumentation scheme
2. Discourse indicators
    - An overt linguistic expression of discourse relations between sub-sentence phrases.
        - Premise: *after, at, because*
        - Conclusion: *therefore, in conclusion, consequently*
        - Contrast: *but, except, never*
3. Sentiment terminology
    - A list of negative and positive words.
4. User models
    - Properties of users such as
        - Quality: *expectations: color quality*
        - Constraints: *cost, size, portability*
        - Context of use: *indoor, sport, travel*
        - Parameters: *age, gender, education*
5. Camera domain: terminology from camera domain[10]:
    - Properties with binary values such as *has a flash*
    - Properties with ranges such as *the number of megapixels, scope of zoom,* and *lens size*

To recognize the textual elements of each tier in the text, the authors used GATE tool (Cunningham et al., 2002). GATE is a framework written in Java used by linguistics and text engineers to combine a wide variety of natural language processing tools. GATE is explained in a bottom-up approach as follows

1. A gazetteer (A list of words that are associated with a central concept) associates textual passages in the corpus that match terms on the lists with an annotation.

2. JAPE rules used the annotation associated by gazetteers, and created annotations that were easily searchable by ANNIC

3. Carefully designed queries used to query (using ANNIC) the annotated database for extract text passages that identified justification against and for buying Canon PowerShot SX220 HS Digital Camera

Wyner et al. (2012) extracted arguments for and against buying Canon PowerShot SX220 HS Digital Camera by using A domain related list of terms related to users and camera, a list of sentiment terminology, and discourse markers.

[10] http://www.co- ode.org/ontologies/photography/

Using associations that are indicative of opinion stances is a novel way to extract arguments from text. Somasundaran et al. (2009) presented an unsupervised opinion analysis method for double-sided debate classification, i.e., recognizing which stance a person takes in an online debate. The authors enumerated the challenges that came with online debate data and developed an approach to tackle them. The challenges were

1. Debate participants switched back and forth between their opinions about sides (two sides).

2. Conflicting sides were discussed within a post.

3. Among different aspects of topics, some were differentiating and the rest were just personal preferences.

4. Although some participants acknowledged (referred) the opinion of other ones, they did not endorse the rival opinions.

To address the above-mentioned issues, the authors created opinion-target pairs in which, the opinion was a word that matched any entry from Subjectivity lexicon (Wilson et al., 2005), and the target was a word that was highly related to a domain topic. To match the opinion tokens with target tokens, the authors build a syntactic rule-based system using Stanford Parser[11]. For instance, dobj(opinion, target) is a rule that paired an opinion with a target that was the direct object of the opinion: *I love (opinion) Firefox (target) and defended (opinion) it (target)*. Once these opinion-target pairs were created, the authors masked the identity of the opinion word, replacing the word with its polarity and transforming each opinion-target pair to a polarity-target pair. For instance *love-Firefox* was masked to $Firefox^{+}$ (read as Firefox positive). The authors measured the probability that the target of an opinion (*positive* or *negative*) appeared in the vicinity of the topic-related opinion:

$$P(topic_j^q \mid target_i^p) = \frac{\#(topic_j^q, target_i^p)}{\#(target_i^p)}; j = \{1,2\}, i = \{1 \ldots M\}, p = \{+,-\}, q = \{+,-\}$$

Where M is the number of unique targets in the web corpus; p, and q represent the polarity. The authors collected the web corpus (development data) used to calculate the $P(topic_j^q \mid target_i^p)$ probability by downloading related weblog and forum webpages using Yahoo API. In the test data, if N is the number of polarity-target pairs in a post, $I_j (j = \{1 \ldots N\})$ is an instance of polarity-target in the post.

[11] http://nlp.stanford.edu/software/lex-parser.shtml.

Somasundaran et al. (2009) created two scores for each polarity-target instance I_j:

$$w_j = p(topic_1^+ \mid target_i^p) + p(topic_2^- \mid target_i^p)$$
$$u_j = p(topic_1^- \mid target_i^p) + p(topic_2^+ \mid target_i^p)$$

For each instance of polarity-target (I_j), the authors looked up w_j and u_i learned for the web corpus, where $target_i^p$ is the polarity-target type of which I_j is an instance. The overall side of a post, given all N polarity-target instances in the post, was determined by maximizing $\sum_{j=1}^{N}(w_j x_j + u_j y_j)$, where x_j and y_j have the values of 0 or 1, and $x_j + y_j = 1; x_j - x_{j-1} = 0, y_j - y_{j-1} = 0$.

For the above mentioned rule classifier, the authors achieved the accuracy ranging from 61% to 66.67% for a variety of dual topics such as *Windows vs Mac*, and *Firefox vs Internet Explorer*.

2.6 Discourse Analysis

Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) is one of the most widely accepted frameworks for discourse analysis. In the framework of RST, a coherent text can be represented as discourse tree whose leaves are non-overlapping text spans called discourse elementary units (EDUs). These are the minimal text units of discourse trees (Feng et al., 2012).

Adjacent nodes can be related through particular discourse relations to form a discourse sub tree, which can then be related to other adjacent nodes in the tree structure. According to RST, there are two types of discourse relations, mononuclear and multi-nuclear (Mann and Thompson, 1988). In mononuclear relations, one of the text spans, the *nucleus*, is more salient than the other, the *satellite*. In multi-nuclear relations, all text spans are equally important for interpretation (Feng et al., 2012).

Marcu (1997) brought discourse parsing into attention for the first time. Since then, many different algorithms and systems (Soricut and Marcu, 2003; Reitter, 2003; Le Thanh et al., 2004; Baldridge and Lascarides, 2005; Subba and Di Eugenio, 2009; Sagae, 2009; Hernault et al., 2010b) have been proposed, which extracted different textual information and adopted various approaches for discourse tree building.

HILDA discourse parser was the first fully implemented feature-based discourse parser that works at the full text level (Hernault et al., 2010). This parser was used as basis for work done by Feng et al. (2012) and became a fully implemented text-level discourse parser with the best-reported performance. The authors used a bottom-up approach to build a discourse tree for the full text:

1. An input text is segmented into EDUs.

2. A binary structure classifier evaluates whether a discourse relation is likely to hold between consecutive EDUs. The two EDUs, which are most probably connected by a discourse relation, are merged into a discourse sub tree of two EDUs.

3. A multi-class *Relation* classifier evaluates which discourse relation label should be assigned to this new sub tree.

4. The *Structure* classifier and the *Relation* classifier are employed in cascade to re-evaluate which relations are the most likely to hold between adjacent spans (discourse sub trees of any size, including atomic EDUs).

This procedure is repeated until all spans are merged, and a discourse tree covering the full text is therefore produced.

Feng (et al., 2012) used the RST Discourse Treebank (RST-DT) (Carlson et al., 2001) as the training data for their discourse parser. RST-DT is a corpus annotated in the framework of RST and includes 24 discourse markers defined by Mann and Thompson (1988). The list of discourse markers extracted from our dataset can be found in Appendix C.

Chapter 3 Background Concepts

One of the aims of Linguistics is to understand the structure of a language. One way to understand the structure of the language is by exploring detailed grammars that attempt to describe the language. However, even a complete set of grammatical rules cannot perfectly describe the language used in communication between people. Statistical Natural Language Processing is a branch of Natural Language Processing (NLP) that seeks to understand the complex structure of the language by finding common patterns that occur frequently by using statistical methods.

As the focus of this thesis, sentiment analysis (also referred as opinion mining) uses statistical NLP, machine learning, and computational linguistics methods in order to extract subjective information and determine the polarity of opinions expressed in text.

The goal of Machine Learning is to develop algorithms that automatically search for common patterns in data. Machine Learning is used in many text analysis tasks, sentiment analysis being one of them. Sentiment classification, as part of sentiment analysis, is the categorization of documents into different sentiments (e.g., positive or negative) using machine-learning algorithms (Manning, C., & Schutze H., 1991).

In classification, human experts annotate training data and class labels (e.g., *positive* or *negative*) are known before the classification algorithms. Accuracy of classification is usually high, but preparing training data is always expensive since the data requires manual annotation by human experts. Alternatively, it is possible to automatically learn from unlabeled data; this task is referred to as unsupervised learning. Clustering is a good example of unsupervised learning. Clustering algorithms simply place similar words into a same group and assign different words to different groups.

In semi-supervised learning, the system is seeded with labeled data and further learning from unlabeled data augments the initial data. Among machine learning algorithms used in statistical NLP, the Naïve Bayes supervised classification is the most commonly used technique, because the technique is fast and it works well with text data
The main concepts used in our work can be divided into five categories:

- Basic Concepts
- Pre-processing
- Feature Selection
- Classification (Machine Learning)
- Annotation and Guidelines

3.1 Basic Concepts

The classical bag-of-word model is often used to represent documents in Machine Learning experiments (Pak & Paroubek, 2010). Bag-of-words is the simplified representation of a document in which the document is represented as a bag (a set which allows repetitions) of its words. To produce bag-of-words, each document in a data set is reduced to a dictionary in which an entry is transformed to a <word, frequency of the word> pair. The bag-of-words model represents a document by reducing it to a vector of 0s and 1s, where each attribute shows presence (1) or absence (0) of a particular word in the document. In this representation, grammar and word order are discarded.

The following example shows the representation of a dataset contacting two documents using the bag-of-words model. First, a dictionary is created for each document, and then a bag-of-word vector is created from joining the dictionaries.

- Document 1: "As much as I hate to admit it, sometimes the future scares me."
- Document 2: "Previous year was the warmest year on the record, climatologists say."
- Dictionary 1: {"as": 2, "much": 1, "I": 1, "hate": 1, "to": 1, "admit": 1, "it": 1, "sometimes": 1, "the": 1, "future": 1, "scares": 1, "me": 1}
- Dictionary 2: {"previous": 1, "year": 2, "was": 1, "the": 2, "warmest": 1, "record": 1, "climatologists": 1, "say": 1}
- Joined Dictionary: {"admit": 1, "as": 2, "future": 1, "hate": 1, "I": 1, "it": 1, "me": 1, "much": 1, "previous": 1, "scares": 1, "sometimes": 1, "the": 3, "year": 2, "warmest": 1}
- Document 1 represented as bag-of-words model: [1, 2, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0]
- Document 2 represented as bag-of-words model: [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 1]
- The dataset represented as bag-of-words vector: [1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 1]

An n-gram is a continuous sequence of *n* words found in a document. An n-gram of size one is referred to as unigram (shown in the above example); similarly, an n-gram of size two is called bigram, n-gram of size the called trigram, n-gram of size four is called 4-gram and so on. We used unigrams in text classification and bigrams, tri-grams, 4-grams, and 5-grams for collocation analysis (section 3.3.14).

3.2 Data pre-processing

Data pre-processing is used to reduce the dimensionality of a dataset (Kouloumpis et al., 2011). Since the bag-of-words model is used to represent a dataset, the number of unique words in the dataset is its dimensionality. We removed *irrelevant* and redundant data to reduce the dimensionality of our dataset. Text pre-processing often starts with removing stop-words. Stop-words are highly frequent words (e.g., *the, of, for*) that are not meaningful to the task of sentiment analysis.

Lists of stop-words are not unique. For the sentiment analysis task, we use a list of stop-words from http://www.site.uottawa.ca/~diana/csi5180/StopWords. The list consists of words such as:

- Determiners, i.e., "the", "a", "an"
- Pronouns, e.g., "I", "me", "myself", "mine", "nobody", "someone", "this", "that"
- Numerical characters, e.g., "3", "45"
- Non-ASCII characters, i.e., non-English characters
- Punctuations marks, e.g., ":", "."
    - We kept two punctuation marks, namely "?", and "!". These punctuation marks emphasize a positive or negative emotional state and will be useful in the sentiment analysis.

Due to the nature of online data, a corpus collected from online forums and social networking sites could contain:

- URLs, e.g., "www.mothering.org"
- Non-English characters
- Emoticons, e.g., ":)", ":D", ":("
- Elongated words, e.g., "siiiimple"
- Slang, e.g., "wow", "lol"
- Incorrectly spelled English words, e.g., "pediatrisian"

There are two approaches for handling URLs in pre-processing: expanding the corpus by importing data from the target URLs or removing the URL. In the first approach, the content of any page visited via any URLs in the corpus is included in the original corpus. In this work, the latter approach was chosen for its simplicity (we removed the URLs from the entire corpus).

Non-English characters can be removed, since these characters are meaningless in the given context and will not contribute to the sentiment analysis task. It is important not to remove emoticons since they carry positive and negative sentiment and can be useful in the sentiment analysis task. Although elongated words can be corrected to reduce the dimensionality of the dataset, elongated words were kept as their original forms. The elongated words emphasize the positive or negative sentiment in a sentience and can be useful to the sentiment analysis task. For instance, in the following sentence "*wowww*" could be corrected to "wow"; however it emphasizes the negativity of the sentence and was kept as "wowww" during the data pre-processing setp. "*Wowww. I'm so sorry. What is the prognosis?* "

Finally, incorrectly spelled words can be corrected. We corrected misspelled words since these words do not emphasize on a sentiment in a sentence and simply are misspelled by

users. We corrected the misspelled words in our dataset by using tools such as Java Language Tool.[12]

3.3 Text Classification

Given a document $d \in \mathrm{X}$, where X is a document space, a set of classes $\mathbb{C} = \{c_1, c_2, c_3 \ldots, c_m\}$, and a training set of documents $< d, c > \in \mathrm{X} \times \mathbb{C}$, text classification is learning a classifier that maps documents to classes (Manning at al., 2008). The classification algorithms used for this thesis are Naive Bayes (NB) and Support Vector Machine (SVM).

3.3.1 Naïve Bayes Classification

Bayesian classifiers predict the probability that a given document *d* being in a particular class *c*, and assume conditional independence.

$$P(c \mid d) = P(c) \prod_{1 \le k \le n_d} P(t_k \mid c) \tag{3.1}$$

In formula 3.1, n_d is the number of terms in document *d*, and $P(t_k \mid c)$ is the probability that term t_k in document *d* belongs to class *c*. There are two different NB classifiers namely, multinomial NB and Bernoulli NB. Both classifiers attempt to find the best class *c* for document *d*. The best class *c* is the one that maximizes $P(c \mid d)$ probability.

$$\begin{aligned} &\arg\max_{c \in C} P(c \mid d) \\ &= \arg\max_{c \in C} \frac{P(d \mid c) P(c)}{P(d)} \\ &= \arg\max_{c \in C} P(d \mid c) P(c) \end{aligned} \tag{3.2}$$

Where $P(c)$ is the fraction of documents with class *c*. $P(c \mid d)$ is the probability of a document *d* being in class *c*. $P(d)$ is the same for all classes and does not affect the *argmax*, therefore was removed from the denominator.

The multinomial model generates one term from the vocabulary (*V*) in each position of the document. Alternatively, the Bernoulli model generates an indicator (e_i) for each term of the vocabulary, either 1 for the presence of the term in document or 0 for the absence (Manning et al., 2008). Given the conditional independence assumption:

[12] Java Language Tool is a Java-based Spell Checker: Retrieved from http://wiki.languagetool.org/java-spell-checker

$$Multinomial: P(d \mid c) = P(< t_1, t_2, \ldots, t_{n_d} > \mid c) = \prod_{1 \le k \le n_d} P(t_k \mid c) \quad (3.3)$$

$$Bernoulli: P(d \mid c) = P(< e_1, e_2, \ldots, e_M >) = \prod_{1 \le i \le M} P(e_i \mid c) \quad (3.4)$$

Where $< t_1, t_2 \ldots, t_{n_d} >$ and $< e_1, e_2, \ldots, e_M >$ are two forms of document representation. In the first form a document is represented as the vector $< t_1, t_2 \ldots, t_{n_d} >$ in which *t* is a term in document *d* that is the part of the vocabulary (V). In this Vocabulary, *n* is the number of terms in the vocabulary and n_d is the number of such terms that appear in document *d*. Furthermore, the sequence of terms in $< t_1, t_2 \ldots, t_{n_d} >$ appear in the same order as they appear in the document *d*.

In the second form of document representation, $< e_1, e_2, \ldots, e_M >$ is the binary vector of dimensionality *M* that indicates whether each term *t* occurs in document *d* or not.

Therefore, in the multinomial model, the document space X is the set of all term sequences in the vocabulary (*V*), whereas in the Bernoulli model, the document space X is $\{0,1\}^M$, the set of 0s and 1s that indicates the presence of term *t* in document *d*.

The probability of a document *d* being in class *c* is calculated as (Manning et al., 2008):

$$Multinomial: P(d \mid c) \propto P(c) \prod_{1 \le k \le n_d} P(t_k \mid c) \quad (3.5)$$

$$Bernoulli: P(d \mid c) \propto P(c) \prod_{1 \le i \le M} P(e_i \mid c) \quad (3.6)$$

For both multinomial and Bernoulli model, $P(c)$ is estimated as $P(c) = \frac{N_c}{N}$, where N_c is the number of documents in class *c* and *N* is the total number of documents. As for the multinomial model, $P(t_k \mid c)$ is the conditional probability of term t_k occurring in a document of class *c*. The letter *k* represents the position of term *t* in document *d*. The conditional probability $P(t \mid c)$ is estimated as the relative frequency of term t in documents with class c:

$$P(t \mid c) = \frac{T_{ct} + 1}{\sum_{t' \in V} T_{ct'} + 1} \quad (3.7)$$

Where, T_{ct} is the number of occurrences of term *t* in training documents from class *c*, including multiple occurrences of a term in a document.

In contrast, the Bernoulli model estimates conditional probability $P(t \mid c)$ as the relative frequency of documents with class *c* that contains term *t* (Manning et al., 2008):

$$p(t \mid c) = \frac{N_{ct} + 1}{N_c + 2} \tag{3.8}$$

Where, N_{ct} is the number of documents of class c that contains term t, and N_c is the number of documents of class c.

3.3.2 Support Vector Machines

Support Vector Machine (SVM) is a learning algorithm that transform the original training data into a space with higher dimensions; then within the new dimensions the SVM algorithm searches for the maximal marginal decision hyper-plane (MMDH) that can separate documents of one class from another one (Han et al., 2012).

For clarity, we explain SVM for a simple case of binary classification where the classes are linearly separable (Figure 4).

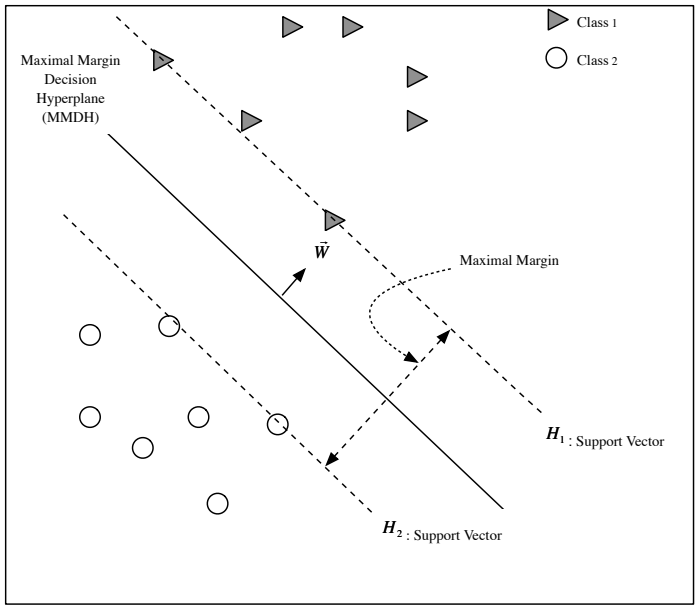

Figure 4 - SVM (Adapted From Han et al, 2012)

Let corpus D be given as $(\vec{X}_1, y_1), (\vec{X}_2, y_2), \ldots, (\vec{X}_{|D|}, y_2)$ where $\vec{X}$ is the feature vector representing document d and y_1 and y_2 ($y_2 = \overline{y}_1$) are two classes associated with any feature vector, and $|D|$ represents the size of corpus D. The role of SVM is to assign the document d, represented as the vector $\vec{X}$, to class y_1 if $\vec{W} \cdot \vec{X} > b$ and assign document d represented as vector the $\vec{X}$ to class y_2 if $\vec{W} \cdot \vec{X} \leq b$.

The hyper plane separating documents with different classes is defined by bias b (parameter that define the decision boundary), and normal vector $\vec{W}$ (Han et al., 2012). Therefore, a separating hyper-plane (decision surface) can be defined as $\vec{W} \cdot \vec{X} + b = 0$ (Figure 4).

Assuming that -1 indicates class y_1 and +1 indicates class $y_2 = \overline{y}_1$ then the SVM linear classifier can be written as (Manning et al., 2008)

$$f(\vec{X}) = sign(\vec{W} \cdot \vec{X} + b) \tag{3.9}$$

If the value of $f(\vec{X})$ for document d represented as vector $\vec{X}$ is -1, the document d belongs to one class, and if the value of $f(\vec{X})$ for document d is +1, the document belongs to another class.

If the document d represented as the vector $\vec{X}$ is far away from the decision surface (hyperplane), then we can be confident in the classification result. To get a better sense, this distance can be measured as the Euclidean distance of $\vec{X}$ from the decision surface (the vertical distance (r) between the document $\vec{X}$ and the decision surface) (Figure 5).

$$\vec{X} - \vec{X}' = yr\frac{\vec{W}}{|\vec{W}|} \tag{3.10}$$

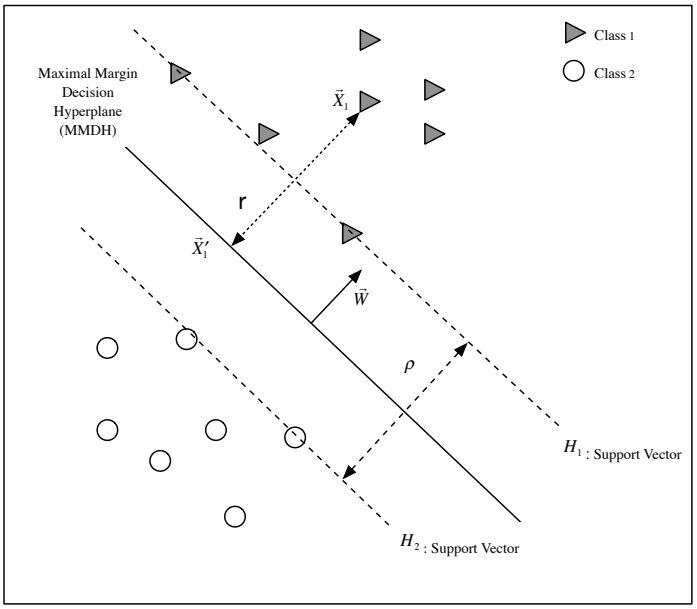

Figure 5 - SVM (Adapted From Han et al. 2012)

In formula 3.10, $\vec{X}'$ is the closest point on the hyperplane to the document *d*, represented as the vector $\vec{X}$. Furthermore, $\frac{\vec{W}}{|\vec{W}|}$ represents the hyperplane unit vector. Therefore, $r\frac{\vec{W}}{|\vec{W}|}$ represents the distance between document *d*, represented as vector $\vec{X}$, and the hyper plane.

3.3.3 Classifier Evaluation

In this section, we describe the evaluation metrics and approaches that assess the quality of a classifier performance. Although several classification metrics are commonly used in text classification, such as Accuracy, Error rate, Precision, Recall, and F-measure, they are all based on Confusion matrix. Hence, we introduce the matrix first.

3.3.4 Confusion matrix

The confusion matrix shows how well classifiers can recognize documents of different classes. For simplicity, we assume that the documents are classified into two classes (i.e., *Positive* and *Negative*). Considering the two-class classification, the confusion matrix consists of true positives, true negatives, false positives, false negatives, and the total of them (Table 1).

- *Positive* (P): the total number of positive documents
- *Negative* (N): the total number of negative documents
- *True positive* (TP): the number of positive documents that are correctly classified by the classifier
- *True negative* (TN): the number of negative documents that are classified correctly by the classifier
- *False positive* (FP): the number of negative documents that are classified as positive
- *False negative* (FN): the number of positive documents that are classified as negative

To generalize, for a corpus annotated with *m* class labels, the confusion matrix is a $m \times m$ matrix. An entry M_{ij} indicates the number of documents of class *i*, labeled as class *j* by the classifier; hence, for a classifier with high accuracy, most of the documents appear along the diagonal of the confusion matrix (Manning, C., & Schutze H., 1991).

		Predicted Class		
		Positive	**Negative**	**Total**
Actual Class	**Positive**	TP	FN	P
	Negative	FP	TN	N
	Total	P	N	P+N

Table 1 - Confusion Matrix

3.3.5 Accuracy

The accuracy of a classifier on a given test set is the fraction of documents classified correctly by the classifier.

$$accuracy = \frac{TP + TN}{P + N} \tag{3.11}$$

3.3.6 Error rate

The error rate is the fraction of the documents that are misclassified by a classifier on a given data set.

$$error = \frac{FP + FN}{P + N} \tag{3.12}$$

3.3.7 Precision

Precision (exactness) of a classifier on a given test set is the fraction of documents classified as positive are actually positive.

$$precision = \frac{TP}{TP + FP} \tag{3.13}$$

3.3.8 Recall

Recall (completeness) is the fraction of positive documents classified as positive (not missed).

$$recall = \frac{TP}{TP + FN} \tag{3.14}$$

3.3.9 F-measure

In order to understand the importance of F-measure, we consider a classifier with perfect Precision or prefect Recall. Precision with score of 1.0 means that all documents which classifier labeled as class A belong to class A; however, it does not give us any information about how many documents are misclassified. On the other hand, a recall of score 1.0 means that every document from class A is classified as class A but it does not give us any information about how many other documents are misclassified as class A. Figure 6 shows the inverse relationship between Precision and Recall.

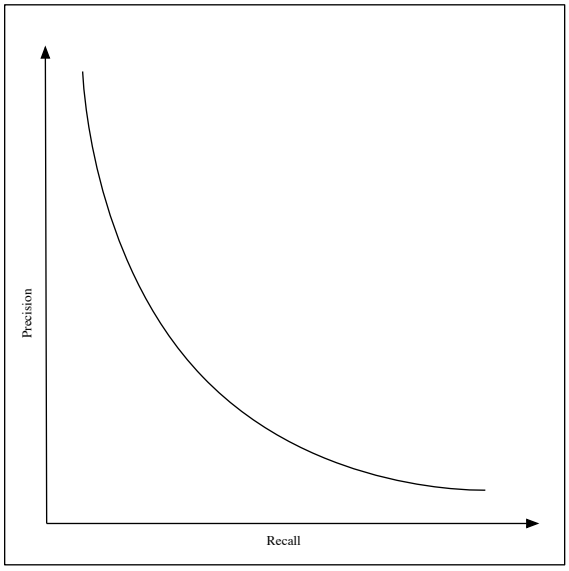

Figure 6 - Precision and Recall (Adapted from Manning et al., 2008)

Therefore, the higher the Precision is, the lower is the corresponding Recall. In order to achieve a balance between precision and recall, F-measure is introduced as a harmonic mean of precision and recall. We use the balanced F-measure, which equally weighs Precision and Recall:

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{3.15}$$

3.3.10 TF-IDF

TF-IDF is a weighing scheme that is assigned to a term in a dataset. TF stands for term frequency, which is the number of time that a term appears in the dataset. IDF stands for inverse document frequency, which is the inverse of the number of documents in the dataset containing term. If a term appears in all documents in the dataset, TF-IDF will be very low for the term. In contrast, if a term appears many times in few number of

documents, TF-IDF will be high for the term. The higher the TF-IDF is for a term in the dataset, the more important is the term for the dataset.

