Emotion mining from text for actionable recommendations detailed survey

Jaishree Ranganathan* and
Angelina A. Tzacheva

Department of Computer Science,
University of North Carolina at Charlotte,
Charlotte, NC, USA
Email: jrangan1@uncc.edu
Email: aatzache@uncc.edu
*Corresponding author

Abstract: In the era of Web 2.0, people express their opinion, feelings and thoughts about topics including political and cultural events, natural disasters, products and services, through mediums such as blogs, forums, and micro-blogs, like Twitter. Also, large amount of text is generated through e-mail which contains the writer's feeling or opinion; for instance, customer care service e-mail. The texts generated through such platforms are a rich source of data which can be mined in order to gain useful information about user opinion or feeling which in turn can be utilised in specific applications such as: marketing, sale predictions, political surveys, health care, student-faculty culture, e-learning platforms, and social networks. This process of identifying and extracting information about the attitude of a speaker or writer about a topic, polarity, or emotion in a document is called sentiment analysis. There are variety of sources for extracting sentiment such as speech, music, facial expression. Due to the rich source of information available in the form of text data, this paper focuses on sentiment analysis and emotion mining from text, as well as discovering actionable patterns. The actionable patterns may suggest ways to alter the user's sentiment or emotion to a more positive or desirable state.

Keywords: actionable pattern mining; data mining; text mining, sentiment analysis.

Reference to this paper should be made as follows: Ranganathan, J. and Tzacheva, A.A. (xxxx) 'Emotion mining from text for actionable recommendations detailed survey', *Int. J. Data Mining, Modelling and Management*, Vol. x, No. x, pp xxx–xxx.

Biographical notes: Jaishree Ranganathan is a PhD student in Computer Science Department at The University of North Carolina, Charlotte. She received her MS in Computer Science in 2017. Her research interests include data mining, text mining, knowledge discovery in databases, natural language processing, machine learning, and social media mining.

Angelina A. Tzacheva is currently a Teaching Associate Professor at The University of North Carolina, Charlotte. Her research interests include data mining, knowledge discovery in databases, distributed knowledge systems, medical imaging, multimedia databases, and bioinformatics.

This paper is a revised and expanded version of a paper entitled [title] presented at [name, location and date of conference].

1 Introduction

According to Merriam-Webster Dictionary (2002), *sentiment* is an attitude, thought, or judgment prompted by feeling. Sentiment is a broad term which encompasses opinion based on an emotional state or reaction. Sentiment Analysis is the process of identifying opinion, emotion, and sentiment expressed by human about products, topic, public and political events, etc. There is wide variety of sources available for extracting sentiment expressed by human including *non-verbal communication*, i.e., facial expression, *speech*, *music* a tool that evokes emotion, *body expression* or *movements*, *images or photos* and *writing* or *text*. Sentiment mining in each of the area is studied extensively. For instance, emotion from music in studies by Kim et al. (2010b) and Yang and Chen (2012) shows state of art reviews, Yang and Chen (2011) and Yang et al. (2008) applied machine learning and ranking-based methods for music emotion recognition. Similarly El Ayadi et al. (2011) provide a survey of speech emotion classification, Zeng et al. (2009) focus on approaches that can handle audio and/or visual recordings of spontaneous displays of affective states, Kleinsmith and Bianchi-Berthouze (2013) provides a literature review on affective body expression and recognition, Cohn and Katz (1998) develop semi-automated methods of discriminating emotion and para-linguistic communication in face and voice (De Silva and Ng, 2000; Devillers et al., 2003; Nguyen et al., 2005; Torao et al., 1997) detect emotion from speech. Kisilevich et al. (2013) propose unsupervised approach combining linguistic features, traditional statistical models and lexicon to identify opinion or sentiment from photo comments. Majority of the research on sentiment analysis is focused on mining sentiment, opinion, and emotion from text. This focus on text can be attributed towards the growth of textual data from sources like e-mail, customer reviews, blogs, micro-blogs like Twitter, etc.

Text mining is a set of methods used to analyse unstructured data and discover patterns that were unknown beforehand (Sinoara et al., 2017). Text mining techniques are important for gleaning emotion from text. Sinoara et al. (2017) provide a systematic mapping about semantics-concerned text mining.

Origin and etymology of emotion dates back to 1579, it is known to have originated from the French word ‘emouvoir’ which means ‘to stir up’. According to Dixon (2003), ‘emotion’ is introduced into academic discussion to replace ‘passion’. Emotion is one of the aspects of our lives that influences day-to-day activities including social behaviour, friendship, family, work, and many others. Emotion mining has its root in many disciplines apart from computer science as follows: human science, psychiatry, nursing, psychology, neuro-science, linguistics, social science, anthropology, communication science, economics, criminology, political-science, philosophy, etc. Emotion mining gained attraction in the field of computer science due to the vast variety of systems that can be developed and promising applications.

According to Yadollahi et al. (2017), following are some of the applications related to emotion mining systems: In customer care services, emotion mining can help marketers gain information about how much satisfied their customers are and what aspects of their service should be improved or revised to consequently make a strong relationship with their end users by Gupta et al. (2013). User's emotions can additionally be used for sale predictions of a particular product. In e-learning applications, the intelligent tutoring system can decide on teaching materials, based on user's feelings and mental state. In human computer interaction, the computer can monitor user's emotions to suggest suitable music or movies says (Voeffray, 2011). Having the technology of identifying emotions enables new textual access approaches such as allowing user's to filter results of a search by emotion. In addition, output of an emotion-mining system can serve as input to other systems. For instance, Rangel and Rosso (2016) use the emotions detected in the text for author profiling, specifically identifying the writer's age and gender. Last but not least, psychologists can infer patients' emotions and predict their state of mind accordingly. On a longer period of time, they are able to detect if a patient is facing depression or stress (De Choudhury et al., 2013) or even thinks about committing suicide, which is extremely useful, since he/she can be referred to counseling services (Luyckx et al., 2012). Emotion Mining can be applied to social network data, such as Twitter, to assess the sentiment and the overall emotion of Tweets, as well as to analyse events (Ranganathan et al., 2017).

The knowledge discovery or data mining is the process of finding the interesting patterns and application of the discovered pattern to specific area of interest. Emotion mining is a knowledge discovery process. Techniques like classification, association and clustering are applied in order to find the interesting pattern from the data. In the era of internet and world wide web (WWW) the amount of data generated is huge which in turn is a major reason for generating large amount of knowledge. It is important that the discovered knowledge or pattern is used for real-life applications including business, medicine, education, military, in a meaningful and useful manner. Emotion mining can be applicable to e-learning applications, student teacher evaluation, smart appliances, psychology, etc. One of the major problem in the knowledge discovery process is reducing the volume of discovered patterns and selecting the appropriate interestingness measure (Dardzinska, 2012). There are two aspects of interestingness of rules in the literature (Liu et al., 1997; Silberschatz and Tuzhilin, 1995; Dardzinska, 2012): objective and subjective measures. Objective measures are data-driven and domain independent. On the other hand, subjective measures which includes unexpectedness, novelty and actionability is user-driven and domain-dependent. Patterns are actionable if the user can perform action using the patterns to his/her advantage. Action rules proposed by Ras and Wieczorkowska (2000) are special types of rules which forms a hint to the users, show a way to reclassify objects with respect to some distinguished attribute called the decision attribute.

This paper provides a comprehensive survey of research works on sentiment analysis especially focused on gleaning human emotion from text. Also, the paper gives idea about specific techniques, tools, resources used in the field of text mining. Finally, this paper aims to review actionable pattern discovery methods, as a means to analyse emotions and provide actionable recommendations to users of emotion mining systems. The actionable recommendations can be applied to: improving customer care services, e-learning applications, human-computer interaction systems, and medical psychology systems.

2 Emotion models and theories

Theorist have classified emotion into two major categories. The first category states that emotions are discrete and according to second category emotion is characterised as dimensional. Discrete emotion theory Figure 1 states that specific core emotions are sub served by independent neural system, on the other hand dimensional model Figure 2 states that all affective states or emotion arise from cognitive interpretations of core neural sensations (Posner et al., 2005).

Figure 1 Discrete model (see online version for colours)

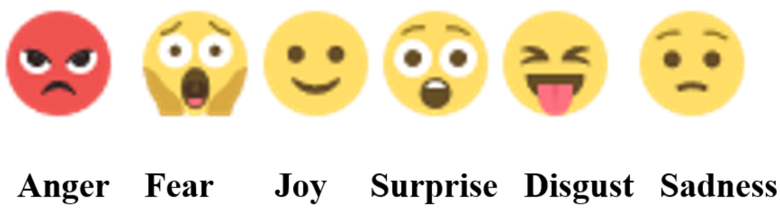
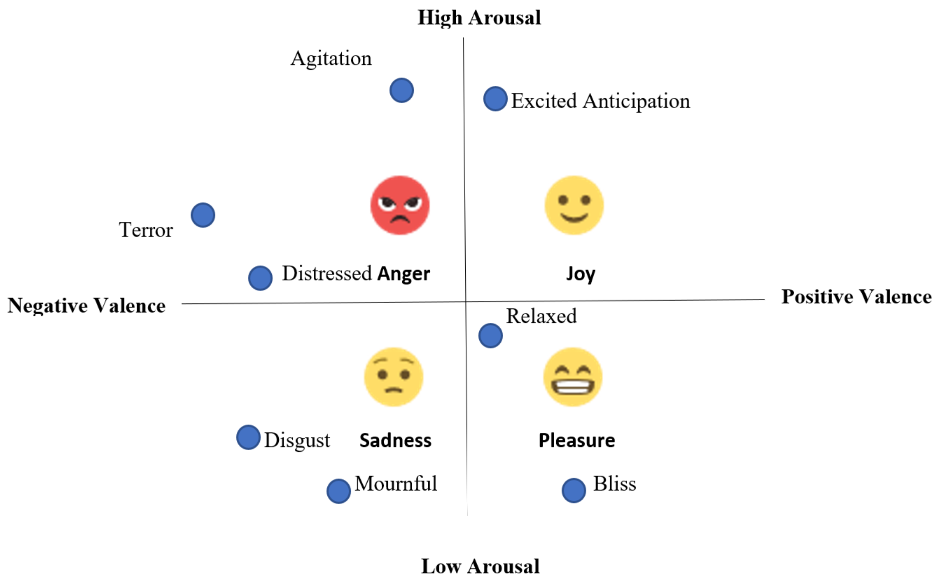


Figure 2 Dimensional model (see online version for colours)



Ekman (1992) characterises emotion into six basic forms as sadness, disgust, enjoyment, anger, fear, and surprise. Plutchik and Kellerman (2013) agreed with Ekman's biologically driven perspective but developed the wheel of emotions on bipolar axes: joy versus sadness, anger versus fear, trust versus disgust and surprise versus anticipation. Shaver et al. (1987) model a hierarchical tree structure for the basic emotions love, joy, surprise, anger, sadness, and fear and the leaves of the tree contain further categorisation for each of these six basic emotions. Lövheim (2012) present a three-dimensional model for monoamine neurotransmitters and emotions. In this model, the monoamine systems are represented as orthogonal axes and the eight basic emotions, labelled according to Tomkins (1962, 1963), are placed at each of the eight possible extreme values, represented as corners of cube.

There are many dimensional models for emotion, following are the widely accepted models as suggested by Rubin and Talarico (2009): circumplex model, vector model, and positive activation – negative activation model.

According to Russell (1980), circumplex model suggests that emotions are distributed in a two-dimensional space, containing arousal and valence dimensions. Bradley et al. (1992), propose a two-dimensional model in which the base dimension is arousal and that the valence determines the direction in which the emotion lies. Watson and Tellegen (1985), develop the positive activation – negative activation model in which they suggest that positive affect and negative affect are two separate systems.

3 Emotion mining researches

This section provides a review of existing research works with respect to emotion classification from text. It is divided into two subsections where, the first section elaborates existing methods on general text classification of emotion, the second section details studies with respect to social media data specifically Twitter data.

3.1 Text-emotion mining

Neviarouskaya et al. (2011) propose a rule-based approach for recognising affective communication in text messages. They use the following emotional states: ‘anger’, ‘disgust’, ‘fear’, ‘guilt’, ‘interest’, ‘joy’, ‘sadness’, ‘distress’, ‘shame’, and ‘surprise’; and communicative functions including: ‘greeting’, ‘thanks’, ‘posing a question’, ‘congratulation’, and ‘farewell’. In this work Neviarouskaya et al. build a special affect database including emoticons, acronym’s, abbreviations, adjectives, nouns, verbs, adverbs, words representing communicative functions and interjections with MySQL 5.0. Human annotators manually label the affect database with emotion categories and intensity values. Their affect analysis model consists of five stages each with manually created rules: symbolic cue analysis, syntactical structure analysis, word-level analysis, phrase-level analysis, and sentence-level analysis. Their system has certain limitations like dependency on the database, failure to disambiguate word meanings and process expression modifiers.

Ho and Cao (2012) use high-order hidden Markov model (HMM) for emotion detection from ISEAR dataset. The idea is to transform the input text into a sequence of events that cause mental states. Then automatically construct HMM based on the training dataset and generate the model to process the sequence of states that cause emotion. By cross-validation their model shows promising results.

Mishne et al. (2012) classify writer’s mood in blog text collected from LiveJournal a free weblog service using Yahoo API. To ensure the proper balance of training data across all moods, they select blog posts containing one of 40 top occurring moods in the entire corpus. They contribute a significant part of the work towards feature selection. Some classic features like frequency counts [words, part-of-speech (POS)], and length of blog post; subjective nature of blogs like semantic orientation, point-wise mutual information (PMI) which is a measure of the degree of association between two terms; features unique to online text like emphasised words, special symbols including punctuation's, and emoticons were used for training the SVMlight model from support vector machine (SVM) package. They attribute subjective nature of the corpus 'annotation' and nature of blog posts as major factors for low accuracy.

Strapparava and Mihalcea (2008) implement five systems for emotion analysis for news headlines using knowledge-based and corpus-based approaches. They evaluate the systems on the dataset of 1,000 newspaper headlines from SemEval 2007 by conducting fine-grained and coarse-grained evaluations. Results show that each of the systems have specific strength and they compare the results with three baseline systems in SEMEVAL emotion annotation task: SWAT (Katz et al., 2007), UPAR7 (Chaumartin, 2007) and UA (Kozareva et al., 2007). UPAR7 obtains best results in terms of fine-grained evaluations whereas the developed system using WordNet-Affect gives best performance in terms of coarse-grained evaluation with highest recall and F-measure.

Gupta et al. (2013) present a method for identifying emotional customer care emails using 'Boostexter' (Schapire and Singer, 2000) classifier which is based on boosting family of algorithms. Many 'weak' moderately accurate base classifiers combined to build a highly accurate classifier in boosting. They also extract salient features from emotional emails which reflect customer frustration, dissatisfaction with the business, threats to leave or take legal action and/or report to authorities. Their results show that the 'Boostexter' system with salient features resulted in a 20% absolute F-measure improvement compared to the baseline system using word ngrams.

Danisman and Alpkocak (2008) develop text classifier using vector space model (VSM) where each document is a vector and terms correspond to dimensions. The basic hypothesis in using VSM for classification is the contiguity hypothesis where documents in the same class form a contiguous space whereas regions of different class do not overlap. The weighting scheme term frequency-inverse document frequency (TF-IDF) is used to calculate each term weight values. They have analysed the effect of emotional intensity and stemming to the classification performance. Results show that VSM performs equally well compared to other well-known classifiers Naive Bayes, SVM and ConceptNet. In addition to the classification model they also developed emotion enabled video player which shows emotional state of the video based on the subtitles.

Hancock et al. (2007) contribute research towards identifying emotions during text-based communication. They conducted experiments with eighty undergraduate students in 40 same sex dyads. The results suggest that irrespective of gender, agreement with the conversation partner was more with positive affect expressers. Also, the following strategies are used to differentiate between positive and negative emotional states: frequency of disagreement, punctuation, negative affect terms, amount of words used. Negation and exclamation points are the two major linguistic cues that helped predict the textual emotion. These findings support the 'Social information processing theory' (Walther, 2008) and provide reasonable insights on how to automatically extract emotions in a text-based communication.

Kim et al. (2010a) evaluate categorical model and dimensional model for four affective states anger, fear, joy, and sadness. They use the following three emotional datasets with sentence-level emotion annotations: SemEval 2007 (Strapparava and Mihalcea, 2007), international survey on emotion antecedents and reactions (ISEAR) (Scherer and Wallbott, 1994), fairy tales (Alm, 2008) and apply VSM with dimensionality reduction variants (latent semantic analysis, probabilistic latent semantic analysis, non-negative matrix factorisation) and dimensional model in MATLAB. Though their experiments show categorical non-negative matrix factorisation and dimensional model have better performances, it is also inferred that either of the techniques perform well on generalised dataset.

Kao et al. (2009) provide a comprehensive survey on the existing research methods (earlier 2009) for emotion detection from text, identify their limitations and propose an integrated system to improve the emotion detection capabilities of the existing systems. They classify textual emotion detection into keyword-based, learning-based and hybrid (combination of keyword, learning methods and other components). According to them, keyword-based method lack the use of linguistic information to detect emotions, learning-based methods still need keywords in the form of features, hybrid methods though they outperform the previous two approaches still limited with the category of emotions. Based on the above studies they propose an integrated architecture that includes semantic analysis, ontology design of emotion models and adopting case-based learning approach.

