Fuzzy Clustering for Semi-Supervised Learning—Case study: Construction of an Emotion Lexicon

Soujanya Poria,[1] Alexander Gelbukh,[2] Dipankar Das,[3] and Sivaji Bandyopadhyay[1]

[1] Computer Science and Engineering Department,
Jadavpur University, Kolkata, India
[2] Center for Computing Research, National Polytechnic Institute,
Mexico City, Mexico
[3] Computer Science and Engineering Department, National Institute of Technology (NIT),
Meghalaya, India
soujanya.poria@gmail.com, www.gelbukh.com
dipankar.dipnil2005@gmail.com, sbandyopadhyay@cse.jdvu.ac.in

Abstract. We consider the task of semi-supervised classification: extending category labels from a small dataset of labeled examples to a much larger set. We show that, at least on our case study task, unsupervised fuzzy clustering of the unlabeled examples helps in obtaining the hard clusters. Namely, we used the membership values obtained with fuzzy clustering as additional features for hard clustering. We also used these membership values to reduce the confusion set for the hard clustering. As a case study, we use applied the proposed method to the task of constructing a large emotion lexicon by extending the emotion labels from the WordNet Affect lexicon using various features of words. Some of the features were extracted from the emotional statements of the freely available ISEAR dataset; other features were WordNet distance and the similarity measured via the polarity scores in the SenticNet resource. The proposed method classified words by emotion labels with high accuracy.

1 Introduction

We consider the classification task, which consists in assigning to each object one label from a predefined inventory of labels. Typical supervised classification task consists in learning the classification rules from a test set of labeled examples, in order to later apply the learned rules to previously unseen examples and thus assign them a specific label.

In contrast, unsupervised classification task, called also clustering, consists in finding internal structure in a set of data items in order to group them together. Such grouping can be interpreted as assigning the items category labels, where belonging of two items to the same group is interpreted as assigning them the same label. Note that unlike in the supervised learning, the unlabeled data items, or at least a large amount of such items, are available to the classifier at the training stage and are used for training, i.e., fining regularities in the dataset. On the other hand, no specific predefined inventory of labels is used in unsupervised classification: the groups themselves are considered equivalent to the labels.

Semi-supervised learning lies in the middle between these two extremes. The task consists in using both a small number of labeled examples and a large number of unlabeled examples in order to learn to assign labels from a predefined inventory to unlabeled examples. The advantage of this task over unsupervised classification is in assigning specific labels, while advantage over supervised classification is in using much smaller training labeled dataset, with the lack of information from labeled examples being compensated by the information extracted from a large set of unlabeled examples.

In this paper we propose a method for semi-supervised learning, which consists in fuzzy clustering the large set of unlabeled examples, and then using the information on the found fuzzy clusters as additional features for supervised learning from labeled examples.

In order to evaluate our proposed method, we applied it to the task of building a large emotion lexicon by extending emotion labels from a small seed labeled lexicon to a larger set of words. Building emotion lexicons is currently a very important task. While emotions are not linguistic entities, the most convenient access that we have to them is through the language (Strapparava and Valitutti, 2004). Huge bodies of natural language texts in Internet contain not only informative contents but also such information as emotions, opinions, and attitudes. Analysis of emotions in natural language texts and other media is receiving considerable and rapidly growing interest from the research community under the umbrella of subjectivity analysis and affective computing.

The majority of subjectivity analysis methods related to emotions are based on text keywords spotting using lexical resources. Various techniques have been proposed for constructing dictionaries of sentiment-related words. For emotion detection, the Affective Lexicon (Strapparava and Valitutti, 2004), a small well-organized dictionary with affective annotation, is currently one of the most widely used resources.

The aspects that govern the lexical-level semantic orientation depend on natural language context (Pang *et al*., 2002), language properties (Wiebe and Mihalcea, 2006), domain pragmatic knowledge (Aue and Gamon, 2005), time dimension (Read, 2005), colors and culture (Strapparava and Ozbal, 2010), as well as many other aspects. Combining all these aspects of emotion orientation is related with human psychology and is a multifaceted problem (Liu, 2010). Although a word may evoke different emotions in different contexts, an emotion lexicon is a useful component for any sophisticated emotion detection algorithm (Mohammad and Turney, 2010) and is one of the primary resources to start with.

The rest of the paper is organized as follows. In Section 2, we discuss related work. An overview of the algorithm is given in Section 3. Section 4 presents the fuzzy clustering step of the algorithm, and Section 5 the final hard clustering step. Section 6 outlines an application of the proposed algorithm to a problem of constructing of an emotion lexicon via semi-supervised learning and presents the experimental results. Finally, Section 7 concludes the paper.