3.3.11 K-fold Cross Validation

K-fold cross validation is often used to estimate how accurately a classifier will perform in practice (Han et al., 2012). First, annotated training data is partitioned into *k* mutually exclusive subsets (folds) with the same class distribution (stratified) and size $D_1, D_2, D_3, \ldots, D_k$.

Then, the training and testing is performed in *k* iterations as follows:

- At first iteration classifier is trained on $D_2, D_3, \ldots, D_k$ and tested on D_1
- At second iteration classifier is trained on $D_1, D_3, \ldots, D_k$ and tested on D_2
- ⋮
- At k^{th} iteration classifier is trained on D_k and tested on $D_1, D_2, D_3, \ldots, D_{k-1}$

The overall accuracy is estimated as the total number of correctly classified documents over *k* iterations divided by the total number of annotated documents.

In this thesis, we chose to use 10-fold cross validation provided by WEKA software package since out dataset was not big enough to be separated into dedicated training and test sets.

3.3.12 Statistical test of significance

Suppose we want to choose between two classifiers C_1 and C_2. Can we compare the two classifiers just based on the result of an evaluation metrics i.e. *error rate*? Is C_1 better than C_2 if $err(C_1) < err(C_2)$? Not necessarily, the difference between C_1 and C_2 may be by chance (Han et al., 2012).

The purpose of a statistical test of significance is to determine whether the difference in an evaluation metrics, i.e., *error rate* between two classifiers is not by chance. Statistical significance testing quantifies probability that the difference between two classifiers is due to luck (hypothesis testing):

- If the probability is low, the two classifiers are different
- If the probability is high either the two classifiers are not different or there is not enough data to determine the difference.

For simplicity, suppose 10-fold cross validation is used to calculate the error rate for classifiers C_1 and C_2. The null hypothesis is that there is no difference between the two classifiers C_1 and C_2.

Then t-statistics is calculated as in (Han et al., 2012):

$$t = \frac{\overline{err}(C_1) - \overline{err}(C_2)}{\sqrt{\text{var}(C_1 - C_2)/k}} \tag{3.16}$$

$$\overline{err}(C_1) = \frac{\sum_{i=1}^{k} err(C_1)_i}{k} \tag{3.17}$$

$err(C_1)_i = Error\ rate\ at\ fold\ i^{th}\ for\ classier\ C_1; k = Number\ of\ folds$

For a single annotated training dataset and 10-fold cross-validation used for both classifiers:

$$\text{var}(C_1 - C_2) = \frac{1}{k-1}\sum_{i=1}^{k}[err(C_1)_i - err(C_2)_i - (\overline{err}(C_1) - \overline{err}(C_2))^2] \tag{3.18}$$

We select a significance level (*sig*) of 1% to 5% or respective confidence limit ($z = \frac{sig}{2}$) of 0.5% to 0.25%. Consulting a t-distribution table we find the value *t* for the corresponding confidence limit (*z*) and degree of freedom (*k-1*).

- If $t > z$ or $t < -z$ then we can reject the hypothesis that the classifiers C_1 and C_2 are the same.
- Otherwise we accept the hypothesis that the classifiers C_1 and C_2 are the same.

3.3.13 Improving Classification Accuracy

We used data selection (i.e., bagging and boosting) and attribute selection (feature selection) as two techniques to improve the accuracy of classification in our machine learning experiments.

Data Selection

An ensemble method combines *k* classifiers $C_1, C_2, C_3, \ldots, C_k$ to create a new classifier with usually higher accuracy. Data sets $D_1, D_2, D_3, \ldots, D_k$ are created from the original annotated data set *D* where D_1 is used to train classifier C_1, D_2 is used to train classifier

C_2, …, and D_k is used to train classifier C_k. For a document d in the corpus D, each classifier $C_i, i = \{1,\ldots,k\}$ then votes by returning a class label. The composite classifier then returns a single class prediction based on each classifier's C_i vote (Han et al., 2012).

In bagging (Breiman, 1996), k classifiers $\{C_1, C_2, C_3, \ldots, C_k\}$ are created on different datasets $\{D_1, D_2, D_3, \ldots, D_k\}$. The training sets are bootstrapped (split from the original dataset with replacement, bootstrapped). Each classifier accuracy is slightly better than chance (weak learner). The composite classifier generates the final prediction by averaging the result of all classifiers.

Boosting is based on PAC (Probably approximately correct) model (Kearns & Valiant, 1988), which implies a weak (base) learning algorithm can be boosted to a strong learning algorithm. The weak algorithm trained *k* times, each time it is fed by a different subset of the original training set *D*. This weak algorithm maintains a distribution of set of weights over the training set. Initially equal weights assigned to each instance of the training set. In each round *k*, more weight is assigned to misclassified instances so that the weak learner forced to focus on the incorrectly classified instances. At the end, all classifiers are combined based on an importance factor.

In boosting, contrary to bagging, a classifier in round *k* depends on the classifier in round *k-1* and focuses on the misclassified instances predicted by the previous classifier. Furthermore, in boosting, the training set instances are weighted, and instances that are misclassified in round *k-1* are weighted more heavily. In bagging, the training set instances are not weighted and the classifier in round *k-1* does not affect the classifier in the next round.

Feature selection

It should be noted that all formulas in this section are from Manning et al., 2008. Feature selection serves two purposes.

- Reduce the size of vocabulary and increase classification performance.
- Improve classification accuracy by removing noise features. Noise features are terms that are rare and occur only in one document. In addition, noise features do not have any information about the respected document class label.

Feature selection also helps to prevent classifier's over-fitting to the training data (Hsu et al., 2003). Mutual Information, Information Gain, and Chai Squared are common heuristics used in feature selection.

Information Gain

Given corpus $D =< d_1, d_2, \ldots, d_n >$ where d_n is a document in the corpus, and $C =< c_1, c_2, \ldots, c_m >$ where c_m is a class label from the set of class labels C, the information gain of term t in corpus D for each class label c is calculated as:

$$IG(D,t,c) = H(p_D) - \sum_{x \in \{D_{t+}, D_{t-}\}} \frac{|x|}{|D|} H(p_x) \tag{3.19}$$

Where

$$H(p_D) = -p_D(c)\log(p_D(c)) \;;\; H(p_x) = -p_x(c)\log(p_x(c)) \tag{3.20}$$

$$p_D(c) = \frac{\#\ documents\ labeled\ as\ c}{|D|} \;;\; p_x(c) = \frac{\#\ documents\ labeled\ as\ c}{|x|}$$

$$|D| = total\ number\ of\ documents\ in\ corpus\ D$$

$$|x| = total\ number\ of\ documents\ in\ corpus\ \text{x}$$

$$D_{t-} = subset\ of\ D\ excluding\ term\ t \;;\; D_{t+} = subset\ of\ D\ including\ term\ t$$

To calculate the information gain for term t, corpus D is split into two corpuses: the first corpus, D_{t+}, which is the corpus including terms t, and the second corpus, D_{t-}, which is the corpus excluding term t. The term x in formula 3.19 represents either of D_{t+} or D_{t-} corpuses. Since term t has binary value (i.e., *0,1*), in terms vector representation of each document in corpus *D:*

$$\sum_{x \in \{D_{t+}, D_{t-}\}} \frac{|x|}{|D|} H(p_x) = \frac{|D_{t+}|}{|D|} H(p_x) + \frac{|D_{t-}|}{|D|} H(p_x) \tag{3.21}$$

The higher the information gain of term t, the more the term t is representative of class c.

Mutual Information

Mutual information (MI) of term t and class c indicates the amount of information term t contains about class c. The higher the mutual information, the more term t is representative of class c. Mutual information is calculated as:

$$MI(U,C) = \sum_{e_t \in \{0,1\}} \sum_{c_t \in \{0,1\}} P(U = e_t, C = e_c) \log\left(\frac{P(U = e_t, C = e_c)}{P(U = e_t)P(C = e_c)}\right) \tag{3.22}$$

Where U is a random variable, which takes the value of e_t; C is a random variable, which takes the value of e_c. e_t and e_c can have possible values of 0 or 1. $e_t = 1$ indicates that a document has term t. In contrast, $e_t = 0$ indicates that a document does not have term t. Similarly, $e_c = 1$ indicates that a document is labeled as class c, and $e_c = 0$ indicates that a document is not labeled as class c.

Formula 3.22 can also be written as:

$$MI(U,C) = \frac{N_{11}}{N}\log(\frac{N \cdot N_{11}}{N_{1c} \cdot N_{u1}}) + \frac{N_{10}}{N}\log(\frac{N \cdot N_{10}}{N_{1c} \cdot N_{u0}}) + \frac{N_{01}}{N}\log(\frac{N \cdot N_{01}}{N_{0c} \cdot N_{u1}}) + \frac{N_{00}}{N}\log(\frac{N \cdot N_{00}}{N_{0c} \cdot N_{u0}}) \quad (3.23)$$

In the above formula, N_{11} is the number of documents that contain term t and are labeled as class c. In contrast, N_{01} is the number of documents that do not contain term t and are labeled as class c. N_{1C} ($N_{1C} = N_{10} + N_{11}$) is the number of documents that contain terms t. in contrast, N_{0C} ($N_{0C} = N_{01} + N_{00}$) is the number of documents that do not contain term t. Finally, N_{U1} ($N_{U1} = N_{11} + N_{01}$) is the number of documents that are labeled as class c, and N_{U0} ($N_{U0} = N_{10} + N_{00}$) is the number of documents that are not labeled as class c.

Chi Squared

The chi squared (χ^2) test indicates wether the occurrence of term t and the occurrence of class c are independent of each other. Given the null hypothesis that expected frequency $E_{e_t e_c}$ and observed frequency $N_{e_t e_c}$ are similar (the occurrence of t and the occurrence of c are independent), a high value of χ^2 indicates the rejection of the null hypothesis:

$$\chi^2(D,t,\ c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \quad (3.24)$$

$E_{e_t e_c}$ is the expected number of occurrences of term t and class c in relation to each other For instance, $E_{11} = NP(t,c) = N \times P(t) \times P(c)$, where $P(t) = \frac{N_{10} + N_{11}}{N}$ and $P(c) = \frac{N_{11} + N_{01}}{N}$. N_{11} is the number of occurrences of term t and class c together. N_{10} is the number of occurrences that term t does not appear in a document of class c. Therefore, the equation can be rewritten as:

$$\chi^2 = \frac{(N_{11} + N_{10} + N_{01} + N_{00}) \times (N_{11}N_{00} - N_{10}N_{01})^2}{(N_{11} + N_{01}) \times (N_{11} + N_{10}) \times (N_{10} + N_{00}) \times (N_{01} + N_{00})} \quad (3.25)$$

Given $\chi^{2'}$ from chi-squared distribution table for confidence level of α (usually 0.05, 0.01, 0.005 or 0.001), if $\chi^2 > \chi^{2'}$ then we can reject null hypothesis that term *t* is not representative of class *c*.

3.3.14 Word Collocations

It should be noted that all formulas in this section are from (Manning & Schutze,1991).

A collocation is a composition of two or more words for which the meaning of the whole cannot be fully predicted by meaning of the parts; *Fast food* and *take a chance* are two examples of collocations. Collocations can be used to generate a dictionary of related words via bootstrapping. This dictionary of domain related words is used to identify domain-relevant documents. There are four principal approaches to finding collocations:

- Frequency
- Mean and variance of distance (offset) between collocating words
- Hypothesis testing
- Mutual information

Counting is the simplest way of finding collocations. Words that appear in the most frequent fixed phrases (e.g., bigram phrases) are strongly related.

Collocations can be extended to phrases consisting of words with flexible relations. Mean and variance of signed distance (offset) between two words is used to find the pattern of varying distance between two words. If this pattern is predictable, there is a collocation between two words. For instance, to find if $w_1 w_2$ are correlated:

- Define collocation window of 3 to 4 words on each side of w_1 and w_2
- Calculate mean of the offset between w_1 and w_2 in the corpus
- Calculate how much the individual offsets deviates from the mean of the offsets

The variance measure is used to calculate how much the individual offsets deviates from the mean:

$$\sigma^2 = \frac{\sum_{i=1}^{n} (d_i - \bar{d})^2}{n-1} \tag{3.26}$$

where n is the number of times that $w_1 w_2$ co-occur, d_i is the offset for co-occurrence i and $\bar{d}$ is the sample mean of the offsets. The distance d_i can be negative if w_2 occur before w_1. If the standard deviation s ($s = \sqrt{\sigma^2}$) is low (considerably lower than the mean), it means that two words occur at relatively the same distance from each other and are collocated.

The problem with frequency and mean/variance method is that if two words often occur together by chance then they will be marked as correlated. The hypothesis testing method tests whether two words occur more often than chance.

- Assume null hypothesis H_0 that two words are not correlated
- Calculate probability p when H_0 is true
- Reject H_0 if p is lower than significant level (usually 0.05 or 0.005)

The principal three principal approaches for hypothesis testing are:

- T-test
- Pearson's chi-squared
- Log likelihood ratio

T-test is statistical test that tells us how probable it is that a certain collocation occurs. To see how to use the t-test, lets calculate the t value for the collocation $w_1 w_2$:

$$P(w_1) = \frac{The\ number\ of\ times\ w_1\ occurs\ in\ the\ corpus}{The\ total\ number\ of\ tokens\ in\ the\ corpus}$$

$$P(w_2) = \frac{The\ number\ of\ times\ w_2\ occurs\ in\ the\ corpus}{The\ total\ number\ of\ tokens\ in\ the\ corpus}$$

$$P(w_1, w_2) = \frac{count(w_1, w_2)}{The\ total\ number\ of\ bigrams\ in\ the\ corpus}$$

$$H_0 : P(w_1, w_2) = P(w_1) \times P(w_2)\ ;\ t = \frac{\overline{x} - \mu}{\sqrt{\frac{\sigma^2}{N}}}$$

Where H_0 is the null hypothesis, which assumes that w_1 and w_2 are not correlated. $\overline{x}$ is the sample mean, which is equal to $P(w_1, w_2)$, μ is the mean of the distribution, which is equal to $P(w_1) \times P(w_2)$. σ^2 is the variance and is equal to $P(1-P)$. Finally, N is the number of unigram tokens in the corpus.

If we look up the t value that corresponds to a confidence level of α from a t-table, we will find t'. If $t < t'$ we accept the null hypothesis (H_0) that w_1 and w_2 are not correlated; otherwise we reject the null hypothesis (H_0) and accept that there is correlation between w_1 and w_2.

The t-test assumes normal distribution of the input data. An alternative collocation measure, which doesn't assume normally distributed probabilities, is χ^2 (chi-squared) test. Chi-squared test is applied to contingency tables (Table 2) by comparing observed

frequencies with frequencies expected for independence. If the difference is large, the null hypothesis of independence is rejected.

	w_1	$\sim w_1$	**Marginal**
w_2	O_{11}	O_{12}	$O_{11}+O_{12}$
$\sim w_2$	O_{21}	O_{22}	$O_{21}+O_{22}$
Marginal	$O_{11}+O_{21}$	$O_{12}+O_{22}$	

Table 2 - Contingency Table for Bigrams

Chi-squared statistic sums the difference between observed and expected frequencies in all squares of contingency table as follows:

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{3.27}$$

where i, j represent the row and the column in the contingency table (Table 2). The size of the contingency table depends on the size of n-gram chosen for collocation analysis. For the sake of simplicity, we will choose a 2-by-2 contingency table (Table 2). Observed frequency (O_{ij}) is calculated as follows:

- O_{11} = The count of bigrams with w_1 as the first word and w_2 as the second word
- O_{12} = The count of bigrams with w_2 as the second word and any word other than w_1 (represented as $\sim w_1$ in table 2) as the first word
- O_{21} = The count of bigrams with w_1 as the first word and any word other than w_2 (represented as $\sim w_2$ in table 2) as the second word
- O_{22} = The count of bigrams with any word other than w_1 (represented as $\sim w_1$ in table 2) as the first word and any word other than w_2 (represented as $\sim w_2$ in table 2) as the second word

Expected frequency (E_{ij}) is calculated with the assumption that null hypothesis is true. Null hypothesis is that the occurrence of two words is independent of each other. Expected frequency is calculated from marginal probabilities as follows:

- $E_{11} = N \times \frac{O_{11}+O_{21}}{N} \times \frac{O_{11}+O_{12}}{N}$
- $E_{12} = N \times \frac{O_{12}+O_{22}}{N} \times \frac{O_{11}+O_{12}}{N}$
- $E_{21} = N \times \frac{O_{11}+O_{21}}{N} \times \frac{O_{21}+O_{22}}{N}$

- $E_{22} = N \times \frac{O_{12} + O_{22}}{N} \times \frac{O_{21} + O_{22}}{N}$

Looking up $\chi^{2'}$ for α from chi-squared distribution table, if calculated χ^2 is smaller than $\chi^{2'}$ then the null hypothesis holds true and we accept than w_1 and w_2 are correlated otherwise we reject the null hypothesis

Chi-squared is suitable for large corpus and is inaccurate if expected frequencies are very small. An alternative to chi-squared test is log-likelihood ratio test. To apply log-likelihood ratio test to collocation discovery we consider two hypotheses:

- H_1 : Occurrence of w_2 is independent of the previous occurrence of w_1
- H_2 : Occurrence of w_2 is dependent of the previous occurrence of w_1

The first hypothesis implies that $P(w_2 \mid w_1) = P(w_2 \mid \sim w_1) = \frac{c_2}{N}$, where c_2 is the number of bigrams with w_2 as the second word, and *N* is the tot al number of bigrams in the corpus. However, the second hypothesis implies that $P(w_2 \mid w_1) \neq P(w_2 \mid \sim w_1)$ where $P_1 = P(w_2 \mid w_1) = \frac{c_{12}}{c_1}$, and $P_2 = P(w_2 \mid \sim w_1) = \frac{c_2 - c_{12}}{N - c_1}$. c_{12} represents the number of bigrams with w_1 as the first word and w_2 as the second word. Furthermore, $c_2 - c_{12}$ represents the number of bigrams with w_2 as the second word and any word other than w_1 as the first word. $N - c_1$ is the number of bigrams with any word other than w_1 as the first word.

Assuming binomial distribution, likelihood functions are calculated as:

- $L(H_1) = b(c_{12}; c_1, \frac{c_2}{N}) b(c_2 - c_{12}; N - c_1, \frac{c_2}{N})$
- $L(H_2) = b(c_{12}; c_1, \frac{c_{12}}{c_1}) b(c_2 - c_{12}; N - c_1, \frac{c_2 - c_{12}}{N - c_1})$

In binomial distribution the number of *k* successes out of *n* trials, given the probability of success *P* in any trial, is calculated as $b(k; n, P) = \begin{pmatrix} n \\ k \end{pmatrix} P^k (1 - P)^{n-k}$.

Log-likelihood ratio test ($\log \lambda = \frac{L(H_1)}{L(H_2)}$) simply tells us how much more likely is one hypothesis over the other one. The higher the value of $-2 \log \lambda$, more likely it is that two terms are collocated.

3.4 Measuring Agreement between Annotators

In supervised learning, class labels are known for the training sample. Annotators label the training data before any classifiers can be built. It is important to measure the reliability of annotation and to know if annotators agreed upon class labels by chance. Kappa statistics is a common measure for agreement between two judges:

$$kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{3.28}$$

Where $P(E) = probability\ that\ two\ judges\ agreed\ by\ chance$ and
$P(A) = observed\ proportion\ of\ times\ that\ two\ judges\ agreed$

The expected chance agreement rate is 50%. However, normally the class label distribution is skewed and it is usual to use marginal statistics to calculate the expected agreement. Summing a row or a column forms marginal statistics in a class labels contingency table. For a three-class classification (*i.e.*, *positive*, *negative*, and *mixed*) kappa calculates as follows (Table 3):

	Judge 2				
Judge 1		**Positive**	**Negative**	**Mixed**	**Marginal**
	Positive	O_{11}	O_{12}	O_{13}	$O_{11}+O_{12}+O_{13}$
	Negative	O_{21}	O_{22}	O_{23}	$O_{21}+O_{22}+O_{23}$
	Mixed	O_{31}	O_{32}	O_{33}	$O_{31}+O_{32}+O_{33}$
	Marginal	$O_{11}+O_{21}+O_{31}$	$O_{12}+O_{22}+O_{32}$	$O_{13}+O_{23}+O_{33}$	**Total**

Table 3 - Generic Confusion Matrix For Three Class Classification

$$Total = (O_{11}+O_{12}+O_{13})+(O_{21}+O_{22}+O_{23})+(O_{31}+O_{32}+O_{33})$$

$$P(A) = \frac{O_{11}+O_{22}+O_{33}}{Total}$$

$$P(E)_{positive} = \frac{(O_{11}+O_{12}+O_{13})\times(O_{11}+O_{21}+O_{31})/Total}{Total}$$

$$P(E)_{negative} = \frac{(O_{11}+O_{12}+O_{13})\times(O_{12}+O_{22}+O_{32})/Total}{Total}$$

$$P(E)_{mixed} = \frac{(O_{31}+O_{32}+O_{33})\times(O_{13}+O_{23}+O_{33})/Total}{Total}$$

$$P(E) = P(E)_{positive} + P(E)_{negative} + P(E)_{mixed}$$

As a rule of thumb, a kappa value over 0.8 is taken as a good agreement. A kappa value between 0.67 and 0.8 is taken as a fair agreement and a kappa value below 0.67 is a taken as agreement by chance (Carletta, J., 1996).

3.4.1 Argumentation

Argumentation is composed of a set of arguments used to attack or support a hypothesis. For instance consider: "Newborn genome screening would be helpful since it can help preventing life-threatening diseases to devolve". The hypothesis is the phrase "Newborn genome screening would be helpful" and the supporting argument is "Newborn genome screening can help prevent life-threatening diseases".

Each argument can be seen as a set of statements supported with some evidence. There are two types of argumentation styles namely, monologic and dialogic argumentation.
In monologic argumentation, a person starts a debate by posting a topic. Debate participants can then post comments for or against the topic; therefore, there is no exchange of comments between participants. In contrast, in dialogic argumentation, participants attack or support other participants' point of views. Most of the online forums enjoy dialogic argumentation style, since participants usually comment on each other's comments in addition to the topic of conversation

In this chapter, basic information to understand the theory behind the experiments was provided. In the next chapter, data collection as a step to provide high quality data for experiments will be explained.

Chapter 4 Data Preparation

In this chapter we explain the data collection, manual annotation of our dataset, and the preliminary statistical analysis that was done on the dataset to get a better feeling about the data.

4.1 Data Collection

Newborn genome screening (NGS) offers screening for serious diseases to all newborn babies. Early identification of these diseases allows treatment that may prevent growth problems, health problems, mental retardation, and sudden infant death. In Ontario, the Guthrie test is administered shortly after birth to test for diseases that are not usually apparent in the neonatal period. Early detection of these diseases is the key to effective treatment and can ease or prevent serious health problems and even save lives. However, not every parent or child's guardian is convinced by the benefits of NGS.

To analyse opinion of the general public, we collected data from medical online forums and blogs. A forum is an online discussion web site where members can post discussions, read and respond to posts by forum other members. A single conversation is called a thread, which includes the initial comment and all the replies (comments) to it. A comment is a set of sentences written by a user in response to a discussion thread. The following is an illustration of the hierarchical structure of a discussion, threads, and replies on a forum (Figure 7):

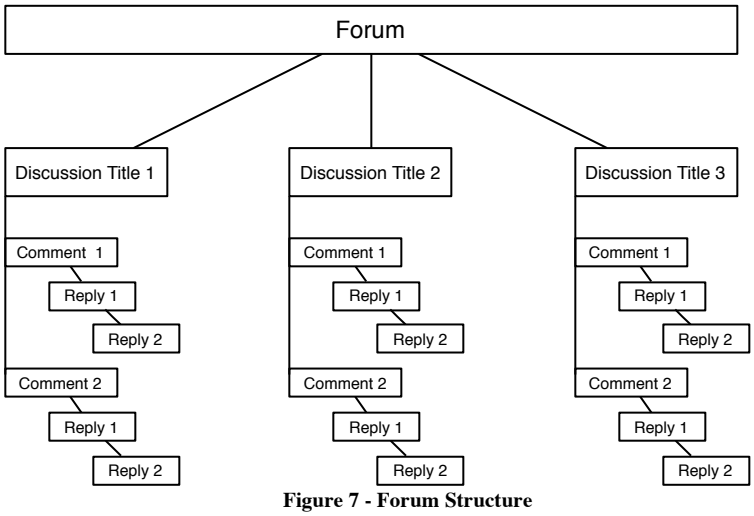

Figure 7 - Forum Structure

A blog is a discussion or information website that usually has a single contributor which can be a person or an organization. People can comment on each post and the comments are usually shown in reverse chronological order. The following is an illustration of the structure posts and comments on posts in a blog (Figure 8):

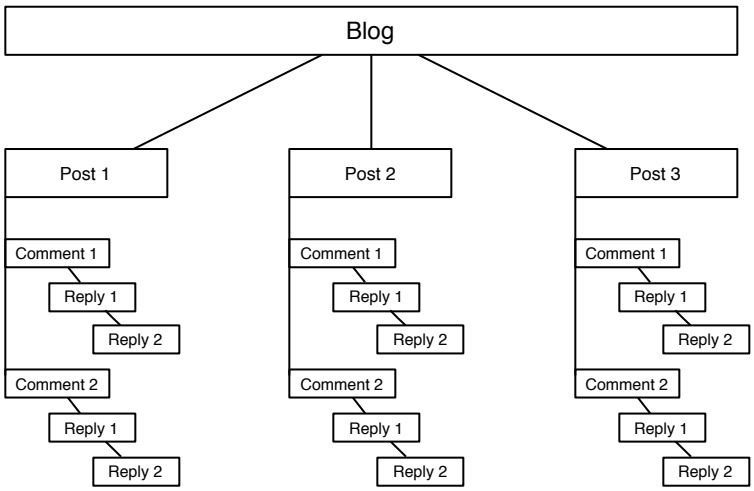

Figure 8 - Blog Structure

Although the hierarchical structure of online forums and blogs are the same, there are key differences between forums and blogs:

- In forums, only members can comment on a discussion title in a thread, whereas in blogs any user can comment on a post; therefore, no login and registration is required for users to comment on a blog post.
- In forums all members can start a discussion thread, whereas in blogs a person or a small group is responsible for posting articles; therefore, blogs centralized and more focused, whereas forums are decentralized and less focused.
- Forums are designed for discussion between many users, whereas blogs are mainly designed for single user post. However, any user can comment on a post.