Chaumartin (2007) proposes a rule-based linguistic system UPAR7 to detect the emotion and valence of news headlines (SemEval 2007 dataset). This system uses statistical analyser (Stanford Parser) to tag word dependencies. Further they use their own enriched version of lexical resources like WordNet, WordNet-Affect, SentiWordNet. The system tries to find the main subject of the news title by using the dependency graph, contrasts, and accentuation's. The rule-based system identifies emotions with 89.43% accuracy and valence with 55% accuracy. However, the recall is low. The difference between the ac-curacies of emotion and valence is due to the fact that it is easier to detect emotions of individual words rather than valence which needs a global understanding of the sentence.

Jain and Kulkarni (2014) provide a short survey of existing methods in textual emotion detection like Kao et al. but their method lacks to show the actual works. They develop a statistical model, i.e., VSM for automatic emotion detection from text using a comprehensive dataset created from ISEAR, WordNet-Affect and WPARD datasets. Their model uses bag-of-words approach because of which it lacks the ability to consider the semantic and syntactic information from the text. Also, they do not provide statistical results to the developed model. But they have handled the negation using 'not' by first finding the emotion of the sentence without considering negation and after that generate the reverse of the resulting emotion.

Lei et al. (2014) present a lexicon-based approach towards social emotion detection. They designed a new algorithm for document selection which has positive effect on the performance of social emotion detection systems. After document selection they exploit words and POS features. POS helps alleviate the problems of emotional ambiguity of words and the context dependence of the sentiment orientations. Finally generate the emotion lexicon based on the features. They gathered 40,897 news articles assigned with ratings over eight social emotions including touching, empathy, boredom, anger, amusement, sadness, surprise, and warmness. Their method outperforms the baseline methods of SWAT (Katz et al., 2007), emotion-topic model (ETM), and emotion-term model (ET) (Bao et al., 2009, 2012).

Mishne et al. (2006) determine the aggregate mood levels across large number of blog postings, i.e., classify the blog posts into one of forty most frequent moods. They estimate the moods of complete blogs by identifying textual features (discriminating terms) and use these features in the learning models to predict the mood intensity in each time slot. Discriminating terms are collected by applying log likelihood measure to quantify the divergence between term frequencies across different corpora. Because of the nature of the dataset they did not achieve good results in the case studies performed.

3.2 Twitter – emotion mining

Wang et al. (2012) automatically create large emotion-labelled dataset by collecting tweets using Twitter streaming API which contain emotion hash-tags. According to Merriam-Webster Dictionary hash-tag is a word or phrase preceded by the symbol # that classifies or categorises the accompanying text (such as a tweet). Their source of the emotion words is psychology paper by Shaver et al. (1987). The list of basic emotional words in [3] were expanded by including lexical variants, e.g., 'surprising' and 'surprised' for 'surprise'. By using this approach, they collected 5 million tweets and further applied certain filtering heuristics as follows: retain only tweets with emotion hash-tags at the end, discard tweets having less than five words, remove tweets with URLs or quotations. After which they had a collection of 2,488,982 tweets. Features like N-gram, adjectives, N-gram position, POS, sentiment/emotion lexicons were explored, and their results show that combination of n-gram, sentiment/emotion lexicons, POS yields higher accuracy with both the machine learning classifiers close to 60%. Also, they validate the effectiveness of larger training dataset by creating sequence of training dataset with increasing size and observe dataset size is directly proportional to accuracy.

Mohammad (2012) developed corpus from Twitter posts using emotion hash-tags like Wang et al. (2012), Hasan et al. (2014) and Roberts et al. (2012) called as Twitter emotion corpus (TEC) consisting of 21,000 tweets. SVMs with sequential minimal optimisation (SMO) classifier was used with uni-gram and bi-gram features. The automatic classifiers obtained an F-score much higher than the random baseline (SemEval – 2007, 1,000 headlines dataset). Similar to Wang et al. (2012), in this paper best results are achieved with higher number of training instances. For example, Joy-NotJoy classifier get the best results compared to Sadness-NotSadness. He also performed experiments to show the effectiveness of cross-domain classification by using the TEC corpus for classifying the newspaper headlines domain.

Hasan et al. (2014) evaluated the use of hashtags like Wang et al. (2012) to automatically label Twitter messages with corresponding emotion tags. They used circumplex model of human affect for defining the emotional states. In this work Hasan et al. validate and confirm that hash-tags are reliable features for automatic emotion labelling. To prove that hash-tags are reliable sources for automatic emotion labelling they conducted two sets of experiment one with novices and other with psychology experts and validated using Fleiss-Kappa results. In this work they also identified that human labelling of emotion using crowd sourcing is inconsistent and unreliable whereas expert labelling gives 87% accuracy with hash-tag labels. The system 'Emotex' was developed to classify Twitter messages achieved 90% accuracy.

Roberts et al. (2012) create emotion corpus from micro-blogging service Twitter. The corpus contains seven emotions annotated across 14 topics including Valentine's day, World Cup 2010, Stock Market, Christmas, etc. The emotions are based on Ekman (1992) six basic emotions and 'love'. The topics of each tweet are obtained by considering the tweet to be associated with a probabilistic mixture of topics using latent dirichlet allocation (LDA) topic modelling technique. The system uses a series of binary SVM classifiers to detect each of the seven emotions annotated in the corpus. Each classifier performs independently on a single emotion, resulting in seven separate binary classifiers implemented using the software available from WEKA. Each classifier uses specific set of features like punctuation, hypernyms, n-grams, and topics. According to the results 'fear' is the best performing emotion and also suggests that this emotion is highly lexicalised with less variation than other emotions, as it has comparable recall but significantly higher precision.

Bollen et al. (2011) analyse the relationship between public mood patterns and social, economic, and other major events in media and popular culture over a time by using sentiment analysis on tweets extracted from micro-blogging platform Twitter. They use profile of mood states (POMS), a psychometric questionnaire composed of 793 adjective terms including synonyms and related word constructs. It is proved that POMS serve as a valid alternative to machine learning. They calculate aggregate of the mood vector for all tweets of a day. Their results show that social, political, cultural, and economical events have significant effect on public mood. The effect of economic events on public mood is equivalent to the degree of public response to rapid changes of economic indicators magnified by media.

Purver and Battersby (2012) used Twitter data labelled with emoticons and hash-tags to train supervised classifiers. They used SVMs with linear kernel and uni-gram features for classification. Their method had better performance for emotions like happiness, sadness, and anger but not good in case of other emotions like fear, surprise, and disgust. They achieved accuracy in the range of 60%.

4 Lexicons

According to Merriam-Webster Dictionary (2002) lexicon is a vocabulary of a language, an individual speaker or group of speakers or a subject. In text mining these lexicons are a useful resource in particular for tasks like sentiment analysis. This section covers some of the widely used lexicons in the sentiment analysis research works.

4.1 Affective norms for English words – ANEW

Bradley et al. (1999) provides a set of normative emotional rating for many English language words. The rating is in terms of pleasure, arousal, and dominance that complements the existing international affective picture system (IAPS) (Lang et al., 1997) and international affective digitised sounds (IADS) (Bradley and Lang, 2007) which are collections of picture and sound stimuli respectively. ANEW is developed by Center for Emotion and Attention (CSEA) at University of Florida. They used self-assessment manikin (SAM) an affective rating system (non-verbal pictorial assessment) to assess the three dimensions of pleasure, arousal, and dominance with students from introductory psychology class. One major advantage of ANEW is that it has been validated by several persons which makes it preferable for psycho-linguistic studies.

4.2 AFINN

Nielsen (2011) created AFINN lexicon, a list of English words including few phrases scored for valence in the range from –5 (very negative) to +5 (very positive) manually. The word list includes obscene words, few positive words, words from public domain 'Original balanced affective word list' by Greg Siegle, internet slang words from urban dictionary including acronyms, and recent additions from 'The compass DeRose guide to emotion words' by Steven J. DeRose. The initial version AFINN-96 includes 1,468 unique words, including few phrases. The latest version AFINN-111 has 2,477 unique words and phrases. Though AFINN has slightly better performance than ANEW, for studies in scientific psycholinguistics' ANEW is preferable as the scoring is validated across several persons.

4.3 Linguistic inquiry word count

Linguistic inquiry word count (LIWC) is an emotion lexicon (program + dictionary) that contains close to 5,000 words by Tausczik and Pennebaker (2010) to calculate the percentage of emotional word categories within a text. Initially they built a list of emotion word categories from dictionaries, thesauruses, questionnaires, and list made by research assistants. Then group of three judges, independently validated the word and emotion category for each word in the initial list. After validation by judges the word list is updated using set of rules as follows: word retained if two out of three judges agreed, word deleted if two of three judges agreed for the word to be excluded, word added to list if two of the three judges agreed. A separate group of judges again conduct the entire process. Initial judging happened between 1992 and 1994. Later a significant revision happened in 1997 and in 2007 to streamline the original program and dictionaries.

4.4 NRC emotion lexicon

Mohammad and Turney (2010, 2013a, 2013b) create the NRC emotion lexicon called 'EmoLex' with 14200 words. To generate the lexicon, they first collect list of words from following sources: Macquarie Thesaurus uni-grams and bi-grams that frequently occur in Google n-gram corpus, words with positive and negative semantic orientation from general inquirer, emotion words from WordNet-Affect lexicon. Turker's through Amazon's mechanical turk annotated the word list based on the emotion they evoke. They perform series of validations to make sure that the manual annotations are right using additional word choice questions for Turker's. This help them identify the annotators are not familiar with the word and ignore results from such annotators. They prove that annotations by crowd-sourcing are of high-quality by comparison with gold standards.

4.5 WordNet-Affect

WordNet-Affect lexicon is an emotional lexicon for affective knowledge. Strapparava et al. (2004), Valitutti et al. (2004) and Strapparava et al. (2006) develop this resource WordNet-Affect on top of WordNet (Miller, 1995) through selection and labelling of synsets representing affective concepts. They create an AFFECT database with 1,093 terms directly or indirectly referring to emotional states, started with adjectives and further extended by adding nouns, verbs, and adverbs. This affect database was then projected to the WordNet-Affect as an affective label. WordNet-Affect contains 2,874 synsets and 4,787 words. An important property of this affective lexicon concerning mainly adjectival interpretation is the stative/causative dimension. An emotional adjective is said causative if it refers to some emotion that is caused by the entity represented by the modified noun (e.g., amusing movie). In a similar way, an emotional adjective is said stative if it refers to the emotion owned or felt by the subject denoted by the modified noun (e.g., cheerful/happy boy). Jindal and Taneja (2017) use WordNet which is the base for WordNet-Affect lexicon for multi label categorisation of text documents.

4.6 Clean balanced emotional tweet

Shahraki and Zaıane (2017) develop a dataset called clean balanced emotional tweet (CBET) especially for Twitter data. They use hash-tags to collect 208,544 general-purpose tweets. After pre-processing they select 3,000 sample tweets for each emotion category and create a dataset with 27,000 samples called the CBET dataset. The lexicon is $V \times E$ matrix, where the word at index (j, i) denotes the degree that the word w_j express the emotion e_i. Each word w_j has a corresponding weight vector calculated as the number of times the word w_j has occurred in tweets that have label e_i as emotion. This CBET is a new lexicon that overcomes some of the drawbacks of previous emotion lexicons.

5 Features for classification

Features are important aspects for any kind of classification task. It is necessary to make good selection of features for achieving higher classification accuracy. There are wide variety of approaches available for feature selection including but limited to information gain, chi-squared, document frequency, sampling-based methods. Seetha et al. (2015) propose a hybrid approach combining Zipf's law-based feature selection and the use of linear SVM weight for feature ranking. Dasgupta et al. (2007) propose an unsupervised feature selection strategey using sampling for regularised least squares classification algorithm.They show that this method performs better compared to other commonly used feature selection methods. This section explains some of the widely used features that are applied for text mining methods.

5.1 N-grams

Word N-grams are extensively in use for machine learning models such as SVM, Naive Bayes and many more text classification algorithms. N-grams are set of co-occurring words, where N = 1, 2, 3, etc. When N = 1, this is referred to as uni-grams, N = 2 is referred as bi-grams and N = 3 is referred as tri-grams.

For instance, consider the sentence 'this is supervised machine learning', The following are the tri-grams: 'this is supervised', 'is supervised machine', and 'supervised machine learning'. The number of N-grams for a sentence is calculated as in equation (1).

$$N - grams_Z = Z - (N - 1) \tag{1}$$

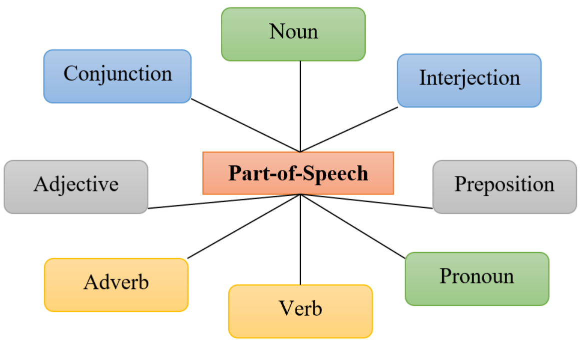
Z is number of words in sentence

N is N-gram.

5.2 Part-of-speech

Based on the use and function, words in English language is categorised into specific types called POS. The following are the major POS in English: noun, pronoun, verb, adverb, adjective, conjunction, preposition, interjection as in Figure 3. For instance, Benamara et al. (2007), study the effect of adjectives and adverbs in the process of sentiment analysis.

Figure 3 Part-of-speech (see online version for colours)



5.3 Frequency count

Most of the text classification systems use word frequencies in the document, 'bag-of-words' model as the feature (Sebastiani, 2002; Mishne et al., 2012). Apart from the word or term frequency (Mishne et al., 2012) used frequency of POS tags, frequency of word lemmas as features for text classification.

5.4 Symbols – emoticons

According to Mishne et al. (2012), emoticons are sequences of printable characters which are intended to represent human emotions or attitudes; often these are sideways textual representation of facial expressions. Some examples are shown in Table 1. Emoticons can be used for sentiment analysis, and emotion detection.

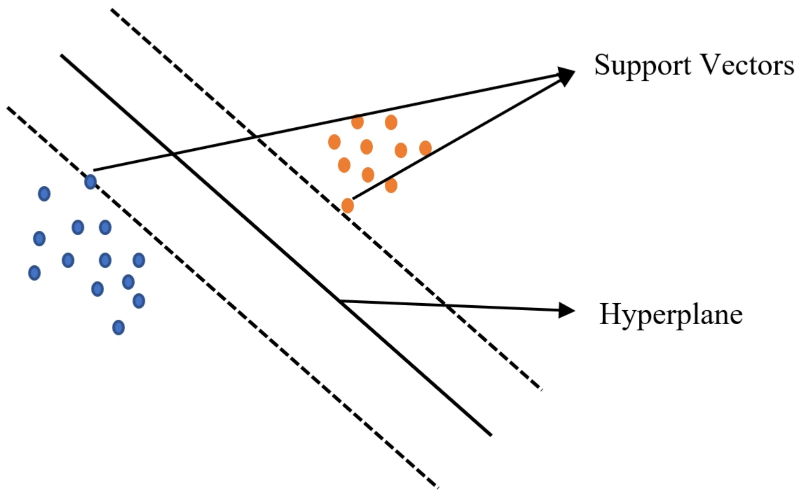
Table 1 Example emoticons (see online version for colours)

Emotion	*Emoticon*	*Emojis*
Happy or smile face	:)	😊
Sad	:(	☹
Neutral	:|	😐
Wink	:)	😉

6 Machine learning methods used for emotion classification

Khan et al. (2010) provide a review of common machine learning algorithms for text classification. Document classification can be divided into three categories based on the available methods as follows: supervised, un-supervised, semi-supervised methods. The automatic classification of documents into predefined categories has observed as an active attention, as the internet usage rate has quickly enlarged (Khan et al., 2010). Some of the common machine learning approaches are SVM, Bayesian classifier, decision tree, K-nearest neighbour (KNN), neural networks, latent semantic analysis, etc. Since this paper focus on text classification this section details specific classifiers widely used in the literature for text classification. In general, supervised learning techniques are used for automatic text classification. Here, pre-defined category labels are assigned to documents based on the likelihood suggested by a training set of labelled documents.

Figure 4 Support vector example (see online version for colours)



6.1 Support vector machines

SVM is a statistical learning method introduced by Vapnik (1999). Brücher et al. (2002), details SVM as follows: the main idea behind SVM is to find a decision surface that best separates the two class of documents in the n-dimensional space.

The samples (documents) that are close to decision surface are called *support vectors* shown in Figure 4 as shown by Meyer (2001). Major advantage of SVM is its superior runtime-behaviour during the categorisation of new documents because only one dot product per new document must be computed. A disadvantage is the fact that a document could be assigned to several categories because the similarity is typically calculated individually for each category. Nevertheless, SVM is a very powerful method and has outperformed other methods in several studies by Joachims (1998), Dumais et al. (1998), Yang and Liu (1999) and Siolas and d'Alché-Buc (2000). The following works use SVM for emotion classification from text (Mishne et al., 2012; Danisman and Alpkocak, 2008; Roberts et al., 2012; Purver and Battersby, 2012).

