Journal of Intelligent Information Systems (2020) 55:51–66
https://doi.org/10.1007/s10844-019-00591-8

Sentiment analysis using rule-based and case-based reasoning

Petr Berka[1]

Received: 12 September 2019 / Revised: 4 December 2019 / Accepted: 5 December 2019 /
Published online: 22 January 2020

Abstract

Sentiment analysis becomes increasingly popular with the rapid growth of various reviews, survey responses, tweets or posts available from social media like Facebook or Twitter. Sentiment analysis can be turned into the question of whether a piece of text is expressing positive, negative or neutral sentiment towards the discussed topic and can be thus understood as a knowledge-based classification problem. A variety of knowledge-based techniques can be used to solve this problem. The paper focuses on two complementary approaches that originate in the area of AI (artificial intelligence), rule-based reasoning and case-based reasoning. We describe basic principles of both approaches, their strengths and limitations and, based on a review of literature, show how these approaches can be used for sentiment analysis.

Keywords Sentiment analysis · Opinion mining · Machine learning · Expert systems

1 Introduction

Sentiment analysis, also called opinion mining, is the task of finding the opinions of authors about specific entities (Liu 2012; Feldman 2013; Cambria et al. 2017). This task becomes increasingly popular with the rapid growth of various reviews, survey responses, tweets or posts available from social media like Facebook or Twitter. When understanding sentiment analysis as a classification problem where the aim is to label pieces of text by a sentiment orientation, we can use a variety of knowledge-based techniques to perform this task and a variety of machine learning methods to create the necessary classification models (knowledge) from data. And indeed, sentiment analysis is one of the typical text mining applications where data mining methods and algorithms are applied to unstructured texts (Srivastava and Sahami 2009; Aggarwal and Zhai 2012). We focus on the human-understandable form of knowledge (rules, cases) in our paper as we believe that the human insight into the used models can bring an extra benefit for the users.

✉ Petr Berka
berka@vse.cz

[1] University of Economics, W. Churchill Sq. 4, Prague, CZ 130 67, Czech Republic

The rest of the paper is organized as follows; Section 2 discusses the ideas of sentiment analysis, Section 3 gives the principles of rule-based and case-based reasoning considering two basic scenarios – knowledge created by domain experts and knowledge learned from data. This section also reviews existing approaches to sentiment analysis using these types of reasoning. Section 4 briefly discusses the main findings of our study and Section 5 concludes the paper.

2 Sentiment analysis

In the simplest case, sentiment analysis corresponds to binary classification task distinguishing between "positive" and "negative" attitude towards the discussed topic. When adding a "neutral" attitude, we are faced with multi-class classification and when expressing the sentiment in an ordered scale, we are solving a regression problem.

Not only the sentiment itself but also the emotions of the author can be analyzed. While the sentiment is usually categorized as positive, negative or neutral, emotions can be defined on a broader scale. Paul Ekman (1999) defines 6 basic emotions: happy, excited, tender, scared, angry and sad. Plutchik (2002) defines 8 basic emotions which come in pairs as positive and negative ones: joy versus sadness; anger versus fear; trust versus disgust; and surprise versus anticipation. Some authors even propose to use audial and visual information for sentiment analysis from multimodal content (Poria et al. 2016), so sentiment analysis can go beyond text mining only.

Sentiment can be analyzed at the document level, at the sentence level or at the aspect level. Document-level analysis is the basic form of sentiment analysis. Here the whole document is supposed to express the opinion of the author towards the discussed entity. But as a single document may contain different opinions about the same entity, sentence-level sentiment analysis gives a more detailed and more correct view. Document-level or sentence-level sentiment analysis works well if the document refers to a single "atomic" entity. Aspect based sentiment analysis is typical for analyzing product reviews, where products (e.g. computers, mobile phones, cars or cameras) have well-defined characteristics, that are assessed by the reviewers. Sentiment expressions are here related to different aspects of the discussed topic and classifying the whole review as either positive or negative towards the product thus may miss the point. An interesting form of sentiment analysis is comparative sentiment analysis; here rather than expressing his attitude towards a particular product, the reviewer compares it with another "referential" product (Feldman 2013).

To perform sentiment analysis automatically we must go through several steps: document collection, document preprocessing and sentiment classification. A general scheme of sentiment analysis as the classification is shown in Fig. 1.

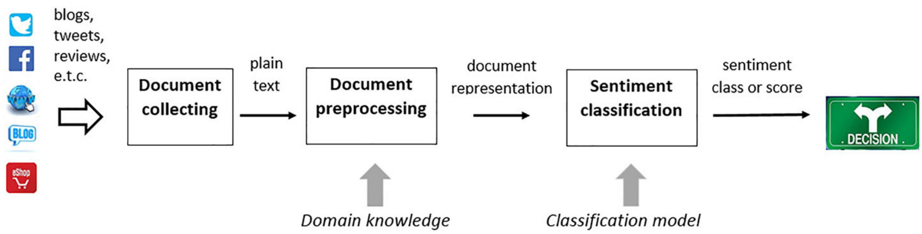

Fig. 1 General scheme of sentiment analysis as a classification task

Different forms of sentiment analysis (document-based, sentence-based, aspect-based) require different forms of textual data representation and preprocessing. For document-level sentiment analysis, the bag-of-word representation might be sufficient. Here, the text is expressed as a set of terms occurring in the document. This representation ignores the position of a term in the document but considers its frequency in the document (a document is represented as a vector of values derived from frequencies of terms in the document). A number of pre-processing steps like word segmentation (to identify distinct words), lemmatization (to remove inflectional ending), stemming (to turn the words into their basic form) and stop-word removing (to remove common words like conjunctions or prepositions that are not related to the content of the document) precede the bag-of-word representation. For sentence-level sentiment analysis, we first have to determine the subjective sentences, i.e. sentences that express the opinion of the author. Part-of-speech tagging that identifies the word categories can be used here to identify e.g. adverbs that are closely related to expressing opinions. Also, various sentiment lexicons can support this step. Then, a similar representation of sentences can be used as in document-level analysis, or the sentences can be represented by the words from the sentiment lexicons (such a representation can be used also for the document level). For aspect-level sentiment analysis, information extraction approaches are necessary. An example can be the named entity recognition, where the names of products, vendors or product attributes must be located in the text.