We were interested in the comments that blog readers write under a blog post. In a blog, similar to a forum thread, comments were about a topic (blog post). Therefore, we assumed that comments from blogs and forums were similar enough that we treated blogs and forums as a single domain.

To find related forums and blogs, we used all the possible combinations of the following search terms as candidate search phrases:

- {Forum, Thread, Bulletin Board, Message Board, BBS}
- {Newborn, infant}
- {Genetic Predisposition Testing, Genetic Screening, Genetic Testing, Genome Sequencing}

For instance, “Newborn Genetic Testing Thread Forum” is a potential search phrase candidate.

We used the Google search engine to search the Internet. The total of 1200 URLs from search results were investigated manually for related user comments in Summer 2013. The top 11 blogs and 18 forums were chosen as relevant URLs. The list of blogs and forums can be found in Appendix D.

After finding related online forums and blogs, we automatically downloaded user comments from the above-mentioned forums and blogs. We used the Selenium Web Driver[13], and the Appache Nutch[14] web crawler. Selenium Web Driver is used to download user comments from blog HTML web pages, whereas Appache Nutch is used to crawl the online forums for related user comments.

For online forums, the time of crawling depends on the set of parameters in Apache Nutch namely, topN and depth. topN is the number of outgoing links from a web page and depth is the number of times a web page should be crawled in order to fetch more documents from the web page. The higher the topN and depth parameters are, the more time it takes to crawl a forum. We set topN to 1000 outgoing links and depth to 10 rounds. A filter is also set up to prevent Nutch to crawl outgoing links with just advertisement and image content or links that point to pages with irrelevant data.
The resulted database from crawling online forums using Apache Nutch was queried using Apache Solr[15] search platform.

We collected a total of 1008 comments from the top 29 online forums and blogs.

4.2 Manual Annotation of Data

In classification (supervised learning, chapter 2), annotating training data is an important step. We performed annotation at two levels, namely, comment-level and EDU-level. Comment-level and EDU-level annotations are used to prepare training data for sentiment analysis. We employed two annotators and provided them with a separate annotation guidelines for each of the two comment-level and EDU-level annotations. The guidelines are reported in Appendix A and discussed below briefly.

NGS requires a blood test to be done shortly after birth to test for treatable diseases that are not usually apparent in the newborn period. Early detection of these diseases is the key to effective treatment and can prevent serious health problems and even save lives.[16]

---

[13] Selenium is a tool that automates browsers. It mimics the behavior of a real user and such interacts with the html page. Retrieved from http://www.seleniumhq.org

[14] Appache Nutch is a well-automated web crawler. Retrieved from http://nutch.apache.org

[15] Solr is a search platform from Apache Lucene project that could be used for high volume text-search. Retrieved from http://lucene.apache.org/solr/

[16] New Born Screening Ontario (2014). Retrieved in October 22, 2014 from Available: http://www.newbornscreening.on.ca

NGS doesn't include entire genome sequencing. It just includes testing for 29 diseases that can be found at http://www.newbornscreening.on.ca. A heel prick (a small sample of blood taken from baby's heel) is the common method to collect required blood sample for NGS. It allows health care providers to collect a small amount of blood on special filter paper. This blood is sent to Newborn Screening Ontario (NSO) where it is tested.

In the guidelines, we emphasized that annotators should ignore any pre-natal NGS done before the birth of an infant since this project focuses on the neonatal NGS. In addition NGS tests not done through the heel prick test should be ignored too since the heel prick is mentioned as standard test procedure by New Born Screening Ontario.

Pre-natal genome sequencing tests do not fall under NGS. The following is a list of such tests collected from our dataset:

- Amniocentesis: amniocentesis is a prenatal test that allows your healthcare practitioner to gather information about your baby's health from a sample of your amniotic fluid. This is the fluid that surrounds your baby in the uterus. [17]

- First Trimester Testing: first trimester screen is a new, optional non-invasive evaluation that combines a maternal blood-screening test with an ultrasound evaluation of the fetus to identify risk for specific chromosomal abnormalities, including Down Syndrome Trisomy-21 and Trisomy-18. [18]

- Prenatal Screening: prenatal screening is routinely offered for Down syndrome, and trisomy 18-risk assessment. [19]

We did not analyse opinions related to the three above-mentioned tests.

4.2.1 Comment-level Annotation

The guidelines for comment-level annotation provide annotators with enough information to distinguish between (Appendix A):

- *Relevant* and *irrelevant* comments
- *Positive*, *negative*, and *mixed* class labels

Comment annotation was done in two stages. First, comments were rated into *relevant* or *irrelevant* ones. Then *relevant* comments were labeled as *positive*, *negative*, or *mixed* ones.

[17] Amniocentesis (2014). Retrieved in October 22, 2014 from http://www.babycenter.com/0_amniocentesis_327.bc

[18] First Trimester Screening (2014). Retrieved in October 22, 2014 from http://americanpregnancy.org/prenataltesting/firstscreen.html

[19] Parental Screening Ontario (2014). Retrieved in October 22, 2014 from http://www.prenatalscreeningontario.ca

A comment is *relevant* if it is explicitly or implicitly expresses opinion or reveals a fact about newborn screening or genome sequencing or other aspects that are related to it. We asked the annotators to tag the topics that were discussed by forum users. The following is the list of major relevant topics manually collected by annotators:

- Privacy Concerns
  - Retention /storage of DNA data in database
  - Heel prick cards
  - Exploitation of data by government or insurance companies
  - Parental consent for use of DNA data
  - Research purposes (public/private)
  - Government involvement
  - Anonymity of DNA data for research purposes
  - DNA Privacy Laws, Mandatory testing laws (e.g. Michigan)

- Ethical Concerns
  - Sharing test results with children when older
  - If the test should be voluntary/compulsory

- Emotional Concerns
  - Hurting kids by taking blood samples – usually parents are not willing to consent to screening for their second infant after watching their first infant crying because of heel prick test

After identifying *relevant* comments, annotators labeled the *relevant* comments as *positive*, *negative*, or *mixed*. A comment is positive if it carries an overall positive sentiment. A positive comment may include positive and negative or mixed sentences. A negative comment is the one that carries an overall negative sentiment. A negative comment may include positive, negative, or mixed sentences.
If the annotators were not clear about the overall sentiment of a comment they annotated the comment as mixed one. The comment-level annotation workflow is illustrated in Figure 9.

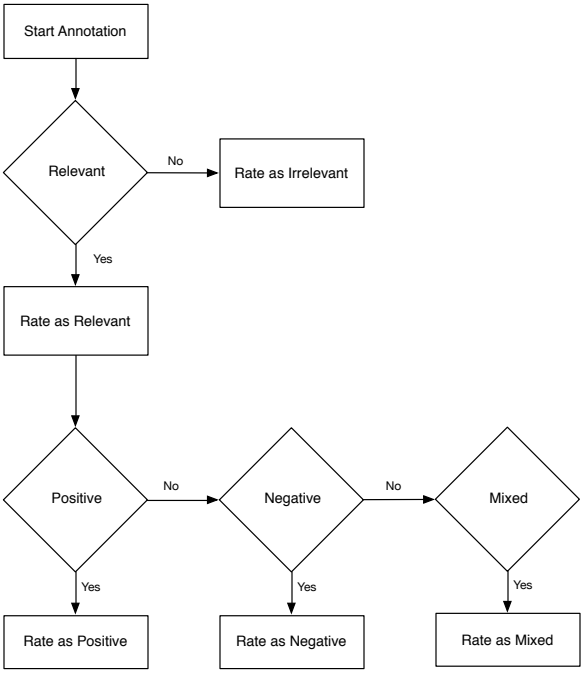

Figure 9 - Comment-Level Annotation Flowchart

Examples of *irrelevant*, *positive*, *negative*, and *mixed* comments from our dataset are provided as follows:

- Irrelevant

  What I find most interesting is that big polar and schizophrenia run in my family. Also I wonder if this could be a factor of severe drug use in my family that affected my son's genes. I fight the autism label because I do know the signs and my son doesn't show them... but since behavioural issues are now added to the spectrum doesn't that make everyone a little autistic...

- Positive

  This was the big test we were holding our breath for, so once it came back healthy we're much more comfortable. I'm just glad it wasn't super-invasive and still gave us the information we wanted!

- Negative

  OK so you have no problem with the government being able to positively identify your kid along with all the other info they are databasing and cross-referencing? And you have no problem with this government owning their genetic code, potentially knowing illness, disabilities, strengths, weaknesses and potential? You really have no problem with that? A trusting soul you are indeed.

- Mixed

  This could obviously be helpful for known genetic conditions limited to very specific gene regions/small amounts of genes, but most genetic characteristics involve so many interactions between genes that this technique seems premature to talk about.

We calculated the kappa statistics to measure the agreement between the two annotators. The overall kappa score was 0.61, which shows a fair agreement. Furthermore, we calculate kappa for each class label (section 3.4) as follows:

	Irrelevant	Positive	Negative	Mixed
Kappa Score	0.588	0.5483	0.504	0.578

Table 4 - Fleiss kappa Score

4.2.2 Annotation of Elementary Discourse Units

We annotated elementary discourse units (EDUs), to achieve three goals:

- Add an extra feature that captures the sentiment of the last EDU for each comment.
- Enhance the existing comment-level classifier by removing *irrelevant* EDUs from *relevant* comments.
- Extract EDUs that contain topic related terms.

Annotators were provided with guidelines to distinguish between:

- *Relevant/irrelevant* EDUs
- *Subjective/objective* EDUs
- *Positive/negative/neutral* EDUs
- EDUs that support "Newborn genome screening" and EDUs, which are against it.

The EDU-level annotation workflow is illustrated in figure 10. In order to annotate EDUs, all comments are decomposed into the consisting EDUs using the Hilda Segmenter Software Package (Feng et al., 2012).

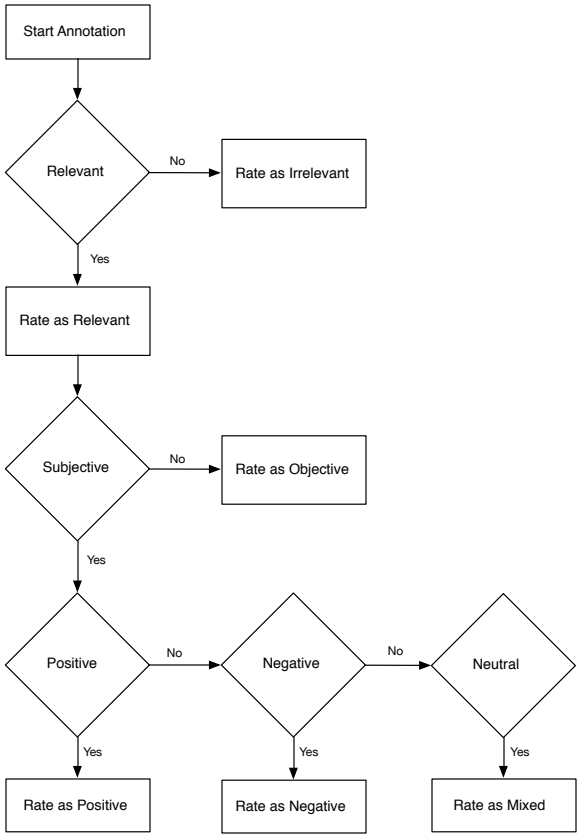

Figure 10 - EDU-Level Annotation Flowchart

Relevant/Irrelevant EDUs

An EDU is defined as *irrelevant*, if its parent sentence is *irrelevant*. A sentence is *relevant* if it is explicitly or implicitly related to an opinion or fact about newborn screening or genome sequencing or other aspects that are related to it. If the writer of a comment expresses subjectivity of someone other than himself/herself, the sentence/EDU is considered as *irrelevant*. While annotators rated each EDU, they took context from the surrounding EDUs into consideration and annotated each EDU relative to the context (i.e., *comment*). We show some examples in Table 5.

Comment	*I'm on the same boat here. Those tests screen for things that are manageable if known, but can be devastating if discovered too late.*
Sentences	*1. I'm on the same boat here.* *2. Those tests screen for things that are manageable if known* *3. But can be devastating if discovered too late.*
Relevant EDUs	*1. Those tests screen for things*
Irrelevant EDUs	*1. I'm on the same boat here.* *2. That are manageable if known* *3. But can be devastating if discovered too late*

Table 5 - Relevant/Irrelevant EDUs

Subjective/Objective EDUs

Subjectivity annotation was done for *relevant* EDUs only. If the primary intention of a sentence was objective presentation of material, the sentence and all its EDUs were considered as objective. A subjective sentence could be formed from objective and/or subjective EDUs. Table 6 shows an example of a subjective comment including objective and subjective EDUs.

Comment	*I had cancer as a child, and while they say the type I had is not genetic, I am terrified that one of my children are going to develop it.*
Sentences	1. *I had cancer as a child* 2. *They say the type I had is not genetic* 3. *I am terrified that one of my children are going to develop it*
EDUs	1. *I had cancer as a child* 2. *They say* 3. *The type I had is not genetic* 4. *I am terrified* 5. *One of my children are going to develop it*
Objective EDUs	1. *I had cancer as a child* 2. *They say* 3. *The type I had is not genetic* 4. *One of my children are going to develop it*
Subjective EDUs	1. *I am terrified*

Table 6 - Objective/Subjective EDUs

Positive/Negative/Neutral EDUs

In our study, according to the annotation guidelines, a positive EDU is a subjective EDU that has positive sentiment. Alternatively, a negative EDU is a subjective EDU carrying negative sentiment. Furthermore, a neutral EDU is a subjective EDU with equal weight of positive and negative sentiment. An EDU can be annotated as neutral if an annotator is indecisive about its positivity/negativity.

Table 7 shows an example of a mixed comment that contains positive, negative, and neutral EDUs as well as objective and *irrelevant* EDUs.

Comment	*I don't remember DD crying about her newborn blood screening, she barely noticed. I think they took it out of her hand too, which has got to be worse. Definitely doing the blood test again, though we'll have to go into the paediatrician to do it since we're having a home birth. The disorders it can catch are just so devastating and catching them right away can prevent death and disability. Not sure what we have to do to get the hearing test done, but we'll find a way to do that too.*
Sentences	1. *I don't remember DD crying about her newborn blood screening, she barely noticed.* 2. *I think they took it out of her hand too, which has got to be worse.* 3. *Definitely doing the blood test again, though we'll have to go into the paediatrician to do it since we're having a home birth.* 4. *The disorders it can catch are just so devastating and catching them right away can prevent death and disability.* 5. *Not sure what we have to do to get the hearing test done, but we'll find a way to do that too.*
EDUs	1. *I do not remember DD crying about her newborn blood screening* 2. *She barely noticed.* 3. *I think* 4. *They took it out of her hand too,* 5. *Which has got to be worse.* 6. *Definitely, Doing the blood test again,* 7. *Though we'll have to go into the paediatrician to do it* 8. *Since we're having a home birth.* 9. *The disorders it can catch are just so devastating* 10. *And catching them right away can prevent death and disability.* 11. *Not sure what we have to do* 12. *To get the hearing test done,* 13. *But we'll find a way* 14. *To do that too.*
Irrelevant EDUs	1. *She barely noticed.* 2. *I think* 3. *To get the hearing test done,* 4. *Since we're having a home birth.* 5. *But we'll find a way* 6. *To do that too.* 7. *Not sure what we have to do* 8. *They took it out of her hand too* 9. *Which has got to be worse.*
Objective EDUs	1. *Though we'll have to go into the paediatrician to do it* 2. *Since we're having a home birth.*
Positive EDUs	1. *And catching them right away can prevent death and disability.*
Negative EDUs	10. *Definitely, Doing the blood test again,* 11. *The disorders it can catch are just so devastating*
Neutral EDUs	1. *I do not remember DD crying about her newborn blood screening*

Table 7 - Example of Comments With Various EDUs

4.3 Data Analysis

After data collection and annotation (explained in chapter 4, and 5 respectively), we calculated some basic statistics about the dataset. A total of 1008 comments were collected from 29 online forums and blogs. The average length of each comment is 124 words. Table 8 shows some basic comment-level statistics.

The Number of Comments	1008
The Number of Discussion Titles	55
Avg. Length of a Comment	124 words
Avg. Length of a Discussion Title	7 words

Table 8 - Comment-Level Statistics

Figure 11 shows the distribution of comment-level class labels after annotation. Majority of comments were annotated as *irrelevant*. *Irrelevant* comments were either *irrelevant* to the subject or the authors were quoting other people's opinion. *Positive*, *negative* and *mixed* comments were imbalanced classes compared to *irrelevant* class.

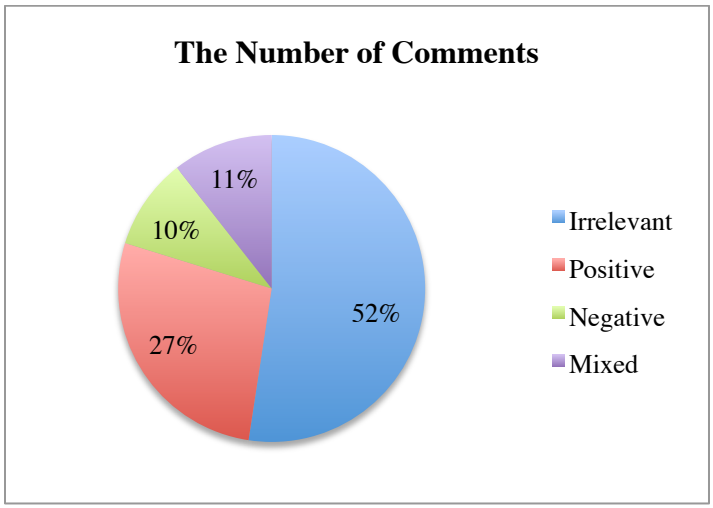

Figure 11 - Distribution of Comment-Level Class Labels

We segmented each comment into EDUs using the Hilda Discourse Parser (section 6.2). Figure 12 shows EDU distribution in class labels. The total number of EDUs was 16214 of which 14091 EDUs were annotated as *irrelevant*, 880 EDUs were annotated as *objective*, 534 EDUs were annotated as *positive*, 312 EDUs were annotated as *negative*, 367 EDUs were annotated as *neutral*, and 32 EDUs were annotated as *mixed*. Since the number of mixed EDUs was very low we merged the mixed EDUs with neutral EDUs and removed the mixed. After merging mixed EDUs with neutral EDUs, the total

number of neutral EDUs became 399. Figure 12 shows the probability distribution of EDUs with *irrelevant*, positive, negative, and neutral classes in the dataset.

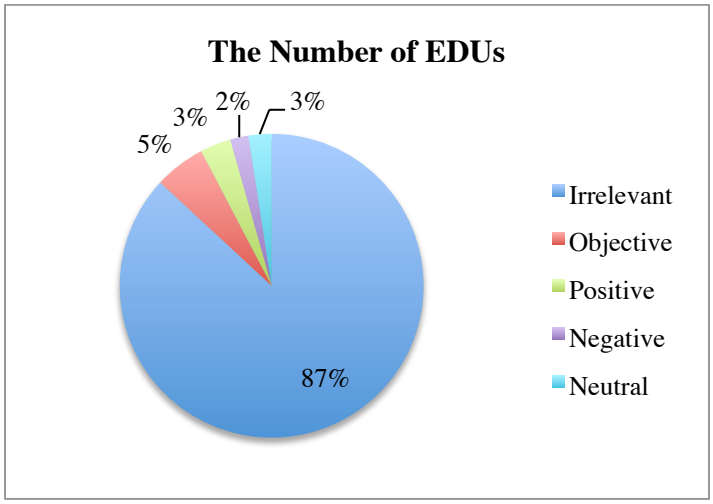

Figure 12 - Distribution of EDU-Level Class Labels

1008 comments were segmented into 16214 EDUs in total. The total of 13061 discourse relations from 18 discourse relation categories are realized. Table 9 shows these relations as well as their counts.

Relation	Count	Relation	Count
Cause	78	Temporal	136
Manner-means	77	Background	405
Contrast	831	Condition	477
Evaluation	611	Same-unit	1091
Topic-change	2	Elaboration	5144
Attribution	2433	Textual-organization	105
Summary	120	Explanation	466
Joint	1309	Comparison	19
Enablement	361	Topic-comment	16

Table 9 - Discourse Relations Counts

In a discourse tree, relations are internal nodes. We calculated the probability distribution of relations, which were direct parents of EDUs (external nodes) as follows:

$$p(relation_i) = \frac{\#\ EDUs\ for\ relation_i}{total\ \#\ EDUs}$$

Table 10 shows the probability distribution calculated using the above formula:

Relation	Probability distribution in EDUs	Relation	Probability distribution in EDUs
Cause	0.00348	Temporal	0.00986
Manner-means	0.00215	Background	0.02534
Contrast	0.04060	Condition	0.03515
Evaluation	0.00986	Same-unit	0.07135
Topic-change	0	Elaboration	0.25978
Attribution	0.190358	Textual-organization	0.00319
Summary	0.00556	Explanation	0.019779
Joint	0.08292	Comparison	0.001873
Enablement	0.03234	Topic-comment	0.000165

Table 10 - Probability Distribution of Discourse Relations

In future work (section 7.2), we will introduce the idea of a rule-based classifier that combines the sentiment of all EDUs in a comment discourse tree to produce a global sentiment for that comment. Since this rule-based classifier uses each relation as an operator to combine sentiments of children of that relation (EDUs or other relations), we were interested in the count of each relation considering class-label of two EDUs that were connected through this relation. For instance consider the following block of text:

With each pregnancy, I have considered the tests that I was offered and almost always came away with the feeling that the results would just lead to more questions and worry and MORE invasive tests.

The above-mentioned block of text is consisted of five EDUs that are connected through discourse relations. Figure 13 shows the EDUs and relations connecting the EDUs in the discourse tree structure:

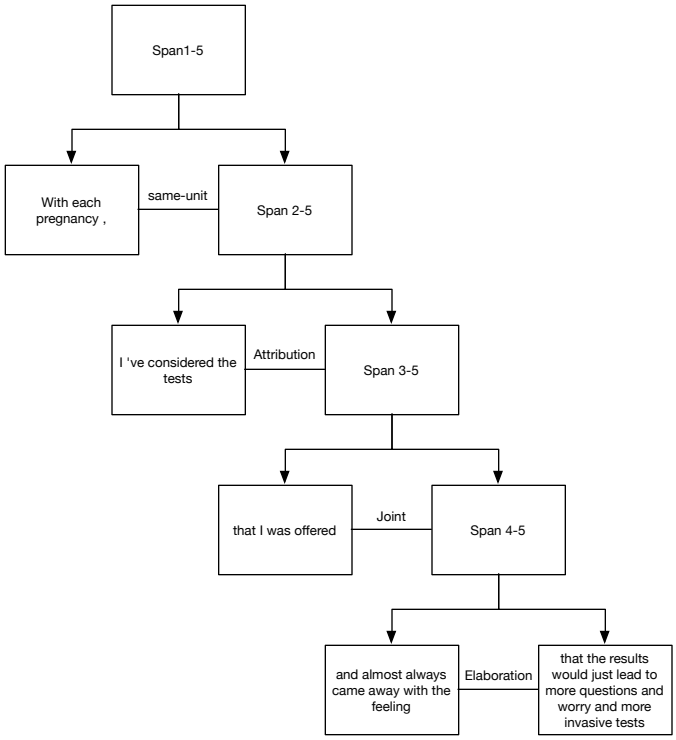

Figure 13 - Discourse Tree

Tables 11 to14 show such statistics for Attribution, Elaboration, Condition, and Contrast. We were particularity interested in the Contrast relation, since this relation produced a negative sentiment if any of its children had a negative sentiment. In tables 11 to 14, empty table cells correspond to the fact that some relations with specific setting do not exist in our dataset.

Table 12 - Counts of Elaboration Relation

Relation	Left EDU	Right EDU	Count
Elaboration	Irrelevant	Irrelevant	1184
		Positive	17
		Negative	16
		Mixed	3
		Neutral	19
		Objective	38
	Positive	Irrelevant	29
		Positive	12
		Negative	
		Mixed	
		Neutral	1
		Objective	10
	Negative	Irrelevant	13
		Positive	1
		Negative	8
		Mixed	
		Neutral	
		Objective	2
	Mixed	Irrelevant	16
		Positive	
		Negative	6
		Mixed	
		Neutral	11
		Objective	4
	Neutral	Irrelevant	
		Positive	4
		Negative	
		Mixed	
		Neutral	
		Objective	
	Objective	Irrelevant	48
		Positive	14
		Negative	3
		Mixed	
		Neutral	5
		Objective	57

Table 11 - Counts of Enablement Relation

Relation	Left EDU	Right EDU	Count
Attribution	Irrelevant	Irrelevant	971
		Positive	32
		Negative	12
		Mixed	2
		Neutral	21
		Objective	41
	Positive	Irrelevant	4
		Positive	10
		Negative	
		Mixed	
		Neutral	
		Objective	1
	Negative	Irrelevant	2
		Positive	
		Negative	2
		Mixed	
		Neutral	
		Objective	
	Mixed	Irrelevant	
		Positive	
		Negative	
		Mixed	
		Neutral	
		Objective	
	Neutral	Irrelevant	7
		Positive	1
		Negative	4
		Mixed	
		Neutral	1
		Objective	
	Objective	Irrelevant	5
		Positive	1
		Negative	1
		Mixed	
		Neutral	
		Objective	6

Table 13 - Counts of Condition Relation

Relation	Left EDU	Right EDU	Count
Condition	Irrelevant	Irrelevant	160
		Positive	5
		Negative	4
		Mixed	1
		Neutral	4
		Objective	2
	Positive	Irrelevant	2
		Positive	5
		Negative	
		Mixed	
		Neutral	
		Objective	1
	Negative	Irrelevant	3
		Positive	
		Negative	1
		Mixed	
		Neutral	
		Objective	1
	Mixed	Irrelevant	1
		Positive	
		Negative	
		Mixed	
		Neutral	
		Objective	
	Neutral	Irrelevant	
		Positive	
		Negative	2
		Mixed	
		Neutral	3
		Objective	1
	Objective	Irrelevant	2
		Positive	
		Negative	1
		Mixed	
		Neutral	
		Objective	2

Table 14 - Counts of Contrast Relation

Relation	Left EDU	Right EDU	Count
Contrast	Irrelevant	Irrelevant	106
		Positive	7
		Negative	1
		Mixed	
		Neutral	1
		Objective	5
	Positive	Irrelevant	2
		Positive	1
		Negative	1
		Mixed	1
		Neutral	
		Objective	2
	Negative	Irrelevant	
		Positive	2
		Negative	2
		Mixed	
		Neutral	
		Objective	
	Mixed	Irrelevant	
		Positive	
		Negative	
		Mixed	
		Neutral	
		Objective	
	Neutral	Irrelevant	
		Positive	1
		Negative	1
		Mixed	
		Neutral	1
		Objective	2
	Objective	Irrelevant	1
		Positive	4
		Negative	3
		Mixed	
		Neutral	
		Objective	1

Chapter 5 Methods and Experimental Setup

In this chapter, we discuss our experiments, the reasons why these experiments were chosen and how they were executed. The main purpose of this thesis was to build a sentiment analysis system that can work well with text harvested from NGS forums and blogs. Furthermore, we developed an argumentation system that could automatically extract the reasons that parents expressed to support or disagree with "Newborn Genome Sequencing".

