Automated Classification of Text Sentiment

Emmanuel Dufourq
Department of Mathematics, University of Cape Town
African Institute for Mathematical Sciences
Cape Town, South Africa
edufourq@gmail.com

Bruce A. Bassett
Department of Mathematics, University of Cape Town
African Institute for Mathematical Sciences
South African Astronomical Observatory, Cape Town,
South Africa
bruce.a.bassett@gmail.com

ABSTRACT

The ability to identify sentiment in text, referred to as sentiment analysis, is one which is natural to adult humans. This task is, however, not one which a computer can perform by default. Identifying sentiments in an automated, algorithmic manner will be a useful capability for business and research in their search to understand what consumers think about their products or services and to understand human sociology. Here we propose two new Genetic Algorithms (GAs) for the task of automated text sentiment analysis. The GAs learn whether words occurring in a text corpus are either sentiment or amplifier words, and their corresponding magnitude. Sentiment words, such as 'horrible', add linearly to the final sentiment. Amplifier words in contrast, which are typically adjectives/adverbs like 'very', multiply the sentiment of the following word. This increases, decreases or negates the sentiment of the following word. The sentiment of the full text is then the sum of these terms. This approach grows both a sentiment and amplifier dictionary which can be reused for other purposes and fed into other machine learning algorithms. We report the results of multiple experiments conducted on large Amazon data sets. The results reveal that our proposed approach was able to outperform several public and/or commercial sentiment analysis algorithms.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Bio-inspired approaches**;

KEYWORDS

Sentiment analysis, genetic algorithm, machine learning

1 INTRODUCTION

The amount of data collected from users around the world and stored for posterity has skyrocketed over the past decade as websites such as Twitter, Amazon and Facebook have facilitated the publication and aggregation of micro opinion pieces that allow individuals to record their sentiments towards things, people and events. This data is clearly of value to researchers, organizations and companies to understand sentiment both as individuals and on average, and as well as to identify trends. The automated detection of emotions and attitudes towards a particular subject, event or entity is what we will call sentiment analysis [16, 23]. Sentiment analysis has been applied to many problem domains; for instance, determining sentiments of consumers towards products, or mining social media to gain an understanding of the public's opinion on matters such as corruption [16, 23, 26].

For adult humans, interpreting the underlying emotions in text is usually performed unconsciously and with apparent ease. We are able to recognize emotions in emails, sentiments in our social media feed and appreciate the subtle nuances of conflicting views in novels. Nevertheless, even for humans text can be notoriously easy to misinterpret. For machines, on the other hand, sentiment analysis is highly non-trivial. From the year 2000 onwards, a number of researchers have begun contributing towards the field of sentiment analysis [23]. This area of research is highly active, increasingly so, due to the vast amount of digital information available, and the amount of sentiments expressed online. With the rapid increase in computational power available in the recent years and the extreme amount of data available online, it is clear that developing novel sentiment analysis methods will be beneficial to organizations in order to enable them to understand what the public feels about their products and services.

In this study, two genetic algorithms (GA) were proposed for text sentiment analysis. The proposed approach optimizes the sentiments of the words in order to correctly classify as much data as possible. This research proposes a new way of representing words, as either a sentiment or an amplifier word, whereby amplifier words intensify the sentiments in sentences. The words are combined to form mathematical expressions in order to determine whether or not a given sentence is positive or negative. The following section describes the GA which was used to create the models which perform sentiment analysis.

2 GENETIC ALGORITHM

A GA [9] is an evolutionary algorithm [8] inspired by "survival of the fittest" in nature that can be used to solve optimization problems. A GA evolves a population of chromosomes which are made up of several genes. The size of the population is a user-defined

parameter. Each chromosome represents a candidate solution to the optimization problem. Each chromosome is evaluated in order to determine how successful it is at solving the optimization problem. The evaluation is obtained by computing the *fitness* of each chromosome. For a maximization problem, a chromosome with a higher fitness is considered as a better one, whereas a chromosome with a smaller fitness is considered as a weaker one.

This study implements GAs to optimize the correct classification (positive or negative) of short pieces of text given words of unknown sentiment in the text. While some sentiment algorithms make use of large dictionaries of words with associated sentiment values [20], the GAs we propose can learn the type and associated sentiment values of words; although the GA can make use of sentiment dictionaries if desired. This is appropriate if there is little training data.

Algorithm 1 illustrates the pseudocode for a GA. An initial population of chromosomes is randomly created in step 2, and each chromosome is evaluated in step 3 to determine if a solution to the optimization problem exists from the initial population. In step 5, the algorithm enters into a generational loop until the maximum number of generations is met, or until a solution to the optimization problem is found. The maximum number of generations is a user-defined parameter.

Algorithm 1: Genetic algorithm

---

input: generation_max: maximum number of GA generations

1 **begin**
2   Create an initial population of chromosomes.
3   Evaluate the initial population.
4   generation ← 0.
5   **while** generation ≤ generation_max **do**
6   generation ← generation + 1.
7   Select the parents.
8   Perform the genetic operators.
9   Replace the current population with the new offspring created in step 8.
10   Evaluate the current population.
11   **return** *The best chromosome.*

---

3 RELATED WORK

GAs have been used before in sentiment analysis studies, though not primarily for actual sentiment determination but rather for feature selection and reduction, e.g. [1]. There the chromosomes had length equal to the total number of features, and the genes were encoded with a *0* or a *1* depending on whether or not that particular feature was to be used or not. The GA optimized which features to use from the original set, and a SVM classifier was then applied to that feature set in order to train and predict the reviews. Genes were encoded in a similar manner for feature selection with the ultimate goal of reducing the number of features in the study of Kalaivani and Shunmuganathan [13].

GAs were also used to optimize features in several other studies, such as that of Paramesha and Ravishankar [25] which used a GA in

order to allocate weights to features. Govindarajan [10] proposed an ensemble approach using Naive Bayes and a GA. Smith [27] proposed the use of GA to reduce the number of features as did Acampora and Cosma [2].