So like in a number of other text mining applications, also in sentiment analysis the analyzed data itself need not be sufficient and domain knowledge is necessary as well. In the case of sentiment analysis, the domain knowledge typically has a form of a sentiment lexicon that contains representative words for different opinion polarities. One group of lexicons categorize words into binary classes (positive or negative); examples of such lexicons are LIWC (www.liwc.net) or GI (www.wjh.harvard.edu/ inquirer). LIWC is a text analysis software that contains, among others, about 1000 words expressing either positive or negative emotions (Tausczik and Pennebaker 2010). GI is an early lexicon that contains more than 11000 words categorized into 183 categories. It contains about 2000 words labeled as positive and about 2300 words labeled as negative (Stone 1966). The second group of sentiment lexicons associate sentiment scores with opinion words. A widely used sentiment lexicon of this type is SentiWordNet (sentiwordnet.isti.cnr.it), an extension of WordNet in which about 147 000 WordNet synsets are associated to three numerical scores Obj(s), Pos(s) and Neg(s), describing how objective, positive, and negative the terms contained in the synset are (Esuli and Sebastiani 2006). The next example of a sentiment lexicon that associates scores with words is SenticNet (senticnet.net). SenticNet, a publicly available resource for concept-level opinion and sentiment analysis provides semantics and sentics (affective information) associated with 30000 common-sense concepts instead of single words. Unlike many other sentiment lexicons, SenticNet is automatically constructed by applying graph mining and dimensionality reduction techniques on the affective common-sense knowledge (Cambria and Hussain 2015). Not only sentiment lexicons play the role of domain knowledge for sentiment analysis tasks. Another example of used domain knowledge are hand-crafted extraction or grammar rules within NLP approaches (Jurafsky and Martin 2008).

The knowledge used for automated sentiment analysis can be obtained from domain experts or from the data (texts to be analyzed). In the latter case, a number of different data mining (text mining) and machine learning techniques can be used. The main drawback of using machine learning algorithms for classification is the necessity to have a sufficient amount of labeled (annotated) training examples. To assure this for sentiment classification can be a problem as, like in other text mining tasks, when using the basic bag-of-words representation, we have a significantly smaller number of examples (documents) than attributes (terms).

Various literature reviews report naïve bayesian classifiers (NB), support vector machines (SVM) and pointwise mutual entropy (PMI) as the most popular machine learning methods, followed by decision rules, k-nn methods and decision trees. NB classifier uses the independence assumption of input attributes t_i (terms in document) given the class c_j (sentiment direction) to compute the conditional probability $P(c_j|t_1, t_2, \ldots, t_n)$, that a document represented by terms $t_1, t_2, \ldots, t_n$ belongs to class c_j based on the product of probabilities $P(t_i|c_j)$ considering the contribution of each term ti independently on the others,

$$P(c_j|t_1, t_2, \ldots, t_n) = \frac{P(c_j) \prod_{i=1}^{n} P(t_i|c_j)}{P(t_1, t_2, \ldots, t_n)} \tag{1}$$

So the complexity of the learned model is only linear in the number of terms and thus NB classifiers are widely used for various text mining tasks. SVM learns a separating hyperplane in the space of transformed attributes. The position of the hyperplane is determined using similar examples that belong to different classes (so-called support vectors). The theory behind this method is rather complex - if the classes are not linearly separable in the space of original attributes, we must transform the attributes using so-called kernel functions – but the separating hyperplane itself has a good intuitive interpretation. PMI is rather a particular criterion than a classification method. PMI, like entropy or chi-square, just assesses the importance of a term t_i for classification of a document into class c_j using a formula derived from the co-occurrence of t_i and c_j in the collection of documents,

$$PMI(c_j, t_i) = \log \frac{P(c_j, t_i)}{P(c_j)P(t_i)} \tag{2}$$

We focus on rule-based and case-based approaches in the paper as we believe, that the interpretable models created here can bring additional benefits when compared to pure "black-box models".

3 Rule-based and case-based reasoning

Knowledge and their usage play a key role in automated decision support because many of the problems to be solved require extensive knowledge about the application domain. The term "knowledge" used here denotes a "piece of expertize" acquired from a domain expert of a respective field or learned from data and represented in a form suitable for automated processing.

Knowledge, its representation and application for automated reasoning are extensively studied in the area of AI (artificial intelligence). Two basic approaches, rule-based reasoning, and case-based reasoning have been proposed to create knowledge-based systems suitable for "intelligent" decision support. Rule-based reasoning (RBR) is closely related to the so-called expert systems, an important research topic in the 1970th and 1980th. A decade ago, the usefulness of the rule-based approach has been recognized in the business rules management systems that apply the basic principles of rule-based reasoning in a particular, business domain. Case-based reasoning (CBR) a counterpart to rule-based reasoning uses the idea that past decisions can be reused to find a solution for a new but similar problem.