2 Related Work

A number of research works aimed to create subjectivity and sentiment lexica (Hatzivassiloglou and McKeown, 1997; Wiebe, 2000; Riloff *et al*., 2003; Baroni and Vegnaduzzo, 2004; Kamps *et al*., 2004; Hu and Liu, 2004; Andreevskaia and Bergler, 2007; Voll and Taboada, 2007; Banea *et al*., 2008; Baccianella *et al*., 2010). Several researchers have contributed to the study of semantic orientation of words (Kobayashi *et al*., 2001; Turney *et al*., 2003; Takamura *et al*., 2005).

However, most of these resources give coarse-grained classification (e.g., *positive*, *negative* or *neutral* sentiment). Other than WordNet Affect and General Inquirer,[1] we are not aware of widely used lexical resources for fine-grained emotion analysis. In particular, the SenticNet resource (Cambria *et al*., 2010; Cambria and Hussain, 2012) currently provides only polarity information but not specific emotion labels.

A number of other related research attempts for detecting emotions are found in the literature. Elliott (1992) considered direct emotion denoting words. Read (2005) used term co-occurrences with emotion seeds. Sidorov and Castro-Sánchez (2006) used a linguistic-based approach. Neviarouskaya *et al*. (2009) hand-crafted rules for detecting emotions. The machine learning approach by Alm *et al*. (2005) used a large number of emotion-related features, including emotion words. Recently, the application of "Mechanical Turk" for generating emotion lexicon (Mohammad and Turney, 2010) was shown to be a promising research direction.

Some results (Yu *et al.*, 2003; Awad *et al*., 2004; Boley and Cao, 2004) show that clustering technique can help to decrease the complexity of Support Vector Machine (SVM) training. However, building the hierarchical structure in these algorithms is computationally expensive. Cervantes *et al*. (2006) presented an SVM classification algorithm based on fuzzy clustering for large data sets. We follow a similar approach and show its effectiveness on the task of classifying emotion words.

3 Overview of the algorithm

The main idea of the method consists in using unsupervised fuzzy clustering to provide additional features and reduce the confusion set for the supervised hard clustering algorithm.

Specifically, the input data of the algorithm consist in a large number of items, or data points (words in our test case) characterized by a number of features (that form feature vectors). A small amount of data items are labeled with the desired categories; the task consists in extending the labels to other data items.

Typically, supervised machine learning algorithms learn from the labeled examples only. For example, in Fig. 1(*a*) shown are data points characterized by one feature (more features would be difficult to show in the figure); four examples are labeled with the category "–" and four with the category "+". Apparently, the data point

[1] http://www.wjh.harvard.edu/inquirer

marked with a question mark should be labeled as “+”, because it lays to the right of the line separating the two sets.

However, analysis of the whole set of data items shown in Fig. 1(*b*), both labeled and unlabeled, suggests that there are two clusters, and the point in question rather belongs to the one with most items labeled as “–”. Indeed, fuzzy clustering assigns membership values in two clusters, characterized by the two centroids shown in Fig. 1(*c*), with the share of membership being larger in the left one than in the right one.

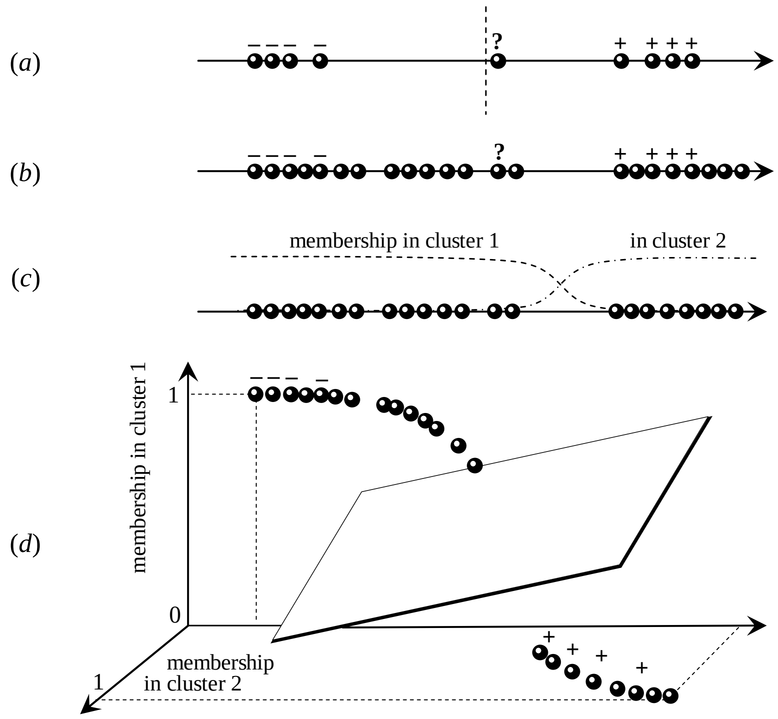

Fig. 1. The idea of the method.