We collected data from a wide range of forums and blogs since the discussion title was rare and it was hard to come by a few forums containing thousands of user comments in this subject.

To start, we provide a brief road map to the experiments done in this thesis:

1. Classified comments as follows:
    - Developed a simple classifier (one-level) comments into *irrelevant/positive/negative/mixed*
    - Developed a more sophisticated classifier (two-levels) to first classify comments into *relevant/irrelevant* ones and then classify the *relevant* comment into *positive/negative/mixed* ones.

2. Compared the results of one-level and two-level classifiers to see whether a more sophisticated classification method actually improved the result.

3. Reduced the dimensionality of each comment by reducing it to *relevant* EDUs and classified comments represented by these EDUs to see if this would reduce the classifier's noise introduced by the presence of the *irrelevant* EDUs in comments.

4. Analysed argumentation in comments by extracting EDUs containing phrases related to the topics (manually collected topics are listed in section 4.2.1). Two EDUs were also extracted before and after the topic-related EDU in order to provide context.

5.1 Classification

As a supervised learning method, to perform classification, we first annotated data (chapter 5), then prepared bag-of-words representation of each comment or EDU. As pre-processing steps to prepare bag-of-words, we removed:

- Stop words
- Non English words
- URLs
- Punctuation except for question mark (?) and exclamation mark (!)

Furthermore, misspelled English words were corrected automatically by using LanguageTool[20] software package. More details about pre-processing can be found in section 3.2. The WEKA (Hall et al., 2009) software package was used as the tool that provided various classification algorithms' implementation.

For both comment and EDU classification, we experimented with bag-of-words, without any extra features, and then added extra features. In this way, we proved that feature engineering (section 5.2) improved the accuracy of the classifiers.

The WEKA software package provided two implementations of support vector machine (SVM) classification algorithms (section 3.3.2). We used SMO[21]implementation of SVM for our experiments. In addition, NB[22] was used as the implementation of Naïve Byes classification algorithm (section 3.3.1). WEKA also provided the facility to create a more complex classifier by combining different classifiers and different combinations of probability estimates for classification. We used the VOTE[23] Meta classifiers (with majority voting rule) to combine the NB, SMO, and LibSVM classifiers to see whether this combination would improve the accuracy of classification.

We wanted to compare SVM, NB, and VOTE classifier results with a baseline. We used WEKA ZeroR[24] classifier as the baseline classifier. It classified all comments into the majority class. In our experiment, the *irrelevant* class had the highest probability distribution and therefore was the majority class.

Finally in order to see how accurately classification algorithms were performed, we used 10-fold cross validation as our training and testing method, because we did not have enough data to keep aside a test set.

5.2 Extra features

Adding extra features such as the number of positive or negative words was shown to improve the accuracy of classification (Wilson et al., 2005; Pak et al., 2010; Kouloumpis et al., 2011; Bobicev et al., 2012; Mohammad et al., 2013). We used 18 extra features in comment classification and 17 extra features in EDU classification. The following is the list of common extra features used for this thesis:

[20] Retrieved 2015-01-01, from https://languagetool.org
[21] Retrieved 2015-01-01, from http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html
[22] Retrieved 2015-01-01, from http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayes.html
[23] Retrieved 2015-01-01, from http://weka.sourceforge.net/doc.dev/weka/classifiers/meta/Vote.html
[24] Retrieved 2015-01-01, from http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/ZeroR.html

- The number of positive words from
    - General Inquirer
    - SentiWordNet
    - Subjectivity Lexicon
    - Depechmode
    - NRC Emotion Lexicon
- The number of negative words from
    - General Inquirer
    - Subjectivity Lexicon
    - SentiWordNet
    - Depechmode
    - NRC Emotion Lexicon
- The number of elongated words
- The number of occurrences of exclamation mark (!) and question mark (?)
- The number of domain specific words
- The number of weak subjective words
- The number of strong subjective words
- The number of negative emoticons
- The number of positive emoticons
- The sentiment of the last EDU in a comment

In General Inquirer (Stone et al., 1966) lexicon, a list of positive and a list of negative words are provided. We used the list of positive and the list of negative words to count the number of positive and negatives words in the corpus.

In SentiWordNet 3.0 (Esuli et al., 2006) each synset (a set of synonyms) is associated with a positive and negative score as illustrated in Table 15.

POS	***A***
ID	*00001740*
PosScore	*0.125*
NegScore	*0*
SynsetTerms	*able#1*
Gloss	*(Usually followed by 'to') having the necessary means or skill or know-how or authority to do something; "able to swim"; "she was able to program her computer"; "we were at last able to buy a car"; "able to get a grant for the project"*

Table 15 - Example of a SentiWordnet Entry

In order to extract the polarity of a word, we calculated the weighted average of the set of scores associated to that word, using the inverse of the word sense rank as its weight. The score is the difference between the positive score and the negative score related to a synset (table 15). If the weighted average is greater than zero the word is positive, otherwise it is negative. Figure 14 shows how the polarity of a word is calculated.

For each SentiWordNet ***enrty***
For each ***synset term***
Compute ***score*** *as* ***PosScore – NegScore***
Create a pair as ***<sense rank, score>***
End
End

For each ***synset term***
For each ***<sense rank, score>*** *pair associated to the* ***synset term***
Compute the ***weight*** *as* $\frac{1}{(sense\ rank)}$
Compute the ***weighted score*** *as* $weight \times score$
End

Compute the ***overall weighted score*** *by adding all the weighted scores calculated in above loop*

Compute the ***overall weight*** *by adding all the weights calculated in the above loop*
Compute the ***weighted average*** *as* $\frac{the\ overall\ weighted\ score}{the\ overall\ weight}$
End

Figure 14 - Pseudo Code Used in SentiWordNet Polarity Detection

Each synset in SentiWordNet has two components, namely the synset and an index (sense rank) associated to it (Table 15). For instance, for able#1, able is the synset term and 1 is the sense rank (the first sense).

In the Subjectivity Lexicon (Wilson et al, 2005), each term is associated with a polarity as shown in table 16. In order to capture the polarity of a word, we simply looked up the value of prior polarity property related to that word. Furthermore, to count the number of weak subjective and strong subjective words, we looked up those values for all the words in a comment or EDU.

Type	*weaksubj*
Length	*1*
Word	*abandoned*
POS	*adj*
Stemmed	*n*
Prior polarity	*negative*

Table 16 - Example of a Subjectivity Lexicon Entry

In Depechemode (Staiano et al., 2014), each entry consists of a set of 8 emotion scores (*Afraid, Amused, Angry, Annoyed, Don't-Care, Happy, Inspired, Sad*). To calculate the polarity of a term, we divided the 8 emotions into to sentiments, namely positive, and negative, as follows:

- Positive: Amused, Happy, Inspired,
- Negative: Afraid, Angry, Sad, Annoyed

Further, we summed up the emotion scores related to positive as the positive sore. Similarly, we summed up the emotion scores for negative as the negative score. If the positive score was greater than the negative score, then the related term was positive, otherwise negative. Table 17 shows a Depechemode entry consisting of the 8 emotions scores associated and a <lemma, part of speach> pair.

Lemma#POS	abuse#n
Afraid	0.093817569
Amused	0.086525206
Angry	0.221250556
Annoyed	0.136038429
Don't-Care	0.112141622
Happy	0.117322572
Inspired	0.103567459
Sad	0.129336587

Table 17 - Example of a Depechemode Entry

In NRC emotion lexicon (Mohammad & Turney, 2013), each word is associated to 8 emotions (*anger, fear, anticipation, trust, surprise, sadness, joy, and disgust*) and two sentiments (*positive, and negative*). Table 18 shows an entry in NRC emotion lexicon.

Abolition	***Anger***	***0***
Abolition	*Fear*	*0*
Abolition	*Anticipation*	*0*
Abolition	*Trust*	*0*
Abolition	*Surprise*	*0*
Abolition	*Sadness*	*0*
Abolition	*Joy*	*0*
Abolition	*Disgust*	*0*
Abolition	*Positive*	*0*
Abolition	*Negative*	*1*

Table 18 - Example of a NRC Emotion Lexicon Entry

To calculate the polarity of a term, we simply looked up the positive and negative score for that term. If the positive score was 1 and the negative score was 0, then the term was positive, otherwise negative.

To generate the list of domain specific words, we used bootstrapping as follows:

1. Removed stop-words from the corpus

2. Assumed an initial list of ten words that were manually selected as *relevant* to the domain (newborn genome sequencing):
    - *Newborn, infant, DNA, genome, genetic, sequencing, testing, screening*
3. Generated 5-grams from the whole corpus
4. Selected 5-grams which contained any of the words from the initial list
5. Ranked all terms produced in the previous step by the means of TF-IDF and chose the first ten terms and added to the initial list
6. Repeated from step 4
7. Stopped after 35 iterations

Using the above-mentioned bootstrapping algorithm, the total of 2432 domain-specific words harvested. After each iteration, we manually checked the output of the bootstrapping algorithm (Figure 15) to confirm the presence of domain specific words. After 35 iterations, we stopped the execution of the algorithm, since there was not major change in the top 2000 words in the algorithms output. Figure 15 shows the bootstrapping process used to generate domain specific terms.

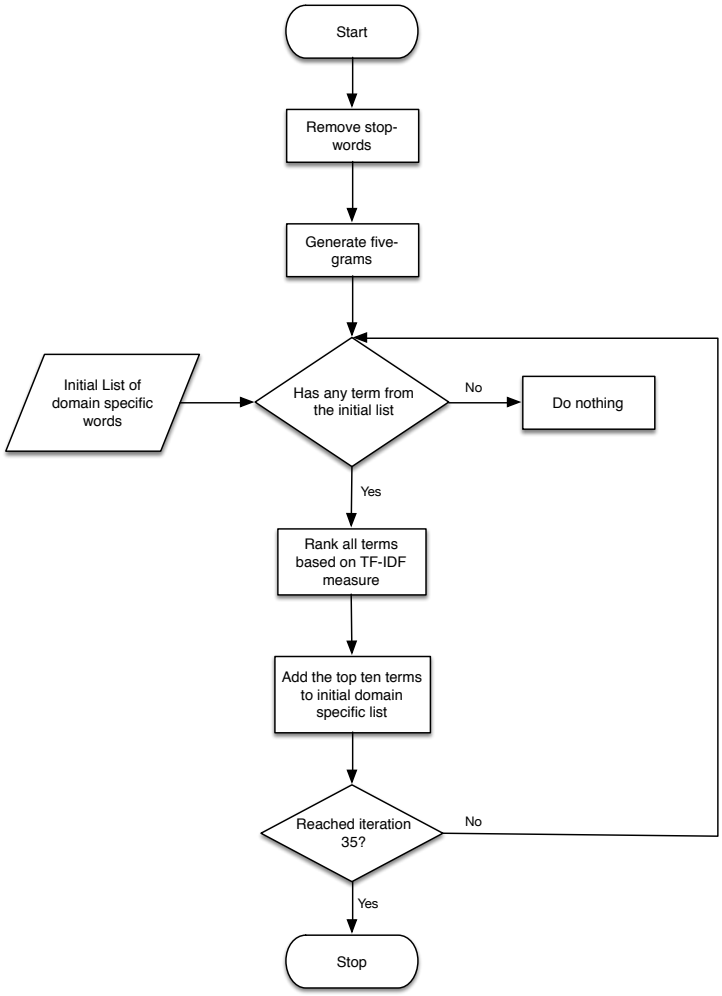

Figure 15 - The Workflow to Generate a Lexicon of Domain Specific Words

The following is the top 10 domains-specific words generated based on bootstrapping algorithm demonstrated in figure 15:

- {'test', 'newborn', 'screening', 'testing', 'genetic', 'tests', 'thanks', 'born', 'heel', 'disease'}

5.3 Classification of Comments

Our goal for comment-level classification was to produce an accurate classifier that classified each comment into *positive, negative, mixed,* or *irrelevant* classes. To classify comments, we create two classifiers:

1. A single classifier (called flat classifier) that classified each comment into *positive, negative, mixed,* or *irrelevant* class.

2. A composite classifier (called two-level classifier, figure 16) consisting of two classifiers:

    a. First classifier classified comments into *relevant* or *irrelevant* classes

    b. Second classifier used the *relevant* comments predicted by the first classifier and classified them into *positive, negative,* or *mixed* classes.

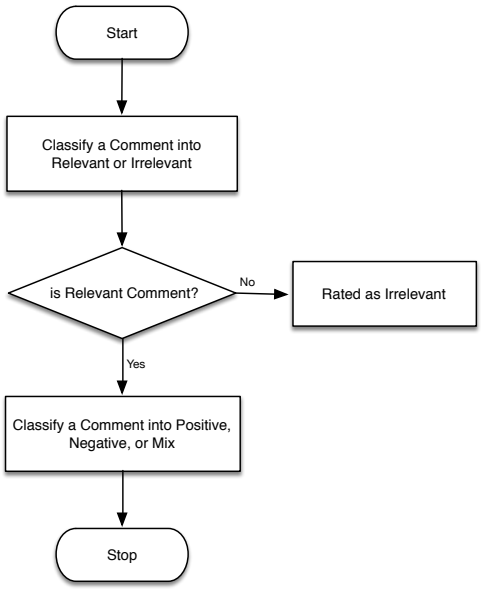

Figure 16 - Two-Level Comment Classifier

3. A composite Classifier (called enhanced two-level classifier, figure 17) consisting of three classifiers:

    a. A classifier to classify comments into *relevant* or *irrelevant* ones.

    b. A classifier to classify EDUs into *relevant* or *irrelevant* ones and reduce *relevant* each comment into a comment consisting of only *relevant* EDUs.

    c. A classifier to classify *relevant* comments into *positive*, *negative*, or *mixed* ones.

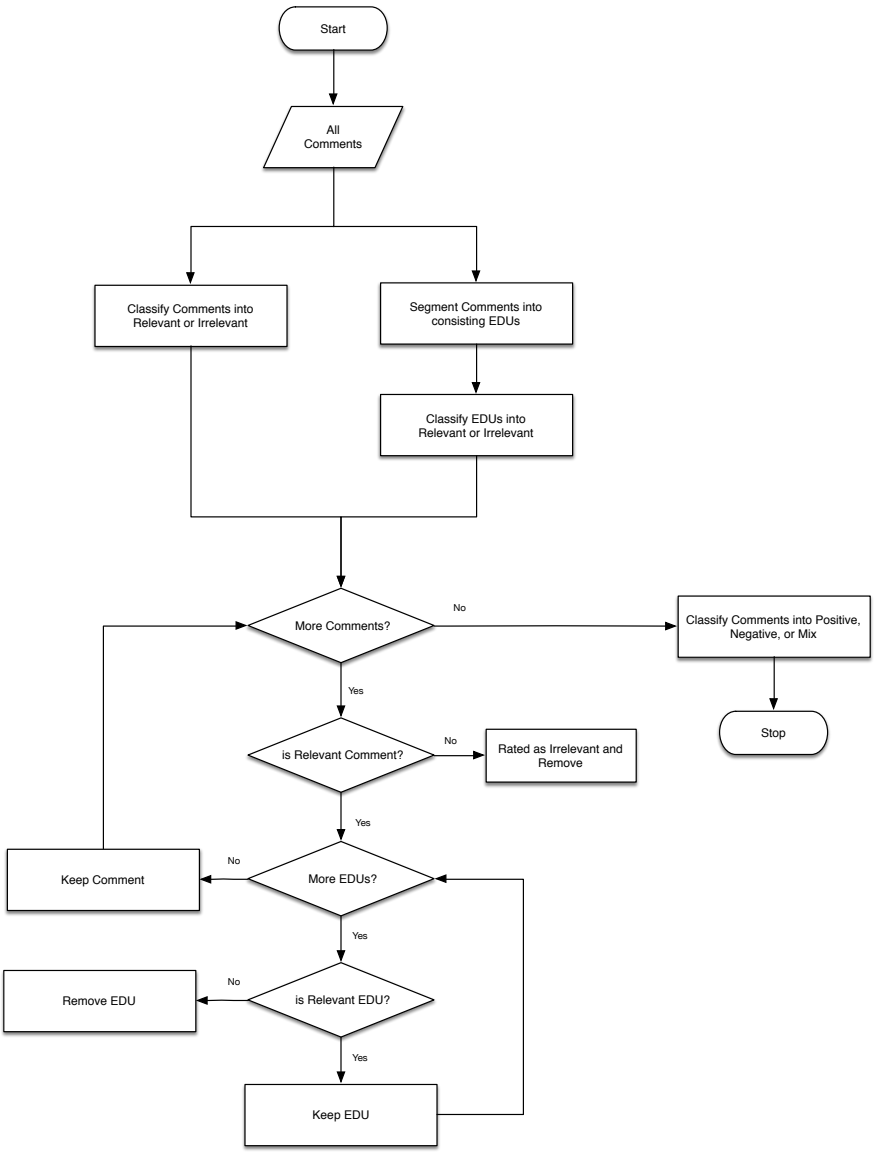

Figure 17 - Enhanced Two-Level Comment Classifier

The SMO, NB, and VOTE classifiers form WEKA (Hall et al., 2009) software package used for the comment-level classification. Furthermore, the WEKA ZeroR classifier was

used as the baseline classifier. The baseline classifier classified all comments into the majority class, which, in comment-level classification, was *irrelevant* class, as mentioned.

The last-EDU-sentiment extra feature was added to the 17 common extra features (section 5.2) for comment-level classification. To generate the last-EDU-sentiment extra feature, all comments were segmented into constituent EDUs, using the Hilda software package (Feng et al., 2012). First, all EDUs were classified into *positive, negative, neutral,* or *irrelevant* (section 5.4), and the predicted class of the last EDU for each comment was picked as the value of this feature for that comment. We experimented with representing comments first by words, which appeared in all the EDUs of each comment, and then by words, which appeared only in the *relevant* EDUs of the comment.

5.4 Classification of EDUs

Our goals for EDU-classification were, first, to predict the sentiment of the last EDU for each comment in order to create an extra feature that captured the sentiment of the last EDU (last-EDU-sentiment feature). Next, we were interested in enhancing the two-level comment classifier (section 6.2.3) by removing *irrelevant* EDUs from *relevant* comments.

We created two classifiers:

- A single classifier that classified each EDU into *positive, negative, neutral,* or *irrelevant* one.

- A composite classifier (called two-level classifier, figure 18) that consisted of two classifiers:
    - The first classifier that classified each EDU into *relevant* or *irrelevant* ones.
    - The second classifier that classified the *relevant* comments (predicted by the first classifier) into *positive, negative,* or *neutral* ones.

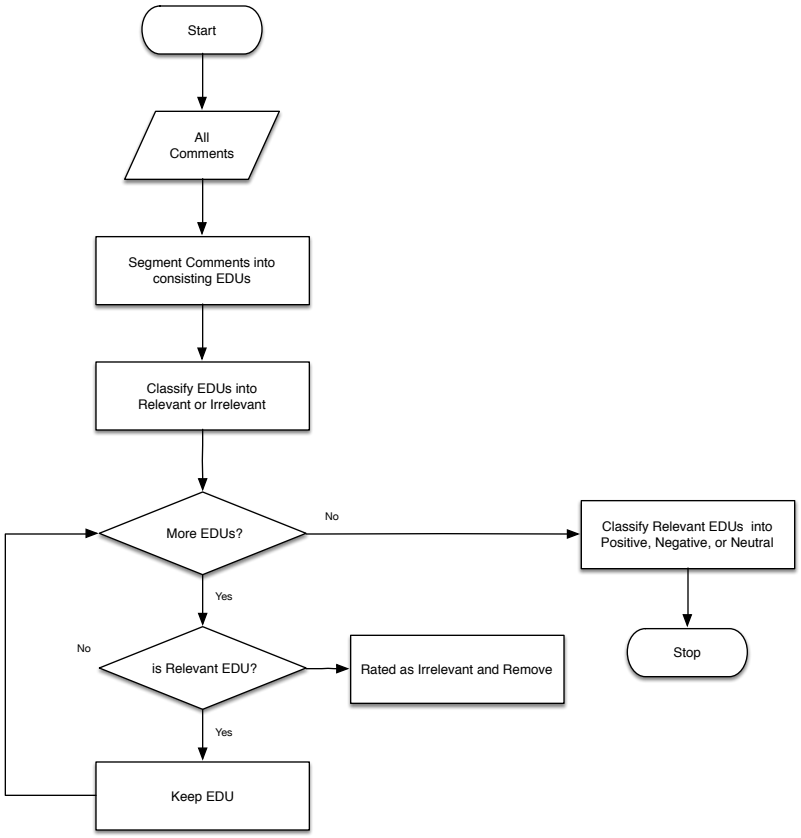

Figure 18 - Two-Level EDU Classifier

We used the VOTE Meta classifier, form WEKA software package, for EDU classification. First, all the comments were segmented into consisting EDUs using Hilda Software Package. SpreadSubsample[25] filter from WEKA software package was used to remove extra *irrelevant* EDUs due to the unbalanced nature of data (chapter 6).

The VOTE Meta classifier was made of two classifiers Naïve Bayes[26], and SMO[27] classifiers. The decision to assign a class to an EDU was made based on the mean of probability distribution learned from the two before-mentioned classifiers.

5.5 Analysis of Argumentation

To extract the reasons that forum users expressed to support or to reject "newborn genome sequencing (NGS)" (Also referred to as "Newborn Genome Screening", NGS in earlier chapters), we studied argumentation as follows: First, two annotators tagged EDUs that contained topic-related content. Then, the annotators rated topic-related EDUs that supported NGS as *pro* and annotated the EDUs that rejected NGS as *con*. All

[25]http://weka.sourceforge.net/doc.stable/weka/filters/supervised/instance/SpreadSubsample.html
[26] http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayes.html
[27] http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html

domain-related EDUs as well as two EDUs before and after those EDUs were extracted. In order to extract the most *relevant* EDUs, we created bigrams, trigrams, four-grams, and five-grams from the extracted EDUs and ranked them based on their frequency in the whole corpus.

It should be noted that single terms (i.e., *unigrams*) would not provide useful information about parents reasoning towards NGS. The most natural choice for us was bigrams; however, bigrams still did not provide much insight into the reasons online forum users rejected or supported NGS. Therefore, the extraction was performed with trigrams, four-grams, and five-grams. We found that five-grams provided the most meaningful information compared to other n-gram. Therefore, we decided to represent the result of our argumentation in the form of five-grams. Furthermore, the top five five-grams were cross-referenced with original user comments to extract the original sentences that contained those five-grams. We report the empirical results in the next chapter.

Chapter 6 Experiments and Empirical Results

In this chapter, we present the experiments we conducted and their results. We will further discuss the results and the improvements gained.

As we mentioned, several experiments were conducted to create a comment classifier that could classify comments into *positive, negative, mixed,* or *irrelevant* ones with high accuracy. In addition, a basic argumentation analysis was conducted to extract the reasons that forum users mentioned in their comments to support or reject newborn genome screening.

6.1 Comment Classification

In this section, all results regarding our attempts to classify comments into *positive, negative, mixed,* or *irrelevant* ones are presented. All the experiments in this section were done for bag-of-words (BOW) and bag-of-words with extra features (BOW-EF). As mentioned in the previous chapter, the experiments performed for comment classification were:

- Created a classifier (called flat classifier) to classify comments into *positive, negative, mixed,* or *irrelevant*
- Created a two-level classifier to classify comments into *relevant* or *irrelevant* first and then classify the *relevant* comments into *positive, negative,* or *mixed* ones.

To report the detailed accuracy (accuracy by class) of classifications, four performance measures were used. The performance measures were Accuracy, Precision, Recall, and F-Measure. We also reported the weighted average of Accuracy, Precision, Recall, and F-measure. The weighted average of a performance measure is the average of the performance measure for each predicted class weighted by the probability distribution of that class. 10-fold cross validation (section 3.3.11) was used to evaluate classifiers. We chose 10-fold cross validation since our dataset was limited to 1008 comments, and there was not enough data available for separate test set and training set.

6.1.1 Flat Comment Classification

We started with the baseline (ZeroR) comment classifier. The results are shown in table 19. ZeroR classifier classified all comments into irrelevant class. The accuracy of the ZeroR classifier was 52.40 %. We classified comments using SMO, NB, and VOTE classifiers from WEKA software package. No extra features were used at this stage. The

comparison of classifier accuracies can be seen in table 19. The detailed accuracy for SMO, NB, and VOTE can be found in Appendix B (B.1.1).

	Accuracy	Precision	Recall	F-Measure
ZeroR	52.40%	0.275	0.524	0.36
SMO	61.92%	0.611	0.619	0.615
NB	59.04%	0.608	0.59	0.595
VOTE	63.96%	0.597	0.64	0.598

Table 19 - Comparison of Flat Comment Classifiers (BOW)

In order to compare the three classifiers (SMO, NB, and VOTE) and their improvement over baseline (ZeroR), we performed statistical tests of significance (t-test, section 3.3.12) with p-value equals to 0.05. The following is the interpretation of the result:

- SMO, NB, and VOTE, outperformed the baseline classifier (ZeroR) using just bag-of-words (BOW) without extra features (table 19).
- NB and SMO were not statistically different. However, the VOTE classifier was statistically different than NB, and SMO.

We performed classification again by adding 18 extra features (section 5.2) to the bag-of-words model. Table 20 shows the comparison of classifiers. The detailed accuracy for SMO, NB, and VOTE can be found in Appendix B (B.1.2).

	Accuracy	Precision	Recall	F-Measure
ZeroR	52.40%	0.275	0.524	0.360
SMO	63.64%	0.585	0.636	0.603
NB	57.65%	0.604	0.576	0.586
VOTE	65.14%	0.597	0.651	0.600

Table 20 - Comparison of Flat Comment Classifiers (BOW-EF)

We explored statistical tests of significance (t-test) for SMO, NB, and Vote using bag-of-words and extra features (BOW-EF) and p-value equals to 0.05. The following is the interpretation of results:

- SMO, NB, and VOTE outperformed the baseline classifier (ZeroR) using just bag-of-words with extra features, BOW-EF (Table 20).
- NB and SMO were not statistically different. However, the VOTE classifier was statistically different than NB, and SMO.