6.2 Naive Bayes classifier

Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes Theorem with strong independence assumptions Khan et al. (2010). Brücher et al. (2002), say that Independence assumption means the order of features is irrelevant and presence of one feature does not affect the presence of other features. The conditional assumption is given in equation (2).

$$p(x|y=c) = \prod_{i=1}^{D} p(x_i|y=c) \tag{2}$$

The computation of Bayes classifier is efficient because of this independence assumption and also limited applicability. Due to its apparently over-simplified assumptions, the naive bayes classifiers often work much better in many complex real-world situations (Khan et al., 2010). The full Bayesian posterior predictive density on the class label Y given an input X and the training data D is given by equation (3) as explained by Murphy (2006).

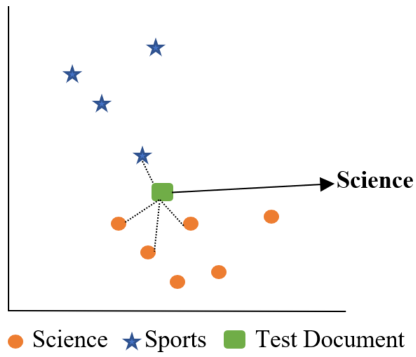
$$p(y=c|x,D) \approx p(y=c|x,\hat{\theta},\hat{\pi}) \propto p(x|y=c,\hat{\theta}_c)p(y=c|\hat{\pi}) \tag{3}$$

But the performance is relatively low compared to SVMs. One of the advantage of Naive Bayes classifier is that it requires only small set of training data to determine the classification instances. Naive Bayes is easy to implement compared to other algorithms, however because of conditional independence assumption it's performance is very poor when features are highly correlated and does not consider frequency of word occurrences. Wang et al. (2012) use Naive Bayes for Twitter emotion classification.

6.3 Vector space model

KNN is vector space classification method. Vector space classification method represents each document as a vector with one real-valued component, for instance TF-IDF (Danisman and Alpkocak, 2008) weight for each term. In general VSM is based on contiguity hypothesis. This hypothesis states that documents in the same class form a contiguous region and regions of different classes do not overlap (Manning et al., 2008). Given a test document, majority class (nearest neighbour) close to the test document is assigned as the class for test document. One of the advantage of K-NN is that, it does not require explicit training data. Because the training phase involved determining the value of 'k' and document pre-processing. KNN is also called as memory-based learning or instance-based learning because it simply memorises examples in the training set and then compares the test document to them. For example, if there are documents of type science and sports. Given a test document K-NN classifies it based on the majority number of classes that are closest to the test document. Consider the example in Figure 5. the test document is close to science document and hence classifies as '*science*'. Kim et al. (2010a) and Jain and Kulkarni (2014) use VSM for text emotion classification.

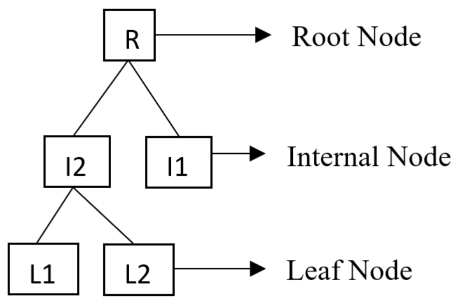
Figure 5 KNN example (see online version for colours)



6.4 Decision tree classifier

The classification problem is solved with the help of tree. The tree has root node, internal node and leaf node as shown in Figure 6. Here leaves represent the document category and branches represent features that lead to the specific category. The root node is the document for classification. According to Khan et al. (2010), main advantage of decision tree is its simplicity in understanding and interpreting, even for non-expert users.

Figure 6 Decision tree nodes



Text classification generally involves more number of features or attributes. Decision tree performs poorly with larger feature set. However, if the feature set is organised and limited according to the requirement then the performance of decision tree is an added advantage to the simplicity and understand-ability.

Table 2 gives a summary of models used in the emotion classification studies.

Table 2 Summary – research works

Author	*Data*	*Method*	*Evaluation*
Neviarouskaya et al. (2011)	Text message	Rule-based	Human annotator
Ho and Cao (2012)	ISEAR	Hidden-Markov model	Cross-validation
Mishne et al. (2012)	Blog	SVM	Accuracy
Strapparava and Mihalcea (2008)	News headlines	Knowledge and corpus-based	Precision, recall, F-measure
Gupta et al. (2013)	Customer care e-mail	Boostexter	Precision, recall, F-measure
Danisman and Alpkocak (2008)	ISEAR	SVM	Kappa, F-measure, accuracy
Hancock et al. (2007)	Text message	Survey	NA
Kim et al. (2010a)	SemEval2007, ISEAR, fairy tales	VSM	Precision, recall, F-measure
Chaumartin (2007)	News headlines	Rule-based lingusitic	Precision, recall, F-measure, accuracy
Jain and Kulkarni (2014)	ISEAR, WordNet-Affect, WPARD	VSM	NA
Lei et al. (2014)	News articles	Lexicon	
Mishne et al. (2006)	Blog	Pace regression module WEKA	Cross validation
Wang et al. (2012)	Twitter	LIBLINEAR, multinomial	Precision, recall,
	data – hashtag	Naive Bayes	F-measure
Mohammad (2012)	Twitter data – hashtag	SVM SMO	Precision, recall, F-measure
Hasan et al. (2014)	Twitter data – hashtag	Emotex	Accuracy
Roberts et al. (2012)	Twitter data	SVM	Precision, recall, F-measure
Bollen et al. (2011)	Twitter data	POMS	Timeline of events
Purver and Battersby (2012)	Twitter – emoticons/ hashtag	SVM linear kernel	Precision, recall, F-measure

7 Actionable pattern discovery

Actionability is a property of the discovered knowledge. Patterns are considered actionable if the user can act upon them, and if this action can benefit the user, or help them to accomplish their goals. Cao (2015) explore the paradigm shift of knowledge discovery from data to actionable knowledge discovery (AKD) and delivery. He observes macro-level (methodological and fundamental issues) and micro-level (technical and engineering issues) perspectives to narrow down the gaps between delivered knowledge and desired knowledge and states that AKD and delivery framework help narrow the gaps in Knowledge discovery process. The author suggests that use of domain knowledge in the data mining process and engaging organisational and social intelligence in the KDD modelling process help furthering the paradigm shift.

Barrett et al. (2011) extract actionable knowledge from data collected in schools that could be valuable to students, teachers, principals, district, state and national administrators. According to Greco et al. (2005) patterns discovered from data are represented in the form of '*if..., then...*' rules called decision rules. These patterns provide information about past events and utilised for prospective decisions. For instance, in medical diagnosis these rules can help identify the relationship between symptoms and sickness and also diagnose new patients based on these past records. Another prospective usefulness of decision rules is getting the desired effect on dependent variables by building strategy of intervention on the independent variables. In the medical example, this can be explained as modifying symptoms to get out from the sickness.

Action rules mining is a method to discover actionable patterns from large datasets. Action rules are rules that describe a possible transition of data from one state to another, or in other words, action rules reclassify data from one category to another (Bagavathi and Tzacheva, 2017).

Dardzinska (2012), summarise the frameworks for generating action rules from Kaur (2005) as follows: loosely coupled and tightly coupled. The loosely coupled framework is often called rule-based. It is based on pairing certain classification rules which have to be discovered first by using for instance algorithms such as LERS (Grzymala-Busse, 1997) or ERID (Dardzinska and Ras, 2005, 2003). The tightly coupled framework is often called object-based and it assumes that action rules are discovered directly from a database (Dardzinska and Ras, 2006; He et al., 2005; Im and Ras, 2008). Classical methods for discovering them follow algorithms either based on frequent sets (called action sets) and association rules mining (Agarwal et al., 1994) or they use algorithms such as LERS or ERID with atomic action sets used as their starting step. Action rules are one way to mine actionable knowledge from large dataset.

7.1 Action rules assumptions

According to Dardzinska (2012), action rules, introduced by Ras and Wieczorkowska (2000) may be utilised by any type of industry maintaining large databases, especially medical, military, education, and business. They are constructed from classification rules which suggest ways to re-classify objects, such as patients, students, or customers to a desired state. However, very often, such a change cannot be done directly to a chosen attribute. Therefore, in a natural way, there comes a need to learn definitions of such an attribute in terms of other attributes. The following are the definitions of action terms, action rules, and their standard interpretation.

Definition 7.1: Atomic action term, means an expression $(a, a_1 \to a_2)$, where a is an attribute, and $a_1, a_2 \in V_a$.

If $a_1 = a_2$ then a is called stable on a_1. In this case, for simplicity reason, we use notation (a, a_1) instead of $(a, a_1 \to a_2)$.

Definition 7.2: Set of action terms, mean the smallest set such that:

1 If t is an atomic action term, then t is an action term.

2 If t_1, t_2 are action terms, then $t_1 * t_2$ is an action term.

3 If t is an action term containing $(a, a_1 \to a_2)$, $(b, b_1 \to b_2)$ as its sub-terms, then $a \neq b$.

Definition 7.3: By the domain of an action term t, denoted by $Dom(t)$, we mean the set of all attribute names listed in t.

Definition 7.4: By an action rule we mean an expression $r = [t_1 \to t_2]$, where t_1 is an action term, and t_2 is an atomic action term.

Additionally, we assume, that $Dom(t_2) = \{d\}$ and $Dom(t_1) \subseteq A$, where A is a set of attributes.

The domain $Dom(r)$ of action rule r is defined as $Dom(t_1) \cup Dom(t_2)$.

Table 3 Information system S

X	*Attribute a*	*Attribute b*	*Attribute c*	*Attribute d*
x_1	a_1	b_1	c_1	H
x_2	a_2	b_2	c_1	H
x_3	a_2	b_1	c_1	A
x_4	a_1	b_1	c_2	A
x_5	a_2	b_1	c_2	A
x_6	a_2	b_2	c_2	H
x_7	a_1	b_1	c_2	A
x_8	a_1	b_2	c_1	A
x_9	a_1	b_1	c_1	H
x_{10}	a_2	b_2	c_1	H

Consider the information system S in Table 3. The following are examples of atomic action terms: $(a, a_2 \to a_2)$, $(b, b_2 \to b_1)$, $(c, c_2 \to c_2)$, $(c, c_3 \to c_3)$, $(d, H \to A)$. Also, consider the following expression in equations (4), (5), and (6). The values of a_2, c_2, c_3 of attributes a and c remain unchanged, while $(b, b_2 \to b_1)$ means that the value of attribute b is changed from b_2 to b_1.

$$(a, a_2 \to a_2) = (a, a_2) \tag{4}$$

$$(c, c_2 \to c_2) = (c, c_2) \tag{5}$$

$$(c, c_3 \to c_3) = (c, c_3) \tag{6}$$

Equations (7) and (8) are example action rules. According to equation (7) r_1 says that if the value a_2 remains unchanged and value b will change from b_2 to b_1 for a given object x, then it is expected that the value d will change from H to A for object x. Clearly, $Dom(r_1) = \{a, b, d\}$. In a similar way, the rule r_2 in equation (8) says that if the value c_2 remains unchanged and value b will change from b_2 to b_1, then it is expected that the value d will change from H to A, and $Dom(r_2) = \{b, c, d\}$.

$$r_1 = [((a, a_2 * (b, b_2 \rightarrow b_1)) \rightarrow (d, H \rightarrow A)] \tag{7}$$

$$r_2 = [((c, c_2 * (b, b_2 \rightarrow b_1)) \rightarrow (d, H \rightarrow A)] \tag{8}$$

Definition 7.5: Standard interpretation N_s of action terms in S = (X, A, V) is defined as follows:

1 If $(a, a_1 \rightarrow a_2)$ is an atomic term, then
$N_s((a, a_1 \rightarrow a_2)) = [\{x \in X : a(x) = a_1\}, \{x \in X : a(x) = a_2\}]$

2 If $t_1 = (a, a_1 \rightarrow a_2) * t$ and $N_s(t) = [Y_1, Y_2]$, then
$N_s(t_1) = [Y_1 \cap \{x \in X : a(x) = a_1\}, Y_2 \cap \{x \in X : a(x) = a_2\}]$.

Now we define $[Y_1, Y_2] \cap [Z_1, Z_2]$ as $[Y_1 \cap Z_1, Y_2 \cap Z_2]$.

Assume that $N_s(t_1) = [Y_1, Y_2]$ and $N_s(t_2) = [Z_1, Z_2]$.

Then $N_s(t_1 * t_2) = N_s(t_1) \cap N_s(t_2)$. Let $r = [t_1 \rightarrow t_2]$ be an action rule, where $N_s(t_1) = [Y_1, Y_2]$, $N_s(t_2) = [Z_1, Z_2]$.

Definition 7.6: By support and confidence of rule r we mean:

1 $sup(r) = min\{card(Y_1 \cap Z_1), card(Y_2 \cap Z_2)\}$

2 $conf(r) = \frac{card(Y_1 \cap Z_1)}{card(Y_1)} . \frac{card(Y_2 \cap Z_2)}{card(Y_2)}$ if $card(Y_1) \neq 0$, $card(Y_2) \neq 0$, $card(Y_1 \cap Z_1) \neq 0$, $card(Y_2 \cap Z_2) \neq 0$

3 $conf(r) = 0$ otherwise.

Now, let us consider equation (7) for support and confidence with example. For the rule r_1 we have:

- $N_s(a, a_2 \rightarrow a_2) = [\{x_2, x_3, x_5, x_6, x_{10}\}, \{x_2, x_3, x_5, x_6, x_{10}\}]$
- $N_s(b, b_2 \rightarrow b_1) = [\{x_2, x_6, x_8, x_{10}\}, \{x_1, x_3, x_4, x_5, x_7, x_9\}]$
- $N_s(a, a_2 \rightarrow a_2) * (b, b_2 \rightarrow b_1) = [\{x_2, x_6, x_{10}\}, \{x_3, x_5\}]$
- $N_s(d, H \rightarrow A) = [\{x_1, x_2, x_6, x_9, x_{10}\}, \{x_3, x_4, x_5, x_7, x_8\}]$.

Therefore, for rule r_1, support $sup(r_1) = 2$, confidence $conf(r_1) = \frac{3}{3} . \frac{2}{2} = 1$.

7.2 *Action rules from classification rules*

Dardzinska (2012) have summarised the action rules loosely coupled framework as follows: Finding useful rules is a very important and extremely interesting task of knowledge discovery in data. Most of the researchers focused on techniques for generating patterns, such as classification rules or association rules, from data sets. They assume that the user should analyse these patterns and infer actionable solutions for specific problems within given domains. The classical knowledge discovery algorithms have the potential to identify enormous number of significant patterns from data. Therefore, people are overwhelmed by a large number of uninteresting patterns which are very difficult to analyse and give time consuming solutions. So, there is still a need to look for new methods and tools with the ability to assist people in identifying rules with useful knowledge. There are two types of interestingness measure: subjective and objective (Adomavicius and Tuzhilin, 1997; Liu et al., 1997; Silberschatz and Tuzhilin, 1995). An objective measure is a data-driven approach for evaluating the quality of association patterns. It is domain-independent and requires minimal input from the users, other than to specify a threshold for filtering low-quality patterns (Tan et al., 2006). An objective measure is usually computed based on the frequency counts tabulated in a contingency table. Subjective interestingness measures include actionability (Adomavicius and Tuzhilin, 1997) and unexpectedness (Silberschatz and Tuzhilin, 1995). When a rule contradicts, surprises, or uncovers new knowledge, it is classified as unexpected. A rule is deemed actionable, if the user can take action to gain an advantage based on this rule. Domain experts basically look at a rule and say that this rule can be converted into an appropriate action.

E-action rules mining is a method which helps people in an intelligent and automatic way to acquire useful information from data. This information can be turned into actions. The approach gives suggestions about how to change certain attribute values of a given set of objects in order to reclassify them according to a user wish.

Two frameworks for mining actionable knowledge can be taken into consideration: rule-based and object-based (Kaur, 2005). In the object-based approach, action rules are extracted directly from a database (Dardzinska and Ras, 2006; He et al., 2005; Im and Ras, 2008), while in rule-based approach (Ras and Wieczorkowska, 2000), the extraction of actionable knowledge is a consequence of using classification rules discovery. It is further subdivided into: methods generating action rules from certain pairs of classification rules (Ras and Tzacheva, 2003; Ras et al., 2005; Tzacheva and Ras, 2005; Tsay and Ras, 2005), and methods generating action rules from single classification rules (Ras and Dardzinska, 2006). For example, algorithm ARAS proposed by Ras et al. (2005) generates sets of terms (built from values of attributes) around classification rules and constructs action rules directly from them.

In most of the algorithms for action rules mining, there is no guarantee that the discovered patterns in the first step will lead to actionable knowledge that is capable of maximising profits. One way to approach this problem is to assign a cost function to all changes of attribute values (Tzacheva and Ras, 2007). If changes of attribute values in the classification part of an action rule are too complex, then they can be replaced by composing such rule with other action rules, as proposed by Tzacheva and Ras (2005). Each composition of these rules uniquely defines a new action rule. Objects supporting each new action rule are the same as objects supporting the action rule replaced by it, but the cost of reclassifying them is lower for the new rule.

E-action rule forms the actionability concept in a better way than action rule (Ras and Wieczorkowska, 2000) by introducing a notion of its supporting class of objects. E-action rules are constructed from certain pairs of classification rules. They can be used not only for evaluating discovered patterns but also for reclassifying some objects in a dataset from one state into a new more desired one.