In our method, we add these membership values as additional features of the feature vectors, as shown in Fig. 1(*d*). If the fuzzy clustering reflects properties of the data relevant for the task, then in this higher-dimensional space the data items are better separated than in the original space, with the new coordinates providing stronger clues to the classifier and the gap being wider than in the original problem.

In addition, in case of multi-category classification, we restrict the confusion set for the classifier to the categories that correspond to the best and second-best membership value for each data point, as explained in Section 5. This requires an additional step of identifying the centroids with the categories; in Fig. 1 we would identify the left centroid with the category “–” and the right one with “+”.

With this, our algorithm consists of the following steps:

1. Fuzzy clustering of the whole set of data points, both labeled and unlabeled ones, into the number of clusters equal to the number of categories;
2. Identifying each cluster with a category label;
3. Restricting the confusion set to the best and second best category according to its membership values in the corresponding clusters;
4. For each pair of categories, training a binary classifier using only the labeled examples for which these two categories were predicted at the step 3 and for which the training label is one of the two (for training, we discard those items whose known label is neither of the two labels to which the confusion set for this item is restricted);
5. Finally, assigning the category to each unlabeled data point by applying the classifier trained for the pair of categories predicted for this data point at the step 3.

In the next sections we explain these steps in detail.

4 Fuzzy Clustering

The first step in our two-step process was unsupervised fuzzy clustering. Specifically, we used the fuzzy c-means clustering algorithm (Bezdek, 1981). Below we give the necessary background on this algorithm in the general case, and then introduce our modification to the general procedure.

4.1 General procedure

Informally speaking, fuzzy clustering consists in optimal grouping of given data points together (that is, clustering) but in such a way that each data point can be shared by one or more clusters by belonging partially to one cluster and partially to another. In addition, each group is characterized by its location in the same space as the data points, that is, via a fictional data point representing the "center" of the cluster.

Formally, fuzzy clustering consists in finding in the feature space a set of points called centroids v_i, $i = 1, ..., c$ (where c is the desired number of clusters), and a set of membership values μ_{ik}, $k = 1, ..., N$ (where N is the number of available data points), that minimize a given objective function $J(\{\mu_{ik}\}, \{v_i\})$. The centroids characterize the constructed clusters, while the membership values μ_{ik} (often called membership functions, being considered as functions $\mu_i(x_k) = \mu_{ik}$) are interpreted as a degree with which a data point x_k belongs to the cluster characterized by the centroid v_i.

Depending on the task, such a degree can in turn be roughly interpreted as a probability of that the data point belongs to some "true" but unobservable hard cluster. In particular, the total membership of one data point in all clusters must be a unity:

$$\sum_{i=1}^{c} \mu_{ik} = 1 \qquad (1)$$

for each $k = 1, ..., N$.

Often (though not in our particular case, see Section 4.2) the function J is taken as

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} \mu_{ik}^{p} \left\| x_k - v_i \right\|^2, \quad p > 1, \tag{2}$$

where the exponential constant p influences the degree of fuzziness of the obtained membership functions, and the notation

$$\left\| u \right\|^2 = \sum_{m=1}^{n} u_m^2 \tag{3}$$

denotes the square of the Euclidean length in the space of n features; u_m are the coordinates of a data point u in this space and n is the number of features given for the task—the dimensionality of the problem.

The standard procedure for fuzzy clustering consists in the following. The objective function (2) together with the constraints (1) is considered as a constraint optimization problem, which can be solved with the method of Lagrangian multipliers. This method consists in reducing a constraint optimization problem to an unconstrained optimization problem in a higher dimensional space: the problem of finding an optimum of the Lagrangian of the original system (1), (2). The Lagrangian is constructed as follows:

$$L(v, \mu, \lambda) = J + \sum_{k=1}^{N} \lambda_k g_k, \tag{4}$$

where λ_k are newly introduced auxiliary variables, v, μ, and λ are shortcuts for the sets of v_i, μ_{ik}, and λ_k, and the functions

$$g_k = 1 - \sum_{i=1}^{c} \mu_{ik} \tag{5}$$

are the penalties for violation the constraints (1).