We showed that the VOTE classifier using bag-of-words with extra features (BOW-EF) achieved an improvement of 13% over baseline classifier (ZeroR) and the difference was statistically significant. We used information gain (section 3.3.13) to show the contribution of each extra feature to the improvement in accuracy of flat comment

classification. Table 21 shows the information gain for extra features in comment-level flat classification. The BOW features are removed from the table:

Info Gain	Rank	Feature
0.06731	5	numPositiveNRC
0.06200	6	numNegativePL
0.06168	7	numPositivePL
0.06056	8	numNegativeNRC
0.05403	9	numPositiveDepmode
0.05323	10	numDSLWords
0.05076	11	numWeakSubjPL
0.04848	12	numPositiveGI
0.04811	13	numNegativeGI
0.04491	14	numStrongSubjPL
0.04779	15	numNegativeDepmode
0.03376	29	numPunctuations
0.03199	30	numNegativeSWN
0.03105	33	numPositiveSWN
0.01172	142	lastEduClassLabel
0	1990	numElongatedWords

Table 21 - Information Gain Calculated for Extra Features in Flat Comment Classification

The rank column in table 21 shows the importance of each extra feature based on the information gain towards class labels. There were total of 2934 features in the BOW-EF. For instance, numPositiveNRC was the 5th important feature out of 2934 features. Furthermore, we compared the accuracies of flat comment classification using BOW and BOW-EF. According to t-test (p-value equals to 0.05), SMO and VOTE improved on accuracy by adding extra features; however, NB performed the worst, when adding the extra features (Table 22).

Flat Comment Classifier	Accuracy (BOW)	Accuracy (BOW-EF)
ZeroR (Baseline)	52.40%	52.40%
SMO	61.92%	63.64%
NB	59.04%	57.65%
VOTE	63.96%	65.14%

Table 22 - Comparison of Flat Comment Classifiers (BOW and BOW-EF)

Comparing each of SMO, NB, and VOTE with baseline (ZeroR) in terms of accuracy, t-test (p-value equals to 0.05) showed that the difference between SMO, NB, VOTE, and baseline (ZeroR) was statistically significant. Furthermore, Adding Extra features to BOW improved the accuracies of SMO, and VOTE classifiers. However, the accuracy of NB using BOW-EF was worse than that of the NB using BOW.

In the next section, we experimented with two-level comment-level classification.

6.1.2 Two-Level Comment Classification

From data analysis (section 4.3), it can be seen that most comments were *irrelevant*; however *irrelevant* comments consisted of sentences that carried sentiments. We hypothesized that a two-level comment classifier that first, classified comments into *relevant* or *irrelevant* and then, classified *relevant* comments into *positive*, *negative*, or *mixed*, would perform better than the flat comment classifier in terms of accuracy. To prove or reject the hypothesis, we developed a two-level comment classifier. The first classifier classified comments into *relevant* or *irrelevant* and the second classifier classified the *relevant* comments into *positive, negative,* or *mixed* (Figure 17).

To measure the performance of the two-level comment classification, two two-level comment classifiers were created. First, we created a best-performing two-level comment classifier, named gold standard two-level comment classifier, by manually removing the irrelevant comments and classifying the *relevant* comment into *positive, negative,* or *mixed*. Therefore, the gold standard two-level comment classifier consisted of just one classifier.

Next, we created a two-level comment classifier that classified comments into *relevant* and *irrelevant* and then classified *relevant* comments into *positive, negative,* or *mixed*.

Gold Standard Two-Level Comment Classification Experiments

The gold standard two-level comment classifier was created to measure how well the two-level comment classifier could perform compared to it. We experimented with SMO, NB, and VOTE classifiers for the gold standard two-level comment classifier. Table 23 shows the results of the classifier from the gold standard two-level classification using bag-of-words (BOW) without any extra features. The detailed accuracy for SMO, NB, and VOTE can be found in Appendix B (B.1.3).

	Accuracy	Precision	Recall	F-Measure
ZeroR	57.5%	0.329	0.575	0.420
SMO	55.16 %	0.533	0.552	0.541
NB	56.28%	0.565	0.563	0.563
VOTE	61.21%	0.560	0.612	0.556

Table 23 - Comparison of the Classifiers from the Gold Standard Two-Level Comment Classification (BOW)

We performed a pair-wise comparison of the NB, SMO, and VOTE classifiers with the baseline classifier (ZeroR). The paired t-test (p-value equals to 0.05) showed that the difference between NB, SMO, VOTE and ZeroR gold standard classifiers was not statistically significant.

In the gold standard two-level classification, only *relevant* comment were classified into positive, negative, and mixed and all *irrelevant* comments were removed manually. Therefore, the gold standard two-level classification can also be seen as a two-level classifier that the first classifier classified all comments into *relevant* or *irrelevant* with perfect accuracy of 100%, and the second classifier classified the relevant comments into *positive, negative,* or *mixed*.

We calculated the Accuracy of the gold standard two-level comment classifier considering the confusion matrix of the VOTE, SMO, and NB classifiers. Table 24 shows the confusion matrix for the gold standard two-level comment classifier based on the VOTE classifier.

	Classified as Positive	Classified as Negative	Classified as Mixed	Classified as Irrelevant
Positive	234	10	12	0
Negative	61	14	16	0
Mixed	62	12	25	0
Irrelevant	0	0	0	488

Table 24 - The Gold Standard Two-Level Comment Classifier Confusion Matrix (VOTE)

Table 25 shows the accuracy of gold standard two-level comment classifiers.

	Accuracy
ZeroR	73.34%
SMO	78.58%
NB	79.12%
VOTE	81.47%

Table 25 - The Gold Standard Two-Level Comment Classification Accuracies (BOW)

We added extra features (section 5.2) to bag-of-words (BOW) and repeated two-level comment classification. This experiment was done to investigate the effect of extra features on the gold standard classification results. Table 26 shows the classification results after adding extra features. The detailed accuracy for SMO, NB, and VOTE can be found in Appendix B (B.1.4).

	Accuracy	Precision	Recall	F-Measure
ZeroR	57.5%	0.329	0.575	0.420
SMO	55.38%	0.531	0.554	0.540
NB	57.39%	0.544	0.544	0.536
VOTE	61.43%	0.560	0.614	0.557

Table 26 - The Comparison of the Classifiers from the Gold Standard Two-Level Comment Classification (BOW-EF)

We performed a pair-wise comparison of the NB, SMO, and VOTE classifiers with the baseline classifier (ZeroR). The paired t-test (p-value equals to 0.05) showed that the difference between NB, SMO, and ZeroR gold standard classifiers was not statistically significant. However, the difference (in terms of accuracy) between VOTE classifier and baseline (ZeroR) classifier was statistically significant.

We calculated the Accuracies of the gold standard two-level comment classifiers considering the confusion matrix of the VOTE, SMO, and NB second classifiers. Table 27 shows the confusion matrix for the gold standard two-level comment classifier based on the VOTE second classifier.

	Classified as Positive	**Classified as Negative**	**Classified as Mixed**	**Classified as Irrelevant**
Positive	235	10	12	0
Negative	63	12	16	0
Mixed	60	12	27	0
Irrelevant	0	0	0	488

Table 27 - The Gold Standard Two-Level Comment Classifier Confusion Matrix (VOTE)

Using the confusion matrix, we calculated the Precision, Recall, and F-measure for gold standard two-level comment classifiers (SMO, NB, and VOTE). The detailed performance by class for each classifier is shown in appendix B.

Table 28 shows the performance of gold standard two-level comment classification using BOW-EF. The detailed accuracy by class can be found in Appendix B (B.1.5).

Gold Standard Two-level Classifiers	**Accuracy**	**Precision**	**Recall**	**F-measure**
SMO	78.7%	0.775	0.786	0.780
NB	79.65%	0.793	0.796	0.794
VOTE	81.58%	0.789	0.786	0.785

Table 28 - The Gold Standard Two-Level Comment Classification Accuracies (BOW-EF)

In conclusion, in gold standard two-level comment classification we removed the *irrelevant* comment manually and classified the *relevant* comment into *positive, negative*, and *mixed*. Using BOW, the SMO, NB, and VOTE classifiers was not statistically significant from the baseline classifier (ZeroR). However, using BOW-EF, the VOTE classifier out-performed the baseline classifier (ZeroR).

Gold Standard Two-Level Classifiers	Accuracy (BOW)	Accuracy (BOW-EF)
ZeroR	73.34%	73.34%
SMO	78.58%	78.70%
NB	79.12%	79.65%
VOTE	81.47%	81.58%

Table 29 - Comparison of Gold Standard Two-Level Comment Classification Accuracies (BOW and BOW-EF)

The paired t-test (p-value equals to 0.05) showed that the difference between gold standard two-level comment classifiers and two-level comment classifiers were not statistically significant. Therefore adding extra features to BOW did not significantly improve the accuracy of the gold standard two-level comment classification.

Two-Level Comment Classification Experiments

We created a two-level classifier comments in two steps. In the first step, comments were classified into *relevant* or *irrelevant*. In the second step, the *relevant* comments, predicted by the first classifier, were classified into *positive, negative,* or *mixed*. We performed two-level comment classification using bag-of words (BOW) as well as bag-of words with extra features (BOW-EF) to investigate the effect of extra features on two-level comment classification. Table 30 shows the results of the first classifier using BOW. The performance by class can be found in Appendix B (B.1.6)

	Accuracy	Precision	Recall	F-Measure
ZeroR	52.24%	0.273	0.522	0.359
SMO	77.30%	0.773	0.773	0.773
NB	76.23%	0.767	0.762	0.762
VOTE	77.4 %	0.762	0.762	0.842

Table 30 - The Comparison of the First Classifier from the Two-Level Comment Classification (BOW)

We performed a pair-wise comparison of the VOTE, SMO, and NB with baseline classifier (ZeroR). The difference between the VOTE, SMO, NB, and ZeroR was statistically significant (t-test with p-value to equals 0.05).

Our goal to create a two-level comment classifier was to see whether we could gain improvement over flat comment classification by classifying only *relevant* comments into *positive, negative*, and *mixed*. Therefore, the role of the first classifier was to remove as much *irrelevant* comment as possible while keeping *relevant* comments. We removed the misclassified instances from the prediction of the first classifier before feeding it to the second classifier as the training dataset. Therefore, the classifier of the choice would be the one with higher average weighted f-measure. The VOTE classifier was chosen as the classifier of the choice as the first classifier. Table 31 shows the confusion matrices for the NB, SMO, and VOTE classifiers.

The First Classifier	Confusion Matrix		
NB	**Classified as Relevant**	**Classified as Irrelevant**	
	364	82	**Relevant**
	136	352	**Irrelevant**
SMO	**Classified as Relevant**	**Classified as Irrelevant**	
	327	119	**Relevant**
	93	395	**Irrelevant**
VOTE	**Classified as Relevant**	**Classified as Irrelevant**	
	301	145	**Relevant**
	66	422	**Irrelevant**

Table 31 - The Comparison of the Confusion Matrices of the First Classifiers in Two-Level Comment Classification (BOW)

We manually removed the misclassified comments from the first classifier prediction and fed the filtered predication from the first classifier to the second classifier. Table 32 shows the results of the second classifiers from two-level comment classification using BOW. The difference between accuracy of these classifiers was not statistically significant. The performance by result can be found in Appendix B (B.1.7)

	Accuracy	Precision	Recall	F-Measure
SMO	60.55%	0.575	0.606	0.586
NB	63.05%	0.606	0.631	0.614
VOTE	63.05%	0.571	0.631	0.576

Table 32 - The Comparison of the Second Classifiers from the Two-Level Comment Classification (BOW)

Table 33 shows the accuracy of two-level comment classification using BOW. In table 33, first classifier is the VOTE classifier for all three two-level classifiers.

Two-Level Classifier	Accuracy
First VOTE, Second SMO	68.52%
First VOTE, Second NB	69.48%
First VOTE, Second VOTE	69.48%

Table 33 - Two-Level Comment Classification Accuracies (BOW)

The two-level comment classification was repeated with bag-of-words with extra features (BOW-EF). Table 34 shows the result of the first classifier for two-level comment classification using bag-of-words with extra features (BOW-EF). The classification performance by class can be found in Appendix B (B.1.8)

	Accuracy	Precision	Recall	F-Measure
ZeroR	52.24%	0.273	0.552	0.359
SMO	78.16%	0.782	0.782	0.781
NB	72.69%	0.750	0.727	0.723
VOTE	78.26%	0.786	0.783	0.782

Table 34 - The Comparison of the First Classifiers from the Two-Level Comment Classification (BOW-EF)

We used information gain (section 3.3.13) to show the information gain of each extra feature towards the classification with relevant/irrelevant class labels. We used 18 extra features in the first classifier of the two-level comment classification. Table 35 shows the information gain for each extra feature in the first level classification of the two-level comment classification (The BOW features are removed for table 35):

InfoGain	Rank	Feature
0.06	6	numNegativePL
0.061	7	numNegativeNRC
0.059	8	numPositiveNRC
0.059	9	numPositiveGI
0.057	10	numPositivePL
0.052	11	numWeakSubjPL
0.051	12	numDSLWords
0.047	13	numNegativeGI
0.046	14	numNegativeDepmode
0.045	16	numStrongSubjPL
0.043	17	numPositiveDepmode
0.037	25	numPunctuations
0.034	27	numPositiveSWN
0.03	30	numNegativeSWN
0.007	859	numElongatedWords
0.005	1874	numNegativeEMOTICON
0	3726	numPositiveEMOTICON
0	9059	lastEduClassLabel

Table 35 - The Information Gain Calculated for Relevant /Irrelevant Comment-Level Classification

The Rank column shows the importance of the feature towards relevant/irrelevant comment classification. For instance, numNegativePL was the 5th important feature out of the total of 2937 features (BOW-EF).

The role of the first classifier was to remove as many *irrelevant* comments as possible while keeping *relevant* comments. We removed the misclassified instances from the prediction of the first classifier before feeding it to the second classifier as the training dataset. Therefore, the classifier of the choice would be the one with higher weighted average f-measure. The VOTE was chosen as the classifier of the choice as the first classifier.

Table 36 shows the confusion matrix for VOTE, NB, and SMO using BOW-EF.

The First Classifier	Confusion Matrix		
NB	**Classified as Relevant**	**Classified as Irrelevant**	
	385	61	**Relevant**
	194	294	**Irrelevant**
SMO	**Classified as Relevant**	**Classified as Irrelevant**	
	332	114	**Relevant**
	91	397	**Irrelevant**
VOTE	**Classified as Relevant**	**Classified as Irrelevant**	
	315	131	**Relevant**
	72	416	**Irrelevant**

Table 36 - The Comparison of the Confusion Matrices of the First Classifiers in Two-Level Comment Classification (BOW)

The two-level comment classification was repeated for the second classifier using bag-of-words with extra features. Table 37 show the comparison of SMO, NB, and VOTE classifiers. Furthermore. The performance by class can be fo9und in Appendix B (B.1.9)

	Accuracy	Precision	Recall	F-Measure
SMO	59.68%	0.560	0.597	0.572
NB	59.95%	0.586	0.599	0.591
VOTE	62.59%	0.575	0.626	0.580

Table 37 - The Comparison of the Second Classifiers from the Two-Level Comment Classification (BOW-EF)

According to t-test (p-value equals to 0.05), the difference between accuracy of NB, SMO, and VOTE classifiers was not statistically significant.

Table 38 shows the accuracy of the two-level comment classification using bag-of-words with extra features (BOW-EF).

Two-Level Classifier	Accuracy
First VOTE, Second SMO	68.63%
First VOTE, Second NB	68.74%
First VOTE, Second VOTE	69.81%

Table 38 - Two-Level Comment Classification Performance (BOW-EF)

Table 39 shows the comparison between two-level comment classifiers using BOW and BOW-EF. The two-level comment classifiers used the VOTE as the first classifier. The accuracy did not improve by adding extra features.

Two-Level Classifier	Accuracy Gold Standard (BOW-EF)	Accuracy (BOW)	Accuracy (BOW-EF)
First Classifier	N/A	**VOTE**	**VOTE**
Second Classifier SMO	77.7%	68.52%	68.63%
Second Classifier NB	79.65%	69.48%	68.74%
Second Classifier VOTE	81.58%	69.48%	69.81%

Table 39 - Two-Level Comment Classification Accuracies (BOW and BOW-EF)

According to t-test with p-value equals to 0.05, the difference between two-level comment classifier using SMO, NB or VOTE, as the second classifier with BOW and BOW-EF was not statistically significant.

The best accuracy achieved for flat comment classification was 65.14% (Table 20), whereas the best accuracy achieved for two-level comment classification was 69.81% (Table 38). Furthermore, the paired t-test (p-value equals to 0.05) showed that the difference between the two comment classifiers was statistically significant.

In conclusion, removing *irrelevant* comments improved the accuracy of the comment classification. Comparing the results presented in tables 20 and 38, we accepted the hypothesis that removing *irrelevant* comments could improve the accuracy of the comment classification.

In the next section we will provide detailed results for EDU-level classification and how these results were used in an attempt to improve the two-level comment classification results.

6.2 EDU Classification

EDU classification was conducted for two reasons:

1. To improve the existing two-level comment-level classifier (section 6.1.2) by removing *irrelevant* EDUs from each *relevant* comment.

2. To create the last-EDU-sentiment extra feature for comment-level classification that would carry the sentiment of the last EDU of each comment.

To perform EDU classification each comment was segmented into consisting EDUs using Hilda[28] discourse parser software package. Then two annotators annotated all EDUs to *relevant* or *irrelevant* classes. The details for EDU annotation can be found in section 4.2.2. The kappa statistics (indicator of agreement between annotators, section 3.4) was 0.67.

[28] Retrieved 2015-01-01, from http://www.cs.toronto.edu/~weifeng/software.html

The same set of features that were used for comment classification (section 5.2) was used for EDU-level classification except for the last-EDU-sentiment feature.

There were total of 16218 EDUs. The total of 14091 EDUs belonged to the *irrelevant* class and 2125 EDUs were in the *relevant* class. From 2125 *relevant* EDUs, 880 EDUs were objective and 1245 EDUs were subjective. The *relevant* subjective EDUs belonged to the *positive, negative, mixed, or neutral* classes (Figure 12). Although annotators manually annotated EDUs into *irrelevant*, *positive*, *negative*, *mixed* or *neutral*, Class labels used for EDU classification were *irrelevant*, *positive, negative,* or *neutral*.
We decided to merge the EDUs that were labeled as mixed with the EDUs, which were labeled as neutral, because almost 0.01% of EDUs were labelled as mixed (32 EDUs out of 16218 EDUs, section 4.3).

To take the imbalance nature of data into account, under-sampling of the *irrelevant* class was performed before EDU classification. Class distribution ratio of 2:1 was maintained by removing some EDUs with *irrelevant* class. This ratio allowed us to keep the number of *irrelevant* EDUs at most twice the total number of *positive, negative, mixed,* and *neutral* EDUs. After under-sampling, the count of EDUs was as follows: 312 EDUs from the negative class, 399 EDUs from the neutral class, 534 EDUs from the positive class, and 2490 EDUs from the *Irrelevant* class (Figure 19).

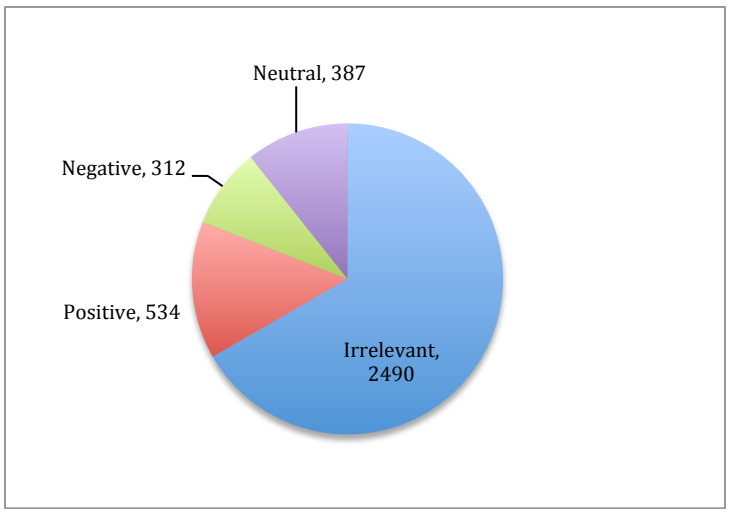

Figure 19 - Count of EDUs after Under-Sampling

Two methods of classification were explored in EDU classification, namely flat and two-level classification. In flat classification, we developed one classifier that classified each EDU into *positive, negative, neutral,* or *irrelevant* one. Furthermore, we experimented with a two-level classifier that classified each EDU into *relevant* or *irrelevant* and then classified the *relevant* EDUs (predicted by the first classifier) into *positive, negative,* or *neutral* ones.

To evaluate the results of classification, 10-fold cross validation was used on the training dataset. In the next section we will discuss flat EDU classification in details.

6.2.1 Flat EDU Classification

In this section, we experimented we the flat EDU classification, and developed three flat EDU classifiers using SMO, NB, and VOTE classifiers form WEKA software package. The classifiers classified EDUs into *irrelevant, positive, negative,* and *neutral* ones. The EDU dataset used in classification was highly imbalanced. Almost 87% of the EDUs were *irrelevant* (Figure 12). We used under-sampling to reduce the number of *irrelevant* data (Figure 19). Although the classification models were built on the under-sampled data, the 10-fold cross validation was done on the original training data.

The baseline classifier (ZeroR) had a high accuracy of 92.87%. However, this classifier classified all EDUs into *irrelevant* class. It was important that the EDU-level classifier performed reasonably well for all classes. We chose the SMO classifier because it had the highest accuracy after baseline and performed well in terms of F-measure (compared to NB and VOTE) in the classification of all classes (Table 40). The performance by class can be found in Appendix B (B.2.1)

	Accuracy	Precision	Recall	F-Measure
ZeroR	92.87%	0.863	0.929	0.894
SMO	82.30%	0.917	0.823	0.862
NB	70.40%	0.905	0.704	0.782
VOTE	81.84%	0.914	0.818	0.858

Table 40 - Comparison of Flat EDU-Level Classifiers (BOW-EF)

With respect to F-measure performance metric, the SMO classifier performed better then the VOTE and NB in classification of *negative* and *neutral* EDUs. The VOTE classifier however performed better in the classification of *positive* EDUs. The NB was the worst of all three.

We used information gain (section 3.3.13) to show the contribution of each extra feature to the improvement in accuracy of flat EDU classification. Table 41 shows the information gain for extra features in EDU-level flat classification (The BOW features are removed from table 41):

InfoGain	Rank	Feature
0.020275	2	numDSLWords
0.019919	3	numNegativeSWN
0.019349	4	numPositiveSWN
0.016857	7	numNegativeDepmode
0.01665	9	numNegativePL
0.014929	10	numStrongSubjPL
0.014719	11	numWeakSubjPL
0.014688	12	numPositiveDepmode
0.013122	13	numNegativeGI
0.012472	15	numNegativeNRC
0.011848	18	numPositiveGI
0.01184	19	numPositiveNRC
0.010523	20	numPositivePL
0.007637	24	numPunctuations
0	2165	numElongatedWords

Table 41 - Information Gain Calculated for Extra Features in Flat EDU Classification

The number of positive words and the number of negative words were amongst the most informative features amongst 17 extra features. The realization of number of positive words and the number of negative words as the most informative features was also intuitive since we positive and negative were two of the class labels.

Due to the low F-measure in negative, neural, and positive classes (Section B.2.1), we experimented with a two-level classifier to see whether we can improve F-measure for positive, negative, and neutral classes by classifying only relevant EDUs into positive, negative, or neutral ones. We will explain the two-level classification experiments in details in the next section.

6.2.2 Two-Level EDU Classification

From data analysis (section 4.3), it can be seen that 87% of the EDUs were *irrelevant*. We hypothesized that a two-level classifier that first, classified comments into *relevant* and *irrelevant* and then, classified *relevant* comments into *positive, negative,* or *mixed*, would perform better than the flat EDU classifier in terms of accuracy.

To prove or reject the hypothesis, we developed a two-level EDU classifier. The first classifier classified EDUs into *relevant* or *irrelevant* and the second classifier classified the *relevant* EDUs into *positive, negative,* or *neutral*.

To measure the performance of the two-level EDU classification, two two-level EDU classifiers were created. First, we created a best-performing two-level EDU classifier, named the gold standard two-level EDU classifier, by manually removing the *irrelevant*

EDUs and classifying the *relevant* EDUs into *positive, negative,* or *neutral*. Therefore, the gold standard two-level EDU classifier consisted of just one classifier.

Next, we created a two-level EDU classifier that classified EDUs into *relevant* and *irrelevant* and then classified *relevant* EDUs into *positive, negative,* or *neutral*. The gold standard two-level EDU classifier was created to measure how well the two-level EDU classifier would perform compared to it. We experimented with NB, SMO, and VOTE classifiers from WEKA software package.

Table 42 shows the result of the second classifier for gold standard two-level EDU classification. We referred to the only classifier in the gold standard two-level EDU classifier as the second classifier since it classified the *relevant* EDUs after the manual removal of the *irrelevant* EDUs. If F-measure was taken as the performance measure, the VOTE classifier performed best in classification of *positive*, and *neutral* EDUs. However, NB classifier performed best in classification of negative EDUs. The performance by class can be found in Appendix B (B.2.2)

	Accuracy	Precision	Recall	F-Measure
ZeroR	47.86%	0.229	0.479	0.310
SMO	60.72%	0.609	0.614	0.604
NB	52.36%	0.528	0.524	0.525
VOTE	59.11%	0.588	0.591	0.584

Table 42 - Comparison of the Second Classifiers from the Gold Standard Two-Level EDU Classification (BOW-EF)

The paired t-test with confidence level of 95% (p-value equals to 0.05) showed that the difference between NB, SMO, VOTE and ZeroR gold standard classifiers was statistically significant. The VOTE and SMO were better than NB (p-value equals to 0.05).

We calculated the Accuracy of the gold standard two-level EDU classifier considering the confusion matrix of the VOTE, SMO, and NB classifiers. Table 43 shows the confusion matrix for the gold standard two-level comment classifier based on the VOTE classifier.

	Classified as Positive	Classified as Negative	Classified as Neutral	Classified as Irrelevant
Positive	397	51	86	0
Negative	92	147	73	0
Mixed	161	46	192	0
Irrelevant	0	0	0	14091

Table 43 - The Gold Standard Two-Level EDU Classifier Confusion Matrix (VOTE)

Table 44 shows the accuracies of the gold standard two-level EDU classification using bag-of-words with extra features (BOW-EF). The performance by class can be found in Appendix B (B.2.3)

	Accuracy	Precision	Recall	F-Measure
SMO	96.81%	0.891	0.891	0.891
NB	96.13%	0.887	0.886	0.886
VOTE	96.68%	0.891	0.891	0.891

Table 44 - The Gold Standard Two-Level EDU Classification Accuracies (BOW-EF)

After experimenting with the gold standard two-level EDU classification, we created a two-level EDU classifier that classified EDUs into *irrelevant, positive, negative*, and *neural* in two steps. In the first step, EDUs were classified into *relevant* or *irrelevant*. In the second step, the *relevant* EDUs (predicted by the first classifier) were classified into *positive, negative,* or *neutral*. The *irrelevant* to *relevant* class label ratio in the EDU dataset was 10 to 1 (Figure 12). We sub-sampled the EDUs with irrelevant class and reduced the ratio of the *irrelevant* to *relevant* class label to 2:1.