Carvalho et al. [5] present a novel GA approach whereby a fixed chromosome is split in two parts, a positive and a negative part. A set of 25 positive and 25 negative words were seeded into the algorithm. Their approach attempts to find which of those words should be added into the respective parts of the GA chromosomes in order to maximize the accuracy of classifying Twitter tweets. A chromosome is then evaluated using a distance measure based on the words in the tweets in relation to the words in the chromosome. Thus, for example, if a particular chromosome is evaluated on some tweet, and the words in the tweet are considered to be nearer (based on the distance measure) to the positive words in the chromosome than to the negative, then the tweet is classified as positive.

Das and Bandyopadhyay [7] make use of a GA for subjectivity detection. Even though this area of research does not deal with sentiments, the research is aligned. Ten features were chosen and a number of predetermined values were assigned to each feature. An example of two features used were parts-of-speech and SentiWordNet values. The former takes up to 45 possible parts-of-speech values; and latter 2 values, positive or negative. The aim behind the research was to optimize the best set of features.

By contrast, the rationale behind the present study is not to propose a new GA feature selection method; instead, the focus is to propose a GA that determines the sentiment of reviews without making use of a feature set. Furthermore, our approach treats each individual piece of text with a sentiment as a mathematical formula made up of unknown variables corresponding to each word in the text. Thus, the goal is to use a GA to simultaneously solve for the unknown variables as a step towards correctly predicting the total sentiment of a piece of text.

4 CLASSIFICATION-VALUE PAIR

In our study, each word is assigned both a 'classification' and a 'value' that we call a classification-value pair in the form '*classification:value*'. Classifications take on one of two types, namely either *sentiment* or *amplifier*. Intuitively this captures the difference between words that carry sentiment directly (e.g. 'horrible', 'sad', 'wonderful') and adjectives/adverbs that modify the sentiment of the following word (e.g. 'very', 'not', 'little'). In addition to this classification every word is given exactly one value associated with that classification, taken from this list:

- Sentiment ∈ {*-1.0, 0.0, 1.0*}
- Amplifier ∈ {*0.5, 1.0, 1.5*}

For this classification-value pair, a sentiment value of -1, 0 and 1 represents a negative, neutral and positive sentiment respectively. The three values for the amplifier represent different intensification values, i.e. a value of 1.5 is a larger amplification than a value of 0.5. These values were selected by conducting various preliminary runs.

A word is referred to as an unknown word if its classification-value is not known. Examples of three classification-value pairs are: *sentiment:1.0*, *amplifier:0.5*, and *sentiment:-1.0*.

The goal of this study is to optimize and determine the classification-value pairs for certain unknown words within a data set, given that, a number of words already have known classification-value pairs. The words which already have a known classification-value pair are stored in a dictionary. Words in a dictionary do not have to be optimized and their classification-value pairs are never altered.

In this study we use two dictionaries, one for sentiment words and one for amplifier words. Known sentiment words were added into the sentiment dictionary, and similarly, known amplifier words were added into the amplifier dictionary. This was done to provide seeds as to guide the GA to converge to the correct solutions. Furthermore, the proposed algorithm can extend these dictionaries in order to create a sentiment lexicon. Details regarding which dictionaries were used are provided in section 7.

5 PROPOSED METHODS FOR OPTIMIZING CLASSIFICATION-VALUE PAIRS

This section describes the use of machine learning in order to optimize the classification-value pairs for the unknown words in the sentences of a data set. We propose Genetic Algorithm for Sentiment Analysis (*GASA*). Each aspect of the GA is explained in terms of how it has been adapted for *GASA* in the following subsections.

5.1 GASA chromosome representation

Each gene within a chromosome is made up of the classification-value pair for an unknown word (not in the sentiment or amplifier dictionary). The length of the chromosome is equal to the number of unknown words in the training corpus. The classification for each unknown word corresponds to a gene in the chromosome, and thus the classification of unknown words: $word_1, word_2, word_3,$ $\ldots, word_n$ is mapped to: $gene_1, gene_2, gene_3, \ldots, gene_n$ — where n represents the number of unknown words in a training corpus. This mapping is never changed.

In order to illustrate the chromosome representation, suppose there are three unknown words: $word_1, word_2, word_3$. Figure 1 illustrates an example of a candidate chromosome of length 3. The illustrated chromosome corresponds to the following classification:

- $word_1$ in gene position 1 is classified as a *sentiment* word with a value of 1.0
- $word_2$ in gene position 2 is classified as an *amplifier* word with a value of 0.5
- $word_3$ in gene position 3 is classified as a *sentiment* word with a value of 0.0

Figure 1: Example of a *GASA* chromosome.

5.2 GASA initial population generation

Prior to creating the initial population, the unknown words have to be input into the GA. The initial population size is set to the same value as the user-defined population size. Suppose the population size is n, then n chromosomes are created during the initial population generation. Each chromosome has a fixed length which is set to the number of unknown training words. The pseudocode for creating a chromosome is presented in algorithm 2. The genes which make up the chromosome are created by randomly selecting either a sentiment or an amplifier classification and assigned a value randomly as described in section 4.

Algorithm 2: Creating a chromosome.

    input : size: the number of unknown words
1 **begin**
2   Initialise the length of the chromosome to size
3   **for** *each gene in the chromosome* **do**
4   Randomly select a classification type.
5   Randomly select a corresponding value for the classification type previously obtained in step 4.

5.3 GASA chromosome evaluation

A chromosome has to be evaluated in order to determine how good it is at solving the optimization problem (namely how well it predicts the overall sentiment of a piece of text). Every chromosome is evaluated on each instance in the data set. In this study, an instance corresponds to text.