Both RBR systems and CBR systems consist of two main parts, knowledge base and inference mechanism. Knowledge base (rule base or case base) contains the domain knowledge, inference mechanism, a domain-independent algorithm, is used for reasoning. The used knowledge representation formalism, of course, determines the respective inference methods.

3.1 Rule-based reasoning

3.1.1 Expert systems

An expert system is “an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution” (Feigenbaum 1979). An expert system (ES) has two main components; a domain-independent inference engine and a knowledge base that stores expert knowledge in a symbolic form. Other features of ES are the ability to process uncertain information and knowledge, dialogue mode of the consultation, and explanation abilities (Giarratano and Riley 1993).

The knowledge represented in expert systems typically has the form of IF-THEN rules. These rules have two basic semantics: procedural (as used in generative systems), i.e.

IF situation THEN action

or declarative (as used in diagnostic systems), i.e.

IF condition THEN conclusion.

The situation or condition is a conjunction of statements that must be fulfilled for the rule to be activated, the action is a list of actions that can be performed if the respective situation occurs and the conclusion is a statement that holds if the condition is true. Rules are used to represent knowledge in a generalized form. As shown in Fig. 2, a rule (the gray rectangle) covers a number of decision situations (points within this rectangle).

The rules are used by the inference engine. In generative systems, the inference engine performs actions that are applicable to the content of working memory, in diagnostic systems, the inference engine derives goals from the rules and the answers to the questions. To find an applicable rule, the inference engine has to search the knowledge base using either forward chaining or backward chaining. Forward chaining is reasoning from facts to the conclusions that can be derived from those facts. Backward chaining is reasoning from goals to be proved to the conditions which support the goals. Only forward chaining can be used in generative systems as the knowledge base does not explicitly contain the goals,

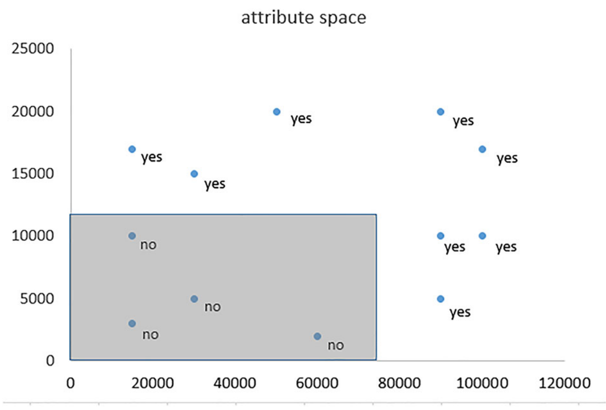

Fig. 2 Expressive power of a rule

both forward and backward chaining can be used in diagnostic systems. It may happen, that more rules are identified as applicable at the same moment. In generative systems, as the actions of different rules can have a contradicting effect on the facts in working memory, a conflict occurs. This conflict must be resolved to identify a single rule that will be applied. More options are available in diagnostic systems; only one rule is applied like in generative systems or all applicable rules are used.

When the applicable rule is found, it is used either to perform the actions (generative systems) or to derive the conclusion (diagnostic systems). The application of a rule in diagnostic systems is based on deduction: if the condition of a rule is true we can infer that the conclusion is true as well. The application of a declarative rule can be enriched by uncertainty processing. When using uncertainty, both the rule itself and the condition can be true only with some degree and thus the derived conclusion is true with some degree as well. When using uncertainty, typically all rules that can be used to derive a conclusion are applied and their particular (uncertain) contributions are combined.

3.1.2 Rule-based classifiers

Rule-based classifiers are typical models created from data using machine learning algorithms. One possibility how to create rules is to transform a decision tree into a set of decision rules in such a way, that each path through the tree from the root to a leaf is turned into one rule, where non-terminal (branching) nodes correspond to the condition of the rule and the labeled leaf corresponds to the conclusion of the rule. There is also a number of algorithms that learn decision rules directly from data. Most of them use the set covering approach in a way called "separate and conquer". During each pass of the main cycle of the algorithm, some examples of the target concept are covered by a single rule and then removed from the training data. This process terminates when there are no uncovered examples in the training set (Fig. 3). A new candidate rule can be created either by rule specialization (adding a new attribute-value pair to the condition of the rule, see e.g. (Clark and Niblett 1989)), or by rule generalization (removing an attribute-value pair from the condition of a rule, see e.g. (Michalski 1969)). The application of the created rule set for classification is very easy and straightforward. The set covering approach assures, that there is only one applicable rule for an example to be classified, so the inference engine has to go sequentially through the list of rules to find this rule in a way similar to forward chaining.

Another approach to rule induction learns weighted decision rules; here each rule is extended by a certainty factor (weight) expressing how strongly the rule contributes to the classification. In this approach the covered examples are not removed during learning, so an example can be covered by more rules. Thus more rules can be used during classification and their contributions have to be combined into the final classification of an example in a similar way that is used for uncertainty processing in expert systems (Berka and Ivánek 1994).

set covering algorithm

1. create a rule that covers some examples of one class and does not cover any examples of other classes
2. remove covered examples from training data
3. if there are some examples not covered by any rule, go to step 1

Fig. 3 Set covering algorithm

The main difference between rules learned from data and rules defined (for expert systems) by domain expert is that rules learned from data directly point from input attributes to classes and no chains of rules can appear in the rule set, i.e. it cannot happen that a statement that appears in one rule in the conclusion appears in another rule in the condition. So the inference in the set of rules learned from data typically has the form of forward chaining as used in expert systems.