Optimal solutions of the original system (1), (2) can be shown to be stationary points of its Lagrangian (4). The converse is not generally true, but in real-life applications it can be assumed to hold. The problem of the presence of maxima or local (but not global) minima of (4) is also conventionally ignored in real-life applications unless the experimental results suggest their presence.

With these assumptions, solving the original system (1), (2) is reduced to finding a stationary point of (4). This, in turn, is reduced to finding a point (v, μ, λ) at which

$$\frac{\partial L}{\partial v_{im}} = \frac{\partial L}{\partial \mu_{ik}} = \frac{\partial L}{\partial \lambda_k} = 0 \tag{6}$$

for all i = 1, ..., c; m = 1, ..., n; k = 1, ..., N, where v_{im} is the m-th coordinate of v_i in the feature space. Note that the last equality is equivalent to the constraints (1), so only the other two values are to be derived separately.

Computing-wise, the stationary point is usually found with a the following iterative procedure:

– An initial point (v, μ, λ) is chosen arbitrarily.
– At each iteration of the algorithm, each of the equations (6) is solved with respect to the corresponding variable (v_{im}, μ_{ik}, or λ_k) assuming the other values to be fixed, and the point is moved to the solution.
– The iterative process stops when the change of the objective function J between iterations becomes smaller than a predefined small constant ε.

The final position of the point (v, μ, λ) is taken as an approximate solution.

For faster convergence, when choosing the initial point, the constraints (1) are enforced by normalizing the initially random values for μ:

$$\mu_{ik} \leftarrow \frac{\mu_{ik}}{\sum_{j=1}^{c} \mu_{ij}}. \tag{7}$$

Thus, the algorithm is completely defined by the equations (6), which are to be solved with respect to v and μ (the auxiliary variables λ are of no interest).

4.2 Modified objective function

To obtain more compact clusters, we used the following modified objective function J; cf (2):

$$J = \sum_{i=1}^{c}\sum_{k=1}^{N} \mu_{ik}^{p} \left(\left\| x_k - v_i \right\|^2 + \rho \sum_{x_r \in N_k} \left\| x_r - v_i \right\|^2 \right), \quad p > 1, \tag{8}$$

where the balancing constant ρ controls the effect of the additional member, and the sets N_k are obtained in a process of intermediate hard clustering. This process of hard clustering consisted in relating each point with the nearest centroid point: $c(x_k)$ = arg min$_i$ $||x_k - v_i||$; then the hard clusters were obtained as $C_i = \{x_k \mid c(x_k) = v_i\}$ and $N_k = C_i$ such that $x_k \in C_i$.

While the analytical solution for the standard objective function (2) is well-known, we had to derive the solution for our modified function (8).

Namely, for our modified objective function, the Lagrangian (4) is given by

$$L = \sum_{i=1}^{c}\sum_{k=1}^{N} \mu_{ik}^{p} \left(\left\| x_k - v_i \right\|^2 + \rho \sum_{x_r \in N_k} \left\| x_r - v_i \right\|^2 \right) + \sum_{k=1}^{N} \lambda_k \left(1 - \sum_{i=1}^{c} \mu_{ik} \right).$$

Then the first part of (6) has the form

$$\frac{\partial L}{\partial v_i} = \sum_{k=1}^{N} \mu_{ik}^{p} \left(\left\| x_k - v_i \right\|^2 + \rho \sum_{x_r \in N_k} \left\| x_r - v_i \right\|^2 \right) = 0$$

for all *i* = 1, ..., *c*, or, in coordinate notation:

$$\begin{aligned} &\frac{\partial}{\partial v_{im}} \sum_{k=1}^{N} \mu_{ik}^{p} \left(\sum_{l=1}^{P} (x_{kl} - v_{il})^2 + \rho \sum_{x_r \in N_k} \sum_{l=1}^{P} (x_{rl} - v_{il})^2 \right) = \\ &\quad \frac{\partial}{\partial v_{im}} \sum_{k=1}^{N} \mu_{ik}^{p} \left((x_{km} - v_{im})^2 + \rho \sum_{x_r \in N_k} (x_{rm} - v_{im})^2 \right) = \\ &\qquad -2 \sum_{k=1}^{N} \mu_{ik}^{p} \left((x_{km} - v_{im}) + \rho \sum_{x_r \in N_k} (x_{rm} - v_{im}) \right) = 0, \end{aligned}$$