The purpose of EDU classification was to improve on comment classification by removing *irrelevant* EDUs from *relevant* comments. If the classifier misclassified irrelevant EDUs it may not be important, since a comment would keep its *relevant* EDUs as well as a few *irrelevant* EDUs that are misclassified as *relevant*. In this case we still managed to reduce the size of comment and keep most of the *relevant* EDUs. However, if the classifier misclassified *relevant* EDUs, the comment would loose most of its *relevant* EDUs and gain few *irrelevant* EDUs. In this case, the comment is reduced to few EDUs with less *relevant* EDUs.

To choose the proper subsampling ratio we experimented with EDU classification with *irrelevant* to *relevant* class ratio of 2:1, 3:1, 4:1, 5:1, 6:1, and finally 10:1. Table 45 shows the confusion matrices for VOTE classifier, and it can be seen from the table that less *relevant* classes are misclassified in 2:1 class ratio.

Irrelevant to Relevant Ratio	Confusion Matrix		
2:1	**Classified as Relevant**	**Classified as Irrelevant**	
	788	457	**Relevant**
	1191	12900	**Irrelevant**
3:1	**Classified as Relevant**	**Classified as Irrelevant**	
	739	506	**Relevant**
	859	13232	**Irrelevant**
4:1	**Classified as Relevant**	**Classified as Irrelevant**	
	707	534	**Relevant**
	667	13424	**Irrelevant**
5:1	**Classified as Relevant**	**Classified as Irrelevant**	
	677	568	**Relevant**
	542	13549	**Irrelevant**
6:1	**Classified as Relevant**	**Classified as Irrelevant**	
	661	584	**Relevant**
	484	13607	**Irrelevant**
10:1	**Classified as Relevant**	**Classified as Irrelevant**	
	561	684	**Relevant**
	275	13816	**Irrelevant**

Table 45 - Confusion Matrices Comparison To Choose Sub-sampling Ratio

Although, the classifier model was built on the subsampled EDU dataset, the 10-fold cross validation was done on the original dataset.

We performed two-level EDU classification using bag-of words with extra features (BOW-EF). Table 46 shows the results of the first classifiers from two-level EDU classification using BOW-EF. The performance by class can be found in Appendix B (B.2.4)

	Accuracy	Precision	Recall	F-Measure
ZeroR	91.88%	0.842	0.918	0.878
SMO	89.25%	0.920	0.893	0.903
NB	81.02%	0.908	0.810	0.846
VOTE	90.18%	0.921	0.902	0.910

Table 46 - Comparison of the First Classifiers from the Two-Level EDU Classification (BOW-EF).

A series of the paired t-test with p-value equals to 0.05, showed that the pair-wise differences between NB, SMO, VOTE and ZeroR gold standard classifiers were

statistically significant. T-test also showed that the difference between SMO, VOTE, and NB was statistically significant.

The purpose of two-level EDU classification was to improve on comment classification by removing *irrelevant* EDUs from *relevant* comments. If the first classifier misclassified irrelevant EDUs it may not be important, since a comment would keep its *relevant* EDUs as well as a few *irrelevant* EDUs that are misclassified as *relevant*. However, if the first classifier misclassified *relevant* EDUs, the comment would loose most of its *relevant* EDUs and gain few *irrelevant* EDUs. Therefore, we ignored the baseline classifier regardless of the fact that it had the highest accuracy.

We chose the VOTE as the first classifier since it had the highest F-measure compared to SMO, NB (Table 47). The second classifier classified *relevant* EDUs, predicted by the first classifier, into *positive, negative,* or *neutral* ones. The performance by class can be found in Appendix B (B.2.5)

	Accuracy	Precision	Recall	F-Measure
ZeroR	41.27%	0.170	0.413	0.241
SMO	63.37%	0.631	0.634	0.629
NB	59.45%	0.590	0.594	0.585
VOTE	62.64%	0.630	0.626	0.618

Table 47 - Comparison of the Second Classifiers from the Two-Level EDU Classification (BOW-EF).

The paired t-test with p-value equals to 0.05, showed that the pair-wise differences between NB, SMO, VOTE and ZeroR classifiers were statistically significant. T-test also showed that the pair-wise differences between SMO, VOTE, and NB were not statistically significant.

Table 48 shows the accuracies of the two-level EDU classification using bag-of-words with extra features (BOW-EF).

	Accuracy
SMO	86.96%
NB	78.24%
VOTE	88.02%

Table 48 - The Two-Level EDU Classification Accuracy (BOW-EF)

Table 49 shows the comparison between two-level EDU classifiers using BOW and BOW-EF.

Two-Level Classifier	Accuracy Gold Standard (BOW-EF)	Accuracy (BOW-EF)
First Classifier	N/A	**SMO**
Second Classifier SMO	96.81%	86.96%
Second Classifier NB	96.13%	78.24%
Second Classifier VOTE	96.68%	88.02%

Table 49 - The Comparison of Accuracy for Two-Level EDU Classification (BOW-EF)

We performed a pair-wise comparison between gold standard two-level EDU classifiers and two-level EDU classifiers. According to t-test (test of significance) with p-value equals to 0.05, the difference between gold standard two-level and two-level EDU classifiers was statistically significant.

6.2.3 Using The Results of EDU Classification in Comment Classification

We used the result of EDU-level classification to investigate the hypothesis that removing *irrelevant* EDUs from a *relevant* comment would improve the performance (accuracy) of two-level comment classification. In order to test the above-mentioned hypothesis, a two-level comment classifier named enhanced two-level comment classifier, and consisted of three classifiers (Figure 17) was developed. In the first level, two classifiers were used in parallel. The first-level comment classifier classified comments into *relevant* or *irrelevant* ones. The first-level EDU classifier classified EDUs into *relevant* or *irrelevant* ones.

In the second level, *irrelevant* EDUs (predicted by the first-level EDU classifier) were removed from *relevant* comments (predicted by the first-level comment classifier). The reduced comments were called enhanced comments. Furthermore, the third classifier, named second-level comment classifier, classified enhanced comments into *positive, negative,* or *mixed* ones.

For the first classifier, we used the first classifier from two-level comment classification using BOW-EF (section 6.1.2). For the second classifier, we used NB, SMO, and VOTE from WEKA software package. Table 50 shows the result of the second classifier from the enhanced two-level comment classification. The performance by class can be found in the Appendix B (B.3.1)

	Accuracy	Precision	Recall	F-Measure
ZeroR	57.18%	0.327	0.572	0.416
SMO	62.31%	0.587	0.623	0.593
NB	53.33%	0.548	0.533	0.539
VOTE	62.32%	0.567	0.623	0.554

Table 50 - Comparison of the Second Classifiers from the Enhanced Two-Level Comment Classification (BOW-EF)

We performed a pair-wise comparison of the VOTE, SMO, and NB with baseline classifier (ZeroR). The difference between the VOTE, and NB was statistically significant (t-test with p-value equals to 0.05). However, SMO, NB, and VOTE were not significantly different from ZeroR (baseline).

We compared the second classifier from the two-level comment classification with that of enhanced two-level comment classification (Table 51).

Second Classifier	Two-Level Comment Accuracy	Enhanced Two-Level Comment Classifier Accuracy
SMO	59.68%	62.31%
NB	59.95%	53.33%
VOTE	62.59%	62.32%

Table 51 - The Comparison of Second Classifier from Two-Level Comment Classification and Enhanced Two-Level Comment Classification

According to t-test (test of significance) with p-value equals to 0.05, the difference between the second classifier (VOTE, SMO, or NB) from the two-level comment classification and the second classifier (VOTE, SMO, or NB) from the enhanced two-level comment classification was not statistically significant. Therefore, removing *irrelevant* EDUs from *relevant* comments did not improve the accuracy of the second classifier.

Table 52 shows the comparison between two-level comment classifier and enhanced two-level comment classifier.

Classifier	Two-Level Comment Classifier Accuracy	Enhanced Two-Level Comment Classifier Accuracy
SMO	68.63%	64.24%
NB	68.74%	60.92%
VOTE	69.81%	64.24%

Table 52 - Comparison of Two-Level Comment Classifiers (BOW-EF)

According to t-test (test of significance) with p-value equals to 0.05, the difference between two-level comment classifier and enhanced two-level comment classifier was not statistically significant. Therefore, removing *irrelevant* EDUs from *relevant* comments did not statistically improve the accuracy of classification.

6.3 Argumentation Results

To extract arguments from comments, we first segmented all comments into EDUs using the Hilda discourse parser. We then asked annotators to annotate each EDU to topic-related and non-topic-related and annotate each topic-related EDU to pro or con. By “Pro”

we meant EDU contained reasoning for "Newborn genome sequencing" and by "con" we meant EDUs did not contain any reasoning related to "newborn genome sequencing".

After annotation, we extracted topic-related EDUs as well as two EDUs before and after topic-related EDUs. The following is the list of major concerns (manually identified by annotators) discussed in the dataset:

- Privacy Concerns
    - Retention /storage of DNA data in database
    - Heel prick cards
    - Exploitation of data by government or insurance companies?
    - Parental consent for use of DNA data
    - Research purposes (public/private)
    - Government involvement
    - Anonymity of DNA data for research purposes
    - DNA Privacy Laws, Mandatory testing laws (e.g. Michigan)

- Ethical Concerns
    - Sharing test results with children when older
    - If the test should be voluntary/compulsory

- Emotional Concerns
    - Hurting kids by taking blood samples – usually parents are not willing to consent to screening for their second infant after watching their first infant crying because of heel prick test

To extract the reasons against and for "newborn genome sequencing", we created n-grams and ranked them based on the frequency of occurrence in the dataset. We experimented with bigrams, trigrams, four-grams, ad five-grams. The most meaningful (meaningful to human) result came from five-grams. The frequency of the top five grams was 9. The top five reasons for "Newborn Genome Sequencing" are listed as follows:

1. Early child special medical treatment
2. Government know potential health benefit
3. Chance feel baby treatable problem
4. Order catch disease early stage
5. Genetic test tell eat avoid

Since we performed pre-processing before creating five-grams, we cross-referenced the above five-grams with the dataset to find more meaningful sentences. The following is the list that was created after cross-referencing with the dataset:

The top five reasons for “Newborn Genome Sequencing” are listed as follows:

1. I would rather know as early as possible if my child needs special medical treatment.

2. The potential health benefits are obvious.

3. If our baby had a treatable problem and we hadn't done it, we'd feel horrible.

4. I also want genetic tests that tell me what medical tests I should get done in order to catch diseases at early stages.

5. I would love to know if I have a genetic condition! Those people were saying it would make life a drag, but I disagree; the more you know about what condition you might get, the more you can do to avoid it.

As for reasons against “newborn genome sequencing”, we listed the top five reasons (base on frequency of five-grams) as follows (The frequency of the top five grams was 9):

1. Body DNA privacy law prevent
2. Public uncomfortable genetic testing promise
3. Newborn extract knowledge consent parent
4. Currently base massive genetic discrimination
5. Heel stick baby injury oppose

The above-mentioned five-grams were cross-referenced with the dataset to find more meaningful sentences. The following is the list that was created after cross-referencing with the dataset:

The top five reasons against “Newborn Genome Sequencing” are listed as follows:

1. Yet I suspect that the legal environment will change substantially as the price of DNA-sequencing continues to drop. “DNA Privacy laws” might prevent schools from openly requesting or using this information in some countries.

2. I believe the simple answer is that polls have shown the vast majority of the public are very uncomfortable with genetic testing, and it can be assumed that society will continue to pass legislation barring discrimination based on genetic tests.

3. Newborn screening is not just limited to the PKU. Most states test for at least 29 conditions. Many test for twice as many or more. Most parents want the testing for newborn conditions but may not like the idea of genomic sequencing. Many states allow religious exemptions. We encourage private testing to keep the DNA

out of the hands of government. Once government has it, they consider it government property and open to future legislative changes.

4. So long as parents get that it's just a snapshot of what could be, not what is or will be I don't see the issue. The only real issue is if schools or employers are allowed to ask what it reported. Then there's the risk of genetic discrimination, something that's already started to happen.

5. The look on my daughter's face when her heel was stabbed (not even close to a prick) made me regret giving the state this test immediately. What good could it possibly serve anyway? Will the state write you a letter and tell you if your baby is sick? How long would you know they were sick before you got that letter? I'm guessing the answers are no and at least a year. I vote skip it.

6.4 Discussion of Results

In this section results of the experiments that were performed will be discussed and compared. The following is the main outline of the experiments done for this thesis:

- Data collection and annotation
- Flat comment classification
- Two-Level comment classification
- Flat EDU classification
- Two-level EDU classification
- An attempt to improve the result of two-level comment classification by removing *irrelevant* EDUs from *relevant* comments
- Simple argumentation attempt to extract the reasons that forum users supported "newborn genome sequencing" or the users rejected "newborn genome sequencing".

The data required for this thesis was collected manually and automatically from many forums, the list of forums can be found in chapter 4. Two annotators performed annotation task at comment-level and EDU-level. At comment-level each comment was annotated to *positive, negative, mixed* or *irrelevant* ones. The kappa measure was 0.61 for comment level.

Forum users tend to talk about various *relevant* and *irrelevant* subjects in a comment. For instance, in a *relevant* comment, a user changed subject to *irrelevant* ones often, but at the end talked about the *relevant* subject. We think that this nature of the comments made it hard for annotators to achieve high agreement over annotation (Table 4). Two annotators perfumed the EDU-level annotation by annotating each EDU into *positive, negative, neutral,* and *irrelevant*. The kappa score for the agreement was 0.67.

Flat comment classification goal was to classify a comment into *positive, negative, mixed, or irrelevant*. We started with a simple classifier that classified comments into *positive, negative, mixed, or irrelevant* ones. The highest accuracy achieved (10-fold cross validation) for flat comment classifier using bag-of-words (BOW) and without any extra features was 63. 96%. In the next step, extra features added to investigate whether the presence of extra features would improve the accuracy. The highest accuracy achieved (10-fold cross validation) using bag-of-words and extra features (BOW-EF) was 65.14%. A total of 18 extra features were used at the comment level. The list of these features can be found in section 5.2. Statistical tests of significance with p-value equals to 0.05 proved that the difference in accuracy between the classifier with extra features (BOW-EF) and the classifier without extra features (BOW) was significant (t-test with p-value equals to 0.05).

In an effort to improve the accuracy of comment-level classification, a two-level classifier consisting of two classifiers was developed. The first classifier classified comments into *relevant* or *irrelevant* ones and the second classifier classified *relevant* comments predicted by first classifier into *positive*, *negative*, or *mixed* ones. The two-level classification was done using BOW and BOW-EF.

A gold standard two-level classifier named gold standard two-level comment classifier was developed. In the first level, *irrelevant* comments were removed manually. The classifier in the second level was called the second classifier. For the second classifier, the highest accuracy that was achieved was 61.21% using the VOTE classifier with BOW (Table 23). Furthermore, the highest accuracy of 61.43% was achieved for the same classifier (VOTE) using BOW-EF (Table 26). The difference between the two gold standard two-level comment classifiers was not statistically significant (p-value equals to 0.05)

In addition to the gold standard two-level classifier, a two-level comment classifier consisting of two classifiers was developed. Table 53 shows accuracy for the first and second classifier of two-level comment classifier.

	Gold Standard Two-Level		**Two-Level**	
	BOW	BOW-EF	BOW	BOW-EF
The First Classifier	N/A	N/A	77.40%	78.26%
The Second Classifier	61.23%	61.43%	63.05%	62.59%

Table 53 - Comparison of the First and Second Classifiers for Two-Level Comment Classification

As it can be seen in table 53, the difference between the first classifiers was not statistically significant (according to the paired t-test with p-value equals to 0.05). Furthermore, the difference between the second classifiers was not statically significant (p-value equals to 0.05). Therefore, adding extra features did not improve the accuracy of the two-level comment classification.

The overall result of two-level comment classification is shown in table 54. The highest accuracy achieved (evaluated by 10-fold cross validation) for the two-level comment classification was 69.48% using BOW and 69.81% using BOW-EF. The difference between the two two-level classifiers was not statistically significant (t-test with p-value equals to 0.05). Therefore, using the 18 extra (section 5.2) features did not improve the two-level comment classification.

Classifier (VOTE)	BOW	BOW-EF
Flat	63.96%	65.14%
Gold Standard Two-Level	81.47%	81.58%
Two-Level	69.48%	69.81%
Enhanced Two-Level	62.59%	62.32%

Table 54 - Comparison of Comment-Level Classifiers

At comment-level classification, two-level classifiers achieved better accuracy than flat classifiers. The two-level comment VOTE classifier using BOW-EF achieved the accuracy of 69.81%. The difference between the flat comment classifier and two-level comment classifier was statistically significant (t-test with p-value equals to 0.05).

In an attempt to improve comment-level classification and in order to perform a simple task of argumentation, we experimented with the EDU-Level classification. In ordered to prepare the training data for classification (supervised learning), all comment were segmented into consisting elementary discourse units (EDUs) by using Hilda discourse parse. Two annotators performed annotation of segmented EDUs into *positive, negative, neutral,* and *irrelevant* EDUs. Almost 87% of EDUs were annotated as *irrelevant* (Figure 12).

We developed a flat classifier that classified EDUs into *positive, negative, neutral,* or *irrelevant* ones. Total of 17 extra features were used in addition to bag-of-words (section 5.2). Under-sampling of *irrelevant* EDUs was performed due to the highly imbalanced nature of data. We achieved the *irrelevant* to *relevant* class ratio of 2:1 after under-sampling, and chose the VOTE as the first classifier for two-level EDU classification.

The purpose of EDU classification was to improve on comment classification by removing *irrelevant* EDUs from *relevant* comments. If the classifier misclassified irrelevant EDUs it may not be important, since a comment would keep its *relevant* EDUs as well as a few *irrelevant* EDUs that are misclassified as *relevant*. In this case we still managed to reduce the size of comment and keep most of the *relevant* EDUs. However, if the classifier misclassified *relevant* EDUs, the comment would loose most of its *relevant* EDUs and gain few *irrelevant* EDUs. In this case, the comment is reduced to few EDUs with less *relevant* EDUs.

To choose the proper subsampling ratio we experimented with EDU classification with *irrelevant* to *relevant* class ratio of 2:1, 3:1, 4:1, 5:1, 6:1, and finally 10:1. Tables 45

show the confusion matrices for VOTE first classifier, and It can be seen the table that less *relevant* classes are misclassified in 2:1 class ratio.

The highest accuracy achieved for flat EDU classification was 81.84% by using VOTE classifier from WEKA software package. Although the accuracy of the flat classifier was high, the f-measure for *positive, negative,* and *neutral* was low. The low f-measure for *positive, negative,* and *neutral* classes was caused due to the highly imbalanced data at the EDU-level.

In order to improve the F-measure for *positive, negative,* and *neutral* classes, a two-level EDU classifier consisting of two classifiers was developed. In the first level, a classifier (called the first classifier) classified EDUs into *relevant* or *irrelevant* ones and in the second level a classifier (called the second classifier) classified the *relevant* EDUs predicted by the first classifier into *positive, negative,* or *neutral* ones. The Accuracy of 90.18% was achieved for the first classifier with VOTE and BOW-EF. The accuracy of 63.37% was achieved for the second classifier with SMO and BOW-EF. The overall accuracy of 88.02% was achieved for the two-level EDU classification with the VOTE as the first classifier and the VOTE as the second classifier using BOW-EF. As it can be seen from the table 55, in EDU classification, we managed to achieve a higher accuracy than baseline (88.02%) by developing a two-level classifier.

Classifier (VOTE)	BOW-EF
ZeroR	92.87%
Flat	81.84%
Two-Level	88.02%

Table 55 - Comparison of EDU Classifiers (BOW-EF)

As an attempt to improve the result of comment-level classification, a two level classifier (called enhanced two-level comment classifier) consisting of three classifiers was developed. The enhanced two-level comment classifier used the result of EDU-level classification to remove the *irrelevant* EDUs for *relevant* comments. The workflow of this classifier can be seen in figure 17. In the first level, all comments were classified into *relevant* or *irrelevant* and in the second level all *relevant* comments were classified into *positive, negative,* or *mixed* ones. However, before the second level classification, EDUs that were predicted as *irrelevant* by the two-level EDU classifier, was removed from each *relevant* comment. The EDU classifier used to remove the *irrelevant* EDUs from a relevant comment was the two-level EDU classifier (accuracy of 88.02%) that was developed previously (table 55).

The accuracy for flat EDU classifier was 81.84%, whereas the accuracy for the two-level EDU classifier was 88.02%. The t-test with p-value equals to 0.05 showed that the difference between these classifiers were statistically significant. Therefore, we chose the two-level EDU classifier over the flat EDU classifier to be used in the enhanced two-level comment classification.

The difference between the two-level comment classification and enhanced two-level comment classification was not statistically significant (t-test with p-value equals to 0,05). However, removing irrelevant EDUs from relevant comments has the potential to reduce the dimensionality of the vocabulary. To summarize all classification accuracies, Table shows the accuracies for comment and EDU classification.

Classification-Level	Classifier	Accuracy
Comment-Level	Flat (BOW)	63.96%
	Flat (BOW-EF)	65.14%
	Gold Standard Two-Level (BOW)	81.47%
	Gold Standard Two-Level (BOW-EF)	81.58%
	Two-Level (BOW)	69.48%
	Two-Level (BOW-EF)	69.81%
EDU-Level	Flat (BOW-EF)	81.84%
	Gold Standard Two-Level (BOW-EF)	96.68%
	Two-Level (BOW-EF)	88.02%
Enhanced Comment-Level	Two-Level (BOW-EF)	62.32%

Table 56 - The Result of All Classifications (EDU and Comment Level)

In addition to classification, a basic task of augmentation analysis was performed to extract the reasons that forum users supported or rejected "newborn genome screening (NGS)". To perform the augmentation analysis, first, two annotators tagged the comments that included topic related word or phrases, and then all topic-related EDUs were annotated as *pro* or *con*. *Pro* EDUs were the ones containing word or phrases that supported NGS and *con* EDUs were the ones that contained word or phrases that disagreed with NGS. To extract the reasons that supported NGS, each EDU annotated as *pro* as well as two EDUs before and after the topic-related EDU were extracted. To extract the most common reason five-grams were generated from the corpus of EDUs extracted in the previous stage. Then all five-grams ranked based on its frequency in the corpus. From section 6.3, it can be seen that early treatment was the first reason that parents agreed with the NGS. Furthermore, to extract the reasons forum users (mainly parents) disagreed with NGS; each EDU annotated as *con* as well as two EDUs before and after the topic-related EDU were extracted. From the corpus of EDUs extracted in the previous step, five-grams were generated and ranked based on their frequency in the corpus.

Chapter 7 Conclusions and Future Work

7.1 Conclusions

In this thesis, sentiment analysis of online forum data on "Newborn Genome Sequencing" was performed. We manually annotated data collected from 29 forums and blogs and built a sentiment analysis system, collected and annotated a new data set. NGS was a rare discussion subject forums and we had to look for potential comments in blogs too.

In a blog, similar to a forum thread, comments were about a topic (blog post). Therefore, we assumed that comments from blogs and forums were similar enough and we treated blogs and forums as a single domain.

The kappa statistics for comment-level annotation was 0.61, whereas the agreement (kappa statistics) for EDU-level annotation was 0.67. The kappa score for comment-level annotation was indicative of the complexity of the data. The annotators found it hard to reach to a consensus over the class label for a comment. The following is the main reasons for the kappa score in comment-annotation (Examples can be found in Appendix A):

- A comment contained positive and negative sentences and a negative comment did not necessarily contain more negative sentences.
- A user could change his or her opinion many times in a comment and be sarcastic about the NGS topic.

The kappa score for the EDU-level annotation (0.67) was higher than the one for comment-level annotation (0.61) since EDUs were shorter than comments and annotators were not confused by the shift in positivity and negativity like comments.

A flat comment-level classifier was developed to classify each comment into *positive, negative, mixed,* or *irrelevant* one. To improve the result of the flat comment classification, we developed a two-level comment to classify comments into *relevant* and *irrelevant* and then the relevant ones into *positive, negative*, and *mixed*. To improve the comment-level classification, we developed 18 extra features. Flat and two-level comment classifiers where developed.

Furthermore, each comment was segmented into constituent elementary discourse units (EDUs) by using the Hilda discourse parse. The segmentation was done in an attempt to:

- Improve the two-level comment classification result.
- Extract the reasons that forum users supported "newborn genome screening" or they disagreed with it.

A flat EDU classifier was developed to classify each EDU into *positive, negative, neutral,* or *irrelevant*. To improve the result of the flat EDU-level classification, we developed a two-level EDU classifier that classified EDUs into *relevant* and *irrelevant* first, and then classified relevant EDUs into *positive, negative,* and *neutral*. For EDU-level classification, we used 17 extra features (the same set of features that was used for the comment-level classification minus lastEDUClassLabel).

In an attempt to improve the result of two-level comment classifier, we used the result of two-level EDU classifier. The enhanced two-level comment classifier was developed that consisted of three classifiers:

1. A classifier that classified comments into *relevant* or *irrelevant* ones.
2. A classifier that rated EDUs into *relevant* or *irrelevant* ones.
3. A classifier that classified each *relevant* comment (including just *relevant* EDUs) into *positive, negative,* or *mixed* one.

Table 56 shows the results achieved for EDU and comment classification

We performed a pair-wise comparison of classifiers at each classification level (flat and two-level), and for each set of features (BOW and BOW-EF). Furthermore, All classification results were obtained on the same data. Therefore t-test provided the most appropriate statistical iterative comparison of classifiers.

Statistical test of significance (t-test with p-value equals to 0.05) showed that the difference between the enhanced two-level comment classifier and the two-level comment classifier was not significant. Although there was no statistically significant improvement in the accuracy of the two-enhanced level comment classification, removing irrelevant EDUs had the potential to reduce the size of the vocabulary.

We probably could improve the low accuracy in comment-level and EDU-level classification by collecting more data. However, the NGS is a rare discussion topic in forums and blogs and in the time frame of the thesis project it was not possible to collect more data.

Finally, a simple task of argumentation was performed to extract the reasons that online forum users supported or rejected "newborn genome sequencing (NGS)". First, two annotators tagged EDUs with topic-related content, and then the annotators rated topic-related EDUs into *pro* or *con* EDUs. *Pro* EDUs were the ones that contained evidence that supported NGS and *con* EDUs were the ones that contained reasoning against NGS. After the annotation, all topic-related EDUs as well as two EDUs before and after the topic-related ones were extracted. Five-grams were generated from the collected data and sorted based on the frequency in the corpus. The top five five-grams (with the frequency of 9) were extracted as the most important reasons for and against NGS.

Considering the challenges that we had while collecting comments from forum and blog related to NGS and kappa score in annotation of comments and EDUs, we were not able

to achieve accuracies higher than 69.81% for comment-level classification. We think that more research is needed to improve the comment-level and EDU-level annotation agreement by developing better annotation guideline that can reflect the fussy nature of the forums and blogs comments in NGS domain. In addition, better sentiment analysis methods can be developed to improve the accuracy of the comment-level classification. In an effort to address the latest we briefly explained the future work in the next section.