3.1.3 Strengths and limitations of rule-based reasoning

The knowledge in the symbolic IF-THEN form is modular and is easy to understand so the knowledge base can be easily interpreted and updated. The rule-based reasoning is transparent and based on a widely used logic formalism. But to obtain the knowledge from domain experts in the form of rules can be very difficult as the experts have to generalize from their experience. The problems related to the often tedious and time-consuming elicitation of experts' knowledge are reported in the AI community as the "knowledge acquisition bottleneck". Especially hard becomes the knowledge acquisition task if the rules should express uncertainties; it is questionable to force the expert to quantify a weak and sometimes not very clear relationship between conditions and conclusions. The numbers (weights, uncertainty factors) assigned to the rules by domain experts need not be consistent with the uncertainty processing methods used during the inference. Another limitation of RBR systems is that the knowledge cannot be automatically updated based on ongoing consultations with the system. The knowledge base thus remains intact until a new knowledge elicitation from the domain expert is performed.

3.2 Case-based reasoning

3.2.1 Case-based reasoning systems

Case-based reasoning builds on Schank's model of dynamic memory from the early 1980th (Schank 1982). The knowledge in CBR systems is represented in the form of previously solved problems called cases. The inference uses analogical reasoning: to solve a new problem most similar case (cases) is (are) used to find the solution. The ratio behind this approach is the idea that successful past solutions of a similar problem can help to solve the new situation.

To express knowledge as "typical" solved problems is more natural for domain experts. A knowledge base in the form of cases can thus be created easier than a knowledge base in the form of rules. Cases represent the domain knowledge in a specific, "local" form. As shown in Fig. 4, each case represents only a small subset of similar situations.

Cases represent typical situations (problems and their solutions) that can occur in the application area. A case can have a form of a vector of numeric values but a case can also have a form of a structured object that lists possible symptoms, questions to be asked during inference and recommended solutions for the problem the case is related to. The key issue in CBR systems is how to express the distance (or similarity) between cases. For cases represented as numeric vectors, we can use measures used for cluster analysis (Euclidean distance, Hamming distance, Chebyshev distance) or measures used for text mining (cosine similarity, Jaccard similarity, Dice similarity). But for cases represented also using categorical attributes or for cases having even a complex structure, those standard measures cannot be used. The simplest distance measure for categorical attributes is the "overlap" distance that can distinguish only between exact match and "some" difference of

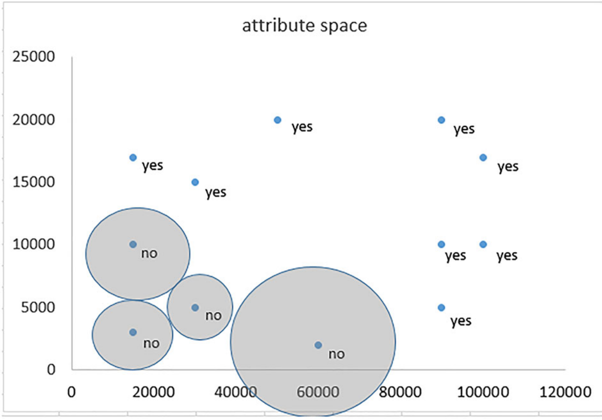

Fig. 4 Expressive power of cases

two values. So to make CBR applicable also to problems in which the case representation goes beyond simple numeric vectors, new distance measures are necessary. Concerning categorical attributes, it is for instance possible to consider a taxonomy of values that helps to define how "far" two values can be. Another method described in (Berka 2011) is based on the idea that the original case representation in the form of a list of values of categorical attributes is transformed into a vector of numeric truth values assigned to every possible value of each attribute in a way similar to binarization of categorical attributes in machine learning. Such transformed representation then enables to use distance measures designed for numeric data.

The basic idea of the case-based inference is to find for the current problem the most similar case(s) solved in the past and to act accordingly. The whole inference cycle of a CBR system usually consists of four steps (Aamodt and Plaza 1994):

– retrieve the most similar case(s),
– reuse the case(s) to attempt to solve the problem,
– revise the proposed solution if necessary,
– retain the new solution as a part of a new case.

3.2.2 Instance-Based Learning

Instance-based learning (lazy learning, memory-based learning) is a group of machine learning methods that learns from data the knowledge suitable for classification in the form of typical examples from the training set. So instead of performing some generalization from data, the learning algorithm just stores "important" examples. There is a variety of possibilities on how to choose such examples within this basic general idea. The possibilities range from storing the whole training data to storing only a few examples representing a particular class (it can be shown, that for linearly separable classes, one example is sufficient to represent a class).

To classify a new example, like in general Case-Based Reasoning, the definition of similarity between examples is the key concept. As the examples are usually represented by attribute-value pairs (values of attribute vectors) the used similarity (or distance) measures can be less complex than in general case-based approach. When classifying an example, the

k-nearest neighbor algorithm

for example $\mathbf{x_x}$

1. find $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}$ k nearest neighbors from instance base
2. assign $y_x = y_j \iff j = argmax_i \sum_{l=1}^{k} \delta(y_i, y_l)$, where $\delta(y_i, y_l) = 1$ iff $y_i = y_l$ else $\delta(y_i, y_l) = 0$

Fig. 5 k-nearest neighbor algorithm with uniform voting

k-nearest neighbor rule is typically used; k cases that are nearest (most similar) to the new example are found and then decide on the class for this example. The decision is based on voting (uniform or weighted) of the k neighbors (Fig. 5).