from which we have (back from coordinate notation to vector notation):

$$\sum_{k=1}^{N} \mu_{ik}^{p} x_k + \sum_{k=1}^{N} \rho \mu_{ik}^{p} \sum_{x_r \in N_k} x_r = \sum_{k=1}^{N} \mu_{ik}^{p} v_i + \sum_{k=1}^{N} \rho \mu_{ik}^{p} \sum_{x_r \in N_k} v_i \ ,$$

or, given that $\sum_{x_r \in N_k} v_i = v_i \sum_{x_r \in N_k} 1 = v_i \mid N_k \mid$, we have:

$$\sum_{k=1}^{N} \mu_{ik}^{p} x_k + \sum_{k=1}^{N} \rho \mu_{ik}^{p} \sum_{x_r \in N_k} x_r = v_i \left(\sum_{k=1}^{N} \mu_{ik}^{p} + \sum_{k=1}^{N} \rho \mu_{ik}^{p} \mid N_k \mid \right).$$

This gives

$$v_i = \frac{\sum_{k=1}^{N} \mu_{ik}^{p} \left(x_k + \rho \sum_{x_r \in N_k} x_r \right)}{\sum_{k=1}^{N} \mu_{ik}^{p} \left(1 + \rho |N_k| \right)} \tag{9}$$

Similarly, the second part of (6) has the form

$$\frac{\partial L}{\partial \mu_{ik}} = p \mu_{ik}^{p-1} \left(\|x_k - v_i\|^2 + \rho \sum_{x_r \in N_k} \|x_r - v_i\|^2 \right) - \lambda_k = 0$$

for all *i* = 1, ..., *c*; *k* = 1, ..., *N*, i.e.,

$$\mu_{ik} = \left(\frac{\lambda_k}{p \left(\|x_k - v_i\|^2 + \rho \sum_{x_r \in N_k} \|x_r - v_i\|^2 \right)} \right)^{\frac{1}{p-1}} . \tag{10}$$

Substituting this value in (1), we have

$$\lambda_k = \frac{p}{\left(\sum_{i=1}^{c} \frac{1}{\left(\left\| x_k - v_i \right\|^2 + \rho \sum_{x_r \in N_k} \left\| x_r - v_i \right\|^2 \right)^{\frac{1}{p-1}}} \right)^{p-1}}, \tag{11}$$

which turns (10) into

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\left\| x_k - v_i \right\|^2 + \rho \sum_{x_r \in N_k} \left\| x_r - v_i \right\|^2}{\left\| x_k - v_j \right\|^2 + \rho \sum_{x_r \in N_k} \left\| x_r - v_j \right\|^2} \right)^{\frac{1}{p-1}}}. \tag{12}$$

It is the values (9) and (12) that we used for the iterative re-calculations at the second step of the algorithm mentioned at the end of Section 4.1 above (the values for λ_k are not really needed, though one can use (11) to calculate them).

5 Classification

Recall that our task was semi-supervised learning: we had a great amount of data items without known category for each item, and a small amount of data items for which the desired category has been manually assigned in the training set; our task consisted in extending this labeling to the whole data set.

The baseline classification method—supervised classification—consisted in using the feature vectors (the same vectors as those assumed in the previous section) of only those data points for which the category was known from the labeled training data set. With those points, a classifier was trained; then this classifier was applied to each data point for which the category was unknown, in order to relate it with some category. With this, each data point to be labeled was processed separately.

In contrast, our semi-supervised method used the internal structure learnt in an unsupervised manner from the raw data set, to help the supervised classifier in making its decisions.

For this, we first conducted fuzzy clustering of the whole available dataset (both labeled and unlabeled data points). Then we extended the *N*-dimensional feature vectors by *c* additional features: the membership values obtained in the fuzzy clustering step. The resulting $N + c$ features were used for hard classification in the usual way. The importance of the additional *c* features was in that they were likely strong predictors of the final class.

To take a further adventure of this fact, we restricted possible outputs of the classifier to only two variants—those that were predicted by the *c* features obtained from the fuzzy clustering: namely, to those two variants that corresponded to the highest and the second highest membership functions.