Assume that chromosome c is being evaluated. Chromosome c is applied to every instance in the data set, and each word within the instances is examined in order to obtain its classification-value pair. Assume that chromosome c is evaluating instance i, whereby the text for instance i is made up of the following words: $w_1, w_2, w_3,$ $\ldots, w_n$, and n denotes the length of instance i.

If a word w_i from instance i is in the sentiment dictionary, then w_i is classified as a sentiment word, and its corresponding value is retrieved from the sentiment dictionary. Similarly, if w_i is in the amplifier dictionary, then w_i is classified as an amplifier word, and its corresponding value is obtained from the amplifier dictionary. If however, w_i is unknown, then its classification-value pair is retrieved from chromosome c.

Once the classification-value pair for every word in an instance of data has been obtained, these classification-value pairs are converted into a mathematical expression in order to obtain the sentiment for the instance. The mathematical expression is evaluated sequentially from left to right. Algorithm 3 presents the pseudocode to evaluate expressions. Amplifier words boost the sentiment words, whereas the sentiment words accumulate each other. If the final word is an amplifier, then that value is simply added onto the result. A positive output denotes a positive sentiment, and a negative output denotes a negative sentiment.

The fitness of a chromosome is determined as the total number of instances for which the sentiment output by the chromosome is equal to the correct sentiment from the data set. Assume that some data set has sentences $s_1, s_2,$ and s_3, and these have correct

Emmanuel Dufourq and Bruce A. Bassett

sentiments of *positive, negative, positive* respectively. If some chromosome evaluates each sentence to: *negative, negative, negative,* then the fitness of that chromosome is one, since, it only correctly classified the second sentence.

Algorithm 3: Pseudocode for arithmetically evaluating a sentence.

input : sentence: the sentence to be evaluated
output : The sentiment for the evaluated sentence

1 **begin**
2   sentiment_count ← 0
3   amplifier_count ← 0
4   **for** *each* word *in the* sentence **do**
5   **if** word *is an amplifier* **then**
6   amplifier_count ← amplifier_count + word's amplifier value
7   **else**
8   **if** amplifier_count *is non-zero* **then**
9   sentiment_count ← sentiment_count + amplifier_count × word's sentiment value
10   **else**
11   sentiment_count ← sentiment_count + word's sentiment value
12   **if** amplifier_count *is non-zero* **then**
13   sentiment_count ← sentiment_count + amplifier_count
14   **return** sentiment_count

5.4 GASA parent selection

Parent selection methods are used to obtain parents from the current population of chromosomes. These parents are used by the genetic operators in order to create offspring. A single parent is obtained when the parent selection method is executed. Once a chromosome has been chosen to be a parent, the selection method can select that particular chromosome again. Three common parent selection methods are fitness-proportionate, rank and tournament selection [4]. For this study, tournament selection was used given that it was shown to be a successful method by Zhong et al. [33].

Algorithm 4 presents the pseudocode for the tournament selection. This selection method has one user-defined parameter, namely, the tournament size. Let k be the tournament size. Tournament selection randomly selects k chromosomes from the current GA population, and compares the fitness of each of the k chromosomes. The chromosome with the highest fitness is returned as the parent chromosome. If a tie occurs, then a random chromosome is selected to break the tie.

5.5 GASA genetic operators

Genetic operators are applied to parents in order to exchange genetic material between the parent chromosomes, and to consequently create novel offspring. The two most common genetic operators are mutation and crossover. Their implementation details for this study are described below.

Algorithm 4: Pseudocode for tournament selection.

input : size: size of the tournament
output : The best chromosome which will be used as a parent

1 **begin**
2   current_best ← null
3   **for** $i ← 1$ **to** size **do**
4   random_chromosome ← randomly select a chromosome from the population
5   Evaluate random_chromosome
6   **if** *fitness of* random_chromosome > *fitness of* current_best **then**
7   current_best ← random_chromosome
8   **return** current_best

5.5.1 GASA mutation. The mutation genetic operator makes use of a single parent chromosome. The classification-value for a single gene in the parent is modified to a new one. A user-defined parameter is associated with the mutation operator, namely the mutation application rate. Figure 2 illustrates the application of the mutation operator on a parent chromosome, and the resulting offspring is illustrated. The second gene in the parent was changed from a classification of "amplifier" with a value of 0.5 to a classification of "sentiment" with a value of 1.0.

parent chromosome

Sentiment: 1.0	Amplifier: 0.5	Sentiment: 0.0

child chromosome

Sentiment: 1.0	Sentiment: 1.0	Sentiment: 0.0

Figure 2: Example of *GASA* mutation. The second gene was selected for mutation and was changed from an amplifier with a value 0.5 to a sentiment with a value of 1.0. The other genes remain unchanged.

5.5.2 GASA crossover. The crossover genetic operator exchanges genetic material between two parent chromosomes: $parent_1$ and $parent_2$, and consequently creates two offspring: $child_1$ and $child_2$. There are several variations of the crossover genetic operator, such as uniform, one-point and two-point crossover.

The crossover method we implement randomly selects a position p in the range $[0, n]$ — where n denotes the length of the chromosome — within the parent chromosomes; the same position p must be selected within the two parents. Two offspring are created, and all the genes except those at position p are copied across to the corresponding offspring without modification. The genes are position

p are swapped, i.e., the gene in position p from $parent_1$ is inserted into position p in $child_2$, and similarly, the gene in position p from $parent_2$ is inserted into position p in $child_1$.

Figure 3 illustrates the application of the proposed crossover operator on two parent chromosomes; the resulting offspring are also illustrated. In this case, the value of p was 1, implying that the first gene was swapped amongst the parent chromosomes.

parent$_1$ chromosome			parent$_2$ chromosome		
Amplifier: 1.0	Amplifier: 0.5	Sentiment: 0.0	Sentiment: 1.0	Sentiment: 0.0	Sentiment: 0.0

child$_1$ chromosome			child$_2$ chromosome		
Sentiment: 1.0	Amplifier: 0.5	Sentiment: 0.0	Amplifier: 1.0	Sentiment: 0.0	Sentiment: 0.0

Figure 3: Example of *GASA* crossover. The first gene was swapped between the parents, i.e. the amplifier in the first gene from parent 1 was swapped with the sentiment in the first gene from parent 2. The result of the crossover is observed in the children chromosomes. All of the other genes remain unchanged.