3.2.3 Strengths and limitations of case-based reasoning

Because domain experts are asked to provide knowledge in the form of particular situations instead in the form of rather general IF-THEN statements, the knowledge acquisition process can be very fast. The possibility to automatically update the case base during the inference also speeds up the development and maintenance of the used knowledge. Similar automatic modification of the knowledge base during usage of an RBR system is generally not possible. The main problems related to CBR are the need for complex distance (similarity) measures and the necessity of fast access to case bases that sometimes can be very huge.

3.3 Comparing both approaches

There is a great conceptual difference between rule-based and case-based approach in the way how knowledge is represented and used for reasoning. The knowledge in the form of particular cases and the way how a case-base is updated better corresponds to the way how humans learn from their experience. Also, the reasoning by analogy principle, implemented in CBR systems better corresponds to human decision-making. The knowledge in the form of rules and deductive reasoning as implemented in RBR systems is on the contrary more typical in various formal systems grounded in logic and mathematics. Further differences between both approaches are at the technical level. To find the relevant piece of knowledge during reasoning, RBR systems apply either backward (goal-driven) or forward (data-driven) chaining to find an applicable rule, CBR systems search the indexed case-base to find a suitable case(s). Once a suitable piece of knowledge is found, it is applied for reasoning. The application of a single rule in RBR systems corresponds to retrieving and reusing a single case in CBR systems. To combine contributions of more applicable rules in RBR systems corresponds to the k-nearest neighbor method used in CBR systems.

When creating the models from data, we have to specify the input parameters for the corresponding machine learning algorithms. The parameters for rule induction algorithms are usually the maximal length of a rule (i.e. maximal number of statements in the condition of a rule), the minimal support (i.e. minimal number of examples that fulfill both condition and conclusion of a rule) and the minimal confidence (i.e. minimal percentage of examples that fulfill conclusion of a rule within group of examples that fulfil condition of this rule). Typical values are not more than 10 for the maximal length, several percent for the minimal support and values between 70% and 100% for minimal confidence. When using instance-based learning, we must specify the value for the parameter “k”, i.e. the number of neighbors that contribute to the classification resp. the number of clusters the examples are assigned to. When specifying the value of "k" we make a rather strong assumption about the expected results (especially if "k" is the number of clusters) so a reasonable approach is to make a series of experiments with a varying value of "k".

Both RBR systems and CBR systems can be used to solve the same decision-making problem, so they can be considered as competitive methods. But they can also be combined either in a sequence or in parallel with the aim to obtain a more precise and more robust solution than that provided by only one of the approaches. The sequential combination means that either RBR system is used first, or CBR system is used first, or there is a switching between both methods. The parallel combination of RBR and CBR is nothing else but creating an ensemble of classifiers as used in the area of machine learning and data mining (Bauer and Kohavi 1999).

3.4 Applications to sentiment analysis

Let's have a look at applications of rule-based and case-based reasoning to sentiment analysis. As we are discussing rule-based and case-based reasoning and in both situations we can acquire the used knowledge from domain experts or learn from data, there are four possibilities how to create and use the suitable knowledge: knowledge in form of rules created by domain experts, knowledge in form of cases created by domain experts, knowledge in form of rules learned from data and knowledge in form of cases learned from data. Table 1 shows how the approaches discussed below fit into these four categories.

Qiu and his co-authors perform rule-based sentence-level sentiment analysis within an approach to dissatisfaction oriented advertising (Qiu et al. 2010). Their idea is to show an advertisement on a web page if this page is associated with topic words towards which the consumer has a negative attitude. The proposed strategy performs three steps. First, topic words the advertisement may be about are extracted from web pages. Then, sentiment towards the extracted topic words is determined and those topic words with negative sentiment are chosen as advertising keywords. In the third step, an advertisement matching the selected advertising keywords is presented. Consider e.g. a text criticizing alcohol consumption. Then topic words would be alcohol and consumption. As the text criticizes drinking alcohol, the sentiment toward the topic words is negative. The matching advertisement will thus advertise a non-alcoholic beverage. Manually created rules that express

Table 1 Application of rule-based and case-based reasoning to sentiment analysis

	Rule-based approach	Case-based approach
Experts	Qiu et al., (2010) Reckman et al., (2013) Romanyshin, (2013) Poira et al., (2014) Chikersal et al., (2015)	Xi et al., (2012)
Data	Rashid et al., (2013) Liu et al, (2015) Ahmed, Danti, (2016) Prabowo, Thelwall, (2013)	Xi et al., (2012) Ohana et al., (2012) Zhou et al. (2015)

relations between a sentiment word S and a topic word T are used for the topic words extraction step. A priority reflecting the certainty of the association between S and T is added to each rule. The priorities allow to handle situations when more extraction rules can be applied; rules are chosen in the descending order of their priorities.

Reckman et al. (2013) use hand-written rules to identify the overall sentiment and sentiment of ambiguous phrases in tweets (they consider positive, negative or neutral sentiment.) Their rules match words or sequences of words (phrases) from their own domain independent taxonomy (lexicon). The rule matching is supported by a morphological dictionary to enable morphological expansion. The rules are organized into lists, where longer (more specific) rules applicable to phrases have higher priorities. They also introduce weights to distinguish the importance of the rules; a list of weak positive and weak negative rules was created that is used only if stronger rules are not applicable.

Romanyshin uses a rule-based approach to clause-level sentiment analysis of reviews in the Ukrainian language (Romanyshin 2013). At first, the plain text is part-of-speech tagged and split into clauses. Then, both sentiment (positive, negative, invertor, intensifier) and emotion (anger, disgust, fear, joy, sadness, surprise) are assigned to each word in a clause using a lexicon for the Ukrainian language. If no matching item is found in the lexicon, the word is assigned the sentiment category “neutral” and the emotion category “none”. Two types of manually created rules are then used to evaluate the final sentiment of a clause; context-independent rules consider only the sentiment categories (e.g. IF intensifier AND positive THEN very_positive), context-dependent rules consider both words from the lexicon and sentiment categories. The approach has been tested on reviews of Ukrainian restaurants.