For example, given $c = 6$ clusters as the target, if a data point had the following membership functions in the clusters that corresponded to the following categories

cluster i	1	2	3	4	5	6
membership μ_i	0.1	**0.2**	0.1	**0.5**	0.0	0.1
category	APPLE	ORANGE	PEAR	BANANA	COCONUT	LEMON

then we forced the hard classifier to choose (using also other features not shown here) only between the categories ORANGE and BANANA for this data point, because these categories corresponded to the clusters to which the given data point was predicted to belong with the best and the second best degree.

However, for this we needed a mapping between fuzzy clusters (centroids) and categories. To find such a mapping, we used a simple majority voting. First, for each data point we selected only one cluster: the one in which it has the greatest membership (in case if several clusters tie, an arbitrary one was chosen). Next, for each cluster, the category was chosen to which the majority of the points associated with it at the previous step belonged; again, ties were resolved by a random choice. While this procedure can potentially result in not one-to-one correspondence between clusters and categories, this did not happen in our experiments.

Since the hard classifier thus needed to choose only between two possible labels, a binary classifier such as SVM was a natural choice. We trained a separate classifier for each pair of categories to choose from, that is, we trained $\binom{c}{2}$ separate binary classifiers: a classifier for two categories C_1, C_2 was trained on all training data points known to belong to C_1 or known to belong to C_2.

Finally, to classify each unlabeled data point, we determined the two categories that constituted the confusion set for it (those with the highest membership functions) and used the corresponding binary classifier.

6 Case study: Semi-supervised Learning of an Emotion Lexicon

We applied our method to the task of semi-supervised learning of an emotion lexicon. A detailed account of the features used for classification and the obtained results can be found in (Poria *et al.*, 2012a, b, 2013).

An emotion lexicon is a dictionary that specifies for each word the main emotion typically communicated by the text where the word is used, for example:

Word	Emotion category	Word	Emotion category
offend	ANGER	*congratulate*	JOY
detestable	DISGUST	*cheerless*	SADNESS
cruelty	FEAR	*puzzle*	SURPRISE

(examples borrowed from WordNet Affect (Strapparava and Valitutti, 2004)). We assumed that words similar in some way, such as in their usage or with similar information associated with them in existing dictionaries, should be related with the same emotion.

With this, we applied the classification technique described in the previous sections to the task of extending the emotion labels from a small existing emotion lexicon to a much larger set of words for which we could collect sufficient information to form the feature vectors.

As a source of labeled examples, we used the mentioned WordNet Affect lexicon. It classifies words and some simple phrases (such as *scare away* or *the green-eyed monster*) into six categories: ANGER, DISGUST, FEAR, JOY, SADNESS, and SURPRISE.

For classification, we used two groups of features for words:

- A number of similarity measures. One set of measures were nine similarity measures based on WordNet (Miller, 1995) calculated with the WordNet::Similarity package were used. Another set were similarity measures based on co-occurrence (more specifically, the distance between occurrences) of the concepts in an emotion-related corpus, specifically, in the International Survey of Emotion Antecedents and Reactions (ISEAR) dataset (Scherer, 2005); see details in (Poria *et al*., 2012a, b). To incorporate a similarity measure as a feature for the feature vectors, we considered each word or concept in our vocabulary as an independent dimension, and the corresponding coordinates for a given word were its similarity values with each word in the vocabulary.
- The data from the ISEAR dataset. This dataset consists of short texts (called statements) describing an emotional situation, each statement being annotated with 40 parameters, including the emotion that the statement describes (though the inventory of the basic emotions used in the ISEAR dataset slightly differs from that using in WordNet Affect). We considered each value of each parameter given in ISEAR as an independent dimension, and the corresponding coordinate value of a concept found in SenticNet was the number of times that this concept was found in the ISEAR statements annotated with this value of the parameter.

The rich set of features facilitated the unsupervised clustering of concepts in such a way that the concepts related to similar emotions we associated with the same fuzzy clusters.

We applied our method to the following sub-corpus and feature set combinations:

- $\boldsymbol{C}$: all words (after stemming) found in the ISEAR dataset. There were 449,060 distinct stemmed words in this dataset. No similarity measure was used.
- $\boldsymbol{C_{Co}}$: the same corpus, but features based on the co-occurrence similarity measure were used for this experiment.
- $\boldsymbol{C_{WA}}$: only words that co-occurred with those from WordNet Affect in an ISEAR statement. There were only 63,280 distinct stemmed words in this sub-corpus.
- $\boldsymbol{C'}$, $\boldsymbol{C'_{Co}}$, $\boldsymbol{C'_{WA}}$: the same sets, but WordNet-based and co-occurrence-based similarity measures were used in these experiments.