7.2 Future work

In the future work, we are interested in investigating possibility of generating a global sentiment for a comment based on combining the sentiments of the consisting EDUs of that comment.

A discourse tree generated by Hilda software package consists of discourse relations as internal nodes and EDUs as external nodes. A discourse relation connects two EDUs, an EDU and a relation, or two relations based on some rules specific to that relation type. For instance, the Contrast relation connects two EDUs with opposite sentiments.

A rule-based classifier that combines the sentiments of EDUs based on the type of relations that connect EDUs can be developed to investigate whether combining the sentiment of EDUs based on the relation types in a discourse three.

Furthermore, the annotation for this thesis was done with Excel tool. The following is the list of changes we encountered during annotation task:

- To save and extract EDUs as tree with a comment at the root and EDUs children.
- To save meta-data related to a comment that would help annotators to tag data in the domain context. For instance, the discussion title of the forum thread that a comment belongs to.
- To provide annotators with a list of class labels (irrelevant, objective, positive, negative, mixed, and neutral) to choose from.

We are interested to develop a JAVA application that can solve the above-mentioned challenges for data annotation.

Bibliography

Anand, P., Walker, M., Abbott, R., Tree, J. E. F., Bowmani, R., & Minor, M. (2011). Cats rule and dogs drool!: Classifying stance in online debate. In Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis (pp. 1-9). Association for Computational Linguistics.

Baccianella, A. E. S. & Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), May, Valletta, Malta: European Language Resources Association (ELRA). ISBN: 2-9517408-6-7

Balahur, A., R. Steinberger (2009). Rethinking Sentiment Analysis in the News: from Theory to Practice and back. In Proceedings of the 1st Workshop on Opinion Mining and Sentiment Analysis.

Barbosa, L., and Feng, J. (2010). Robust sentiment detection on Twitter from biased and noisy data. In Proceedings of Coling.

Beineke, P., Hastie, T., & Vaithyanathan, S. (2004, July). The sentimental factor: Improving review classification via human-provided information. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (p. 263). Association for Computational Linguistics.

Bifet, A., and Frank, E. (2010). Sentiment knowledge discovery in Twitter streaming data. In Proceedings of 13th International Conference on Discovery Science.

Bobicev, V., Sokolova, M., Jafer, Y. & Schramm, D. (2012). Learning Sentiments from Tweets with Personal Health Information. In L. Kosseim & D. Inkpen (eds.), Canadian Conference on AI (pp. 37-48), : Springer. ISBN: 978-3-642-30352-4

Biyani, P., Caragea, C., Mitra, P., & Yen, J (2014). Identifying Emotional and Informational Support in Online Health Communities. In 25th International Conference on Computational Linguistics, 2014.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guille- mot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. (2005). The AMI Meetings Corpus. In *Proceedings of the Measuring Behavior Symposium on "Annotating and measuring Meeting Behavior"*.

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). A framework and graphical development environment for robust NLP tools and applications. In Proceedings of *ACL* (pp. 168-175).

Cabrio, E., & Villata, S. (2012). Combining textual entailment and argumentation theory for supporting online debates interactions. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2 (pp. 208-212). Association for Computational Linguistics.

Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. Computational linguistics, 22(2), 249-254.

Davidov, D.; Tsur, O. and Rappoport, A. (2010). Enhanced sentiment learning using twitter hashtags and smileys. In Proceedings of Coling.

Esuli A., Sebastiani F. (2006). SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In Proceedings of 5th Conference on Language Resources and Evaluation, pp. 417-422

Eysenbach, G. (2009). Infodemiology and Infoveillance: Framework for an Emerging Set of Public Health Informatics Methods to Analyze Search, Communication and Publication Behaviour on the Internet. Journal of Medical Internet Research, 11(1), e11. doi:10.2196/jmir.1157

Fellbaum, C. (2005). WordNet and wordnets. In K. Brown (ed.), Encyclopedia of Language and Linguistics (pp. 665-670), Oxford: Elsevier.

Feng, V. W., & Hirst, G. (2012). Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (pp. 60-68). Association for Computational Linguistics.

Han, J., Kamber, M., Pei, J. (2012). Data Mining Concepts and Techniques. Morgan Kaufman Publishers.

Hatzivassiloglou, V., & McKeown, K.R. (1997). Predicting the semantic orientation of adjectives. Proceedings of the 35th Annual Meeting of the Association for Computational Linguis- tics and the 8th Conference of the European Chapter of the ACL (pp. 174-181). New Brunswick, NJ: Association for Computational Linguistics.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification.

Jansen, B. J.; Zhang, M.; Sobel, K.; and Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. Journal of the American Society for Information Science and Technology 60(11):2169–2188.

Janyce Wiebe et al., "Coding Manual for Distinguishing Subjective and Objective Sentences in Text: Draft 2". March 4th 1999.

Kouloumpis, E., Wilson, T., & Moore, J. (2011). Twitter sentiment analysis: The good the bad and the omg!. ICWSM, 11, 538-541.

Kearnsand M. & Valiant G (1988). LearningBooleanformulaeorfiniteautomata is as hard as factoring. Technical Report TR-14-88, Harvard University Aiken Computation Laboratory.

Kearnsand M. & Valiant G (1994).CryptographiclimitationsonlearningBoolean formulae and finite automata. Journal of the Association for Computing Machinery, 41(1):67–95, January 1994.

Mukherjee, S., & Bhattacharyya, P. (2012). Sentiment Analysis in Twitter with Lightweight Discourse Analysis. In Proceedings of COLING (pp. 1847-1864).

Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, *29*(3), 436-465.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009). The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

Manning, C., Raghavan, P., and Schutze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Manning, C., & Schutze H. (1991). Foundation of Statisrical Natural Language Processing. The MIT Press.

New Born Screening Ontario (2014). Reprieved, October 22, 2014 from http://www.newbornscreening.on.ca

NRC Emotion Lexicon, Saif M. Mohammad and Peter D. Turney. (2013). NRC Technical Report, Ottawa, Canada.

Osman, D., J. Yearwood, P. Vamplew. (2010). Automated opinion detection: Implications of the level of agreement between human raters. Information Processing and Management, 46, 331–342.

O'Connor, B.; Balasubramanyan, R.; Routledge, B.; and Smith, N. (2010). From tweets to polls: Linking text sentiment to public opinion time series. In Proceedings of ICWSM.

Plutchik, R. (1984). Emotions: A general psychoevolutionary theory. *Approaches to emotion*, *1984*, 197-219.

Pak, A. & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining, LREC.

Pang, B. & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the ACL (pp. 271-278).

Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

Pennebaker, J. W., Francis, M. E., & Booth, R. J. (2001). Linguistic inquiry and word count: LIWC 2001. Mahway: Lawrence Erlbaum Associates, 71, 2001.

Peter D. Turney and Michael L. Littman. (2003). Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems, 21(4):315– 346.

Polanyi, L., & Zaenen, A. (2006). Contextual valence shifters. In Computing attitude and affect in text: Theory and applications (pp. 1-10). Springer Netherlands.

Riloff E. & J. Wiebe. (2003). Learning extraction patterns for subjective expressions. In Proceedigns of EMNLP-2003.

Riloff, E. & Jones, R. (1999, July). Learning dictionaries for information extraction by multi-level bootstrapping. In AAAI/IAAI (pp. 474-479).

Riloff, E., & Shepherd, J. (1997). A corpus-based approach for building semantic lexicons. arXiv preprint cmp-lg/9706013.

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu, (2013). NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets, In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013), Atlanta, USA.

Somasundaran, S., & Wiebe, J. (2009). Recognizing stances in online debates. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 (pp. 226-234). Association for Computational Linguistics.

Spertus, E. (1997). Smokey: Automatic recognition of hostile messages. Proceedings of the Conference on Innovative Applications of Artificial Intelligence (pp. 1058-1065). Menlo Park, CA: AAAI Press.

Staiano, J., & Guerini, M. (2014). DepecheMood: a Lexicon for Emotion Analysis from Crowd-Annotated News. Proceedings of ACL-2014.

Stone, P. J., Dunphy, D. C., Smith, M. S., & Ogilvie, D. M. (1966). The General Inquirer: A Computer Approach to Content Analysis. Cambridge, MA: MIT Press.

Strapparava, C. & Valitutti, A. (2004). WordNet-Affect: An affective extension of WordNet. Proceedings of the 4th International Conference on Language Resources and Evaluation (pp. 1083-1086).

Strapparava, C., R. Mihalcea. (2008). Learning to Identify Emotions in Text. In Proceedings of the 2008 ACM symposium on Applied Computing.

Somasundaran, S., Ruppenhofer, J., & Wiebe, J. (2007). Detecting arguing and sentiment in meetings. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue* (Vol. 6).

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational linguistics, 37(2), 267-307.

Thelwall, M., Buckley, K., and Paltoglou, G. (2012). Sentiment strength detection for the social web. Journal of American Society for Information Science, 63(1):163–173.

Tumasjan, A.; Sprenger, T. O.; Sandner, P.; and Welpe, I. (2010). Predicting elections with twitter: What 140 characters reveal about political sentiment. In Proceedings of ICWSM.

Valitutti, A., Strapparava, C. & Stock, O. (2004). Developing affective lexical resources. Psychology Journal.

Wachsmuth, H., Trenkmann, M., Stein, B., & Engels, G. (2014). Modeling Review Argumentation for Robust Sentiment Analysis. In Proceedings of the 25th International Conference on Computational Linguistics COLING.

Wiebe J., Bell M., Maple J. (1999). "Coding Manual for Distinguishing Subjective and Objective Sentences in Text: Draft 2"

William, M., & Thompson, S. (1988). Rhetorical structure theory: Towards a functional theory of text organization. *Text*, *8*(3), 243-281.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 347-354). Association for Computational Linguistics.

Wolf, F., Gibson, E., & Desmet, T. (2004). Discourse coherence and pronoun resolution. Language and Cognitive Processes, 19(6), 665-675.

Wyner, A., Schneider, J., Atkinson, K., & Bench-Capon, T. J. (2012). Semi-Automated Argumentative Analysis of Online Product Reviews. In *COMMA* (pp. 43-50).

Yang, C., Lin, K. H., & Chen, H. H. (2007). Emotion classification using web blog corpora. In Web Intelligence, IEEE/WIC/ACM International Conference on (pp. 275-278). IEEE.

Appendix A Annotation Guidelines

A.1 Introduction

Newborn genome screening (NGS) offers genome screening for serious diseases to all babies. Early identification of these diseases allows treatment that may prevent growth problems, health problems, mental retardation, and sudden infant death.

This guideline provides information to annotators to distinguish between NGS and any other tests done on babies after birth. To make annotators familiar with newborn genome sequencing domain some basic domain specific information is provided as question answering.

A.1.1 What is Newborn Screening (NGS) Testing?

NBS is a blood test done shortly after birth to test for treatable diseases that are not usually apparent in the newborn period. Early detection of these diseases is the key to effective treatment and can prevent serious health problems and even save lives. [29]

[29] Retrieved from http://www.newbornscreening.on.ca

A.1.2 What conditions screened in Ontario?

The following is the list of conditions tested in Ontario under NGS:

Test	LIS Code
Amino Acidemias:	
• Phenylketonuria and Variants / Biopterin defects • Maple Syrup UrineDisease • Homocystinuria (Hypermethioninemias) • Citrullinemias / Argininosuccinic Aciduria • Tyrosinemias • Amino Acidopathies, other	NBS-PKU NBS-MSUD NBS-HCY NBS-CIT NBS-TYR NBS-AA
Organic Acidemias:	
• Propionic / Methylmalonic Acidemias • Isovaleric Acidemia / 2 Methylbutyric Acidemia • Glutaric Acidemia Type 1 • 3 Methylcrotonic / Hydroxymethylglutaric / Methylglutaconic / 2-Methyl, 3-Hydroxybutyric Acidemias, or β-Ketothiolase Deficiency	NBS-C3 NBS-C5 NBS-C5DC NBS-C5OH
Fatty Acid Oxidation Defects:	
• Medium Chain Acyl Co-A Dehydrogenase Deficiency / Glutaric Acidemia Type 2 • Very Long Chain Acyl CoA-Dehydrogenase Deficiency • Long Chain Hydroxyl Acyl Dehydrogenase / Trifunctional Protein Deficiencies • Carnitine Uptake Defect • Fatty Acid Oxidation Disorders, other	NBS-MCAD NBS-VLCAD NBS-LCHAD NBS-CUD NBS-FA
Galactosemia	NBS-GALT
Biotinidase Deficiency	NBS-BIOT
Endocrine Disorders:	
• Congenital Hypothyroidism • Congenital Adrenal Hyperplasia	NBS-CH NBS-CAH
Sickle Cell and other Hemoglobinopathies	NBS-HGB
Cystic Fibrosis	NBS-CF
Severe Combined Immune Deficiency	NBS-SCID

Table 57 - List of Diseases on the NSO Newborn Screening Report[30]

[30] Newborn Screening Ontario. Retrieved from http://www.newbornscreening.on.ca/data/1/rec_docs/853_CHO0095-NSM-Pages-Jan2015-WEB.pdf

A.1.3 When is Newborn Screening (NBS) done?

NBS is done at least twenty-four hours after your baby is born, or before you leave the hospital.

A.1.4 What methods are used to conduct the Newborn Screening and when is the test conducted?

A heel prick (a small sample of blood taken from baby's heel) allows health care providers to collect a small amount of blood on special filter paper. This blood is sent to Newborn Screening Ontario (NSO) where it is tested [31]

A.1.5 What tests are not considered to be Newborn Screening?

Any NBS that is done before the birth of infant should be ignored by annotators since this project focuses on after birth NBS. In addition NBS test not done through heel prick test should be ignored too since heel prick is mentioned as standard test procedure by New Born Screening Ontario[32]

The followings are examples of tests should be ignored since done before the birth of infant:

- Amniocentesis
    - Amniocentesis is a prenatal test that allows your healthcare practitioner to gather information about your baby's health from a sample of your amniotic fluid. This is the fluid that surrounds your baby in the uterus. [33]

- First trimester testing
    - The First Trimester Screen is a new, optional noninvasive evaluation that combines a maternal blood-screening test with an ultrasound evaluation of the fetus to identify risk for specific chromosomal abnormalities, including Down Syndrome Trisomy-21 and Trisomy-18.[34]

- Penta Screening:

[31] New Born Screening Ontario. Retrieved from http://www.newbornscreening.on.ca

[32] New Born Screening Ontario. Retrieved from http://www.newbornscreening.on.ca

[33] Amniocentesis. Retrieved from http://www.babycenter.com/0_amniocentesis_327.bc

[34] Prenatal Screening Ontario. Retrieved from http://www.prenatalscreeningontario.ca/Default.aspx?cid=63&lang=1

- Prenatal screening is routinely offered for Down syndrome, and trisomy 18-risk assessment. [35]

A.1.6 Does Newborn Screening testing include entire genome sequencing or partial?

NGS doesn't include entire genome sequencing. It just includes testing for 29 diseases (Table 57)

A.2 Annotation

Annotation is done at comment level and EDU level. A comment consists of sentences and sentences consist of EDUs. A discourse structure is created for each comment. The discourse structure is a tree whose leaves correspond to *elementary discourse units (EDUs)*, and whose internal nodes correspond to contiguous text spans (called discourse spans). Discourse relations (markers) relate EDUs in the discourse tree. For instance, consider the sentence: "I partially agree with NGS, but I would do it". In this sentence, "I partially agree with NGS" and "but I would do it", are two EDUs and "but" is a discourse relation (Contrast relation). Annotators will be provided with the list of EDUs for each comment and should not worry about how to detect an EDU in a comment.

A comment is a set of sentences written by a user in response to a discussion opened in a forum as a thread. A thread is a set of comments under specific discussion title. A forum is a set of threads. Each discussion has a title and a user usually comments relative to the discussion title.

When annotators rating each comment, they should take the discussion title into consideration and annotate the comment relative to context.

A.2.1 Comment Annotation

In this section, we provide detailed guidelines on how to annotate a comment. Comment Annotation is done in two steps. First, each comment is annotated into *relevant* or *irrelevant*. Secondly, each relevant comment is annotated as *positive*, *negatives* or *mixed*.

A.2.1.1 Relevant/Irrelevant

A comment is relevant if it is explicitly or implicitly related to an opinion, or fact about newborn genome screening or sequencing (NGS) or other aspects that are related to NGS. The discussion title under which the comment appears should also be relevant.

[35] Retrieved from http://www.questdiagnostics.com/testcenter/testguide.action?dc=TS_Penta_Screen

The followings are phrases that are considered as relevant

- Newborn screening
- Newborn genome sequencing
- Newborn DNA testing/sequencing/screening
- Retention /storage of DNA data in database
- Heel prick cards
- Exploitation of data by government or insurance companies?
- Parental consent for use of DNA data
- Research purposes (public/private)
- Government involvement
- Sharing test results with children when older
- DNA Privacy Laws, Mandatory testing laws (e.g. Michigan)
- If the test should be voluntary/compulsory
- Anonymity of DNA data for research purposes

The following three examples are provided to show the relevant and *irrelevant* comments and how to annotate a comment as *relevant* or *irrelevant*. Comments should be analyzed considering the thread discussion title under which they appear. In each example, the discussion title, the comment, and analysis of the comment are provided. The sentences that used to decide whether a comment is *relevant* or *irrelevant* are highlighted as yellow.

1. **Relevant comment:**
   - **Discussion Title:** *"Genome Sequencing For Babies Brings Knowledge And Conflicts"*

     - **Comment***: "I have a family history of Alzheimer's disease. I have seen what it does and its sadness is a part of my life. I am already burdened with the knowledge that I am at risk. Knowing for sure will allow me to make some very difficult choices with my doctors. Data gathering. The immediate result not necessarily a cure in our lifetimes. Perhaps when we are long gone some incurable diseases would be curable. That's the spirit.* "*There is no gene for the human spirit.*"

     - **Analysis***:* User agrees with genome sequencing by mentioning that data collection in this manner can provide information for research and finding cure for incurable diseases. Therefore, The whole comment is relevant and the yellow-highlighted sentences are relevant sentences.

2. **Relevant comment:**
   - **Discussion Title:** "*New born blood screen/heel test*"

     - **Comment**: "*I allowed my DD my first child to undergo this test since I knew nothing of my FILs background; she cried so much I decided not*

to do it the second time. Well, DS was born and the state added two more tests that were pertinent to my family's background. I agreed to the test, and he turned out to have a low thyroid so I said lets do something about this, and the doctors said, no, it is not low functioning enough? WTF? So with #3 and #4, I just said no. No tests, and they are fine, thank you very much!"

- **Analysis**: The above comment is clearly relevant since user is talking about his/her experience with heel prick test.

3. **Irrelevant comment:**
   - **Discussion Title:** *"Genome Sequencing For Babies Brings Knowledge And Conflicts."*

   - **Comment**: My mother had 0% chance of giving birth. Several surgeries later here I am. My father had a sex change operation. Am I a miracle? No, I'm just a bunch of atoms moving around randomly. Life itself is a miracle; individual lives are just a small part of that miracle. I don't share your beliefs that individuals are special unique snowflakes. You should go adopt every child in Ethiopia if you care so much about human life. Otherwise keep typing away at your keyboard and passing judgment upon others.

   - **Analysis**: **There is no key word of phrases related to NGS in the above comment.**

A.2.1.2 Positive/Negative/Mixed

In this section annotators are required to annotate each comment into *positive*, *negative*, or *mixed* classes. A *positive* comment is one that has *positive* sentiment. In contrast, a *negative* comment is one carrying *negative* sentiment. A *mixed* comment is one that an annotator is indecisive about the positivity, negativity of it.

1. **Positive Comment:**
   - **Discussion Title:** *"First trimester genetic testing"*

   - **Comment**: I had screening done at 11.5 weeks. We both had agreed years ago that when we finally got pregnant, we'd screen. It would definitely be a key factor in our decisions, but I digress. I think I had that MaterniT21 test, not sure. They drew blood, and screened for the trisomy. That, in addition to the NT scan checking the neck and nasal bone, all added up to super low risk for me. This was the big test we were holding our breath for, so once it came back healthy we're much more comfortable. I'm just glad it wasn't super invasive and still gave

us the information we wanted! Also, cost: apparently my insurance covered the whole thing. I didn't even have copay.

- **Analysis**: The user is talking about newborn screening and shows *positive* sentiment about the result

2. **Negative Comment:**
   - **Discussion Title:** *"Newborn Screening Procedures - Recommended or Not?"*

   - **Comment**: Yeah it's that test. And you kind of confirmed my tin-foil hat concerns. The information is passed on to the Government where I'm sure it will be kept on file. While I will agree it's highly unlikely in this Country, in the past, dictators have killed off entire populations based off trivial issues (Saddam Hussein and the Kurds / Hitler and the Jews / Gaddafi and his people / Egypt and the people). So while I agree it probably won't happen, I'm not too keen on having genetic information stored with Government offices in the unlikely event that some future U.S. dictator decides to wipe out a mass of people based off a genetic 'deformity' that is deemed unfit for human survival.

   - **Analysis**: The user is concerned about the privacy issue if government keeps the results of NGS, and is worried if government wipes out newborns with genetic anomalies. Therefore, the user is showing *negative* sentiment towards NGS.

3. **Mixed Comment:**
   - **Discussion Title:** *"Newborn Screening Procedures - Recommended or Not?"*

   - **Comment**: These are my philosophical thoughts on the issue:
   - You need to define exactly what your purpose is, since you could be screwed either way you turn. If you don't take the screening you may be ramping up the chances of problems from the government. That may be ok with you if your purpose is to take a stand against government intervention. You just need to know that . . . with a government that you can't trust giving the info to, you could be setting yourself up for a fight if you refuse. Knowing that will help you define your purposes and actions. The real issue is not that the government is storing the data. The real issue is that the government is too big and interferes in things it shouldn't be interfering in. That is the problem that needs to be addressed with your energies, which is what the Ron Paul revolution is about . . . taking over every aspect of local and state government so that the government interference can be removed.

- **Analysis**: In the above comment, the user is arguing for and against NGS. The user does not providing definitive answer whether he/she supports or rejects NGS. Therefore, this comment can be considered as *mixed* one.

In the next section we provide detailed guideline on how to annotated EDUs.

A.2.2 EDU Annotation

In this section, we provide detailed information on how to annotate EDUs. This guide line does not cover how to segment a comment into consisting sentences and EDUs, since for each comment a list of EDUs are provided to annotators.

EDU annotation is done in three steps. First annotators required marking each EDU as *relevant* or *irrelevant*. Secondly, annotators should annotate each relevant EDU as *objective* or *subjective* and finally, each subjective EDU should be annotated as *positive*, *negative*, *mixed* or *neutral*.

For each step, we provide examples to clarify the guideline. An example is an individual comment and has the following structure:

- The thread discussion title under which a comment appears.
- The comment
- The list of EDUs for the comment
- The analysis section that shows the reasoning for EDU annotation

A.2.2.1 Relevant/Irrelevant

An EDU is relevant if it is explicitly or implicitly related to an opinion, or fact about newborn genome screening or sequencing (NGS) or other aspects that are related to NGS. The discussion title under which the EDU appears should also be relevant.

The followings are phrases that are considered as relevant

- Newborn screening
- Newborn genome sequencing
- Newborn DNA testing/sequencing/screening
- Retention /storage of DNA data in database
- Heel prick cards
- Exploitation of data by government or insurance companies?
- Parental consent for use of DNA data

- Research purposes (public/private)
- Government involvement
- Sharing test results with children when older
- DNA Privacy Laws, Mandatory testing laws (e.g. Michigan)
- If the test should be voluntary/compulsory
- Anonymity of DNA data for research purposes

The following example is provided to clarify the annotation of EDUs into relevant or *irrelevant* ones. It should be reminded that EDUs in a comment should be analyzed considering the comment context (thread discussion title). The yellow highlighted sections consist of relevant EDU(s) in a comment.

- **Example:**
  - **Discussion Title:** *"Genome Sequencing For Babies Brings Knowledge And Conflicts."*

    - **Comment:** One could imagine a day where knowing someone's entire genome sequence at birth, you could really begin to think about structuring their health care, their dietary choices, their exercise choices ... early in life, in a way that would have an impact on truly lifelong health, Guttmacher says. Gee, wouldn't one want to do that ANYWAYS? People are very stupid sometimes...also, human DNA itself doesn't tell the whole story. There are lots of epigenetic and environmental factors involved in the expression of genes, not to mention the genome of all the other non-human organisms in the body like bacteria. This could obviously be helpful for known genetic conditions limited to very specific gene regions/small amounts of genes, but most genetic characteristics involve so many interactions between genes that this technique seems premature to talk about.

    - **EDUs:**
      1. One could imagine a day
      2. Where knowing someone 's entire genome sequence at birth,
      3. You could really begin to think about structuring their health care, their dietary choices, their exercise choices ... early in life,
      4. In a way
      5. That would have an impact on truly lifelong health,
      6. Guttmacher says.
      7. Gee,
      8. Would not one want to do that ANYWAYS?
      9. People are very stupid sometimes ... also,
      10. Human DNA
      11. Itself does not tell the whole story.
      12. There are lots of epigenetic and environmental factors
      13. Involved in the expression of genes,

14. Not to mention the genome of all the other non-human organisms in the body like bacteria.
15. This could obviously be helpful for known genetic conditions
16. Limited to very specific gene regions/small amounts of genes,
17. But most genetic characteristics involve so many interactions
18. Between genes that this technique seems premature to talk about.

- **Analysis:** User used few NGS related key phrases that made yellow highlighted EDUs as relevant ones. The user argued for and against NGS. The EDUs 2,3,and 5 argues for NGS. The user used key phrases such as "genome sequencing", "birth", "health", and "DNA". The user also mentioned the positive impact of NGS on lifelong health. However, the EDUs 12,13,14,15,16,17,and 18 argue against NGS. The user used key phrases such as "genome", "genes", and "genetic". The user also mentioned that NGS is premature and could produce wrong diagnoses. Furthermore, the EDUs that are not highlighted are the *irrelevant* EDUs. It can be seen that in such EDUs no NGS related key phrases were used.

A.2.2.2 Subjective/Objective

In this section we provide detailed guideline on how to annotate an EDU into *subjective* or *objective* one.

If primary intention of an objective EDU is the presentation of material that is factual to the reporter, the sentence is objective. Otherwise, the sentence is subjective (Wiebe et al., 1999). Subjectivity annotation should be done just for relevant EDUs. The following three examples are provided to clarify how to annotate subjective versus objective EDUs in a comment (Wiebe et al., 1999):

- **Example:**
  - **Discussion Title:** *"Newborn Screening Procedures - Recommended or Not?"*

  - **Comment:** "Our first 3 children were hospital births that included all the test and interventions that went along with that. We felt completely helpless and out of control. The hospital staff used fear to manipulate use into doing what they wanted. After 3 horrible experiences in the hospital, my wife started studying childbirth with a passion. Our last 4 children have been born at home with no testing or medical intervention. Unless there was some type of complicating condition, we would never give birth in a hospital again. Granted, home-births are not for everyone. A woman should feel comfortable with wherever

she decides to give birth. Education is the key to having a happy and successful birth, whether in the hospital or at home**.**

- **EDUs:**
    1. Our first 3 children were hospital births that included all the test
    2. And interventions
    3. That went along with that.
    4. We felt completely helpless and out of control.
    5. The hospital staff used fear to manipulate use into doing what they wanted.
    6. After 3 horrible experiences in the hospital,
    7. My wife started studying childbirth with a passion.
    8. Our last 4 children have been born at home with no testing or medical intervention.
    9. Unless there was some type of complicating condition,
    10. We would never give birth in a hospital again.
    11. Granted, home-births are not for everyone.
    12. A woman should feel comfortable with wherever she decides to give birth.
    13. Education is the key to having a happy and successful birth, whether in the hospital or at home.