Poria et al. present a rule-based approach to detect explicit aspects and implicit aspect clues (indirect aspect expressions) from consumer reviews (Poria et al. 2014). They designed an aspect parser that uses two groups of manually created rules: rules for sentences having subject verb and rules for sentences that do not have subject verb. The considered aspect categories are functionality, weight, price, appearance, behavior, performance, quality, service, size. To determine a sentiment related to the aspect categories, SenticNet is used as a concept-level opinion lexicon.

Chickersal et. al use a rule-based approach to analyze sentiments of tweets (Chikersal et al. 2015). Their idea is to treat sentiment analysis as a multi-class classification problem because they assign each tweet to “positive”, “negative” or “unknown” class. Two types of rules are used in the presented approach: emoticon-related and sentiment lexicon-related. The manually created rules are applied using following strategy: a tweet containing only positive emoticons is classified as “positive”, a tweet containing only negative emoticons is classified as “negative”. The sentiment lexicon-related rules must be used for tweets without any emoticons. In this case, a tweet containing more than two positive words and no negation or negative words is classified as “positive”, a tweet containing more than two negative words and no positive words is classified as “negative”. Tweets for which no rule is applicable are classified as “unknown”.

Rashid et al. analyze opinions from student feedback data on faculty performance (Rashid et al. 2013). Their idea is to extract features (aspects) together with related opinion words. So they performed aspect-level sentiment analysis, where the aspects are related to the quality of the educational process (the extracted features are e.g. lecture preparation, lecture delivery or course material). POS tagging is applied to tag the features as nouns and opinion words as adjectives. To find frequently occurring features together with the respective opinion words, the association rule mining algorithm apriori is applied to the data. The found rules are tested using some testing data to identify rules with high accuracy when assigning opinion word to a feedback feature, those rules can then be used for the opinion extraction from new feedback data.

Liu et al. (2015) try to solve the problem of how to select (given a set of aspect extraction rules, a set of opinion words and a set of labeled reviews) a subset of extraction rules that can be used to extract aspects from reviews across domains. Their algorithm works in three steps. In the rule evaluation step quality of each rule is assessed based on the precision and recall when applying this rule to the labeled reviews. In the rule ranking step, the rules are ordered by their precision, rules with the same precision are ordered by their recall. A greedy selection of rules is the last step; rules are added from the ranked list one by one into the resulting list of rules. Whenever a new rule is added, the resulting list created so far is evaluated by the F-score (a harmonic average of precision and recall). After adding all rules to the resulting list, this list is pruned by removing rules that do not improve the F-score. The authors distinguish three types of extraction rules: type 1 rules use opinion words to extract aspects, type 2 rules use aspects to extract aspects, type 3 rules use aspects and opinion words to extract new opinion words. The three types of rules are processed separately by the proposed algorithm. The initial set of extraction rules is created manually, but as the proposed algorithm then works automatically, we classify this paper as an approach creating rules from data.

Ahmed and Danti are using rule-based machine learning algorithms for sentiment analysis (Ahmed and Danti 2016). They first classify opinion words using the SentiWordNet lexicon into seven classes: strong-positive positive, weak-positive, neutral, weak-negative, negative and strong-positive. The classified opinion words that are found in a document are then used as inputs for the subsequent learning step. The proposed approach is applied to on-line book reviews and on-line political polls.

Prabowo and Thelwall (2013) conducted a comparative study to evaluate the effectiveness of different classifiers for sentiment analysis on the document level using bag-of-word representation. Their work focuses on rule-based methods and SVM. They consider rules in the form IF antecedent THEN consequent where the antecedent is a conjunction of words and consequent is either positive or negative sentiment related to the antecedent. The rules have been obtained using different machine learning approaches: by using sentiment lexicon (here the positive and negative words from a sentiment lexicon are considered for an antecedent), by querying the web (here the query consists of words from an antecedent of a rule and a sentiment bearing word), and by some standard rule learning algorithms. They also propose to use more classifiers in a sequence; if one classifier fails to classify the document, the document is passed to the next one.

Xi at al. (2012) analyze a micro-blog opinion corpus to construct a quick emergency response model. They consider sentiment intensity of micro-blogs on all three levels; document level, sentence level, and word level. To assess the word emotional strength, they use a Chinese lexicon HowNet. The emotional strength of a sentence is computed from the emotional strength of corresponding words and the emotional strength of a document is computed from the emotional strength of corresponding sentences. Based on this, they create a micro-blog case library referring to crisis situations of various categories (political, economical, cultural, social). An emergency plan is associated with every case. When looking for an appropriate emergency plan for a new micro-blog, CBR inference engine first retrieves from the case library micro-blog cases based on emotional strength similarity. Then, the semantic similarity of the keywords is used to narrow the results. The response plan is then modified until approved by the expert. So the case base is constructed semi-automatically and we thus consider this approach in Table 1 as partially manual and partially automated.