For example, classification rules found from a bakery's data can be very useful to describe who is good client (whom to offer some additional promotions) and who is bad client (whom to watch to minimise loses). However, if shop managers need to improve their understanding of customers and seek for specific actions to improve the services, mere classification rules are not sufficient. We suggest using classification rules for introducing a new method connected with action based on their condition features in order to get a desired effect on their decision feature. When we look the bakery example again, the strategy of action would consist of modifying some condition features in order to improve our understanding of customers behaviour and then improve the services.

E-action rules are useful in many other fields, including medical diagnosis. In medical diagnosis, e.g., in children flat foot problem, classification rules can explain the relationships between symptoms and sickness and help to predict the diagnosis of a new patient. E-action rules are useful in providing a hint to a doctor what symptoms have to be modified or eliminated in order to recover a certain group of patients with better prognoses in their illness.

7.2.1 System DEAR

Dardzinska (2012), summarises an algorithm called system DEAR. The aim of this algorithm is to move objects between classes by changing some flexible attributes. System DEAR identifies the optimal set of attributes to be changed, the new values, and calculates the support and confidence for each of the extended action rule.

Assume that there is a decision system with only one decision attribute, seen as method of treatment. Its domain contains values being integers. This decision attribute classifies objects (patients) with respect to the prognoses for patients. The cardinality of the image $d(X) = \{d_i : d(x) = d_i$ for some $x \in U\}$ is called the rank of attribute d and is denoted by $r(d)$.

Observe that the decision d determines the partition $CLASS_s(d) = \{X_1, X_2, X_{r(d)}\}$ of the set of objects X, where $X_k = d^{-1}(\{d_i\})$ for $1 \leq d_i \leq r_d$. $CLASS_s(d)$ is called the classification of objects in S determined by the decision d. As mentioned before, objects correspond to patients. Also, the patients in $d^{-1}(\{d_1\})$ are better prognoses for a hospital than patients in $d^{-1}(\{d_2\})$ for any $d_2 \leq d_1$. The set $d^{-1}(\{r(d)\})$ represents the patients with prognoses for complete recovery. Clearly the goal of any hospital or medical centres is to maximise the number of recovered patients. It can be achieved by shifting some patients from group $d^{-1}(\{d(2)\})$ to $d^{-1}(\{d_1\})$, for any $d_2 \leq d_1$. Namely, through special methods of treatment offered by medical centres, values of flexible attributes of some patients can be changed and the same all these patients can be moved from a group of worse prognoses ranking to a group of better prognoses.

Assume now that for any two collections of sets X, Y, we write, $X \subseteq Y$ if $(\forall x \in X)(\forall y \in Y)(x \subseteq y)$. Let $S = (X, A_{St} \cup A_{Fl} \cup \{d\})$ be a decision table and $B \subseteq A_{St} \cup A_{Fl}$. We say that attribute d depends on B if $CLASS_s(B) \subseteq CLASS_s(d)$, where $CLASS_s(B)$ is a partition of X generated by B (Pawlak, 2012).

Definition 7.7: Assume that attribute d depends on B where $B \in A_{St} \cup A_{Fl}$. The set B is called d-reduct in S if there is no proper subset C of B such that d depends on C.

The concept of d-reduct in S was introduced to induce rules from S describing values of the attribute d depending on minimal subsets of $A_{St} \cup A_{Fl}$ which preserve the confidence of extracted rules. In order to induce rules in which the **then** part consists of the decision attribute d and the **if** part consists of attributes belonging to $A_{St} \cup A_{Fl}$, sub-tables $(X, B \cup d)$ of S where B is a d-reduct in S should be used for rules extraction.

Let us consider the information system S in Table 4. The set a, c is the set of stable attributes, b is flexible attribute and d is the decision attribute. Also, Dardzinska (2012) assume that H denotes patients of good prognoses and L denotes patients of week prognoses.

Table 4 DEAR – information system S

X	*Attribute a*	*Attribute b*	*Attribute c*	*Attribute d*
x_1	a_0	b_3	c_0	L
x_2	a_0	b_2	c_1	L
x_3	a_0	b_3	c_0	L
x_4	a_0	b_2	c_1	L
x_5	a_2	b_1	c_2	L
x_6	a_2	b_1	c_2	L
x_7	a_2	b_3	c_2	H
x_8	a_2	b_3	c_2	H

$L(r)$ means all attributes listed in the conditional part of rule r. For instance, consider the rule below, then $L(r) = \{a, b\}$.

- $r = [(a, a_1) * (b, b_2) \to (d, H)]$.

$d(r)$ denote the decision value of a rule. In the above example $d(r) = H$. If r_i, r_j are rules and $B \subseteq A_{St} \cup AFl$ is set of attributes, then $r_i/B = r_j/B$ means that the conditional parts of rules r_i, r_j restricted to attributes B are the same. For instance consider the rules r_i, r_j below, then $r_i/(a, c) = r_j/(a, c)$.

- $r_i[[(a, a_1) * (b, b_0) * (c, c_3) \to (d, H)]$
- $r_j[[(a, a_1) * (b, b_2) * (c, c_3) \to (d, H)]$.

Based on below CLASS values, $CLASS(\{a, b\}) \subseteq CLASS_s(d)$ and $CLASS(\{b, c\}) \subseteq CLASS_s(d)$.

- $CLASS_s(\{d\}) = \{\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{x_7, x_8\}\}$
- $CLASS_s(\{a\}) = \{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7, x_8\}\}$
- $CLASS_s(\{b\}) = \{\{x_1, x_3, x_7, x_8\}, \{x_2, x_4\}, \{x_5, x_6\}\}$
- $CLASS_s(\{a, b\}) = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}\{, x_7, x_8\}$
- $CLASS_s(\{c\}) = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6, x_7, x_8\}\}$
- $CLASS_s(\{b, c\}) = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}\{, x_7, x_8\}\}$.

It can be easily checked that both $\{b, c\}$ and $\{a, b\}$ are d-reducts in S. Rules can be directly derived from d-reducts and the information system S. In this example, the following optimal rules are obtained:

- $(a, a_0) \to (d, L)$
- $(b, b_2) \to (d, L)$
- $(b, b_1) \to (d, L)$
- $(c, c_0) \to (d, L)$
- $(c, c_1) \to (d, L)$
- $(a, a_2) * (b, b_3) \to (d, H)$
- $(b, b_3) * (c, c_2) \to (d, H)$.

Now, let us assume that $(a, v \to w)$ denotes the fact that the value of attribute a has been changed from v to w. Similarly, the term $(a, v \to w)(x_i)$ means that $a_{x_i} = v$ has been changed to $a_{x_i} = w$. In other words, the symptom (a, v) of a patient x_i has changed to symptom (a, w).

Assume now that $S = (X, A_{St} \cup A_{Fl} \cup \{d\})$ is a decision table. Assume that rules r_1, r_2 have been extracted from S, B_1 is a maximal subset of A_{St} such that $r_1/B_1 = r_2/B_2$, $d(r_1) = d_1$, $d(r_2) = d_2$ and $d_1 \leq d_2$. Also, assume that $(b_1, b_2, ..., b_p)$ is a list of all attributes in $L(r_1) \cap L(r_2) \cap A_{Fl}$ on which r_1, r_2 differ and

- $r_1(b_1) = v_1$, $r_1(b_2) = v_2$, ..., $r_1(b_p) = v_p$
- $r_2(b_1) = w_1$, $r_2(b_2) = w_2$, ..., $r_2(b_p) = w_p$.

By (r_1, r_2) action rule on $x \in X$:

- $((b_1, v_1 \to w_1) * (b_2, v_2 \to w_2) * ... * (b_p, v_p \to w_p))(x) \to ((d, d_1 \to d_2))(x)$

If the value of rule on x is true then the rule is valid. Otherwise it is false. Let $X_{(r_1)}$ be the set of all patients in X supporting the rule r_1. If (r_1, r_2) action rule is valid on $x \in X_{(r_1)}$ then the action rule supports the new profit ranking d_2 for object x.

Consider the rules in Table 5. The rules representation is given in below:

- $r_1 = (a_1 * b_1 * c_1 * e_1 \to H)$
- $r_2 = (a_1 * b_2 * g_2 * h_2 \to L)$.

Table 5 DEAR – small part of information system S

a-stable	*b-flexible*	*c-stable*	*e-flexible*	*g-stable*	*h-flexible*	*d-decision*
a_1	b_1	c_1	e_1			H
a_1	b_2			g_2	h_2	L

Assume that object x_i supports rule r_1 which means that it is classified as H. In order to re-classify x_i to class L, it is required to change the following values b from b_1 to b_2, $g_{(x)} = g_2$ and h to h_2. This is the meaning of the extended action rule (r_1, r_2) given below:

- $(a_1 * (b, b_1 \to b_2) * c_1 * e_1 * (g, g_2) * (h, \to h_2))(x) \to (d, H \to L)(x)$.

Support and confidence of the extended r_1, r_2 action rule is given as follows:

- $sup(r_1, r_2) = card[(a, a_1) * (b, b_1) * (c, c_1) * (e, e_1) * (g, g_2) * (d, H)]$.
- $conf(r_1, r_2) = \frac{card((a,a_1)*(b,b_1)*(c,c_1)*(e,e_1)*(g,g_2)*(d,H))}{card((a,a_1)*(b,b_1)*(c,c_1)*(e,e_1)*(g,g_2))} \cdot \frac{card((a,a_1)*(b,b_2)*(c,c_1)*(g,g_2)*(h,h_2)*(d,L))}{card((a,a_1)*(b,b_2)*(c,c_1)*(g,g_2)*(h,h_2))}$

For any extended (r_1, r_2) action rule support and confidence can be defined in a similar way.

7.2.2 System DEAR2

Tsay and Ras (2005) propose system DEAR2, which is an action tree algorithm used for generating e-action rules. This algorithm finds the stable attribute with least number of values and using that attribute splits the set of rules recursively. Once the stable attributes are processed, the final subsets are split further based on decision attribute (Dardzinska, 2012). This method generates an action tree which is used to construct E-action rules from the leaf nodes of the same parent. The algorithm DEAR consists of two main steps as shown in Algorithm 1.

Algorithm 1 System DEAR2

1: **Build action-tree:** *Divide the rule table R, taking into consideration all stable attributes.*
2: a) *Find the domain $Dom(a)$ of each attribute $a \in A_{St}$ from the initial table R.*
3: b) *Partition the current table into sub-tables containing only rules supporting values of stable attributes in the corresponding sub-tables.*
4 *Build action-tree: Divide each lowest sub-table into new sub-tables containing rules with the same decision value.*
5 *Build action-tree: Represent each leaf as a set of rules which do not contradict on stable attributes and also define decision value d_i. The path from the root to that leaf gives the description of objects supported by these rules.*
6: **Generate action rules:** *Compare all unmarked leaf nodes of the same parent from action rules.*
7 *Generate action rules: Calculate the support and confidence for each rule. If both are above minimal thresholds, the rule is extracted and added to knowledge base.*

An action-tree has two types of nodes: a leaf node and non-leaf node. At a non-leaf node in the tree, the set of rules is partitioned along the branches and each child node gets its corresponding subset of rules. Leaf represent a set of rules, which define decision value d_i and do not contradict on stable attributes. The path from the root to that leaf gives the description of objects supported by these rules. This algorithm is explained with an example decision system S in Table 6.

Let $A_{St} = \{a, b\}$ be the stable attributes and $A_{Fl} = \{c, d\}$ be the flexible attributes. The goal of system DEAR2 is reclassify objects from class (d, H) to class (d, A). Table 7 shows the knowledge base of certain rules R extracted from Table 6. Sample certain rule for first row of Table 7 is given in equation (9).

$$(a, a_2) \rightarrow (d, A) \tag{9}$$

Table 6 DEAR2 – decision system S

X	*Attribute a*	*Attribute b*	*Attribute c*	*Decision d*
x_1	a_1	b_2	c_1	H
x_2	a_1	b_2	c_1	H
x_3	a_2	b_1	c_1	A
x_4	a_2	b_1	c_1	A
x_5	a_1	b_2	c_2	A
x_6	a_1	b_2	c_2	A
x_7	a_1	b_0	c_1	H
x_8	a_1	b_0	c_1	H
x_9	a_1	b_0	c_3	H
x_{10}	a_1	b_1	c_2	H
x_{11}	a_2	b_2	c_3	A
x_{12}	a_2	b_0	c_1	A

Table 7 Knowledge base T1

Set of objects	*Attribute a*	*Attribute b*	*Attribute c*	*Decision d*
x_3, x_4, x_{11}, x_{12}	a_2			A
x_1, x_2, x_7, x_8	a_1		c_1	H
x_7, x_8, x_9	a_1	b_0		H
x_3, x_4		b_1	c_1	A
x_5, x_6		b_2	c_2	A

The algorithm starts with Table 7. Attribute a is used to split the table into two sub-tables Tables 8 and 9 defined by values a_1, a_2. The attribute a is chosen because $card[V_a] < card[V_b]$.

Table 8 Knowledge base T2

Set of objects	*Attribute a*	*Attribute b*	*Attribute c*	*Decision d*
x_1, x_2, x_7, x_8	a_1		c_1	H
x_7, x_8, x_9	a_1	b_0		H
x_3, x_4		b_1	c_1	A
x_5, x_6		b_2	c_2	A

Table 9 Knowledge base T3

Set of objects	*Attribute a*	*Attribute b*	*Attribute c*	*Decision d*
x_3, x_4, x_{11}, x_{12}	a_2			A
x_3, x_4		b_1	c_1	A
x_5, x_6		b_2	c_2	A

It is evident that all objects in Table 9 have the same decision attribute (d, A), therefore no action rules can be generated and this table is no longer considered for action. Table 8 has another stable attribute b and contains different decision values, it is divided further into Tables 10, 11 and 12.

Table 10 Knowledge base T4

Set of objects	*Attribute a*	*Attribute b*	*Attribute c*	*Decision d*
x_1, x_2, x_7, x_8	a_1		c_1	H
x_7, x_8, x_9	a_1	b_0		H

Table 11 Knowledge base T5

Set of objects	*Attribute a*	*Attribute b*	*Attribute c*	*Decision d*
x_1, x_2, x_7, x_8	a_1		c_1	H
x_3, x_4		b_1	c_1	A

Table 12 Knowledge base T6

Set of objects	*Attribute a*	*Attribute b*	*Attribute c*	*Decision d*
x_1, x_2, x_7, x_8	a_1		c_1	H
x_5, x_6		b_2	c_2	A

There are no stable attributes at this point and Table 10 have same decision values, hence cannot be divided further. In Table 11 there is only one value of flexible attribute $c = c_1$, so this table cannot be partitioned. Table 12 is partitioned into two sub-tables, Tables 13 and 14.

Table 13 Knowledge base T7

Set of objects	*Attribute a*	*Attribute b*	*Attribute c*	*Decision d*
x_1, x_2, x_7, x_8	a_1		c_1	H

Table 14 Knowledge base T8

Set of objects	*Attribute a*	*Attribute b*	*Attribute c*	*Decision d*
x_5, x_6		b_2	c_2	A

Thus, the path from root to the leaf described by d gives the information of rules and objects supported by them. For example, the path $[a = a_1], [b = b_2]$ and $[d = H]$ leads to Table 13. Similarly, the path $[a = a_1], [b = b_2]$ and $[d = A]$ leads to Table 14. The action rule generated by comparing the pairs of rules in the above-mentioned tables is given in equation (10). Support of rule r is given by $sup(r) = min\{4, 2\} = 2$ and confidence is given by $conf(r) = 1.\frac{2}{3} = \frac{2}{3}$.

$$r = [[(a, a_1) * (c, c_1 \rightarrow c_2)] \rightarrow (d, H \rightarrow A)] \tag{10}$$

The action-tree algorithm proposed here requires the extraction of all classification rules from the decision system before any action rule is constructed and has $O(k^2)$ complexity in worst case, where k is the number of classification rules.

7.3 E-action rules – ARAS algorithm

Dardzinska (2012) have summarised the e-action rules proposed by Tsay et al. (2005) for automatic analysis of discovered classification rules. E-action rules hint how to re-classify some objects in a data set from one state into another more desired one. Let $S = (X, A_{St} \cup A_{Fl} \cup d)$ be a decision system, where $d \notin A_{St} \cup A_{Fl}$. To improve the efficiency of the algorithm, when the number of attributes is large, we can extract rules from sub-tables $(X, B \cup d) \subseteq S$, where B is a d – reduct of the system S.

Assume that r_i, r_j are the rules extracted from S. The idea of extended action rule (e-action rule) was given by Tsay et al. (2005) and extended by Ras et al. (2009). Formal definition of extended action rule is given below. We assume that:

1 $B_{St} \in A_{St}$ is maximal, such that $r_i/B_{St} = r_j/B_{St}$

2 $d(r_i) = d_i, d_{(r_j)} = d_j$ and $d_i \leq d_j$

3 $(\forall a \in [A_{St} \cap L(r_i) \cap L(r_j)])[a(r_i) = a(r_j)]$

4 $(\forall m \leq q)(\forall e_m \in [A_{St} \cap (L(r_j) \setminus L(r_i))])[e_m(r_j) = u_j]$

5 $(\forall m \leq r)(\forall c_m \in [A_{Fl} \cap (L(r_j) \setminus L(r_i))])[c_m(r_j) = t_j]$

6 $(\forall m \in p)(\forall b_m \in [A_{Fl} \cap L_(r_i) \cap L_(r_j)])([b_m(r_i) = v_m] * [b_m(r_j) = w_m])$.