- **Analysis**: In the above example, yellow highlighted EDUs are subjective EDUs and the rest are objective EDUs. In the subjective EDUs, the user express feelings like fear. It can be seen that Objective EDUs provide factual information.

- **Example:**
    - **Discussion Title:** *"Genome Sequencing For Babies Brings Knowledge And Conflicts"*

    - **Comment:** Didn't anyone ever watch GATTACA?
      As a Ph.D. in the biomedical sciences, I would not want to know all this information on my self or my child. Our capacity to understand all these data is still very limited. Our DNA and biology may outline our blue print much about us, but so do our environment and the choices we make in life**.**

    - **EDUs:**
        1. Did not anyone ever watch GATTACA?
        2. As a Ph. D. in the biomedical sciences,
        3. I would not want to know all this information on my self or my child.
        4. Our capacity to understand all these data is still very limited.

5. Our DNA and biology may outline our blue print much about us,
6. But so does our environment and the choices
7. We make in life.

- **Analysis**: In the above example, yellow highlighted EDUs are subjective EDUs and the rest are objective EDUs. In the subjective EDUs, the user expresses *negative* sentiment such as "I would never …". It can be seen that Objective EDUs provide factual information.

- **Example**
  - **Discussion Title:** *"Newborn Genome Sequencing- Would You Want To Know If You Could?"*

  - **Comment:** I had cancer as a child, and while they say the type I had is not genetic, I am terrified that one of my children are going to develop it. I am torn as to whether or not I would get the test. My husband would probably be against it though. Having kids (and worrying about them) is terrifying business!

  - **EDUs:**
    1. I had cancer as a child,
    2. And while they say
    3. The type
    4. I had is not genetic,
    5. I am terrified
    6. That one of my children is going to develop it.
    7. I am torn as
    8. To whether or not I would get the test.
    9. My husband would probably be against it though.
    10. Having kids
    11. (And worrying about them)
    12. Is terrifying business!

  - **Analysis:** In the above example, yellow highlighted EDUs are subjective EDUs and the rest are objective EDUs. In the subjective EDUs, the user expresses *negative* feeling such as "I am torn", "I am terrified". It can be seen that Objective EDUs provide factual information.

It is also important to consider whose subjectivity are we considering while tagging a sentence to subjective or objective one (Wiebe et al., 1999).

- If subjectivity is originated from the writer of a comment himself/herself, the sentence is subjective.

- If the writer of a comment expresses subjectivity of someone other than himself/herself, the sentence is "private-state" or "speech-event". Private-state sentences can be identified if they contain verbs such as *hope, believe, feel, see, expect, know, etc*. It should be noted that private-state and speech-event are not considered subjective. The followings are two example:

    - Private-state: "*He believes that heel prick test hurt his dear daughter so he will pass on this test for his son*."

    - Speech-event: "*He said he agrees with newborn genome sequencing*. "

Annotators should take into account that our dataset does not have much of "private-state" or "speech-event" instances since. Users usually express their own opinion when commenting on a discussion title and rarely express opinion of somebody else. In the next section, we provide detailed information on how to annotate EDUs into *positive, negative, mixed,* or *neutral*.

A.2.2.3 Positive/Negative/Mixed/Neutral

In this section annotators are required to tag each EDU to one of *positive, negative, mixed,* or *neutral* classes. A *positive* EDU is the subjective EDU that has *positive* sentiment. A *negative* EDU is the subjective EDU carrying *negative* sentiment. A *mixed* EDU is the subjective EDU that the annotator is indecisive about the positivity, negativity, or neutrality of it. Finally, a *neutral* EDU is a subjective EDU with equal weight of *positive* and *negative* sentiment. The followings are examples to clarify how to annotate EDUs into *positive, negative, mixed,* or *neutral*:

- **Positive EDU Example:**
    - **Discussion Title:** *"First trimester genetic testing"*

    - **Comment**: I had screening done at 11.5 weeks. We both had agreed years ago that when we finally got pregnant, we'd screen. It would definitely be a key factor in our decisions, but I digress. I think I had that MaterniT21 test, not sure. They drew blood, and screened for the trisomy. That, in addition to the NT scan checking the neck and nasal bone, all added up to super low risk for me. This was the big test we were holding our breath for, so once it came back healthy we're much more comfortable. I'm just glad it was not super invasive and still gave us the information we wanted! Also, cost: apparently my insurance covered the whole thing. I didn't even have copay.

- **EDUs:**
    1. I had screening done at 11.5 weeks.
    2. We both had agreed years ago that
    3. When we finally got pregnant,
    4. We 'd screen.
    5. It would definitely be a key factor in our decisions,
    6. But I digress.
    7. I think
    8. I had that MaterniT21 test, not sure.
    9. They drew blood,
    10. And screened for the trisomy.
    11. That, in addition to the NT scan
    12. Checking the neck and nasal bone, all added up to super low risk for me.
    13. This was the big test
    14. We were holding our breath for, so once it came back healthy
    15. We are much more comfortable.
    16. I am just glad
    17. It was not super invasive
    18. And still gave us the information
    19. We wanted! Also, cost:
    20. Apparently my insurance
    21. Covered the whole thing
    22. I did not even have copay.

- **Analysis**: The user is talking about newborn screening and shows *positive* sentiment about the result in the yellow highlighted EDUs.

- **Negative EDU Example**
    - **Discussion Title:** *"Newborn Genome Sequencing- Would You Want To Know If You Could?"*

    - **Comment:** I had cancer as a child, and while they say the type I had is not genetic, I am terrified that one of my children are going to develop it. I am torn as to whether or not I would get the test. My husband would probably be against it though. Having kids (and worrying about them) is terrifying business!

    - **EDUs:**
        13. I had cancer as a child,
        14. And while they say
        15. The type
        16. I had is not genetic,
        17. I am terrified
        18. That one of my children is going to develop it.

19. I am torn as
20. To whether or not I would get the test.
21. My husband would probably be against it though.
22. Having kids
23. (And worrying about them)
24. Is terrifying business!

- **Analysis:** In the above example, yellow highlighted EDUs are *negative* EDUs. In the *Negative* EDUs, the user expresses *negative* feeling such as "I am torn", "I am terrified".

- **Mixed EDU Examples:**
  - **Discussion Title:** *"Genome Sequencing For Babies Brings Knowledge And Conflicts"*

  - **Comment:** I have a family history of Alzheimer's disease. I have seen what it does and its sadness is a part of my life. I am already burdened with the knowledge that I am at risk. Knowing for sure will allow me to make some very difficult choices with my doctors. Data gathering. The immediate result not necessarily a cure in our lifetimes. Perhaps when we are long gone some incurable diseases would be curable. That's the spirit. "There is no gene for the human spirit.

  - **EDUs:**
    1. I have a family history of Alzheimer 's disease.
    2. I have seen what it does and its sadness is a part of my life.
    3. I am already burdened with the knowledge that I am at risk.
    4. Knowing for sure will allow me to make some very difficult choices with my doctors.
    5. Data gathering.
    6. The immediate result not necessarily a cure in our lifetimes.
    7. Perhaps when we are long gone some incurable diseases would be curable.
    8. That 's the spirit.
    9. There is no gene for the human spirit.

  - **Analysis:** In the above comment, the yellow highlighted EDU is *mixed* since The EDU is subjective and the user is not sure about his decision to do NGS for his kid.

- **Neutral Example:**
  - **Discussion Title:** *"Genome Sequencing For Babies Brings Knowledge And Conflicts"*

- **Comment**: We are moving towards a day when our corporate healthcare system will sequence our children in the womb so they can be born with pre-existing conditions. If we are lucky they might release the results of the test to us, but I suspect they will keep it to themselves unless it enhances their profit margins. Heaven forbid they ever identify the Liberal/Conservative gene or there will be genocide.

- **EDUs:**
    1. We are moving towards a day
    2. When our corporate healthcare system will sequence our children in the womb so they can be born with pre-existing conditions.
    3. If we are lucky they might release the results of the test
    4. To us,
    5. But I suspect
    6. They will keep it to themselves
    7. Unless it enhances their profit margins.
    8. Heaven forbid
    9. They ever identify the Liberal/Conservative gene or there will be genocide.

- **Analysis:** In the above comment, the yellow highlighted EDU is *neutral* since the EDU is subjective and speculative. It should be noted that speculation can be considered as *neutral* (Wilson et al., 2005).

A.2.3 Annotation Workflow

The Comment annotation workflow is as follows:

1. All comments are to be annotated as *irrelevant/relevant*
2. All relevant comments are to be annotated as *positive/negative/ mixed*

The EDU annotation workflow is as follows:

1. All EDUs are to be annotated as *irrelevant/relevant* regardless of whether they are subjective/objective
2. All relevant EDUs are to be annotated as *subjective/objective*
3. All subjective EDUs are to be annotated as *positive/negative/mixed/neutral*.

Appendix B Classification Performance By Class

B.1 Comment Classification

B.1.1 Flat Comment Classification (BOW)

Precision	Recall	F-Measure	Class
0.542	0.578	0.56	Positive
0.253	0.222	0.237	Negative
0.25	0.222	0.235	Mixed
0.786	0.794	0.79	Irrelevant
0.611	0.619	0.615	Weighted Average

Table 58 - Flat Comment Classifier Performance by Class (SMO, BOW)

Precision	Recall	F-Measure	Class
0.5	0.641	0.562	Positive
0.182	0.156	0.168	Negative
0.33	0.354	0.341	Mixed
0.8	0.692	0.742	Irrelevant
0.608	0.59	0.595	Weighted Average

Table 59 - Flat Comment Classifier Performance by Class (NB, BOW)

Precision	Recall	F-Measure	Class
0.652	0.469	0.545	Positive
0.270	0.111	0.157	Negative
0.354	0.172	0.231	Mixed
0.677	0.920	0.780	Irrelevant
0.597	0.640	0.598	Weighted Average

Table 60 - Flat Comment Classifier Performance by Class (VOTE, BOW)

B.1.2 Flat Comment Classification (BOW-EF)

Precision	Recall	F-Measure	Class
0.545	0.520	0.532	Positive
0.244	0.122	0.163	Negative
0.260	0.131	0.174	Mixed
0.735	0.894	0.807	Irrelevant
0.585	0.636	0.603	Weighted Average

Table 61 - Flat Comment Classifier Performance by Class (SMO, BOW-EF)

Precision	Recall	F-Measure	Class
0.507	0.559	0.532	Positive
0.259	0.167	0.203	Negative
0.224	0.364	0.277	Mixed
0.795	0.704	0.747	Irrelevant
0.604	0.576	0.586	Weighted Average

Table 62 - Flat Comment Classifier Performance by Class (NB, BOW-EF)

Precision	Recall	F-Measure	Class
0.621	0.500	0.554	Positive
0.294	0.056	0.093	Negative
0.366	0.152	0.214	Mixed
0.687	0.941	0.794	Irrelevant
0.597	0.651	0.600	Weighted Average

Table 63 - Flat Comment Classifier Performance by Class (VOTE, BOW-EF)

B.1.3 Gold Standard Comment Classification (Second classifiers, BOW)

Precision	Recall	F-Measure	Class
0.743	0.723	0.733	Positive
0.280	0.231	0.253	Negative
0.369	0.455	0.407	Mixed
0.565	0.563	0.563	Weighted Average

Table 64 - The NB Classifier Performance by Class from The Gold Standard Two-Level Comment Classification

Precision	Recall	F-Measure	Class
0.675	0.738	0.705	Positive
0.256	0.220	0.237	Negative
0.420	0.374	0.396	Mixed
0.533	0.552	0.541	Weighted Average

Table 65 - The SMO Classifier Performance by Class from The Gold Standard Two-Level Comment Classification (BOW)

Precision	Recall	F-Measure	Class
0.655	0.914	0.763	Positive
0.389	0.154	0.220	Negative
0.472	0.253	0.329	Mixed
0.560	0.612	0.556	Weighted Average

Table 66 - The VOTE Classifier Performance by Class from The Gold Standard Two-Level Comment Classification (BOW)

B.1.4 Gold Standard Comment Classification (Second classifiers, BOW-EF)

Precision	Recall	F-Measure	Class
0.735	0.746	0.74	Positive
0.28	0.231	0.253	Negative
0.396	0.444	0.419	Mixed
0.567	0.574	0.570	Weighted Average

Table 67 - The NB Classifier Performance by Class from the Gold Standard Two-Level Comment Classification (BOW-EF)

Precision	Recall	F-Measure	Class
0.594	0.905	0.717	Positive
0.286	0.080	0.125	Negative
0.15	0.037	0.059	Mixed
0.437	0.553	0.457	Weighted Average

Table 68 - The SMO Classifier Performance by Class from the Gold Standard Two-Level Comment Classification (BOW-EF)

Precision	Recall	F-Measure	Class
0.656	0.918	0.765	Positive
0.353	0.132	0.192	Negative
0.500	0.273	0.353	Mixed
0.560	0.614	0.557	Weighted Average

Table 69 - The VOTE Classifier Performance by Class from the Gold Standard Two-Level Comment Classification (BOW-EF)

B.1.5 The Overall Gold Standard Comment Classification (BOW-EF)

Precision	Recall	F-Measure	Class
0.734	0.746	0.740	Positive
0.280	0.230	0.253	Negative
0.396	0.444	0.419	Mixed
1.000	1.000	1.000	Irrelevant
0.793	0.796	0.794	Weighted Average

Table 70 - The Gold Standard Two-Level Comment Classification Performance by Class (NB, BOW-EF)

Precision	Recall	F-Measure	Class
0.673	0.750	0.709	Positive
0.260	0.208	0.231	Negative
0.409	0.363	0.385	Mixed
1.000	1.000	1.000	Irrelevant
0.775	0.786	0.780	Weighted Average

Table 71 - The Gold Standard Two-Level Comment Classification Performance by Class (SMO, BOW-EF)

Precision	Recall	F-Measure	Class
0.656	0.917	0.765	Positive
0.352	0.131	0.192	Negative
0.500	0.236	0.321	Mixed
1.000	1.000	1.000	Irrelevant
0.789	0.786	0.785	Weighted Average

Table 72 - The Gold Standard Two-Level Comment Classification Performance by Class (VOTE, BOW-EF)

B.1.6 Two-Level Comment Classification (First Classifiers, BOW)

Precision	Recall	F-Measure	Class
0.723	0.814	0.766	Relevant
0.808	0.715	0.759	Irrelevant
0.767	0.762	0.762	Weighted Average

Table 73 - The First Classifier Performance by Class from the Two-Level Comment Classification (NB, BOW)

Precision	Recall	F-Measure	Class
0.779	0.733	0.755	Relevant
0.768	0.809	0.788	Irrelevant
0.773	0.773	0.773	Weighted Average

Table 74 - The First Classifier Performance by Class from the Two-Level Comment Classification (SMO, BOW)

Precision	Recall	F-Measure	Class
0.833	0.67	0.743	Relevant
0.744	0.877	0.805	Irrelevant
0.787	0.778	0.775	Weighted Average

Table 75 - The First Classifier Performance by Class from the Two-Level Comment Classification (VOTE, BOW)

B.1.7 Two-Level Comment Classification (Second Classifiers, BOW)

Precision	Recall	F-Measure	Class
0.718	0.822	0.767	Positive
0.321	0.250	0.281	Negative
0.407	0.324	0.361	Mixed
0.575	0.606	0.586	Weighted Average

Table 76 - The Second Classifier Performance by Class from the Two-Level Comment Classification (SMO, BOW)

Precision	Recall	F-Measure	Class
0.755	0.836	0.794	Positive
0.378	0.236	0.291	Negative
0.397	0.419	0.408	Mixed
0.606	0.631	0.614	Weighted Average

Table 77 - The Second Classifier Performance by Class from the Two-Level Comment Classification (NB, BOW)

Precision	Recall	F-Measure	Class
0.684	0.921	0.785	Positive
0.357	0.139	0.2	Negative
0.455	0.27	0.339	Mixed
0.571	0.631	0.576	Weighted Average

Table 78 - The Second Classifier Performance by Class from the Two-Level Comment Classification (VOTE, BOW)

B.1.8 Two-Level Comment Classification (First Classifiers, BOW-EF)

Precision	Recall	F-Measure	Class
0.787	0.744	0.765	Relevant
0.777	0.816	0.796	Irrelevant
0.782	0.782	0.781	Weighted Average

Table 79 - The First Classifier Performance by Class from the Two-Level Comment Classification (SMO, BOW-EF)

Precision	Recall	F-Measure	Class
0.665	0.863	0.751	Relevant
0.828	0.602	0.698	Irrelevant
0.750	0.727	0.723	Weighted Average

Table 80 - The First Classifier Performance by Class from the Two-Level Comment Classification (NB, BOW-EF)

Precision	Recall	F-Measure	Class
0.814	0.706	0.756	Relevant
0.761	0.852	0.804	Irrelevant
0.786	0.783	0.782	Weighted Average

Table 81 - The First Classifier Performance by Class from the Two-Level Comment Classification (VOTE, BOW-EF)

B.1.9 Two-Level Comment Classification (Second Classifiers, BOW-EF)

Precision	Recall	F-Measure	Class
0.697	0.827	0.757	Positive
0.400	0.316	0.353	Negative
0.339	0.235	0.277	Mixed
0.560	0.597	0.572	Weighted Average

Table 82 - The Second Classifier Performance by Class from the Two-Level Comment Classification (SMO, BOW-EF)

Precision	Recall	F-Measure	Class
0.740	0.777	0.758	Positive
0.333	0.250	0.286	Negative
0.404	0.444	0.424	Mixed
0.586	0.599	0.591	Weighted Average

Table 83 - The Second Classifier Performance by Class from the Two-Level Comment Classification (NB, BOW-EF)

Precision	Recall	F-Measure	Class
0.687	0.909	0.783	Positive
0.421	0.211	0.281	Negative
0.417	0.247	0.310	Mixed
0.575	0.626	0.580	Weighted Average

Table 84 - The Second Classifier Performance by Class from the Two-Level Comment Classification (VOTE, BOW-EF)

B.2 EDU Classification

B.2.1 Flat EDU Classification (BOW-EF)

Precision	Recall	F-Measure	Class
0.159	0.504	0.242	Positive
0.198	0.429	0.271	Negative
0.155	0.344	0.213	Neutral
0.971	0.847	0.905	Irrelevant
0.914	0.818	0.858	Weighted Average

Table 85 - Flat EDU-Level VOTE Classifier Performance by Class (BOW-EF)

Precision	Recall	F-Measure	Class
0.146	0.482	0.224	Positive
0.243	0.429	0.311	Negative
0.164	0.359	0.225	Neutral
0.975	0.853	0.910	Irrelevant
0.917	0.823	0.862	Weighted Average

Table 86 - Flat EDU-Level SMO Classifier Performance by Class (BOW-EF)

Precision	Recall	F-Measure	Class
0.133	0.431	0.203	Positive
0.077	0.490	0.133	Negative
0.077	0.320	0.124	Neutral
0.966	0.726	0.829	Irrelevant
0.905	0.704	0.782	Weighted Average

Table 87 - Flat EDU-Level NB Classifier Performance By Class (BOW-EF)

B.2.2 Gold Standard Two-Level EDU Classification (Second Classifiers, BOW-EF)

Precision	Recall	F-Measure	Class
0.631	0.778	0.697	Positive
0.617	0.485	0.543	Negative
0.567	0.445	0.499	Neutral
0.609	0.614	0.604	Weighted Average

Table 88 - The Second Classifier Performance by Class from the Gold Standard Two-Level EDU Classification (SMO, BOW-EF)

Precision	Recall	F-Measure	Class
0.600	0.566	0.582	Positive
0.491	0.455	0.473	Negative
0.459	0.521	0.488	Neutral
0.528	0.524	0.525	Weighted Average

Table 89 - The Second Classifier Performance by Class from the Gold Standard Two-Level EDU Classification (NB, BOW-EF)

Precision	Recall	F-Measure	Class
0.611	0.743	0.671	Positive
0.602	0.471	0.529	Negative
0.547	0.481	0.512	Neutral
0.588	0.591	0.584	Weighted Average

Table 90 - The Second Classifier Performance by Class from the Gold Standard Two-Level EDU Classification (VOTE, BOW-EF)

B.2.3 The Overall Gold Standard Two-Level EDU Classification (BOW-EF)

Precision	Recall	F-Measure	Class
0.600	0.565	0.582	Positive
0.491	0.455	0.472	Negative
0.459	0.521	0.488	Neutral
0.918	0.918	0.918	Irrelevant
0.887	0.886	0.886	Weighted Average

Table 91 - The Gold Standard Two-Level EDU Classifier Performance by Class (NB, BOW-EF)

Precision	Recall	F-Measure	Class
0.621	0.747	0.678	Positive
0.627	0.528	0.573	Negative
0.564	0.481	0.519	Neutral
0.918	0.918	0.918	Irrelevant
0.893	0.893	0.893	Weighted Average

Table 92 - The Gold Standard Two-Level EDU Classifier Performance by Class (SMO, BOW-EF)

Precision	Recall	F-Measure	Class
0.610	0.743	0.670	Positive
0.602	0.471	0.528	Negative
0.547	0.481	0.511	Neutral
0.918	0.918	0.918	Irrelevant
0.891	0.892	0.891	Weighted Average

Table 93 - The Gold Standard Two-Level EDU Classifier Performance by Class (VOTE, BOW-EF)

B.2.4 Two-Level EDU Classification (First Classifiers, BOW-EF)

Precision	Recall	F-Measure	Class
0.398	0.633	0.489	Relevant
0.966	0.915	0.940	Irrelevant
0.920	0.893	0.903	Weighted Average

Table 94 - The First Classifier Performance by Class from the Two-Level EDU Classification (SMO, BOW-EF)

Precision	Recall	F-Measure	Class
0.250	0.670	0.364	Relevant
0.966	0.823	0.888	Irrelevant
0.908	0.810	0.846	Weighted Average

Table 95 - The First Classifier Performance by Class from the Two-Level EDU Classification (NB, BOW-EF)

Precision	Recall	F-Measure	Class
0.427	0.614	0.504	Relevant
0.964	0.927	0.946	Irrelevant
0.921	0.902	0.910	Weighted Average

Table 96 - The First Classifier Performance by Class from the Two-Level EDU Classification (VOTE, BOW-EF)

B.2.5 Two-Level EDU Classification (Second Classifiers, BOW-EF)

Precision	Recall	F-Measure	Class
0.624	0.778	0.693	Positive
0.600	0.431	0.502	Negative
0.542	0.489	0.514	Neutral
0.590	0.594	0.585	Weighted Average

Table 97 - The Second Classifier Performance by Class from the Two-Level EDU Classification (NB, BOW-EF)

Precision	Recall	F-Measure	Class
0.651	0.768	0.704	Positive
0.623	0.575	0.598	Negative
0.613	0.515	0.560	Neutral
0.631	0.634	0.629	Weighted Average

Table 98 - The Second Classifier Performance by Class from the Two-Level EDU Classification (SMO, BOW-EF)

Precision	Recall	F-Measure	Class
0.623	0.803	0.702	Positive
0.678	0.467	0.553	Negative
0.604	0.527	0.563	Neutral
0.630	0.626	0.618	Weighted Average

Table 99 - The Second Classifier Performance by Class from the Two-Level EDU Classification (VOTE, BOW-EF)

B.3 Enhanced Comment Classification

B.3.1 Enhanced Comment Classification (Second Classifiers, BOW_EF)

Precision	Recall	F-Measure	Class
0.707	0.655	0.680	Positive
0.270	0.250	0.260	Negative
0.352	0.451	0.395	Mixed
0.548	0.533	0.539	Weighted Average

Table 100 - The Second Classifier Performance by Class from the Enhanced Two-Level Comment Classification (NB, BOW-EF)

Precision	Recall	F-Measure	Class
0.690	0.854	0.764	Positive
0.465	0.294	0.360	Negative
0.404	0.268	0.322	Mixed
0.587	0.623	0.593	Weighted Average

Table 101 - The Second Classifier Performance by Class from the Enhanced Two-Level Comment Classification (SMO, BOW-EF)

Precision	Recall	F-Measure	Class
0.656	0.937	0.772	Positive
0.407	0.162	0.232	Negative
0.458	0.155	0.232	Mixed
0.567	0.623	0.554	Weighted Average

Table 102 - The Second Classifier Performance by Class from the Enhanced Two-Level Comment Classification (VOTE, BOW-EF)

Appendix C Discourse Markers

The list of discourse markers:

1. Elaboration
2. Enablement
3. Contrast
4. Attribution
5. Joint
6. Background
7. Span
8. Same-unit
9. Condition
10. Evaluation
11. Explanation
12. Generalization
13. Similarity
14. Cause-effect
15. Example
16. Violated-expectations
17. Temporal-sequence
18. Temporal
19. Textual-organization
20. Summary
21. Cause
22. Topic-comment
23. Manner-means
24. Comparison
25. Topic-change

Appendix D Forums and Blogs

The list of Blogs and forums crawled for this thesis:

Blogs:

1. http://www.npr.org/blogs/health/
2. http://scienceblogs.com/geneticfuture/
3. http://www.guardian.co.uk/science/blog/
4. http://www.dailypaul.com/
5. http://www.babyrabies.com/
6. http://healthland.time.com/
7. http://www.deseretnews.com/article/
8. http://blogs.babycenter.com/mom_stories/
9. http://www.theglobeandmail.com/news/national
10. http://www.eyeondna.com/
11. http://www.futurepundit.com/archives/

Forums:

1. http://forum.cysticfibrosis.com/threads/
2. http://www.mothering.com/community/t/
3. http://forums.naturalparenting.com.au/
4. http://www.parentingforums.org/forum.php/
5. http://forums.familyeducation.com/
6. http://neogaf.com/forum/
7. http://www.bubhub.com.au/community/forums/
8. http://babyandbump.momtastic.com/parenting-forums/
9. http://www.huggies.co.nz/forum/
10. http://tribes.tribe.net/postpartummammas/thread/
11. http://www.marijuana.com/threads/
12. http://www.diaperswappers.com/forum/
13. http://www.christianparentsforum.com/forums/
14. http://www.justmommies.com/babies/newborn/
15. http://forum.smartcanucks.ca/canadian-parents/
16. http://www.themommyplaybook.com/forum/index.php/
17. http://www.godaycare.com/childcareforum/
18. http://fosterparentforum.org/