Stacked denoising autoencoders for sentiment analysis: a review

Hesam Sagha, Nicholas Cummins, Björn Schuller

Angaben zur Veröffentlichung / Publication details:

Sagha, Hesam, Nicholas Cummins, and Björn Schuller. 2017. "Stacked denoising autoencoders for sentiment analysis: a review." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7 (5): e1212. https://doi.org/10.1002/widm.1212.

Stacked Denoising Autoencoders for Sentiment Analysis: A review

Hesam Sagha*, Nicholas Cummins†, Björn Schuller‡

Article Type:

Focus Article

Abstract

Deep learning has been proven to outperform many conventional machine learning algorithms (e. g., Support Vector Machines) in many fields such as image processing and text analyses. This is due to its outstanding capability to model complex data distributions. However, as networks become deeper, there is an increased risk of overfitting and higher sensitivity to noise. Stacked Denoising Autoencoders (SDAs) provide an infrastructure to resolve these issues. In the field of sentiment recognition from textual contents, SDAs have been widely used (especially for domain adaptation), and have been consistently refined and improved through defining new alternate topologies as well as different learning algorithms. A wide selection of these approaches are reviewed and compared relatively in this article. The results from the reviewed works indicate the promising capability of SDAs to perform sentiment recognition on a multitude of domains and languages.

INTRODUCTION

Affects are part of human life and are expressed through verbal (speech and its content) and non-verbal (written opinions, facial gestures, body movement) communications. Within this context, the analysis of sentiments from textual contents has gained increasing attention. This is due, in part, to: (i) the ease of access to abundant web-based data collections, (ii) less privacy invasion associated with data collection (when compared with audio-visual data collection), and (iii) its benefits in a variety of applications, such as product reviews and recommendation systems. Cattell [1] defined sentiments as "an acquired and relatively permanent major neuropsychic disposition to react emotionally, cognitively, and conatively toward a certain object (defined as person, thing, condition, place, event) in a certain stable fashion,

*Chair of Complex & Intelligent Systems, University of Passau, Germany

†Chair of Complex & Intelligent Systems, University of Passau, Germany

‡Head of the Chair of Complex & Intelligent Systems, University of Passau, Germany
Department of Computing, Imperial College London, London/U.K

with awareness of the object and the manner of reacting." In another words, sentiments involve emotional dispositions formed toward an object over time [2,3]. Examples of sentiments include romantic love, loyalty, friendship, patriotism, hate, as well as more transient, acute emotional responses, to social losses (sorrow, envy) and gains (pride, gratitude) [4].

In the computer science domain, sentiment analysis is commonly defined as "the task of identifying the polarity and subjectivity of documents using a combination of machine learning, information retrieval, and natural language processing techniques." [5]. Sentiments are typically considered as high-level categories of either 'positive', 'negative', or 'neutral', and relatively little research has attempted to classify a broader range of affects.

Typically, sentiment analysis systems first map 'text' into a set of high-dimensional numerical feature vectors (e. g., Bag of Words), Then a machine learning paradigm is applied to learn a relationship between these feature vectors and their corresponding target labels (e. g., positive/negative sentiment).

A commonly occurring issue in sentiment analysis systems relates to the often high dimensionality of the extracted feature vectors. Such issues include the addition of noisy or unrelated features and the increase in the number of training parameters, and can act as a catalyst for overfitting on training data; consequently, hampering the creation of robust models. In this regard, *Autoencoders* can be used to enhance system performance. Autoencoders operate by mapping data into a low dimensional space which represents a latent structure that keeps useful information and can be remapped to the original feature vector with minimum reconstruction error.

Further, real-world data is often corrupted by noise and outliers. For example: tweets (a 160 word posting made on the website *Twitter*) may have bizarre words, acronyms, and initialism; and, product reviews may contain non-related texts. These types of noise can affect the sentiment recognition performance either at the stage of training a model (through improper estimation of parameters) or at the stage of applying the trained model on a new (noisy) sample. Therefore, cleaning and extracting a useful representation of data which is robust to noise, before training a model, is paramount. *Denosing Autoencoders* (DAs) can provide such functionalities. DAs have also been used in *domain adaptation*, where a model is trained on one (labelled) corpus and applied on another contextually different (unlabelled) one. In this case, a DA estimates a shared latent representation of both corpora and tries to match the distribution of the latent representations.