Assume now that rules r_i, r_j are extracted from S and $r_i/A_{St} = r_j/A_{St}$, $d(r_i) = d_i$, $d(r_j) = d_j$ and $d_i \leq d_j$. We also have the assumption that $(b_1, b_2, ..., b_m)$ is list of all attributes in $Dom(r_i) \cap Dom(r_j) \cap A_{Fl}$ on which r_i and r_j differ and $r_i(b_1) = v_1, r_i(b_2) = v_2, ..., r_i(b_m) = v_m, r_j(b_1) = w_1, r_j(b_2) = w_2, ..., r_j(b_m) = w_m$.

Definition 7.8: By r_i, r_j action rule on $x \in X$, mean an expression of the form: $r = [(b_1, v_1 \to w_1) * (b_2, v_2 \to w_2) * ... * (b_m, v_m \to w_m)](x) \to (d, d_i \to d_j)(x)$.

Object $x \in X$ supports r_i, r_j extended action rule r in system $S = (X, A_{St} \cup A_{Fl} \cup \{d\})$ if the following conditions are satisfied:

1 $(\forall i \leq p)[b_i \in L(r)][b_i(x) = v_i]\Lambda d(x) = d_1$

2 $(\forall i \leq p)[b_i \in L(r)][b_i(y) = w_i]\Lambda d(y) = d_2$

3 $(\forall j \leq p)[a_j \in A_{St} \cap L(r_j)][a_j(x) = u_j]$

4 $(\forall j \leq p)[a_j \in A_{St} \cap L(r_j)][a_j(y) = u_j]$

5 objects x, y support rules r_1, r_2 respectively.

Definition 7.9: By the support of rule r, we mean the number of all objects in S satisfying the left side of the rule consisting of conditional parts of terms. $sup(r) = card[(b_1, v_1) * (b_2, v_2) * ... * (b_m, v_m) * (d, d_i)]$.

For computing the confidence of extended (r_1, r_2) action rule we divide the number of objects supporting (r_1, r_2) action rule by the number of objects supporting left hand side of this rule and multiply it by the confidence of the classification rule r_2. Values of stable attributes listed in r_1 do not have to be considered at all.

Definition 7.10: The confidence of extended (r_1, r_2) action rule is equal to $conf((r_1, r_2)) = \frac{sup(r)}{sup(L(r))}.conf(r_2)$.

Let us consider the example extended action rule r in form:

$$r = [(a, a_1) * (c, c_1 \rightarrow c_2) \rightarrow (d, H \rightarrow A)] \tag{11}$$

The support of this rule is $sup(r) = 4$, and the confidence is $conf(r) = \frac{4}{3}.\frac{2}{3} = \frac{2}{3} =$ 66%.

7.3.1 Action rules based on agglomerative strategy – ARAS algorithm

It is expensive to construct action rules from pairs of classification rules and also, there are chances of constructing their classification parts. Ras and Dardzinska (2006) show that single classification rules are sufficient to build action rules. In Dardzinska (2012), summarises a simple LERS-type algorithm for constructing action rules from single classification rule. LERS proposed by Grzymala-Busse (1997) is a classic example of bottom-up strategy which constructs rules with a conditional part of the length $k + 1$ after all rules with a conditional part of length k have been constructed. System ARAS assumes that LERS is used to extract classification rules. By using LERS as the pre-processing module for ARAS, the overall complexity of the algorithm is decreased. This algorithm was proposed in Ras et al. (2007).

Consider the information system S in Table 15. The set $A_{St} = \{a, b, c\}$ constitute the stable attributes, and $A_{Fl} = \{e, f, g\}$ constitute the flexible attributes of system S. Using system LERS (Grzymala-Busse, 1997) to extract classification rules. The goal is to reclassify object (d, A) to either $(d, I) = d_I$ or $(d, E) = d_E$.

Table 15 ARAS – information system S

X	*Attr. a*	*Attr. b*	*Attr. c*	*Attr. e*	*Attr. f*	*Attr. g*	*Decision d*
x_1	a_1	b_0	H	e_1	f_1	g_0	I
x_2	a_2	b_0	H	e_2	f_1	g_2	A
x_3	a_3	b_0	H	e_3	f_1	g_2	A
x_4	a_1	b_0	L	e_3	f_1	g_0	A
x_5	a_1	b_1	H	e_2	f_1	g_0	A
x_6	a_2	b_0	H	e_3	f_2	g_0	A
x_7	a_2	b_2	L	e_3	f_1	g_1	A
x_8	a_2	b_0	L	e_3	f_1	g_1	E

The following are the four certain classification rules extracted by using system LERS (Grzymala-Busse, 1997) on decision system S in Table 15.

- $r_1 = [(b_0 * c_H * f_1 * g_0) \to d_I]$
- $r_2 = [(a_2 * b_0 * e_3 * f_1) \to d_E]$
- $r_3 = [e_1 \to d_I]$
- $r_4 = [(b_0 * g_1) \to d_E]$.

Action rule schemas associated with r_1, r_2, r_3, r_4 and the reclassification task either $[d, (d_A \to d_I)]$ or $[d, (d_A \to d_E)]$ are:

- $r_{1[d_A \to d_I]} = [(b_0 * c_H * (f, \to f_1) * (g, \to g_0)] \to (d, d_A \to d_I]$
- $r_{2[d_A \to d_E]} = [(a_2 * b_0 * (e, \to e_3) * (f, \to f_1)] \to (d, d_A \to d_E]$
- $r_{3[d_A \to d_I]} = (e, \to e_1) \to (d, d_A \to d_I]$
- $r_{4[d_A \to d_E]} = [(b_0 * (g, \to g_1)] \to (d, d_A \to d_E]$.

We can show that:

- $Sup(r_{1[(d_A \to d_I)]}) = \{x_2, x_3, x_6\}$
- $Sup(r_{2[(d_A \to d_E)]}) = \{x_2, x_6\}$
- $Sup(r_{3[(d_A \to d_I)]}) = \{x_2, x_3, x_4, x_5, x_6, x_7\}$
- $Sup(r_{4[(d_A \to d_E)]}) = \{x_2, x_3, x_4, x_6\}$.

Assuming that:

- $X(r_1, d_A) = Sup(r_{1[(d_A \to d_I)]})$
- $X(r_2, d_A) = Sup(r_{2[(d_A \to d_E)]})$
- $X(r_3, d_A) = Sup(r_{3[(d_A \to d_I)]})$
- $X(r_4, d_A) = Sup(r_{4[(d_A \to d_E)]})$.

by applying ARAS algorithm we get:

- $(b_0 * c_H * a_1)^* = \{x_1\} \not\subseteq X(r_1, d_A)$
- $(b_0 * c_H * a_2)^* = \{x_2, x_6\} \subseteq X(r_1, d_A)$
- $(b_0 * c_H * f_2)^* = \{x_6\} \subseteq X(r_1, d_A)$
- $(b_0 * c_H * g_1)^* = \{x_7, x_8\} \not\subseteq X(r_1, d_A)$
- $(b_0 * c_H * g_2)^* = \{x_2, x_3\} \subseteq X(r_1, d_A)$.

Algorithm ARAS will construct two action rules for the action rule schema:

- $[b_0 * c_H * (f, f_2 \to f_1) * (g, \to g_0)] \to (d, d_A \to d_I)$
- $[b_0 * c_H * (f, \to f_1) * (g, g_2 \to g_0)] \to (d, d_A \to d_I)$.

In a similar way remaining action rules can be constructed. The complexity of ARAS is lower when compared to the complexity of system DEAR. System DEAR (Tzacheva and Ras, 2005) takes all possible pairs of classification rules within each cluster (groups classification rules of non-conflicting rules) and tries to build action rules from them. ARAS algorithm treats each classification rule describing target decision value as a seed and grabs other classification rules describing non-target decision values to form a cluster. Then it builds decision rules automatically from them. Rules grabbed into a seed are only compared with that seed. So the number of pairs of rules which have to be checked is greatly reduced. The complexity of second module of ARAS is $O(k.n)$, where n is the number of classification rules extracted by LERS and k is the number of clusters. Whereas time complexity of DEAR system is $O(n.n)$, where n is number of classification rules extracted by LERS.

7.4 Action rules tightly coupled – object-based

Dardzinska (2012) has summarised the tighlty coupled framework for action rules as follows: in Ras (1996), we can find first steps taken with problem of mining action rules without pre-existing classification rules which is similar to the Apriori algorithm proposed by Agarwal et al. (1994). Ras and Wieczorkowska (2000) propose changes to stable attributes in their paper. In general, to avoid unnecessary changes to stable attributes and to rule out action rules with such changes, very high cost is assigned. Ras et al. (2008) and Tzacheva (2010) propose there is cost associated with changes to stable or flexible attributes.

Im et al. (2010) propose a method that extracts action rules directly from attribute values in incomplete information systems without using pre-existing conditional rules. This means that they use pre-existing classification rules or generate rules using rule discovery algorithms such as LERS (Grzymala-Busse, 1997) or ERID (Dardzinska and Ras, 2005), then construct action rules either from certain pairs of rules or from a single classification rule. The methods in (He et al., 2005; Ras and Dardzinska, 2006; Ras et al., 2007) do not formulate actions directly from existing classification rules. Actions are built as the effect of possible changes in classification rules. Thus, the extraction of classification rule during action rule formulation is inevitable.

The algorithm ARD (Dardzinska, 2012) is used for for constructing action rules, similar to ERID. Identifying the relationships between granules defined by the indiscernibility relation on system's objects is the main goal of algorithm ARD. Papers of Dardzinska and Ras (2006) and Ras and Dardzinska (2006, 2010) present a new strategy for discovering action rules directly from the decision system. To present this method, it is sufficient to show how terms of length greater than one are built. Only positive marks yield action rules. Action terms of length k are built from unmarked action terms of length $k - 1$ and unmarked atomic action terms of length one.

Assume that $S = (X, A \cup \{d\}, V)$ is a decision system, and λ_1, λ_2 denote minimum support and minimum confidence respectively. Each attribute $a \in A$ defines in a unique way the set $C_s(a) = \{N_s(t_a)$:t_a is an atomic action term built from elements in $V_a\}$.

7.4.1 Marking strategy

$\forall N_s(t_a) \in C_s(a)$:

- If $L(N_s(t_a))$ = 0 or $R(N_s(t_a))$ = 0 or $L(N_s(t_a * t_d))$ = 0 or $R(N_s(t_a * t_d))$ = 0 then t_a is marked negative.
- If $L(N_s(t_a)$=R(N$_s(t_a))$ then t_a stays unmarked.
- If card $(L(N_s(t_a * t_d))) < \lambda_1$ then t_a is marked negative.
- If card $(L(N_s(t_a * t_d))) \geq \lambda_1$ and $conf(t_a \to t_d) < \lambda_2$ then t_a stays unmarked.
- If card $(L(N_s(t_a * t_d))) \geq \lambda_1$ and $conf(t_a \to t_d) \geq \lambda_2$ then t_a is marked positive.

From all marked forms, action rule $t_a \to t_d$ is taken into consideration.

Table 16 Decision system S

X	*Attribute a*	*Attribute b*	*Attribute c*	*Decision d*
x_1	a_1	b_1	c_1	H
x_2	a_2	b_2	c_3	H
x_3	a_2	b_1	c_3	A
x_4	a_3	b_1	c_2	A
x_5	a_2	b_1	c_2	A
x_6	a_2	b_2	c_2	H
x_7	a_3	b_1	c_2	A
x_8	a_1	b_2	c_1	A
x_9	a_1	b_1	c_3	H
x_{10}	a_2	b_2	c_3	H

Consider the decision system S in Table 16. $\{a, c\}$ are stable attributes denoted by A_{St} and $\{b, d\}$ are flexible attributes denoted by A_{Fl}. We are interested in object re-classification from decision class (d, H) to (d, A). Assume that threshold for minimal support is λ_1 = 2 and for minimal confidence λ_2 = 0.25.

7.4.2 First loop

Building all atomic action terms for S. For the decision attribute $\{d\}$ is S:

$$N_s(t_{11}) = [\{x_1, x_2, x_6, x_9, x_{10}\}, \{x_3, x_4, x_5, x_7, x_8\}] \tag{12}$$

For classification attributes, both stable and flexible, in S:

$$t_2 = (a, a_2 \to a_2)t_3 = (a, a_3 \to a_3)t_4 = (c, c_1 \to c_1)$$

$$t_5 = (c, c_2 \to c_2)t_6 = (c, c_3 \to c_3)t_7 = (b, b_1 \to b_1)$$

$$t_8 = (b, b_1 \to b_2)t_9 = (b, b_2 \to b_1)t_{10} = (b, b_2 \to b_2)$$

$$N_s(t_1) = [\{x_1, x_8, x_9\}, \{x_1, x_8, x_9\}] \tag{13}$$

$$N_s(t_2) = [\{x_2, x_3, x_5, x_6, x_{10}\}, \{x_2, x_3, x_5, x_6, x_{10}\}] \tag{14}$$

$$N_s(t_3) = [\{x_4, x_7\}, \{x_4, x_7\}] \tag{15}$$

$$N_s(t_4) = [\{x_1, x_8\}, \{x_1, x_8\}] \tag{16}$$

$$N_s(t_5) = [\{x_4, x_5, x_6, x_7\}, \{x_4, x_5, x_6, x_7\}] \tag{17}$$

$$N_s(t_6) = [\{x_2, x_3, x_9, x_{10}\}, \{x_2, x_3, x_9, x_{10}\}] \tag{18}$$

$$N_s(t_7) = [\{x_1, x_3, x_4, x_5, x_7, x_9\}, \{x_1, x_3, x_4, x_5, x_7, x_9\}] \tag{19}$$

$$N_s(t_8) = [\{x_1, x_3, x_4, x_5, x_7, x_9\}, \{x_2, x_6, x_8, x_{10}\}] \tag{20}$$

$$N_s(t_9) = [\{x_2, x_6, x_8, x_{10}\}, \{x_1, x_3, x_4, x_5, x_7, x_9\}] \tag{21}$$

$$N_s(t_{10}) = [\{x_2, x_6, x_8, x_{10}\}, \{x_2, x_6, x_8, x_{10}\}] \tag{22}$$

- Equations (13), (14), (16), (17), (18) and (19) are not marked as Y_1 = Y_2.
- Equation (15) is marked negative as card ($Y1 \cap Y_2$) = 0.
- Equation (20) is not marked as sup = 2 but $conf$ = 0.04 < λ_2.
- Equation (21) is marked positive as sup = 3 and $conf$ = 0.5.
- Equation (22) is not marked as sup = 3 but $conf$ = 0.18 < λ_2.

7.4.3 Second loop

Building action terms of length two from all possible unmarked atomic action terms.

$$N_s(t_1 * t_4) = [\{x_1, x_8\}, \{x_1, x_8\}] = N_s(t_4) \tag{23}$$

$$N_s(t_1 * t_5) = [\{\emptyset\}, \{\emptyset\}] \tag{24}$$

$$N_s(t_1 * t_6) = [\{x_9\}, \{x_9\}] \tag{25}$$

$$N_s(t_1 * t_7) = [\{x_1, x_9\}, \{x_1, x_9\}] \tag{26}$$

$$N_s(t_1 * t_8) = [\{x_1, x_9\}, \{x_8\}] \tag{27}$$

$$N_s(t_1 * t_{10}) = [\{x_8\}, \{x_8\}] \tag{28}$$

$$N_s(t_2 * t_4) = [\{\emptyset\}, \{\emptyset\}] \tag{29}$$

$$N_s(t_2 * t_5) = [\{x_5, x_6\}, \{x_5, x_6\}] \tag{30}$$

$$N_s(t_2 * t_6) = [\{x_2, x_3, x_{10}\}, \{x_2, x_3, x_{10}\}] \tag{31}$$

$$N_s(t_2 * t_7) = [\{x_3, x_5\}, \{x_3, x_5\}] \tag{32}$$

$$N_s(t_2 * t_8) = [\{x_3, x_5\}, \{x_2, x_6, x_{10}\}] \tag{33}$$

$$N_s(t_2 * t_{10}) = [\{x_2, x_6, x_{10}, \{x_2, x_6, x_{10}\}] \tag{34}$$

$$N_s(t_4 * t_7) = [\{x_1\}, \{x_1\}] \tag{35}$$

$$N_s(t_4 * t_8) = [\{x_1\}, \{x_8\}] \tag{36}$$

$$N_s(t_4 * t_{10}) = [\{x_8\}, \{x_8\}] \tag{37}$$

$$N_s(t_5 * t_7) = [\{x_4, x_5, x_7\}, \{x_5, x_7\}] \tag{38}$$

$$N_s(t_5 * t_8) = [\{x_4, x_5, x_7\}, \{x_6\}] \tag{39}$$

$$N_s(t_5 * t_{10}) = [\{x_6\}, \{x_6\}] \tag{40}$$

$$N_s(t_6 * t_7) = [\{x_3, x_9\}, \{x_3, x_9\}] \tag{41}$$

$$N_s(t_6 * t_8) = [\{x_3, x_9\}, \{x_2, x_{10}\}] \tag{42}$$

$$N_s(t_6 * t_{10}) = [\{x_2, x_{10}\}, \{x_2, x_{10}\}] \tag{43}$$

$$N_s(t_7 * t_{10}) = [\{\emptyset\}, \{\emptyset\}] \tag{44}$$

$$N_s(t_8 * t_{10}) = [\{\emptyset\}, \{x_2, x_6, x_8, x_{10}\} \tag{45}$$

- Equations (25), (26), (34), (35), (39), (40), (42) and (43) are marked negative as card $(Y_2 \cap Z_2)$ = 0.
- Equations (24), (29), (37), (44) and (45) are marked negative as card (Y_1) = 0.
- Equations (30) and (31) not marked as Y_1 = Y_2.
- Equations (32), (33) and (38) are marked as negative as card $(Y_1 \cap Z_1)$ = 0.
- Equations (36) and (41) are marked negative as sup = 1.
- Equation (23) is marked but no rule.
- Equation (27) is *marked negative* as sup = 2 and $conf$ = 1.