6 EXPERIMENTAL SETUP

This section describes the experimental set up which was used in order to evaluate the performance of *GASA*. *GASA* was programmed in Java and the experiments were conducted at the University of Geneva on the Baobab cluster.

6.1 Data sets

Based on the literature surveyed, there is no consistency in terms of the number of data sets used in previous studies. Furthermore, the total number of reviews also varies from one study to another. For example, Che et al. [6] used a data set containing only 878 reviews, whereas the data set used in the study of Wang et al. [32] had 108891 reviews — the number of reviews largely differ between these two. Pang et al. [24] and Govindarajan [10] used 2000 reviews, whereas Acampora and Cosma [2] used 95084. Paramesha and Ravishankar [25] used 2243. Carvalho et al. [5] used two data sets which had 359 and 1908 reviews. There is no study which guides researchers to a set of recommended benchmark data sets for sentiment analysis.

For this study, eight data sets were constructed from several Amazon data sets. The Amazon data sets were obtained from Leskovec and Krevl [14] and McAuley et al. [18]. Each instance in the Amazon data sets is made up of a short summary and a review which were provided by a user. An example of an instance is as follows:

- **summary:** Pittsburgh - Home of the OLDIES
- **review:** I have all of the doo wop DVD's and this one is as good or better than the 1st ones. Remember once these performers are gone, we'll never get to see them again. Rhino did an excellent job and if you like or love doo wop and Rock n Roll you'll LOVE this DVD !

Four Amazon data sets were randomly selected, namely: Cell Phones and Accessories (Cellphone), Office Products (Office Prod),

Grocery and Gourmet Food (Foods) and Video Games. From these four large data sets, eight data sets (four summary and four review data sets) were created for this study. The summary data sets were created by randomly selecting 1000 positive and 1000 negative summaries from a problem domain. Similarly, the review data sets were created by randomly selecting 1000 positive and 1000 negative reviews from a problem domain. For example, the created Cellphone review data set had reviews selected from the Amazon Cellphone data set only. Similarly, the created Foods summary data set had summaries selected from the Amazon Foods data set only.

These data sets were created since they represent different problem domains and contain a similar number of instances as compared to that presented in [22, 24] and allow for a large number of experiments to be performed. The eight data sets used in this study are listed in table 1. Stop words and other irrelevant words were not removed from the data. For these to have no contribution to the overall sentiment, GASA must classify them as either 'sentiment' or 'amplifier' with a value of 0.

Table 1: Data sets used in this study. Four review and four summary data sets were created. Each data set had 2000 instances.

Data set	Number of positive/ negative instances
Cellphone	1000/1000 reviews
Office Prod	1000/1000 reviews
Foods	1000/1000 reviews
Video Games	1000/1000 reviews
Cellphone	1000/1000 summaries
Office Prod	1000/1000 summaries
Foods	1000/1000 summaries
Video Games	1000/1000 summaries

6.2 Experimental parameters

The parameters used for the *GASA* experiments are presented in table 2. These parameters were obtained by preliminary runs. The following section presents the results obtained by *GASA* on several experiments.

Table 2: *GASA* parameters. These were obtained from preliminary runs.

GASA Parameter	Value
Population Size	200
Parent selection method	Tournament
Tournament size	7
Maximum number of generations	500
Crossover rate	60%
Mutation rate	40%

7 RESULTS AND DISCUSSION

This section is split into three subsections. The purpose of the algorithm discussed the first subsection was to determine whether a given word was a sentiment or an amplifier word. The second

subsection describes experiments whereby the goal was to determine the value of the sentiment word. And finally, the third subsection presents the results when *GASA* was compared to other sentiment analysis algorithms. The results from the first two experiments demonstrate *GASA*'s ability to generate a sentiment lexicon whereas the third experiment illustrates how *GASA* performs as a sentiment analysis algorithm.

The amplifier dictionary was seeded with two words: "not" and "never"; each had a value of "-1", representing negation. The sentiment dictionary was obtained from [11] which we refer to as "HuLiu-6786". This dictionary contains 6786 known sentiment words labelled as either positive or negative.

7.1 Predicting 'sentiment' or 'amplifier'

Several experiments were conducted whereby the classification problem was converted into a 2-class problem, namely sentiment and amplifier classes. Each word in the HuLiu-6786 dictionary was considered as a sentiment word. Ten-fold cross-validation was used on the HuLiu-6786 dictionary whereby during each experiment, 9 folds from the dictionary were used for training, and the remaining fold from the dictionary was used for testing. *GASA* had to predict whether each word in the test fold was a sentiment or an amplifier. The training and testing data was made up of dictionary words and not reviews or summary data.

GASA's fitness function did not take into consideration (during the evolutionary process) whether or not a training dictionary word was correctly classified as a sentiment or not. Thus, during the evolutionary process for these experiments, *GASA* did not directly optimize the chromosomes in order to correctly distinguish between the two classes. Instead, *GASA*'s goal was to correctly classify the overall sentiment of as many instances (reviews or summary data) as possible.

For these experiments, two data sets were created from the Cellphone, Office Products, Foods and Video Games data sets. Namely, all of the summary data combined and all of the review data combined. Thus, the two combined data sets had 8000 instances each.

The results for these experiments are presented in tables 3 and 4. Each row in the table represents a particular word frequency condition; this is followed by the corresponding average number of dictionary words in the training data which met the word frequency condition and the average test accuracy across the 10-folds. Note that this test accuracy is in terms of the 2-class problem of distinguishing between a sentiment or amplifier word as described above.