Finally, a one layer DA may not have sufficient capability to model the nonlinear structure of data, and therefore, could yield a high reconstruction error. In this case, using deep structures with more parameters is a promising solution. However, having more parameters necessitates having more training data as well as more computational power and space. To keep the number of training parameters as low as possible both *Stacked Autoencoders* (SAEs) and *Stacked DAs* (SDAs) have been proposed [6]. SAEs and SDAs are types of DA with *deep topologies*, which during the training phase, the layers are trained one by one and stacked on top of each other. Therefore, at each step only the parameters of *one* layer are being tuned.

In this article, we review different types, topologies, and learning methods of using Autoencoders for sentiment analysis in (multi-domain) (multi-lingual) texts. The structure of the remainder of this review is as follows: we first offer a brief overview of common Sentiment Analysis Corpora. We then describe Auto-Encoders (AEs), Denoising Autoencoders (DAs), and Stacked Denoising Autoencoders (SDAs). Then, marginalised SDAs (mSDAs)

Table 1: List of most popular datasets used for sentiment analysis, which have been used by the reviewed works in this paper.

Name	Language	Content	Class (# of samples) Positive/Negative/Neutral
Amazon Reviews Dataset [7]	English	Reviews of Books, Electronics, Kitchen appliances, DVDs	1000 / 1000 / 0
Movie Review [8]	English	Movie Reviews	1000 / 1000 / 0
IMDb [9]	English	Movie Reviews	25000 / 25000 / 0
SemEval2013 [10] (Task 2)	English	Twitter	5349 / 2186 / 6440

and variants thereof such as, Hybrid Heterogeneous Transfer Learning (HHTL), and Stacked Instance-wise Denoising Autoencoders are presented. We then review the use of SDAs within an ensemble of classifiers, and finally compare relatively the models before concluding the article.

Sentiment Analysis Corpora

Before commencing the review, it is worthwhile introducing the characteristics of commonly used databases. Four popular datasets (Amazon Review database, Movie Review, IMDb, and SemEval2013) are commonly used within the context of sentiment analysis using autoencoders. Table 1 offers an overview of the key aspect associated with these databases.

LITERATURE REVIEW

Autoencoders are a type of neural network which consists of an *encoding phase*, in which, feature vectors are mapped onto an abstract lower (or higher) dimensional space, in such a way that the original feature vector can be reconstructed in a subsequent *decoding phase* with minimal reconstruction error (see Fig. 1(a)). An Autoencoder consists of an N dimensional input layer, an H dimensional hidden layer, and an N dimensional output layer, noting that N denotes the feature space dimensionality.

During the training phase of an Autoencoder, each sample from $\mathbf{X} = \{\mathbf{x_i}\}$ ($\mathbf{x_i} \in R^N$, $i = [1 \dots M]$, where M denotes the number of samples) is set for both the input and the output layers, and an optimisation is performed to learn the network weights that minimise the reconstruction error. This error could be computed as the *squared error (Frobenius norm)*, the *Bergman Divergence*, or in the case of probabilistic representation of the feature vector $\mathbf{x_i}$, the *Kullback Leibler divergence*, or the *cross-entropy*.

For an Autoencoder, the objective function can be expressed as:

$$\mathcal{L}(\mathbf{W}) = \sum_i D(\mathbf{x_i}, \tilde{\mathbf{x}}_\mathbf{i}) \qquad s.t. \quad \mathbf{h_i} = g(\mathbf{W}\mathbf{x_i} + \mathbf{b}), \tilde{\mathbf{x}}_\mathbf{i} = f(\mathbf{W}'\mathbf{h_i} + \mathbf{b}'), \quad (1)$$

where $\tilde{\mathbf{x}}_\mathbf{i}$ is the reconstructed vector, $D(\mathbf{x}_\mathbf{i}, \tilde{\mathbf{x}}_\mathbf{i})$ is the distance between $\mathbf{x}_\mathbf{i}$ and $\tilde{\mathbf{x}}_\mathbf{i}$, $\mathbf{W} \in R^{H\times N}, \mathbf{W'} \in R^{N\times H}, \mathbf{b} \in R^H$, and $\mathbf{b'} \in R^N$ are the parameters to be learned, and g and f are nonlinear activation functions (e. g., $tanh(.)$). Once the network is trained, outputs of the hidden layer, $\mathbf{h}_\mathbf{i}$, are used for training and applying a subsequent classifier. Note that, if f and g are linear functions and $H < N$, the unique and global minimum can be described in terms of principle component analysis and least square regression [11].