A case-based approach to sentiment classification of customer reviews as positive or negative is described in Ohana et al. (2012). Here, the cases are learned from data using supervised techniques. To create the case base, labeled training documents are evaluated by five different sentiment lexicons. Correct classification of a training document by at least one lexicon results in creating and adding the respective case into the case base. Each case has two main parts, case description, and case solution. The case description represents the basic characteristics of the document, the case solution represents successful past predictions done by the case. A case is represented by two types of features. The first type encodes document statistics like total words and sentences or average sentence size, the second type encodes writing style like the ratio of unique words or the stop-words ratio. All features are numeric and are normalized using min-max normalization, so Euclidean distance can be used to measure the similarity. To classify a new document k most similar cases (k-neighbors) are retrieved (in the reported experiments, k was set to 1, 2 or 3) and the lexicons related to the cases' solutions are reused. The resulting sentiment score of the classified document is obtained by querying the lexicons of the retrieved k-neighbors for sentiment information of terms present in the document.

Zhou et al. combine sentiment analysis and analogical (case-based) reasoning to obtain so-called latent customer needs from product reviews (Zhou et al. 2015). The first step of their approach consists of segmenting reviews into sentences and labeling each sentence as positive or negative using user-provided ratings. Then, association rules mining is used to extract product attributes and to identify use-cases that will enter the case base. In the third step, a fuzzy SVM model is created to predict the sentiment of product reviews. In the last step, CBR is used for analogical reasoning between ordinary and extraordinary use cases to elicit latent customer needs. Ordinary use cases are cases (reviews) that express the customer needs explicitly while extraordinary use cases do not. During the CBR inference, ordinary use cases are reused and customized to elicit latent customer needs automatically. But as domain experts are involved in validating the elicited needs we can again consider the approach as partially manual.

4 Discussion

Rule-based and case-based reasoning approaches have a long tradition in creating knowledge-based decision support systems in a number of different application areas like medical diagnosis, technical diagnosis, business, finance or chemistry. The presented examples of rule-based and case-based reasoning applied to sentiment analysis illustrate the variety of particular tasks, where these techniques have been used. As can be seen from the reviewed work, rule-based approaches are used both to determine the aspect-sentiment relationship and to classify the sentiment orientation of whole documents or sentences. In the former case, rules can be created manually or with the support of rule-learning algorithms, in the latter case, machine learning algorithms are mostly used. Concerning the applications of case-based reasoning to sentiment analysis, in most cases, the models are created from data.

There is no general guideline on whether to prefer rule-based or case-based reasoning. When compared with rule-based systems, knowledge acquisition for case-based reasoning systems is easier and faster, but case-based reasoning requires fine-tuned distance measures. So the decision usually depends on problems related to obtaining the knowledge for a particular task.

We cannot find any explicit comparison of rule-based and case-based approaches in the reviewed papers and it does not make much sense to compare classification accuracies achieved on different data. We cannot draw any general claims about superiority of an algorithm without a detailed empirical comparison. Papers (Prabowo and Thelwall 2013; Ravi and Ravi 2015) that report on empirical comparison of various machine learning methods used for sentiment analysis show, that the classification accuracy of rule-based or case-based (k-NN) approaches is comparable to classification accuracy of the other methods This confirms the well known no-free-lunch theorem saying that there exists no algorithm that will systematically outperform all the other algorithms. So the understandability of classification models based on rules or cases can bring an extra benefit to the users, an insight into the decision-making process.

5 Conclusions

Increasing attention is paid in the machine-learning community to interpretable machine learning. Here, interpretation is understood as the process of giving explanations to humans (Otte 2013; Doshi-Velez and Kim 2017). So the created models should not only be accurate but also understandable for the domain experts and end-users. Some types of models are interpretable "per se", some types of models need complex transformations to become interpretable. Following the idea of interpretability, we intentionally narrow the scope of our review to methods that provide interpretable models directly. So we omit a number of examples where neural networks, SVM o probabilistic approaches are used for sentiment analysis.

The reviewed papers show how to use either rule-based or case-based approach separately. A combination of both approaches is reported from various domains (Cabrera and Edye 2010; Lee 2008; Yang and Shen 2008) but, with the exception of using rule-based and case-based models (among other models) when creating ensembles of classifiers (Wang et al. 2014), we haven't found an attempt to integrate rule-based and case-based approaches for sentiment analysis in an interpretable way. So following the general idea of using both approaches in a sequential way we propose an example scenario in which the system tries to retrieve similar cases first to solve the problem and only if no such case can be found (i.e. there is no sufficiently similar case in the case-base), the general rules are applied. This way of reasoning, that well corresponds to the human decision-making process, can be inspiring for sentiment analysis.

Acknowledgements The paper was processed with the contribution of long term institutional support of research activities by Faculty of Informatics and Statistics, University of Economics, Prague.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations and system approaches. *AI Communications*, *7*(1), 39–59.

Aggarwal, C., & Zhai, C. (Eds.) (2012). *Mining text data*. Berlin: Springer.

Ahmed, S., & Danti, A. (2016). Effective sentimental analysis and opinion mining of web reviews using rule based classifiers. In Behera, H.S., & Mohapatra, D.P. (Eds.) *Computational intelligence in data mining* (pp. 171–197). India: Springer.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Machine Learning*, *36*(1/2), 105–139.

Berka, P. (2011). NEST: a compositional approach to rule-based and case-based reasoning. *Advances in Artificial Intelligence*.

Berka, P., & Ivánek, J. (1994). Automated knowledge acquisition for PROSPECTOR-like expert systems. In Bergadano, J., & deRaedt, L. (Eds.) *Proceedings of European conf. on machine learning ECML'94* (pp. 339–342). Springer LNAI 784, Springer.

Cabrera, M., & Edye, E. (2010). Integration of rule based expert systems and case based reasoning in an acute bacterial meningitis clinical decision support system. *International Journal of Computer Science and Information Security*, *7*(2), 112–118.