The word frequency condition is read as follows: a value of '> 10' means that the experiment only took into consideration the training dictionary words which occurred at least 10 times in the review/summary data. In table 3, there were an average of 446 training dictionary words which had a frequency of 10. Similarly, a value of '> 250' means that the experiment only took into consideration the training dictionary words which occurred at least 250 times in the review/summary data. There were an average of 23 words in the training data which had a frequency of 250. The condition '> 0' implied that a dictionary word had to occur at least once in the review/summary data.

The purpose of using the word frequency condition was to determine the effect on the number of times a word was present in the training data and *GASA*'s ability to correctly classify the words in the test set which also had such a frequency.

Table 3: Test accuracy (%) results on the two class problem (sentiment and amplifier) on all of the review data combined into a single data set. Ten-fold cross-validation was used.

Word Frequency	Total number of dictionary words	Accuracy (%)
> 0	1917	51.39
> 10	446	54.80
> 15	334	55.47
> 20	260	56.66
> 100	56	64.52
> 250	23	72.00

Table 4: Test accuracy (%) results on the two class problem (sentiment and amplifier) on all of the summary data combined into a single data set. Ten-fold cross-validation was used.

Word Frequency	Total number of dictionary words	Accuracy (%)
> 0	626	55.11
> 10	76	68.42
> 15	56	67.86
> 20	41	75.61

When all of the words are taken into consideration, i.e. a frequency value '> 0' , *GASA* achieved an accuracy of 51.39% and 55.11% on the review and summary data respectively. The accuracy improved when the word frequency condition was increased. In terms of the review data, the accuracy went from 51.39% to 72.00% when the word frequency was increased from 'greater than 0' to 'greater than 250'.

For the combined summary data, when the words had a frequency of at least 20 the accuracy was 75.61% as opposed to an accuracy of 55.11% for a frequency condition greater than 0. The combined review data set had more words than the combined summary data set because the summaries are short text. For this reason, the conditions were stopped at 20 for the combined summary data. Words from the dictionary which occur with a small frequency are more challenging for *GASA* to correctly classify as a sentiment or amplifier since they occur infrequently in the data. Nonetheless, the findings reveal that *GASA* is able to extend a sentiment and amplifier lexicon provided that the words occur with a large frequency in the training data.

7.2 Predicting the value of the sentiment

A set of experiments was conducted in order to determine how effective *GASA* would be at classifying the sentiment value of a set of words instead of sentences. In order to achieve this, the HuLiu-6786 dictionary was used, and a certain percentage of the words

in the dictionary were considered as unknown. The problem was converted into a 2-class classification problem, namely positive and negative sentiment values. Thus, in terms of the classification-value pair, only the "value" aspect was taken into consideration.

The HuLiu-6786 dictionary contained more sentiment words than were present in the data sets, and thus only the dictionary words found in the training data sets were considered — this set was named S.

Ten-fold cross-validation was used in the following manner: 10 folds were randomly created from S, 9 folds were seeded into GASA and the algorithm was executed as defined in section 5. At the end of the GA generational loop the algorithm had to predict the sentiment value of the words in the test fold as either "positive" or "negative". The predictions were then compared against the correct values in order to determine the accuracy. Similarly to the experiment described in subsection 7.1, GASA did not directly optimize the chromosomes in order to correctly distinguish between the two classes. GASA's objective was to correctly classify the overall sentiment of as many instances (reviews or summary data) as possible.

Table 5: Test accuracy (%) results on the two class problem (positive and negative sentiment) on all of the review data combined into a single data set. Ten-fold cross-validation was used.

Word Frequency	Total number of dictionary words	Accuracy (%)
> 0	1917	36.62
> 10	446	44.39
> 15	334	46.71
> 20	260	49.23
> 100	56	69.64
> 250	23	82.61

Table 6: Test accuracy (%) results on the two class problem (positive and negative sentiment) on all of the summary data combined into a single data set. Ten-fold cross-validation was used.

Word Frequency	Total number of dictionary words	Accuracy (%)
> 0	626	53.04
> 10	76	86.84
> 15	56	92.86
> 20	41	95.12

For these experiments, the same data sets which were described in subsection 7.1 were used. The results for these experiments are presented in tables 5 and 6. Subsection 7.1 describes how to interpret the tables. Dictionary words in the combined summary data set which had a frequency value of at least 20 resulted in an accuracy of 95.12%. Sentiment words had to appear a greater number of times in the combined review data set in order to achieve a higher accuracy; more precisely, words which had a frequency of at least 250 times

resulted in an accuracy of 82.61%. These results reveal that GASA is able to extend a sentiment lexicon provided that the words occur frequently.

7.3 Comparison of GASA with commercial Sentiment Tools

How good is GASA? To check we compared GASA seeded with the HuLiu-6786 dictionary to other sentiment analysis methods including AlchemyAPI [12], MeaningCloud [19], NLTK [21], Lexalytics [15], LingPipe [3], Stanford sentiment analysis [17, 28], SentiStrength [30, 31] and Dandelion API [29]. AlchemyAPI, MeaningCloud, Lexalytics and Dandelion are commercial APIs. LingPipe and SentiStrength have both commercial and non-commercial licences. The results of the comparison are presented in table 7.

In terms of the summary data, the top 3 ranking methods in order of performance were LingPipe, AlchemyAPI and GASA with an average test accuracy of 77.75%, 72.88% and 69.92% respectively. When comparing GASA to the four commercial APIs, AlchemyAPI achieved the best accuracy, while GASA outperformed Dandelion, Lexalytics and MeaningCloud.

In terms of the review data sets, the top three performing methods were LingPipe, AlchemyAPI and SentiStrength. GASA was outperformed by two commercial API, namely AlchemyAPI and Dandelion. GASA achieved higher test accuracy when compared to two commercial APIs, namely, Lexalytics and MeaningCloud.