The processing of textual data with Autoencoders suffers from scalability with the high dimensionality of vocabulary size as well as task-irrelevant words. Zhai and Zhang addressed this problem by introducing supervision via the loss function of Autoencoders (Semisupervised Bergman Divergence Autoencoder: SBDA) [12]. In particular, they trained a linear classifier on the labelled data, and then defined a loss function for their Autoencoder with the weights learned from the linear classifier:

$$D(\mathbf{x}_\mathbf{i}, \tilde{\mathbf{x}}_\mathbf{i}) = (\theta^T(\mathbf{x}_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{i}))^2, \tag{2}$$

where $\theta \in R^N$ are the parameters of the linear classifier. Their experimental results on Amazon Reviews dataset and IMDb dataset show this addition substantially improves system accuracy over standard DAs. However, this technique only aids reconstruction for the features for which the linear classifier is sensitive to.

Denoising Autoencoders

To increase the robustness to noisy data, DAs were originally proposed by LeCun [13]. The input $\mathbf{x}_\mathbf{i}$ is stochastically corrupted with some noise ($\breve{\mathbf{x}}_\mathbf{i} \leftarrow q(\mathbf{x}_\mathbf{i})$, q: corrupting function) and is then fed to the input layer, but the original vector, $\mathbf{x}_\mathbf{i}$, is kept for the output layer (i. e., denoising data), see Fig. 1(b). Although DA provides a representation which is supposedly more robust to noise, the learnt parameters are susceptible to the level of noise applied to the original vectors; high level noise will harm the learning of robust representations [14]. To overcome this issue, a *scheduled DA* (ScheDA) has been introduced, in which the network is trained on gradually decreasing noise level for corrupting feature vectors [15]. Initially, a high level of noise forces the network to learn global and coarse-grained features. Then, by decreasing the noise levels, finer representations are learnt. Alternatively, *Composite DAs* are proposed, in which, at each stage of the training, data with specific noise level is presented to the network, and only a subset of hidden layers is tuned [16].

DAs have been employed for cross language sentiment classification tasks where there is a lack of labels on the target language [17]. In this method a machine translation was used to generate texts from the source language (English) to the target language (Chinese). DAs are then used to reduce the effect of noisy source text and target translations. Finally, a decision fusion is made on the two classifiers, which are trained on the target and source denoised representations. This approach, however, does not consider common sentiment information between the two languages. In another topology, on top of the Autoencoders' hidden layers, two sets of weights were trained for each language, W_E and W_C, for reconstructing (parallel) English and Chinese features [18]. The additional weights improved the performance with respect to the previous approach on a product review database (NLP&CC 2013 CLSC dataset[1]).

[1] http://tcci.ccf.org.cn/conference/2013/dldoc/evsam03.zip

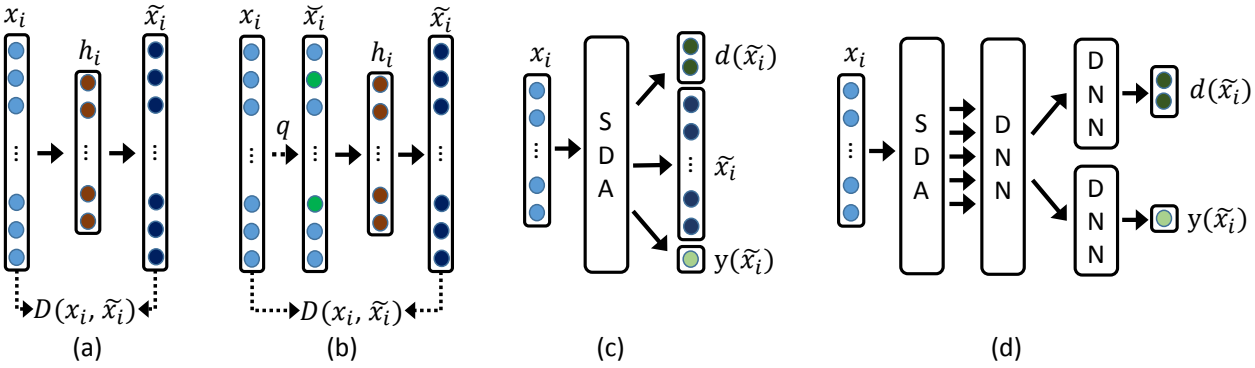

Figure 1: (a) Autoencoder, (b) Denoising Autoencoder, (c) SDA with Domain and Sentiment Supervision (SDA-DSS), (d) Domain Adversarial Neural Network (DANN) with 'SDA representation'. *d(.)*: Domain, *y(.)*: label.

Stacked Denoising Autoencoder

Deep neural network architectures (i. e., neural networks with more than one hidden layer) are highly popular in the machine learning community due to their high capability for modelling data [19]. However, having more layers means that more parameters are required to be tuned during the training phase. Therefore, there is a risk of overfitting to the training data as well as the network falling into a local minimum. Additionally, tuning more parameters brings computational issues such as memory limitation and increased training time.