7.4.4 Third loop

Building action terms of length three, four etc. from all possible unmarked shorter terms. It is repeated until we reach the fix point. In our example the algorithm stops, as we cannot form any other action terms. Two rules we obtained are given below:

$$r_1 = [(b, b_2 \rightarrow b_1)] \rightarrow (d, H \rightarrow A) \tag{46}$$

with $sup(r_1)$ = 3 and $conf(r_1)$ = 0.5

$$r_2 = [(a, a_1 * (b, b_1 \rightarrow b_2)] \rightarrow (d, H \rightarrow A) \tag{47}$$

with $sup(r_2)$ = 2 and $conf(r_2)$ = 1.

7.5 Association action rules

The task of association rule mining is to find certain association relationships among a set of objects in large databases Dardzinska (2012). The association relationships are described as rules. Association rules discovery was first introduced by Agrawal et al. (1993). A simple example of association rule given by Agrawal et al. (1993), is as follows: "90% of transactions that purchase bread and butter also purchase milk." The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

In literature of data mining according to Dardzinska (2012) there are a lot of papers on designing scalable algorithms for mining association rules. For instance, Agarwal et al. (1994) give a more formal definition of association rule. Let $I = \{i_1, i_2, ..., i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subset I$. Each transaction has a unique identifier called transaction ID. A transaction T contains X, where X is set of some items in the item set I, if $X \subset I$. An association rule is an implication of the form $X \rightarrow Y$, where $X \subset I$ and $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ holds in the transaction set D with confidence c if c% of transactions in D that contain X also contain Y. The rule $X \rightarrow Y$ has a support s in the transaction set D if s% of transactions in D contain $X \cup Y$. They propose new algorithms for discovering association rules between items in a large sales transaction database. Similarly Gouda and Zaki (2001), Han et al. (2004), Savasere et al. (1995) and Zaki (2000) propose specific scalable algorithms for large amount of data.

Wang et al. (2002), suggest how to build a new model for promotion strategies to new customers, with the goal of maximising the profit for a given set of transactions and pre-selected target items. They use association rule to generate recommendation action. Padmanabhan and Tuzhilin (1998) and Wang et al. (2003) propose methods to generate interesting patterns by incorporating knowledge in the process of searching for patterns in data. They focus on providing methods which generate unexpected patterns with respect to intuition. Prasanna et al. (2016) generate stock rules based on association rule mining using enhanced apriori algorith with modified gentic algorithm. The proposed approach generates strong association rules based on the fact that the premature convergence of the genetic algorithm is controlled by using crossover and mutation genetic operations.

Genrally most of traditional rule mining systems extract huge amount of rules which poses problem for the end user of the results in finding the most useful or appropriate rules. Tzacheva (2012) propose an algorithm to explore local action rules based on schemas. This method helps avoid the extensive post processing step to prune the lasrge volume of rules generated. Similary Zhang et al. (2010) use second order mining from association rules to find genuine useful knowledge for decision making. They use both objective and subjective measures to extract intelligent knowledge.

There are two categories of association rules according to Paul and Hoque (2010), namely conventional positive association rules and irregular association rules. The patterns that occur frequently and represent routine decisions based on set of facts are called conventional positive association rules, on the other hand patterns that represent rare decisions based on the same set of facts are called irregular association rules. They propose a level wise search algorithm based on action and non-action type data to find irregular association rules. These irregular patterns help identify the wrong or illegal practice or decision in applications including health-care, banking, and others to improve the system. The proposed algorithm uses maximum confidence constraint to form rules.

Imbalanced datasets are likely to prune most of the rules from the minority class and affect the accuracy of classification (Arunasalam and Chawla, 2006). Zhang et al. (2008) propose an efficient algorithm to mine novel association rules called combined association rules on imbalanced datasets. Formal definition of combined association rule is given below.

Definition 7.11: Let T be a dataset where each tuple is described by a schema $S = (S_{D1}, ..., S_{Dm}, S_{A1}, ..., S_{An}, S_C)$, in which $S_D = (S_{D1}, S_{D2}, ..., S_{Dm})$ are m non-actionable attributes, $S_A = (S_{A1}, S_{A2}, ..., S_{An})$ are n actionable attributes, and S_C is a class attribute. Suppose itemset $D \subseteq I_D$, I_D is the itemset of any items with attributes $(S_{D1}, S_{D2}, ..., S_{Dm})$, itemset $A \subseteq I_A$, I_A is the itemset of any items with attributes $(S_{A1}, S_{A2}, ..., S_{An})$, C is 1-itemset of class attribute, a combined association rule is represented as $D + A_1 \rightarrow C_{k1}$ ... $D + A_i \rightarrow C_{ki}$, here $+$ means itemsets appearing simultaneously.

Ras et al. (2008) introduce the notion of action frequent action sets, and present methods of building action rules from frequent action sets to achieve the following objectives: to extract action rules directly from a decision system without using pre-existing classification rules, and to extract action rules that have minimal attribute involvement.

7.5.1 Frequent action sets

Let t_a is an atomic action set, where $Ns(t_a)$ = $[Y_1, Y_2]$ and a $\in A$. We say that t_a is called frequent if card (Y_1) $\geq \lambda_1$ and card (Y_2) $\geq \lambda_2$. The operation of generating $(k + 1)$ element candidate action sets from frequent k-element action sets is performed in two steps:

- *Merging step:* Merge pairs (t_1, t_2) of frequent k-element action sets into $(k + 1)$ element candidate action set if all elements in t_1 and t_2 are the same except the last element.
- *Pruning step:* Delete each $(k + 1)$ element candidate action set t if either of it is not an action set or some k-element subset of f is not a frequent k-element action set.

Now, if t is a $(k + 1)$ element candidate action set, $N_s(t) = [Y_1, Y_2]$, $card(Y_1) \geq \lambda_1$ and $card(Y_2) \geq \lambda_2$, then t is a frequent $(k + 1)$ element action set. We say that t is a frequent action set in S if t is a frequent k-element action set in S, for some k.

Assume now that the expression $[t - t_2]$ denotes the action set containing all atomic action sets listed in t but not listed in t_1. The set AARS (λ_1, λ_2) of association action rules in S is constructed in the following way.

Let t be a frequent action set in S and t_1 is its subset. Any action rule $r = [(t - t_1) \rightarrow t_1]$ is an association action rule in AARS (λ_1, λ_2) if $conf(r) \geq \lambda_2$.

Consider the information system in Table 16. Assume λ_1 = 2 and λ_2 = 0.4. The following frequent action sets can be constructed.

Table 17 Frequent action sets

Frequent action set	*Support*
$(a,\ a_1)$	3
$(a,\ a_2)$	5
$(a,\ a_3)$	2
$(b,\ b_1)$	6
$(b,\ b_2)$	4
$(b,\ b_1 \rightarrow b_2)$	6
$(b,\ b_2 \rightarrow b_1)$	4
$(c,\ c_1)$	2
$(c,\ c_2)$	4
$(c,\ c_3)$	4
$(d,\ H)$	5
$(d,\ A)$	5

Building pairs from all possible atomic terms: as shown in Table 18.

Table 18 Possible action sets

Possible action set	*Support*
$(a,\ a_1)$ * $(b,\ b_1)$	2
$(a,\ a_1)$ * $(b,\ b_2)$	1 (not frequent)
$(a,\ a_1)$ * $(b,\ b_1 \rightarrow b_2)$	2
$(a,\ a_1)$ * $(b,\ b_2 \rightarrow b_1)$	1 (not frequent)
$(a,\ a_1)$ * $(c,\ c_1)$	2
$(a,\ a_1)$ * $(c,\ c_2)$	0 (not frequent)
$(a,\ a_1)$ * $(c,\ c_3)$	1 (not frequent)
$(a,\ a_1)$ * $(d,\ H)$	2
$(a,\ a_1)$ * $(d,\ A)$	1 (not frequent)
$(a,\ a_2)$ * $(b,\ b_1)$	2
$(a,\ a_2)$ * $(b,\ b_2)$	3
$(a,\ a_2)$ * $(b,\ b_1 \rightarrow b_2)$	2
$(a,\ a_2)$ * $(b,\ b_2 \rightarrow b_1)$	3
$(a,\ a_2)$ * $(c,\ c_1)$	0 (not frequent)
$(a,\ a_2)$ * $(c,\ c_2)$	2
$(a,\ a_2)$ * $(c,\ c_3)$	3
$(a,\ a_2)$ * $(d,\ H)$	3
$(a,\ a_2)$ * $(d,\ A)$	2
...	...
$(b,\ b_1)$ * $(c,\ c_1)$	1 (not frequent)
$(b,\ b_1)$ * $(c,\ c_2)$	3
$(b,\ b_1)$ * $(c,\ c_3)$	2
...	...
$(a,\ a_2)$ * $(b,\ b_2 \rightarrow b_1)$ * $(c,\ c_2)$ * $(d,\ H \rightarrow A)$	2

Association action rules can be constructed from all frequent action sets. For instance, we can generate action rule $[(a,\ a_2)$ * $(b,\ b_2 \rightarrow b_1)$ * $(c,\ c_2)] \rightarrow (d,\ H \rightarrow A)$ from the last frequent action set listed above. We can also construct simple association action rule, calculate the cost of association action rule, and give a strategy to construct simple association action rules of lowest cost.

7.6 Representative association action rules

Dardzinska (2012), summarise the concept of representative association rules as follows: Kryszkiewicz (1998) introduce the concept of representative association rules. These representative association rules form a small subset of association rules from which the remaining association rules can be generated. Ras et al. (2008) and Saquer and Jitender (2000) present similar approach for action rules.

Definition 7.12: By a cover of association rule $r : (t_1 \rightarrow t)$ we mean $cov(r) = cov(t_1 \rightarrow t) = \{t_1 * t_2 \rightarrow t_3 : t_2, t_3$ are not overlapping subterms of $t\}$.

For instance, assume that $r : [(a, a_1 \rightarrow a_2) \rightarrow (b, b_1 \rightarrow b_2) * (c, c_1 \rightarrow c_2) * (d, d_1 \rightarrow d_2)]$ is an association rule. Then, $((a, a_1 \rightarrow a_2) * (b, b_1 \rightarrow b_2) * (c, c_1 \rightarrow c_2)) \in cov(r)$.

Property 1: If r $\in AAR_s(\lambda_1, \lambda_2)$, then each rule $r_k \in cov(r)$ also belongs to $AAR_s(\lambda_1, \lambda_2)$.

From the definition of $AAR_s(\lambda_1, \lambda_2)$ we have: $sup(r) \geq \lambda_1$ and $conf(r) \geq \lambda_2$.

$$r_k = (t_1 * t_2) \rightarrow t_4 \tag{48}$$

$$r = (t_1 * t_2) * t_3 * t_4 \tag{49}$$

$$N_s(t_i) = [Y_i, Z_i] \tag{50}$$

where $i \in \{1, 2, 3, 4\}$.

Consider equations (48), (49), and (50). Since, $\dfrac{card[Y_1 \cap Y_2 \cap Y_3 \cap Y_4]}{card[Y_1]} \geq \lambda_1$ then, $\dfrac{card[Y_1 \cap Y_2 \cap Y_4]}{card[Y_1 \cap Y_2]} \geq \lambda_1$. It comes from the fact, that: $card[Y_1 \cap Y_2 \cap Y_4] \geq card[Y_1 \cap Y_2 \cap Y_3 \cap Y_4]$ and $card[Y_1 \geq card[Y_1 \cap Y_2]$. In a similar way we show that $\dfrac{card[Z_1 \cap Z_2 \cap Z_4]}{card[Z_1 \cap Z_2]} \geq \lambda_1$. The same, $sup(r_k) \geq \lambda_1$.

Now assume, that: $conf(r) = \dfrac{card[Y_1 \cap Y_2 \cap Y_3 \cap Y_4]}{card[Y_1]} . \dfrac{card[Z_1 \cap Z_2 \cap Z_3 \cap Z_4]}{card[Z_1]} \geq \lambda_2$. Clearly, $\dfrac{card[Y_1 \cap Y_2 \cap Y_4]}{card[Y_1 \cap Y_2]} . \dfrac{card[Z_1 \cap Z_2 \cap Z_4]}{card[Z_1 \cap Z_2]} \geq \lambda_2$. The same $conf(r_k) \geq \lambda_2$.

Property 2: Representative association rules $RAAR_s(\lambda_1, \lambda_2)$ form a least set of representative association action rules that covers all association action rules $AAR_s(\lambda_1, \lambda_2)$.

Assume that $r \in RAAR_s(\lambda_1, \lambda_2)$ and there exists $(r_k \in (t_1 \rightarrow t) \in AAR_s(\lambda_1, \lambda_2)$ such that $r_k \neq r$ and $r \in cov(r_k)$. Since $r \in cov(r_1)$ then r is not in $RRAR_s(\lambda_1, \lambda_2)$.

Property 3: All association rules $AAR_s(\lambda_1, \lambda_2)$ can be derived from representative association action rules $RAAS_s(\lambda_1, \lambda_2)$ by means of cover operator.

$$r : (t \to s) \in AAR_s(\lambda_1, \lambda_2) \tag{51}$$

$$t = t_1 * t_2 * ... * t_n \tag{52}$$

where $\{t_i\}_{i\in\{1,2,...,n\}}$.

Assume that equations (51) and (52) are atomic action sets. It means that $sup(r) \geq \lambda_1$ and $conf(r) \geq \lambda_2$.

$$r_i(t) = ((t - t_i \to s * t_i) \tag{53}$$

Let equation (53) for any atomic action set t_i in t. Clearly support and confidence are given by equations (54) and (55).

$$sup(r_i(t)) = sup(r) \tag{54}$$

$$conf(r_i(t)) \leq conf(r) \tag{55}$$

Now we show how to construct representative association action rule from which r can be generated. It consists of two main steps: First we:

1 Find t_i in t such that $conf(r_i(t)) \geq \lambda_2$.

2 If succeeded then $t := (t - t_i), s := s * t_i$ and we go back to step 1. Otherwise procedure stops.

In next step we extend the decision part of the rule generated in previous step. Assuming that $(t \to s)$ in such rule, and $T = \{t_1, t_2, ..., t_m\}$ is a set of all atomic action terms not listed in s, we:

1 Find t_i in T such that $sup(t \to s * t_i) \geq \lambda_1$.

2 If succeeded then $T := T - t_i, s := s * t_i$ and we go back to step 1. Otherwise procedure stops.

The resulting association action rule is a representative rule from which the initial rule r can be generated.

7.7 Cost and feasibility

Ras and Tzacheva (2005) and Tzacheva and Tsay (2008), introduce the notion cost and feasibility of an action rule. The notion of cost and feasibility is summarised as follows in Dardzinska (2012): Assume that S is an information system. Let $b \in B$ is flexible attribute and b_1, b_2 are values of b. By ρ (b_1, b_2) mean any number from the open interval $(0, 1) \cup \{+\infty\}$ which describes the cost to change the value from b_1 to b_2 by the user of the information system S.

- The value of $\rho(b_1, b_2 \approx 0)$ is interpreted that the change of values from b_1 to b_2 is quite trivial.

- The value of $\rho(b_1, b_2 \approx 1)$ is interpreted that the change of values from b_1 to b_2 is very difficult to be achieved.
- The value of $\rho(b_1, b_2 \approx +\infty)$ is interpreted that the change is not feasible.

Also, if $\rho(b_1, b_2) < \rho(b_3, b_4)$, then change of values from b_1 to b_2 is more feasible than the change from b_3 to b_4.

The values $\rho(b_i, b_j)$ are given by the user of information system and they should be seen as atomic values needed to introduce the notion of the feasibility of an action rule.

Assume now that equation (56) is a (r_1, r_2) action rule (Dardzinska and Ras, 2006; Ras and Dardzinska, 2011).

$$\begin{aligned} r &= [(b_1, v_1 \to w_1) * (b_2, v_2 \to w_2) * ... * (b_m, v_m \to w_m)](x) \\ &= (d, d_1 \to d_2)(x) \end{aligned} \tag{56}$$

Definition 7.13: By the cost of rule r denoted by $cost(r)$ means value in equation (57)

$$cost(r) = \sum\{\rho(v_i, w_i) : 1 \le i \le n\}. \tag{57}$$

Rule r is feasible if $cost(r) < \rho(d_1, d_2)$, which means that $cost(r)$ has to be a finite number and the cost of the conditional part of the rule has to be a finite number and the cost of the conditional part of the rule has to be lower than the cost of the decision part of the rule.