Cambria, E., Das, D., Bandyopadhyay, S., Feraco, A. (Eds.) (2017). *A practical guide to sentiment analysis*. Berlin: Springer.

Cambria, E., & Hussain, A. (2015). Sentic computing. A Common-Sense-Based Framework for Concept-Level Sentiment Analysis Springer.

Chikersal, P., Poria, S., Cambria, E. (2015). SeNTU: sentiment analysis of tweets by combining a rule-based classifier with supervised learning. In *Proceedings of the 9th Int. workshop on semantic evaluation, Denver* (pp. 647–651).

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*, 261–283.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv:1702.08608.

Ekman, P. (1999). Basic emotions. In Dalgleish, T., & Power, M.J. (Eds.) *Handbook of cognition and emotion* (pp. 45–60). New York: Wiley.

Esuli, A., & Sebastiani, F. (2006). Sentiwordnet: a publicly available lexical resource for opinion mining. In *Proceedings fifth international conference on language resources and evaluation LREC*, (Vol. 6 pp. 417–422).

Feigenbaum, E.A. (1979). Themes and case studies of knowledge engineering. In Michie, D. (Ed.) *Expert systems in the micro electronic age*: Edinburgh University Press.

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, *56*(4), 82–89.

Giarratano, J., & Riley, G. (1993). *Expert systems. Principles and programming*. Boston: PWS Publishing Company.

Jurafsky, D., & Martin, J.H. (2008). *Speech and language processing*. Englewood Cliffs: Prentice Hall.

Lee, G.H. (2008). Rule-based and case-based reasoning approach for internal audit of bank. *Knowledge-Based Systems*, *21*, 140–147.

Liu, B. (2012). Sentiment analysis and opinion mining. Morgan & Claypool Publishers.

Liu, Q., Gao, Z., Liu, B., Zhang, Y. (2015). Automated rule selection for aspect extraction in opinion mining. In *Proceedings of the 24th international joint conference on artificial intelligence (IJCAI 2015)* (pp. 1291–1297).

Michalski, R.S. (1969). On the quasi-minimal solution of the general covering problem. In *Proceedings 5th Int. symposium on information processing FCIP'69, Bled, Yugoslavia* (pp. 125–128).

Ohana, B., Delany, S.J., Tierney, B.A. (2012). Case-based approach to cross domain sentiment classification. In Agudo, B.D., & Watson, I. (Eds.) *Case-based reasoning research and Development ICCBR 2012*. Berlin: Springer.

Otte, C. (2013). Safe and interpretable machine learning, a methodological review. In Moewes, C. (Ed.) *Computational intelligence in intelligent data analysis* (pp. 111–122). Berlin: Springer. SCI 445.

Plutchik, R. (2002). Emotions and life: perspectives from psychology, Biology, and Evolution. Amer Psychological Assn, USA.

Poria, S., Cambria, E., Howard, N., Huang, G.-B., Hussain, A. (2016). Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing*, *174*, 50–59.

Poria, S., Cambria, E., Ku, L.W., Gui, C., Gelbukh, A. (2014). Rule-based approach to aspect extraction from product reviews. In *Proceedings of the second workshop on natural language processing for social media, Dublin* (pp. 28–37).

Prabowo, R., & Thelwall, M. (2013). Sentiment analysis: a combined approach. *Journal of Infometrics*, 143–157.

Qiu, G., He, X., Zhang, F., Shi, Y., Bu, J., Chen, C. (2010). DASA dissatisfaction-oriented advertising based on sentiment analysis. *Expert Systems with Applications*, *37*, 6182–6191.

Rashid, A., Asif, S., Butt, N.A., Ashraf, I. (2013). Feature level opinion mining of educational student feedback data using sequential pattern mining and association rule mining. *Int. J. of Computer Applications*, *81*(10), 31–38.

Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, *89*, 14–46.

Reckman, H., Baird, C., Crawford, J., Crowell, R., Micciulla, L., Sethi, S., Veress, F. (2013). Rule-based detection of sentiment phrases using SAS sentiment analysis. In *Proceedings second joint conference on lexical and computational semantics, Atlanta* (pp. 513–519).

Romanyshin, M. (2013). Rule-based sentiment analysis of ukrainian reviews. *Int. J. of Artificial Intelligence & Applications*, *4*(4), 103–111.

Schank, R. (1982). *Dynamic memory a theory of learning in computers and people*. New York: Cambridge University Press.

Srivastava, A.N., & Sahami, M. (2009). Text mining: classification, clustering and applications. Chapman & Hall/CRC Press.

Stone, P.J. (1966). *The general inquirer. a computer approach to content analysis*. Cambridge: MIT Press.

Tausczik, Y.R., & Pennebaker, J.W. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, *29*(1), 24–54.

Wang, G., Sun, J., Ma, J., Xu, K., Gu, J. (2014). Sentiment classification: the contribution of ensemble learning. *Decision Support Systems*, *57*, 77–93.

Xi, M., Wu, H., Niu, Z. (2012). A quick emergency response model for microblog public opinion crisis based on text sentiment intensity. *Journal of Software*, *7*(6), 1413–1420.

Yang, M., & Shen, Q. (2008). Reinforcing fuzzy rule-based diagnosis of turbomachines with case-based reasoning. *Int. J. of Knowledge-Based and Intelligent Engineering Systems*, *12*(2), 173–181.

Zhou, F., Jiao, R.J., Linsey, J.S. (2015). Latent customer needs elicitation by use case analogical reasoning from sentiment analysis of online product reviews. *Journal of Mechanical Design, 137*.