For evaluation, we used the membership value obtained at the step of fuzzy clustering for the class that corresponded to the label chosen at the step of final hard clustering as a confidence measure. In each corpus, we selected top 100 words with the best confidence measure, and calculated the accuracy of the final hard classification on this set. We compared the accuracy achieved by our method with the accuracy achieved for the same words by the baseline method: SVM without the fuzzy clustering step. The results are shown in Table 1.

Table 1. Accuracy (%) of the baseline (SVM only) and the proposed classifiers for top 100 confidence words on different subcorpora.

Sub-corpus	SVM only	Fuzzy + SVM	Sub-corpus	SVM only	Fuzzy + SVM
C_{Co}	84.10	87.44	C'_{Co}	86.77	87.44
C	83.22	88.01	C'	85.19	90.78
C_{WA}	88.23	92.56	C'_{WA}	91.67	95.02

One can observe from the table that with each combination of a sub-corpus and the feature sets employed in the experiment, our method (denoted as Fuzzy + SVM in the table) significantly outperforms the baseline (SVM only) method.

7 Conclusions

Semi-supervised learning consists in using the inner structure of the set of unlabeled examples to aid supervised learning for classification basing on a small number of labeled examples.

We have proposed a two-step process for semi-supervised learning. At the first step, unsupervised fuzzy clustering of all available data is performed. The resulting membership functions are then used for two purposes: to reduce the confusion set for each data item and as additional features in the feature vectors. At the second step, a set of binary classifiers for the reduced confusion sets are trained in the extended feature space and are applied to assign the labels to the unlabeled data points.

We tested our method on an important task: construction of emotional lexicon. In this task, data items were words (we experimented with almost half million words) and a rich set of features were extracted from an emotion-related corpus. In addition, a number of similarity measures were used as features, namely, for each word and each similarity measure that we used, the similarity values between the given word and all words in the vocabulary were used as individual features. This gave a very large feature set suitable for unsupervised clustering.

Our experiments have shown that our suggested method outperforms the baseline classification technique, which was SVM without prior fuzzy clustering. In the future, we plan to conduct similar experiments on other classification tasks, in order to estimate the limitations and applicability of our method to a wider class of classification problems.

Acknowledgements. The work was partially supported by the Governments of India and Mexico under the CONACYT-DST India (CONACYT 122030) project “Answer Validation through Textual Entailment”, the Government of Mexico under the CONACYT 50206-H project and SIP-IPN 20121823 project through Instituto Politécnico Nacional, as well as the Seventh Framework Programme of European Union, project 269180 “Web Information Quality Evaluation Initiative (WIQ-EI)”.

References

1. Alm, Ovesdotter, C., Roth, D., Richard S. 2005. Emotions from text: Machine learning for text-based emotion prediction. In Proceedings of HLT-EMNLP, 579–586.
2. Andreevskaia A., Bergler S. 2007. CLaC and CLaC-NB: Knowledge-based and corpus-based approaches to sentiment tagging. 4th International Workshop on SemEval, 117–120.
3. Aue, A., Gamon, M. 2005. Customizing sentiment classifiers to new domains: A case study. In Proc. of RANLP.
4. Awad, M., Khan, L., Bastani F., Yen I. L. 2004. An Effective support vector machine (SVMs) Performance Using Hierarchical Clustering. Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’04), 663–667.
5. Baccianella, S., Esuli, A., Sebastiani, F. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. LRE, 2200–2204.
6. Banea, C., Mihalcea, R., Wiebe, J. 2008. A Bootstrapping Method for Building Subjectivity Lexicons for Languages with Scarce Resources. LREC.
7. Baroni, M., Vegnaduzzo, S. 2004. Identifying subjective adjectives through web-based mutual information. Proceedings of the German Conference on NLP.
8. Bezdek, J. C. 1981. Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York.
9. Boley, D., Cao, D. 2004. Training support vector machine Using Adaptive Clustering. In Proc. of SIAM Int. Conf on Data Mining, Lake Buena Vista, FL, USA.
10. Cambria, E., Speer, R., Havasi, C., Hussain, A. 2010. SenticNet: A publicly available semantic resource for opinion mining. In: Proc. of AAAI CSK, 14–18.
11. Cambria, E., Hussain, A. 2012. Sentic computing: Techniques, tools, and applications. Dordrecht, Netherlands: Springer, 153 pp.
12. Cervantes, J., Xiaoou Li, Wen Yu. 2006. Support Vector Machine Classification Based on Fuzzy Clustering for Large Data Sets. MICAI 2006, LNAI 4293, 572–582.
13. Elliott, C. 1992. The affective reasoner: A process model of emotions in a multi-agent system. Ph.D. thesis, Institute for the Learning Sciences, Northwestern University.
14. Hatzivassiloglou, V., McKeown, K. R. 1997. Predicting the semantic orientation of adjectives. 35th Annual Meeting of the ACL and the 8th EACL, 174–181.
15. Hu, M., Liu, B. 2004. Mining and summarizing customer reviews. In Proceedings of the ACM SIGKDD, 168–177.
16. Kamps, J., Marx, M., Mokken, R. J., Rijke, M. de. 2004. Using wordnet to measure semantic orientation of adjectives. In Proceedings of the 4th LREC 2004, IV, 1115–1118.
17. Kobayashi, N., Inui, T., Inui, K. 2001. Dictionary-based acquisition of the lexical knowledge for p/n analysis (in Japanese). In Proceedings of Japanese Society for Artificial Intelligence, SLUD-33, pp. 45–50.
18. Liu, B. 2010. Sentiment Analysis: A Multi-Faceted Problem. IEEE Intelligent Systems.
19. Miller, A. G. 1995. WordNet: a lexical database for English. In Communications of the ACM, vol. 38 (11), 39–41.