Appendices A and B illustrates several examples of the review data used. The predicted sentiment on the sample reviews from GASA and other sentiment analysis methods is presented. The reviews were randomly selected in order to illustrate cases where GASA correctly and incorrectly classified the sentiment.

8 EXTENDING GASA (CA-GASA)

When determining the classification for an unknown word w, the GASA algorithm does not take into consideration the words before and after w, i.e. it is context independent. This ignores the fact that many words have different meanings — with different sentiments. How can we begin to allow for multiple, context-dependent sentiments? We propose to allocate a context-dependent classification to an unknown word w; an approach we call Context Aware GASA (CA-GASA). In order to achieve this, several modifications to GASA are required. The primary modification lies within the representation of the chromosomes.

Each gene contains two principle parts, the context classification and the context-free classification. When a word in an instance of data is evaluated the classification-value pair is obtained from either the context classification or context-free classification. Two lists of words are used in order to make this decision, namely $list_{next}$ and $list_{previous}$. When a CA-GASA chromosome is evaluated on an instance of data i on a word w, the context classification-value pair is allocated if the word w is surrounded by the words in $list_{next}$ and $list_{previous}$. If this is not the case, then the context-free classification-value pair is allocated. This process is further discussed below. Figure 4 illustrates an example of a CA-GASA chromosome.

Table 7: Test accuracy (%) illustrating a comparison between other commercial and non commercial sentiment analysis methods and GASA. The 70/30 holdout method was used, and all of the data sets had a size of 2000. Types "S" and "R" denote summary and review data sets respectively. Watson refers to Alchemy API, MC to MeaningCloud, LEX to Lexalytics, LP to LingPipe, SS to SentiStrength, DL to Dandelion and SD to Stanford sentiment analysis.

Data set	Type	Watson	MC	NLTK	LEX	LP	SS	DL	SD	GASA
Cellphone	S	75.67	60.00	60.67	37.50	80.17	68.00	55.17	54.83	74.33
Office Prod	S	71.50	59.83	59.83	39.83	80.00	69.83	53.50	55.83	70.67
Foods	S	68.83	52.67	57.67	40.00	74.50	64.50	50.17	51.00	63.33
Video Games	S	75.50	59.83	57.67	38.33	76.33	66.17	55.00	51.33	71.33
Average Summary		*72.88*	*58.08*	*58.96*	*38.92*	*77.75*	*67.13*	*53.46*	*53.25*	*69.92*
Cellphone	R	70.17	62.33	64.17	58.50	81.50	67.67	65.00	54.17	69.17
Office Prod	R	71.50	64.83	64.83	60.17	81.50	72.00	68.33	53.67	68.67
Foods	R	71.17	62.83	63.83	64.00	78.83	70.67	69.67	52.83	64.83
Video Games	R	69.17	62.00	71.83	53.00	79.17	68.17	67.00	58.17	66.00
Average Review		*70.50*	*63.00*	*66.17*	*58.92*	*80.25*	*69.63*	*67.50*	*54.71*	*67.17*

In order to enable multiple classification-value pairs to be associated with a word, a new gene encoding is used. For a word w, each gene has the following properties:

- The word w.
- The context rule which is defined as follows:
  - The maximum possible size of the next context words, denoted as $next_{size}$.
  - The maximum possible size of the previous context words, denoted as $previous_{size}$.
  - The list of the next context words, denoted as $list_{next}$.
  - The list of the previous context words, denoted as $list_{previous}$.
  - The number of words to look ahead and compare with $list_{next}$, denoted as $number_{ahead}$.
  - The number of words to look behind and compare with $list_{previous}$, denoted as $number_{behind}$.
  - The context classification-value pair.
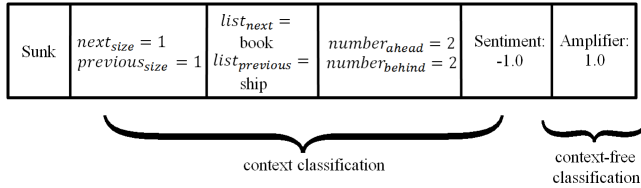- The context-free classification-value pair.

Figure 4: Example of a *CA-GASA* chromosome.

8.1 CA-GASA chromosome evaluation

Assume that a *CA-GASA* chromosome c is being applied to an instance of data. A word w in the sentence is evaluated as follows. Starting from the word w, look at the next $number_{ahead}$ words from w and add them to the $list_x$. Once again, starting from the word w, look at the previous $number_{behind}$ words from w and add them to the $list_y$.

Let the size of $list_x$ be denoted as $size_x$, and let the size of $list_y$ be denoted as $size_y$. Let the number of words in the intersection between $list_x$ and $list_{next}$ be denoted by a, and let the number of words in the intersection between $list_y$ and $list_{previous}$ be denoted by b.

If $\frac{a+b}{size_x+size_y} \geq 0.5$, then the word w is classified by the context classification value. Otherwise, the word w is classified by the context-free classification value.

The *CA-GASA* chromosome in figure 4 has one next context word, "ship", and one previous context word, "book". Assume the sentence "the ship sunk" is being evaluated, and the classification-value for the word "sunk" is being determined. In this case $list_x$ is empty, and $list_y$ = {ship, the}. Consequently, $size_x = 0$ and $size_y = 2$. The intersection between $list_x$ and $list_{next}$ is empty, and the intersection between $list_y$ and $list_{previous}$ is {ship}, and thus, $a = 0$ and $b = 1$. The word "sunk" is classified by the context classification-value since $\frac{0+1}{0+2} \geq 0.5$, i.e. "sunk" is classified as a sentiment word with a value of -1.0 (since the context classification-value pair in the figure is a sentiment with value of -1.0).

8.2 CA-GASA Results

From table 8, when HuLiu-6786 was seeded into the proposed methods, it is observed that *GASA* outperformed *CA-GASA* on 2 summary data sets, and tied in the other data sets, whereas *CA-GASA* outperformed *GASA* on 3 of the review data sets. One drawback of *CA-GASA* is that the search space is significantly larger than *GASA* and as a result the training time is much longer. Given this drawback, *CA-GASA* was not tested against the other sentiment analysis algorithms.