Consider d is a decision attribute, assume that $D_s[(d, d_1 \to d_2)]$ denotes the set of all action rules in S having the term $(d, d_1 \to d_2)$ on the decision site. Among all action rules in $D_s[(d, d_1 \to d_2)]$ we have to choose a rule with the smallest cost value. However it can still happen that the rule we chose has the cost value not acceptable by the user of the information system S. The cost of the action rule in equation (58) might be high only because the cost value of one of its sub-terms in the conditional part of the rule is high.

$$\begin{aligned} r_i &= [(b_1, v_1 \to w_1) * (b_2, v_2 \to w_2) * ... * (b_m, v_m \to w_m)](x_i) \\ &\to (d, d_1 \to d_2)(x_i) \end{aligned} \tag{58}$$

Ras and Tzacheva (2005) propose a heuristic procedure to find the low-cost action rule. Assume that $(b_i, v_i \to w_i)$ is the sub-term that increases the cost of action rule in (58). In this scenario, we may look for an action rule in $D_s[(b_i, v_i \to w_i)]$ with the smallest cost value. Assume that equation (59) is feasible rule.

$$\begin{aligned} r_j &= [(b_{i1}, v_{i1} \to w_{i1}) * (b_{i2}, v_{i2} \to w_{i2}) * ... * (b_{im}, v_{im} \to w_{im})](x_j) \\ &\to (b_i, v_i \to w_i)(x_j) \end{aligned} \tag{59}$$

Since objects x_i, x_j are coming from the same information system S, we can compose r_i with r_j getting a new feasible rule given: $(r_i, r_j) = [(b_1, v_1 \to w_1) * (b_{i1}, v_{i1} \to w_{i1}) * (b_{i2}, v_{i2} \to w_{i2}) * ... * (b_{in}, v_{in} \to w_{in}) * ... * (b_m, v_m \to w_m)](x) \to (d, d_1 \to d_2)(x)$.

The cost of this new action rule (r_i, r_j) is lower than the cost of (r_i). However, if support of this rule is equal to 0, then it has no value for user. Otherwise, we can recursively follow this method looking for cheaper rules reclassifying objects from the group d_1 into the group d_2. Each successful step will produce a rule which is cheaper than the previous one. Obviously, this heuristic procedure has to end.

It may seem that if $D_s[(d, d_1 \rightarrow d_2)$ contains all action rules which reclassify objects from one group d_1 into the group d_2 then any new action rule obtained as the result of the proposed recursive strategy is already in that set. This statement is agreeable but practically $D_s[(d, d_1 \rightarrow d_2)$ never contains all such rules.

Firstly, it is too expensive to generate all possible rules from an information system and secondly even if we have such rules it is still too expensive to generate all possible action rules from them. Thus the author justifies the applicability of the heuristic.

7.8 Event condition action rule

Traditional database systems do not initiate operations on their own, instead they are passive and respond to user queries. Later due to significant increase in the volume of data, managing data was complex. Thus the passive databases were converted to active databases to repond indepentantly to data-related events. Goldin et al. (2004) say that this behaviour is described by event-condition-action rules (ECA). There are three components in ECA rules: event, condition and action. The event is the happening to which the rule responds, condition examines the context of the event, if the relevant event and condition occurred then action denotes the task to be carried out (Paton and Diaz, 1999).

Qiao et al. (2007) present graphical ECA rules with temporal events to specify real-time constraints. They used smart home application to validate the generated rules. In smart homes, sensors collect data about human movement and interaction to appliances which is then sent to a real-time active database. Authors believe that with ECA rules, real-time active database will have enhanced capabilities to detect complex events and contexts to differentiate between situations, thus anticipating potential hazardous situations and intelligently advise safety and living standards for person inside the monitored smart home.

7.9 Meta action

Meta-actions are referred to as higher level concepts by Touati et al. (2014) and Tzacheva and Ras (2010). Consider the medical example from Subsection 7.2.1, in order to move a patient from worst prognoses state to good prognoses state requires some treatment procedures to be changed or some medication to be changed. This actionable knowledge is represented by meta-actions. A more formal definition of meta-action given by Touati et al. (2014) is as follows:

Definition 7.14 (meta-actions): Meta-actions associated with an information system S are defined as higher level concepts used to model certain generalisation of action rules. Meta-actions, when executed, trigger changes in values of some flexible attributes in S.

Tzacheva and Ras (2010) present a strategy for generating association action rules and action paths by introducing the use of Meta-actions and influence matrix (Wang et al., 2006). According to Tzacheva and Ras (2010) some higher-level actions called meta-actions are required to trigger the change of flexible attributes in order to move undesirable objects into a desirable group. For example, if a patient is suffering from certain disease, then without proper drug or treatment it is not possible to get rid of the disease. The treatment or medicine in this is an example of meta-action. They use Influence matrix to identify which candidate association action rule and action paths are valid with respect to meta-actions and hidden correlation between classification attributes and decision attributes. Action paths are a sequence of action terms as described in Subsection 7.1.

7.10 Meta actions for sentiment analysis and business recommendations

As described in Subsection 7.9 meta actions are higher level actions that are used to activate action rules. According to Ras et al. (2017) action rules are a set of atomic actions for achieving expected result, Meta-actions are the actions that need to be executed in order to trigger corresponding atomic actions.

Ras et al. (2017) model a net promoter score (NPS) recommender system for driving business revenue mainly based on action rules and meta actions. NPS is a standard metric for measuring customer satisfaction. This system utilised around 400,000 records of the customer satisfaction telephone surveys containing details related to customer details, survey details and benchmark questions. action rules, knowledge in actionable format is collected from customers using a business and also from customers using semantically similar business. The concept of decision reducts (minimal set of attributes that keep the characteristics of the full dataset (Ras et al., 2017) is used to choose critical benchmarks). The triggers (meta actions) for action rules are extracted based on aspect-based sentiment analysis (Hu and Liu, 2004) and text summarisation of the customer text comments in the survey. Feature-opinion pairs are identified with Stanford Parser. They also performed feature clustering based on pre-defined list of seed words. Meta actions are generated by dividing feature class into several subclasses.

Tarnowska et al. (2018) also explain application of decision reducts theory to solve business problem. Similar to Ras et al. (2017), this paper focus on business recommendations to improve NPS of companies. They detail the application area – customer loyalty improvement, machine learning techniques used to develop the knowledge-based system and visualisation techniques for the interactive recommender system.

8 Actionable recommendations for emotion mining

In relation to emotion mining, actionable patterns may suggest a way to alter the user's emotion from a negative, or neutral to a more positive emotion, or a desirable state/attitude. For example, for customer care services, recommendation systems for online shopping, or smart phones that are able to recognise human emotions, emotion altering actionable patterns include: suggesting calming music, playing mood enhancing movie, changing the background colours to suiting ones, or calling caring friends (for smart phones).

In our work Ranganathan et al. (2017) the primary intent of the action rules generated is to provide viable suggestions on how to make a twitter user feel more positive. For Twitter social network data, actionable recommendations may include – how to increase user's friends count, how to increase the user's follower's count, and how to change the overall sentiment from negative to positive, or from neutral to positive.

In the future, we plan to apply action rules in order to suggest ways to alter specific negative emotions such as: sadness, fear, anger to more positive ones such as: joy, trust, surprise. We plan to apply this method to Twitter social network data, as well e-learning, student-teacher evaluations, and Amazon customer satisfaction surveys.

9 Conclusions

Emotions and feelings accompany us throughout the span of our lives and colour the way we build and maintain the basis for interactions with people in a society (Neviarouskaya et al., 2011), and through computer-based systems, including human-computer interaction. With rise of social media such as blogs, forums, social networking sites like Twitter, Facebook, and proliferation of online product reviews, the need for sentiment analysis techniques and emotion detection from text has been ever increasing. Additional applications include: customer care services, recommendation systems for online shopping, text messages, e-learning, and student teaching evaluations, as well as the smart phones and technology of the future, which is able to detect and recognise human emotions. Mining for actionable knowledge and providing actionable recommendations, which can alter emotions from negative to positive is a challenging and important subject, that benefits all emotion recognition systems.

References

Adomavicius, G. and Tuzhilin, A. (1997) 'Discovery of actionable patterns in databases: the action hierarchy approach', *KDD*, pp.111–114.

Agarwal, R., Srikant, R. et al. (1994) 'Fast algorithms for mining association rules', *Proc. of the 20th VLDB Conference*, pp.487–499.

Agrawal, R., Imieliński, T. and Swami, A. (1993) 'Mining association rules between sets of items in large databases', *ACM Sigmod Record*, Vol. 22, No. 2, pp.207–216.

Alm, E.C.O. (2008) 'Affect in text and speech'.

Arunasalam, B. and Chawla, S. (2006) 'CCCS: a top-down associative classifier for imbalanced class distribution', *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.517–522.

Bagavathi, A. and Tzacheva, A.A. (2017) 'Rule based systems in a distributed environment: survey', *Proceedings of International Conference on Cloud Computing and Applications (CCA17), 3rd World Congress on Electrical Engineering and Computer Systems and Science (EECSS'17)*, pp.1–17.

Bao, S., Xu, S., Zhang, L., Yan, R., Su, Z., Han, D. and Yu, Y. (2009) 'Joint emotion-topic modeling for social affective text mining', *Ninth IEEE International Conference on Data Mining, ICDM'09*, pp.699–704.

Bao, S., Xu, S., Zhang, L., Yan, R., Su, Z., Han, D. and Yu, Y. (2012) 'Mining social emotions from affective text', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 24, No. 9, pp.1658–1670.

Barrett, J., Caruthers, E., German, K., Hamby, E., Lofthus, R., Srinivas, S. and Ells, E. (2011) 'From data to actionable knowledge: a collaborative effort with educators', *Proceedings of KDD*.

Benamara, F., Cesarano, C., Picariello, A., Recupero, D.R. and Subrahmanian, V.S. (2007) 'Sentiment analysis: adjectives and adverbs are better than adjectives alone', in *Proceedings of International AAAI Conference on Weblogs and Social Media (ICWSM)*.

Bollen, J., Mao, H. and Pepe, A. (2011) 'Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena', *International AAAI Conference on Web and Social Media ICWSM*, Vol. 11, pp.450–453.

Brücher, H., Knolmayer, G. and Mittermayer, M-A. (2002) 'Document classification methods for organizing explicit knowledge'.

Bradley, M.M. and Lang, P.J. (2007) *The nternational Affective Digitized Sounds (IADS-2) Affective Ratings of Sounds and Instruction Manual*, Tech. Rep. B-3, University of Florida, Gainesville, FL.

Bradley, M.M. and Lang, P.J. (1999) 'Affective norms for English words (ANEW): instruction manual and affective ratings'.

Bradley, M.M., Greenwald, M.K., Petry, M.C. and Lang, P.J. (1992) 'Remembering pictures: pleasure and arousal in memory', *Journal of Experimental Psychology Learning, Memory, and Cognition*, Vol. 18, No. 2, p.379.

Cao, L. (2015) 'Actionable knowledge discovery and delivery', *Metasynthetic Computing and Engineering of Complex Systems*, Vol. 5, pp.287–312.

Chaumartin, F. (2007) 'UPAR7: a knowledge-based system for headline sentiment tagging', *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp.422–425.

Cohn, J.F. and Katz, G.S. (1998) 'Bimodal expression of emotion by face and voice', *Proceedings of the Sixth ACM International Conference on Multimedia Face/Gesture Recognition and their Applications*, pp.41–44.

Danisman, T. and Alpkocak, A. (2008) 'Feeler: emotion classification of text using vector space model', *AISB 2008 Convention Communication, Interaction and Social Intelligence*, Vol. 1, p.53.

Dardzinska, A. and Ras, Z.W. (2003) 'On rules discovery from incomplete information systems', in Lin, T.Y., Hu, X., Ohsuga, S. and Liau, C. (Eds.): *Proceedings of ICDM'03 Workshop on Foundations and New Directions of Data Mining*, IEEE Computer Society, Melbourne, Florida, pp.31–35.

Dardzinska, A. and Ras, Z.W. (2005) 'Extracting rules from incomplete decision systems: system ERID', *Foundations and Novel Approaches in Data Mining*, pp.143–153.

Dardzinska, A. and Ras, Z.W. (2006) 'Cooperative discovery of interesting action rules', *International Conference on Flexible Query Answering Systems*, pp.489–497.

Dardzinska, A. (2012) *Action Rules Mining*, Vol. 468, Springer.

Dasgupta, A., Drineas, P., Harb, B., Josifovski, V. and Mahoney, M.W. (2007) 'Feature selection methods for text classification', *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.230–239.

De Choudhury, M., Gamon, Mi., Counts, S. and Horvitz, E. (2013) 'Predicting depression via social media', *International AAAI Conference on Weblogs and Social Media (ICWSM)*, Vol. 13, pp.1–10.

De Silva, L.C. and Ng, P.C. (2000) 'Bimodal emotion recognition', *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp.332–335.

Devillers, L., Lamel, L. and Vasilescu, I. (2003) 'Emotion detection in task-oriented spoken dialogues', *Proceedings 2003 International Conference on Multimedia and Expo, ICME 03*, Vol. 3, pp.3–549.

Dixon, T. (2003) *From Passions to Emotions The Creation of a Secular Psychological Category*, Cambridge University Press.

Dumais, S., Platt, J., Heckerman, D. and Sahami, M. (1998) 'Inductive learning algorithms and representations for text categorization', *Proceedings of the Seventh International Conference on Information and Knowledge Management*, pp.148–155.

Ekman, P. (1992) 'An argument for basic emotions', *Cognition & Emotion*, Vol. 6, Nos. 3–4, pp.169–200.

El Ayadi, M., Kamel, M.S. and Karray, F. (2011) 'Survey on speech emotion recognition: features, classification schemes, and databases', *Pattern Recognition*, Vol. 44, No. 3, pp.572–587.

Goldin, D., Srinivasa, S. and Srikanti, V. (2004) 'Active databases as information systems', *Proceedings of International Database Engineering and Applications Symposium, IDEAS 04*, pp.123–130.

Gouda, K. and Zaki, M.J. (2001) 'Efficiently mining maximal frequent itemsets', *Proceedings IEEE International Conference on Data Mining, ICDM 2001*, pp.163–170.

Greco, S., Matarazzo, B., Pappalardo, N. and Slowinski, R. (2005) 'Measuring expected effects of interventions based on decision rules', *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 17, Nos. 1–2, pp.103–118.

Grzymala-Busse, J.W. (1997) 'A new version of the rule induction system LERS', *Fundamenta Informaticae*, Vol. 31, No. 1, pp.27–39.

Gupta, N., Gilbert, M. and Fabbrizio, G.D. (2013) 'Emotion detection in email customer care', *Computational Intelligence*, Vol. 29, No. 3, pp.489–505.

Han, Ji., Pei, J., Yin, Y. and Mao, R. (2004) 'Mining frequent patterns without candidate generation: a frequent-pattern tree approach', *Data Mining and Knowledge Discovery*, Vol. 8, No. 1, pp.53–87.

Hancock, J.T., Landrigan, C. and Silver, C. (2007) 'Expressing emotion in text-based communication', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.929–932.

Hasan, M., Agu, E. and Rundensteiner, E. (2014) 'Using hashtags as labels for supervised learning of emotions in twitter messages', *ACM SIGKDD Workshop on Health Informatics*, New York, USA.

He, Z., Xu, X., Deng, S. and Ma, R. (2005) 'Mining action rules from scratch', *Journal of Expert Systems with Applications*, Vol. 29, No. 3, pp.691–699.

Ho, D.T. and Cao, T.H. (2012) 'A high-order hidden Markov model for emotion detection from textual data', *Pacific Rim Knowledge Acquisition Workshop*, pp.94–105.

Hu, M. and Liu, B. (2004) 'Mining and summarizing customer reviews', *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.168–177.

Im, S. and Ras, Z.W. (2008) 'Action rule extraction from a decision table: ARED', *International Symposium on Methodologies for Intelligent Systems*, pp.160–168.

Im, S., Ras, Z. and Wasyluk, H. (2010) 'Action rule discovery from incomplete data', *Knowledge and Information Systems*, Vol. 25, No. 1, pp.21–33.

Jain, M.C. and Kulkarni, V.Y. (2014) 'TexEmo: conveying emotion from text-the study', *International Journal of Computer Applications*, Vol. 86, No. 4.

Jindal, R. and Taneja, S. (2017) 'A lexical-semantics-based method for multi label text categorisation using word net', *International Journal of Data Mining, Modelling and Management*, Vol. 9, No. 4, pp.340–360.

Joachims, T. (1998) 'Text categorization with support vector machines: learning with many relevant features', *European Conference on Machine Learning*, pp.137–142.

Kao, E.C-C., Liu, C-C., Yang, T-H., Hsieh, C-T. and Soo, V-W. (2009) 'Towards text-based emotion detection a survey and possible improvements', *IEEE International Conference on Information Management and Engineering, ICIME'09*, pp.70–74.

Katz, P., Singleton, M. and Wicentowski, R. (2007) 'Swat-mp: the semeval-2007 systems for task 5 and task 14', *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp.308–313.

Kaur, H. (2005) 'Actionable rules: issues and new directions', *World Enformatika Conference WEC*, No. 5, pp.61–64.

Khan, A., Baharudin, B., Lee, L.H. and Khan, K. (2010) 'A review of machine learning algorithms for text-documents classification', *Journal of Advances in Information Technology*, Vol. 1, No. 1, pp.4–20.

Kim, S.M., Valitutti, A. and Calvo, R.A. (2010a) 'Evaluation of unsupervised emotion models to textual affect recognition', *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pp.62–70.