20. Mohammad, S., Turney, P.D. 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In Proc. of NAACL-HLT, Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, 26–34.
21. Neviarouskaya, A., Prendinger, H., Ishizuka, M. 2009. SentiFul: Generating a Reliable Lexicon for Sentiment Analysis. In ACII'09, IEEE, pp. 363–368.
22. Pang, B., Lillian, L., Shivakumar, V. 2002. Thumbs up? Sentiment classification using machine learning techniques. In the Proc. of EMNLP, 79–86.
23. Poria, S., Gelbukh, A., Cambria, E., Das, D., Bandyopadhyay, S. 2012. Enriching SenticNet Polarity Scores through Semi-Supervised Fuzzy Clustering. In Proc. of the SENTIRE 2012 workshop at IEEE ICDM 2012.
24. Poria, S., Gelbukh, A., Cambria, E., Yang, P., Hussain, A., Durrani, T. 2012. Merging SenticNet and WordNet-Affect Emotion Lists for Sentiment Analysis. In Proc. of the 11th International Conference on Signal Processing, IEEE ICSP 2012, Beijing.
25. Poria, S., Gelbukh, A., Das, D., Bandyopadhyay, S. 2013. Extending SenticNet with Affective Labels for Concept-based Opinion Mining. IEEE Intelligent Systems, submitted.
26. Read, J. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In Proceedings of the ACL Student Research Workshop.
27. Riloff, E., Wiebe, J., Wilson, T. 2003. Learning subjective nouns using extraction pattern bootstrapping. In Proceedings of the Seventh CoNLL-03), 25–32.
28. Scherer K. R. 2005. What are emotions? And how can they be measured? Social Science Information, 44(4):693–727.
29. Sidorov, G., Castro-Sánchez, N.A. 2006. Automatic emotional personality description using linguistic data. Research in computing science 20:89–94.
30. Strapparava, C., Ozbal, G. 2010. The Color of Emotions in Texts. Proceedings of the 2nd Workshop on Cognitive Aspects of the Lexicon (CogALex 2010), 28–32, Beijing.
31. Strapparava, C., Valitutti, A. 2004. Wordnet affect: an affective extension of wordnet. Language Resource and Evaluation.
32. Takamura, H., Inui, T., Okumura, M. 2005. Extracting Semantic Orientations of Words using Spin Model. In 43rd ACL, 133–140.
33. Turney, P. D., Littman, Michael L. 2003. Measuring praise and criticism: Inference of semantic orientation from association. ACM TIS, 21(4):315–346.
34. Voll, K., M. Taboada. 2007. Not All Words are Created Equal: Extracting Semantic Orientation as a Function of Adjective Relevance. In Proceedings of the 20th Australian Joint Conference on Artificial Intelligence, Gold Coast, Australia, pp. 337–346.
35. Wiebe, J. M. 2000. Learning subjective adjectives from corpora. In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI 2000), 735–740.
36. Wiebe, J., Mihalcea, R. 2006. Word sense and subjectivity. In Proceedings of COLING/ACL, Sydney, Australia, 1065–1072.
37. Yu H., Yang J., Han Jiawei. 2003. Classifying Large Data Sets Using SVMs with Hierarchical Clusters. In Proc. of the 9th ACM SIGKDD.