In order to address the large training time, it would be of interest to determine if an approach could be proposed in order to find which words in some data set have more than one meaning, and to create the context classification for those words only. This would reduce the complexity of *CA-GASA* whilst retaining the ability to perform word disambiguation. Unknown words which only express one sentiment regardless of the context could be represented using *GASA*, and words which have more than one sentiment could use the *CA-GASA* representation.

Table 8: Test accuracy (%) on the summary and review data showing a comparison between GASA and CA-GASA. Ten-fold cross-validation was used, and all of the data sets had a size of 2000. Both GASA and CA-GASA were seeded with the HuLiu-6786 sentiment dictionary.

	Summary Data		Review Data	
Data set	**GASA**	**CA-GASA**	**GASA**	**CA-GASA**
Cellphone	73.55	71.75	69.10	70.05
Office Prod	71.75	70.70	71.00	68.20
Foods	65.75	65.75	67.15	67.35
Video Games	69.40	69.40	65.45	67.00

9   CONCLUSION

Being able to determine the sentiment of text is a useful ability to businesses and other entities in order to gain an understanding of people's opinions on their products and services. This study proposed a GA approach in order to classify the sentiment of sentences by optimizing unknown words as either a sentiment or an amplifier word. This study proposes a way to represent the sentiments and amplifiers through the use of simple mathematical expressions in order to evaluate the final sentiment of sentences. The experimental results revealed that *GASA* was able to outperform certain commercial APIs.

One advantage of *GASA* is that the algorithm can grow a sentiment dictionary which can be reused or further improved upon. The experiments suggested that if a particular word appeared a large number of times in the training data set, then the proposed method is likely to correctly classify its sentiment.

We also proposed *CA-GASA*, and the rationale behind this modification was to provide the ability to allocate a sentiment based on the context for which the words are in. This method requires additional work in order to reduce the complex search space. It would also be of interest in order to investigate if an ensemble of *GASA* chromosomes could outperform the accuracy of a single one.

A   CORRECTLY CLASSIFIED BY *GASA*

This section presents a random sample of the reviews which were correctly classified by *GASA*. The reviews were obtained from the Video Game data set. The reviews were lemmatized using Stanford CoreNLP [17]. The sentence "we thought this movie was quite entertaining" is lemmatized as follows: "we think this movie be quite entertaining". The results are compared to the following four sentiment analysis entities: NLTK, Alchemy API, SentiStrength and MeaningCloud. The reviews are as follows:

Text: "*have purchase the original when release several year ago, and thoroughly enjoy it, I be excite to see a new version out for 2006. it live up to the challenge I expect from this game. it be difficult to find truly challenging puzzle game, and you will not be disappoint with this. my only disappoint come after solve it as the original provide a video of the programmer, although this version do offer the opportunity to replay and select from several final outcome.*"

Result: Correctly classified by *GASA* as positive.

Text: "*have not finish it yet, but it sure be a lot of fun! yes, there be/will be even better game. and yes, maybe GTA V be better (it have at least three time more budget so no surprise there). but Watch Dogs be truly a great game! some have unrealistically high expectation or be real hater and should just stick to buy gta every five year instead of buy each game and then give they bad review over and over... before write another obvious review: yes, WE all KNOW that "gta5" be NOT "watch dog" "call of duty" be NOT "candy crush" and "Far Cry 3" be NOT "Tetris""*"

Result: Correctly classified by *GASA* as positive.

Text: "*I recommend this game. very little communication loss or problem. if problem occur they be very good as about replace what you lose. the game be fun to play.*"

Result: Correctly classified by *GASA* as positive.

Text: "*I buy this game because I remember how Madden use to be. I have hear the gameplay be break (it be, horribly), but I think, hey, if I get use to the stupid game glitch at least it will look good. no. wow be I wrong. everything about this game be atrocious . the only thing that look crisp in this game be the score, which incidentally be the only reason I know it be display in hd. every time a player score a touchdown he literally run through the stand and disappear into the mesh. I could go on, but just think that I pay more than $ 5 for this game be make I extremely angry right no.*"

Result: Correctly classified by *GASA* as negative.

Text: "*I play game on a desktop pc in my office, and on a laptop in the living room. I also frequently rebuild my machine and upgrade hardware. I be interested in the online community that will follow this game, however I will not support such strict measure in copy protection. I have cancel my pre-order and may purchase later once this limited activation process be remove.*"

Result: Correctly classified by *GASA* as negative.

B   INCORRECTLY CLASSIFIED BY *GASA*

This section presents a random sample of the reviews which were incorrectly classified by *GASA*. Refer to appendix A for details regarding the data, lemmatization and the algorithms used for comparison. The reviews are as follows:

Text: "*I like final Fantasy 13 but it just do not draw I in that well and the story be slightly confusing. with that be say I absolutely love final Fantasy 13-2! the story be weird because of time travel, but I think it be awesome. they seem to have fix everything about the first one. I also like the fact that this game have multiple ending and new game plus! buy it!*"

Result: The correct classification for the review is positive. *GASA* classified it as negative. Three of the algorithms correctly classified the text as positive, and one algorithm classified it as negative.

Text: "*outstanding shooter game. it be set in world war two and you better not run out in the open. good graphic and story.*"

Result: The correct classification for the review is positive. *GASA* classified it as negative. Only one algorithm correctly classified the text. Two algorithms classified it as neutral and the remaining algorithm classified it as negative.

Text: "*they sit a little bit loose on the controller's peg, so there be some extra play. not shabby, not stellar. put some more life into a controller that you think you be go to lose.*"

Result: The correct classification for the review is positive. *GASA* classified it as negative. All of the algorithms incorrectly classified the text as negataive.