Kim, Y.E., Schmidt, E.M., Migneco, R., Morton, B.G., Richardson, P., Scott, J., Speck, J.A. and Turnbull, D. (2010b) 'Music emotion recognition: a state of the art review', *Proc. ISMIR*, pp.255–266.

Kisilevich, Sl., Rohrdantz, C., Maidel, V. and Keim, D. (2013) 'What do you think about this photo?: A novel approach to opinion and sentiment analysis of photo comments', *International Journal of Data Mining, Modelling and Management*, Vol. 5, No. 2, pp.138–157.

Kleinsmith, A. and Bianchi-Berthouze, N. (2013) 'Affective body expression perception and recognition: a survey', *IEEE Transactions on Affective Computing*, Vol. 4, No. 1, pp.15–33.

Kozareva, Z., Navarro, B., Vázquez, S. and Montoyo, A. (2007) 'UA-ZBSA: a headline emotion classification through web information', *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp.334–337.

Kryszkiewicz, M. (1998) 'Representative association rules', *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp.198–209.

Lövheim, H. (2012) 'A new three-dimensional model for emotions and monoamine neurotransmitters', *Medical Hypotheses*, Vol. 78, No. 2, pp.341–348.

Lang, P.J., Bradley, M.M. and Cuthbert, B.N. (1997) *International Affective Picture System (IAPS) Technical Manual and Affective Ratings*, pp.39–58, NIMH Center for the Study of Emotion and Attention.

Lei, J., Rao, Y., Li, Q., Quan, X. and Wenyin, L. (2014) 'Towards building a social emotion detection system for online news', *Future Generation Computer Systems*, Vol. 7, pp.438–448.

Liu, B., Hsu, W. and Chen, S. (1997) 'Using general impressions to analyze discovered classification rules', *KDD*, pp.31–36.

Luyckx, K., Vaassen, F., Peersman, C. and Daelemans, W. (2012) 'Fine-grained emotion detection in suicide notes: a thresholding approach to multi-label classification', *Biomedical Informatics Insights*, Vol. 5, pp.BII–S8966.

Manning, C.D., Raghavan, P. and Schütze, H. (2008) 'Introduction to information retrieval', Vol. 1.

Merriam-Webster Dictionary (2002) *Merriam-Webster* [online] http://www.mw.com/home.htm.

Meyer, D. (2001) 'Support vector machines', *R News*, Vol. 1, No. 3, pp.23–26.

Miller, G.A. (1995) 'WordNet: a lexical database for English', *Communications of the ACM*, Vol. 38, No. 11, pp.39–41.

Mishne, G., De Rijke, M. et al. (2006) 'Capturing global mood levels using blog posts', *AAAI Spring Symposium Computational Approaches to Analyzing Weblogs*, Vol. 6, pp.145–152.

Mishne, G. et al. (2012) 'Experiments with mood classification in blog posts', *Proceedings of ACM SIGIR 2005 Workshop on Stylistic Analysis of Text for Information Access*, Vol. 19, pp.321–327.

Mohammad, S.M. and Turney, P.D. (2010) 'Emotions evoked by common words and phrases: using mechanical Turk to create an emotion lexicon', *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pp.26–34.

Mohammad, S.M. and Turney, P.D. (2013a) 'Crowdsourcing a word – emotion association lexicon', *Computational Intelligence*, Vol. 29, No. 3, pp.436–465.

Mohammad, S.M. and Turney, P.D. (2013b) *NRC Emotion Lexicon*, NRC Technical Report.

Mohammad, S.M. (2012) '# emotional tweets', *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1 Proceedings of the Main Conference and the Shared Task, and Volume 2 Proceedings of the Sixth International Workshop on Semantic Evaluation*, pp.246–255.

Murphy, K.P. (2006) *Naive Bayes Classifiers*, Vol. 18, University of British Columbia.

Neviarouskaya, A., Prendinger, H. and Ishizuka, M. (2011) 'Affect analysis model: novel rule-based approach to affect sensing from text', *Journal of Natural Language Engineering*, Vol. 17, No. 1, pp.95–135.

Nguyen, T., Bass, I., Li, M. and Sethi, I.K. (2005) 'Investigation of combining SVM and decision tree for emotion classification', *Seventh IEEE International Symposium on Multimedia (ISM'05)*, p.5.

Nielsen, F. (2011) 'A new ANEW: evaluation of a word list for sentiment analysis in microblogs', *arXiv preprint arXiv 1103.2903.*.

Padmanabhan, B. and Tuzhilin, A. (1998) 'A belief-driven method for discovering unexpected patterns', *KDD*, Vol. 98, pp.94–100.

Paton, N.W. and Diaz, O. (1999) 'Active database systems', *ACM Computing Surveys (CSUR)*, Vol. 31, No. 1, pp.63–103.

Paul, R. and Hoque, A.S.M.L. (2010) 'Mining irregular association rules based on action & non-action type data', *Fifth International Conference on Digital Information Management (ICDIM)*, pp.63–68.

Pawlak, Z. (2012) *Rough Sets Theoretical Aspects of Reasoning About Data*, Vol. 9, Springer Science & Business Media.

Plutchik, R. and Kellerman, H. (2013) *Emotion, Psychopathology, and Psychotherapy*, Vol. 5, Academic Press.

Posner, J., Russell, J.A. and Peterson, B.S. (2005) 'The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology', *Development and Psychopathology*, Vol. 17, No. 3, pp.715–734.

Prasanna, S. and Ezhilmaran, D. (2016) 'Association rule mining using enhanced apriori with modified GA for stock prediction', *International Journal of Data Mining, Modelling and Management*, Vol. 8, No. 2, pp.195–207.

Purver, M. and Battersby, S. (2012) 'Experimenting with distant supervision for emotion classification', *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp.482–491.

Qiao, Y., Zhong, K., Wang, H. and Li, X. (2007) 'Developing event-condition-action rules in real-time active database', *Proceedings of the 2007 ACM symposium on Applied Computing*, pp.511–516.

Ranganathan, J., Irudayaraj, A.S. and Tzacheva, A.A. (2017) 'Action rules for sentiment analysis on Twitter data using spark', *2017 IEEE International Conference on Data Mining Workshops (ICDMW*, pp.51–60.

Rangel, F. and Rosso, P. (2016) 'On the impact of emotions on author profiling', *Information Processing & Management*, Vol. 52, No. 1, pp.73–92.

Ras, Z.W. (1996) 'Cooperative knowledge-based systems', *Intelligent Automation & Soft Computing*, Vol. 2, No. 2, pp.193–201.

Ras, Z.W. and Dardzinska, A. (2006) 'Action rules discovery, a new simplified strategy', *International Symposium on Methodologies for Intelligent Systems*, pp.445–453.

Ras, Z.W. and Dardzinska, A. (2010) 'Action rules discovery without pre-existing classification rules', *International Conference on Rough Sets and Current Trends in Computing*, pp.181–190.

Ras, Z.W. and Dardzinska, A. (2011) 'From data to classification rules and actions', *International Journal of Intelligent Systems*, Vol. 26, No. 6, pp.572–590.

Ras, Z.W. and Tzacheva, A.A. (2003) 'Discovering semantic inconsistencies to improve action rules mining', *Intelligent Information Processing and Web Mining*, pp.301–310.

Ras, Z.W. and Tzacheva, A.A. (2005) 'In search for action rules of the lowest cost', *Monitoring, Security, and Rescue Techniques in Multiagent Systems*, pp.261–272.

Ras, Z.W. and Wieczorkowska, A. (2000) 'Action-rules: how to increase profit of a company', *European Conference on Principles of Data Mining and Knowledge Discovery*, pp.587–592.

Ras, Z.W., Dardzinska, A., Tsay, L-S. and Wasyluk, H. (2008) 'Association action rules', *IEEE International Conference on Data Mining Workshops, ICDMW 08*, pp.283–290.

Ras, Z.W., Tarnowska, K.A., Kuang, J., Daniel, L. and Fowler, D. (2017) 'User friendly NPS-based recommender system for driving business revenue', *International Joint Conference on Rough Sets*, pp.34–48.

Ras, Z.W., Tsay, L-S. and Dardzińska, A. (2009) 'Tree-based algorithms for action rules discovery', *Mining Complex Data*, pp.153–163.

Ras, Z.W., Tzacheva, A.A., Tsay, L-S. and Giirdal, O. (2005) 'Mining for interesting action rules', *IEEE International Conference on Intelligent Agent Technology*, pp.187–193.

Ras, Z.W., Wyrzykowska, E. and Wasyluk, H. (2007) 'ARAS: action rules discovery based on agglomerative strategy', *International Workshop on Mining Complex Data*, pp.196–208.

Roberts, K., Roach, M.A., Johnson, J., Guthrie, J. and Harabagiu, S.M. (2012) 'EmpaTweet: annotating and detecting emotions on Twitter', *Language Resources and Evaluation (LREC)*, Vol. 12, pp.3806–3813.

Rubin, D.C. and Talarico, J.M. (2009) 'A comparison of dimensional models of emotion: evidence from emotions, prototypical events, autobiographical memories, and words', *Memory*, Vol. 17, No. 8, pp.802–808.

Russell, J.A. (1980) 'A circumplex model of affect', *Journal of Personality and Social Psychology*, Vol. 39, No. 6, p.1161.

Saquer, J. and Jitender, S. (2000) 'Using closed itemsets for discovering representative association rules', *International Symposium on Methodologies for Intelligent Systems*, pp.495–504.

Savasere, A., Omiecinski, E.R. and Navathe, S.B. (1995) 'An efficient algorithm for mining association rules in large databases'.

Schapire, R.E and Singer, Y. (2000) 'BoosTexter: a boosting-based system for text categorization', *Journal of Machine Learning*, Vol. 39, Nos. 2–3, pp.135–168.

Scherer, K.R. and Wallbott, H.G. (1994) 'Evidence for universality and cultural variation of differential emotion response patterning', *Journal of Personality and Social Psychology*, Vol. 66, No. 2, p.310.

Sebastiani, F. (2002) 'Machine learning in automated text categorization', *ACM computing surveys (CSUR)*, Vol. 34, No. 1, pp.1–47.

Seetha, H., Murty, M.N. and Saravanan, R. (2015) 'Effective feature selection technique for text classification', *International Journal of Data Mining, Modelling and Management*, Vol. 7, No. 3, pp..165–184.

Shahraki, A.G. and Zaıane, O.R. (2017) 'Lexical and learning-based emotion mining from text'.

Shaver, P., Schwartz, J., Kirson, D. and O'Connor, C. (1987) 'Emotion knowledge: further exploration of a prototype approach', *Journal of Personality and Social Psychology*, Vol. 52, No. 6, p.1061.

Silberschatz, A. and Tuzhilin, A. (1995) 'On subjective measures of interestingness in knowledge discovery', *KDD*, Vol. 95, pp.275–281.

Sinoara, R.A., Antunes, J and Rezende, S.O. (2017) 'Text mining and semantics: a systematic mapping study', *Journal of the Brazilian Computer Society*, Vol. 23, No. 1, p.9.

Siolas, G. and d'Alché-Buc, F. (2000) 'Support vector machines based on a semantic kernel for text categorization', *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Vol. 5, pp.205–209.

Strapparava, C. and Mihalcea, R. (2007) 'Semeval-2007 task 14: affective text', *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp.70–74.

Strapparava, C. and Mihalcea, R. (2008) 'Learning to identify emotions in text', *Proceedings of the 2008 ACM Symposium on Applied Computing*, pp.1556–1560.

Strapparava, C., Valitutti, A. et al. (2004) 'Wordnet affect: an affective extension of wordnet', *Language Resources and Evaluation LREC*, Vol. 4, pp.1083–1086.

Strapparava, C., Valitutti, A. and Stock, O. (2006) 'The affective weight of lexicon', *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pp.423–426.

Tan, P-N. et al. (2006) *Introduction to Data Mining*, Pearson Education, India.

Tarnowska, K.A., Ras, Z.W., Daniel, L. and Fowler, D. (2018) 'Visual analysis of relevant features in customer loyalty improvement recommendation', *Advances in Feature Selection for Data and Pattern Recognition*, pp.269–293.

Tausczik, Y.R. and Pennebaker, J.W. (2010) 'The psychological meaning of words: LIWC and computerized text analysis methods', *Journal of Language and Social Psychology*, Vol. 29, No. 1, pp.24–54.

Tomkins, S. (1962) *Affect Imagery Consciousness Volume I The Positive Affects*, Springer Publishing Company.

Tomkins, S.S. (1963) *Affect Imagery Consciousness, 2 The Negative Affects*, Tavistock/Routledge.

Torao, Y., Naruki, S., Kaori, Y. and Masahiro, N. (1997) 'An emotion processing system based on fuzzy inference and subjective observations', *Journal of Information Sciences*, Vol. 101, Nos. 3–4, pp.217–247.

Touati, H., Ras, Z.W., Studnicki, J. and Wieczorkowska, A.A. (2014) 'Mining surgical meta-actions effects with variable diagnoses' number', *International Symposium on Methodologies for Intelligent Systems*, pp.254–263.

Tsay, L-S. and Ras, Z.W. (2005) 'Action rules discovery: system DEAR2, method and experiments', *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 17, Nos. 1–2, pp.119–128.

Tsay, L.S., Ras, Z.W. and Dardzinska, A. (2005) 'Mining e-action rules', *Mining Complex Data*, pp.85–90.

Tzacheva, A.A. and Ras, Z.W. (2005) 'Action rules mining', *International Journal of Intelligent Systems*, Vol. 20, No. 7, pp.719–736.

Tzacheva, A. and Ras, Z.W. (2007) 'Constraint based action rule discovery with single classification rules', *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pp.322–329.

Tzacheva, A.A. and Ras, Z.W. (2010) 'Association action rules and action paths triggered by meta-actions', *2010 IEEE International Conference on Granular Computing (GrC)*, pp.772–776.

Tzacheva, A.A. and Tsay, L-S. (2008) 'Tree-based construction of low-cost action rules', *Fundamenta Informaticae*, Vol. 86, Nos. 1/2, pp.213–225.

Tzacheva, A.A. (2010) 'Summaries of action rules by agglomerative clustering', *Advances in Intelligent Information Systems*, pp.259–271.

Tzacheva, A.A. (2012) 'Rule schemas and interesting association action rules mining', *International Journal of Data Mining, Modelling and Management*, Vol. 4, No. 3, pp.244–254.

Valitutti, A., Strapparava, C. and Stock, O. (2004) 'Developing affective lexical resources', *PsychNology Journal*, Vol. 2, No. 1, pp.61–83.

Vapnik, V.N. (1999) 'An overview of statistical learning theory', *IEEE Transactions on Neural Networks*, Vol. 10, No. 5, pp.988–999.

Voeffray, S. (2011) 'Emotion-sensitive human-computer interaction (HCI): state of the art-seminar paper', *Emotion Recognition*, pp.1–4.

Walther, J.B. (2008) 'Social information processing theory', *Engaging Theories in Interpersonal Communication Multiple Perspectives*, pp.391–404.

Wang, K., Zhou, S. and Han, J. (2002) 'Profit mining: from patterns to actions', *International Conference on Extending Database Technology*, pp.70–87.

Wang, K., Jiang, Y. and Lakshmanan, L.V.S. (2003) 'Mining unexpected rules by pushing user dynamics', *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.246–255.

Wang, K., Jiang, Y. and Tuzhilin, A. (2006) 'Mining actionable patterns by role models', *Proceedings of the 22nd International Conference on Data Engineering, ICDE 06*, pp.16–16.

Wang, W., Chen, L., Thirunarayan, K. and Sheth, A.P. (2012) 'Harnessing twitter 'big data' for automatic emotion identification', *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on Social Computing (SocialCom)*, pp.587–592.

Watson, D. and Tellegen, A. (1985) 'Toward a consensual structure of mood', *Psychological Bulletin*, Vol. 98, No. 2, p.219.

Yadollahi, A., Shahraki, A.G. and Zaiane, O.R. (2017) 'Current state of text sentiment analysis from opinion to emotion mining', *ACM Computing Surveys (CSUR)*, Vol. 50, No. 2, p.25.

Yang, Y-H. and Chen, H.H. (2011) 'Ranking-based emotion recognition for music organization and retrieval', *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 19, No. 4, pp.762–774.

Yang, Y-H. and Chen, H.H. (2012) 'Machine recognition of music emotion: a review', *ACM Transactions on Intelligent Systems and Technology (TIST)*, Vol. 3, No. 3, p.40.

Yang, Y. and Liu, X. (1999) 'A re-examination of text categorization methods', *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.42–49.

Yang, Y-H., Lin, Y-C., Su, Y-F. and Chen, H.H. (2008) 'A regression approach to music emotion recognition', *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16, No. 2, pp.448–457.

Zaki, M.J. (2000) 'Scalable algorithms for association mining', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, No. 3, pp.372–390.

Zeng, Z., Pantic, M., Roisman, G.I. and Huang, T.S. (2009) 'A survey of affect recognition methods: audio, visual, and spontaneous expressions', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 1, pp.39–58.

Zhang, H., Zhao, Y., Cao, L. and Zhang, C. (2008) 'Combined association rule mining', *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp.1069–1074.

Zhang, Y., Zhang, L., Liu, Y. and Shi, Y. (2010) 'Mining intelligent knowledge from a two-phase association rules mining', *International Journal of Data Mining, Modelling and Management*, Vol. 2, No. 4, pp.403–419.