Text: "*this headset rock, that be all you need to know. unless you need more, here be some pro's and con's. pro's 1. affordable price2. long cable3. lightweight design4. adjustable microphone5. Chat Boost / Independent Game and Chat Sound6. stereo ExpanderCon's 1. slightly complicated set - up2. cheap ear Cushions The hiss sound people complain about be negligible, it disappear after just a few minute of use and be otherwise drown out by game. amazing quality and durability at a affordable price, highly recommend. UPDATE : 2/18/11 the microphone serve I very well for only 17 day before the Right Earphone cease to function. a day later, the right earphone completely fall off. I get off recommend the product initially, but the fact that the headset can stop work so quickly be terrible. I be return these for a refund and be go to try another headset. I no longer recommend these headphone.*"

Result: The correct classification for the review is negative. *GASA* classified it as positive. Three of the algorithms correctly classified the text as negative whereas the other classified it as neutral.

Text:"*Brunswick pro Bowling be not worth much. game be slow, unnatural, and poor scripting. I can not believe someone can not make a better program then the Wii bowling which be do pretty good. do not waste you money. I want sport game that actually feel and act like real sport. how hard can that be?*"

Result: The correct classification for the review is negative. *GASA* classified it as positive. Three of the algorithms correctly classified the text as negative whereas the other classified it as positive.

ACKNOWLEDGMENTS

REFERENCES

[1] Ahmed Abbasi, Hsinchun Chen, and Arab Salem. 2008. Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. *ACM Transactions on Information Systems* 26, 3, Article 12 (June 2008), 34 pages.
[2] G. Acampora and G. Cosma. 2014. A hybrid computational intelligence approach for efficiently evaluating customer sentiments in E-commerce reviews. In *Intelligent Agents (IA), 2014 IEEE Symposium on.* 73–80.
[3] Alias-i. 2008. LingPipe 4.1.0. http://alias-i.com/lingpipe. (2008).
[4] Tobias Blickle and Lothar Thiele. 1996. A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evolutionary Computation* 4, 4 (Dec. 1996), 361–394.
[5] Jonnathan Carvalho, Adriana Prado, and Alexandre Plastino. 2014. A Statistical and Evolutionary Approach to Sentiment Analysis. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 02 (WI-IAT '14).* IEEE Computer Society, Washington, DC, USA, 110–117.
[6] Wanxiang Che, Yanyan Zhao, Honglei Guo, Zhong Su, and Ting Liu. 2015. Sentence Compression for Aspect-Based Sentiment Analysis. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on* 23, 12 (Dec 2015), 2111–2124.

[7] A Das and S Bandyopadhyay. 2010. Subjectivity Detection using Genetic Algorithm. In *1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA10).*
[8] Agoston E. Eiben and J. E. Smith. 2003. *Introduction to Evolutionary Computing.* SpringerVerlag.
[9] David E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning* (1st ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
[10] M Govindarajan. 2013. Sentiment Analysis of Movie Reviews using Hybrid Method of Naive Bayes and Genetic Algorithm. *International Journal of Advanced Computer Research* 3, 4 (2013).
[11] Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04).* ACM, New York, NY, USA, 168–177.
[12] International Business Machines Watson. 2005. AlchemyAPI Service. http://www.alchemyapi.com/. (2005).
[13] P. Kalaivani and K. L. Shunmuganathan. 2015. Feature Reduction Based on Genetic Algorithm and Hybrid Model for Opinion Mining. *Scientific Programming* 2015, Article 12 (Jan. 2015), 1 pages.
[14] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. (June 2014).
[15] Lexalytics. 2003. https://www.lexalytics.com/. (2003).
[16] Bing Liu. 2015. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions.* Cambridge University Press.
[17] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations.* 55–60.
[18] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring Networks of Substitutable and Complementary Products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15).* ACM, New York, NY, USA, 785–794.
[19] MeaningCloud. 2015. MeaningCloud is a brand by MeaningCloud LLC, a s|ngular company. https://www.meaningcloud.com/. (2015).
[20] Cataldo Musto, Giovanni Semeraro, and Marco Polignano. 2014. A Comparison of Lexicon-based Approaches for Sentiment Analysis of Microblog Posts. In *Proceedings of the 8th International Workshop on Information Filtering and Retrieval co-located with XIII AI*IA Symposium on Artificial Intelligence (AI*IA 2014).* 59–68.
[21] NLTK. 2010. http://text-processing.com/. (2010).
[22] Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.*
[23] Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* 2, 1-2 (Jan. 2008), 1–135.
[24] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10 (EMNLP '02).* Association for Computational Linguistics, Stroudsburg, PA, USA, 79–86.
[25] K Paramesha and C Ravishankar, K. 2013. Optimization of Cross Domain Sentiment Analysis Using Sentiwordnet. *International Journal in Foundations of Computer Science & Technology* 3, 5 (2013).
[26] Kumar Ravi and Vadlamani Ravi. 2015. A Survey on Opinion Mining and Sentiment Analysis. *Knowledge-Based Systems* 89, C (Nov. 2015), 14–46.
[27] P.W.H. Smith. 2010. Using genetic algorithms in word-vector optimisation. In *Computational Intelligence (UKCI), 2010 UK Workshop on.* 1–5.
[28] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP).* Citeseer, 1631–1642.
[29] SpazioDati S.r.l. 2013. Dandelion API. http://dandelion.eu/. (2013).
[30] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment Strength Detection for the Social Web. *Journal of the Association for Information Science and Technology* 63, 1 (Jan. 2012), 163–173.
[31] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment in Short Strength Detection Informal Text. *Journal of the Association for Information Science and Technology* 61, 12 (Dec. 2010), 2544–2558.
[32] Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10).* ACM, New York, NY, USA, 783–792.
[33] Jinghui Zhong, Xiaomin Hu, Jun Zhang, and Min Gu. 2005. Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06),* Vol. 2. 1115–1121.