A way to avoid these issues is to train each layer one by one, and then stack them on top of each other whilst keeping the weights of each trained layers static. This approach is known as Stacked Denoising Autoencoders (SDA) [6]; each hidden layer represents a level of abstraction which can be used for a classification or regression task [20]. The steps to train a SDA are shown in Algorithm 1. Usually the representation at the final layer is considered for further analysis [6], however, concatenating all the abstract representations and the original data vector can also be utilised [21] (to use information in all the abstract representations). Note that, during network construction, once a layer is trained, it receives the uncorrupted output of the previous layer.

The initialisation of the weights of a deep network plays a great role in avoiding local minima. *Stacked Sparse Autoencoders* have been used to initialise the weights layer by layer, and then tune the weights of a SDA by standard backpropagation [22]. It has been shown that this approach improves accuracy over random weight initialisation on the IMDb dataset [23].

SDAs have also been used in domain adaptation for sentiment recognition of reviews of different products [24]. In this paradigm, all the corpora are presented to the SDA to extract a shared representation of all sources; classification is then performed on this representation. Moreover, SDA showed a significant improvement on sentiment relevance detection for cross corpus analysis [25].

A variation of SDA (named SDA with Domain and Sentiment Supervision: SDA-DSS) was proposed by Liu et al. [26]. In this approach the last layer of the stacked Autoencoder, besides performing the reconstruction of the inputs, predicts either the domain (0: source, 1: target), the sentiment label, or both through a softmax function, see Fig 1(c). All three augmentations of this technique showed an improvement over standard SDA on the Amazon

Input : X: Data, L: #layers
Output: F: feature vector
Definitions: W: encoding weights, b: encoding bias, W': decoding weights, b': decoding bias, $q(.)$: corrupting function;
$l \leftarrow 1$;
$\tilde{X}_1 \leftarrow X$;
while $l \leq L$ **do**
  initialise new layer with W_l, b_l, W'_l, b'_l;
  $\breve{X}_l \Leftarrow q(\tilde{X}_l)$ // Corrupt the input;
  train W_l, b_l, W'_l, b'_l with input $\breve{X}_l$ and output X;
  $\tilde{X}_{l+1} \leftarrow f(\tilde{X}_l | W_l, b_l)$ // Generate features for the next layer;
  $l \leftarrow l + 1$;
end
$F \leftarrow \tilde{X}_{L+1}$ OR concat$(\tilde{X}_1, \ldots, \tilde{X}_{L+1})$

Algorithm 1: Stacked Denoising Autoencoder training algorithm.

Reviews dataset.

Marginalised Stacked Denoising Autoencoders

Although SDAs mediate certain problems of Autoencoders, they still suffer from two limitations: (i) high computational costs (through stochastic gradient descent learning); and (ii) a lack of scalability to high dimensional features. Additionally, there is a need to generate a great number of noisy samples, which are corrupted on different features, and supply them to the network during training. *Marginalized SDA* [27,28] (mSDA) helps in overcoming these limitations by using a closed-form solution to estimate the network weights and implicitly apply denoising *without* generating a single noisy sample.

A standard mSDA consists of multiple layers of *linear* denoisers, in which the objective function is:

$$\mathcal{L}(\mathbf{W}) = \frac{1}{m} \sum_{i=1}^{m} \|\mathbf{x_i} - \mathbf{W}\breve{\mathbf{x}}_i\|^2, \tag{3}$$

where $\mathbf{W} : R^{N+1} \rightarrow R^{N+1}$, and we assume that $\mathbf{x}_i = [\mathbf{x}_i; 1]$. The solution to this equation is expressed as a closed-form ordinary least squares:

$$\mathbf{W} = \mathbf{P}\mathbf{Q}^{-1}, \text{ where } \mathbf{Q} = \breve{\mathbf{X}}\breve{\mathbf{X}}^T \text{ and } \mathbf{P} = \mathbf{X}\breve{\mathbf{X}}^T. \tag{4}$$

Having $k \rightarrow \infty$ copies of $\mathbf{X} \in R^{N \times M}$ corrupted by noise, $\mathbf{P} \in R^{N \times N}$ and $\mathbf{Q} \in R^{N \times N}$ converge to their expected values $E[\mathbf{P}]$ and $E[\mathbf{Q}]$. Therefore:

$$\mathbf{W} = E[\mathbf{P}]E[\mathbf{Q}]^{-1}. \tag{5}$$

Let $\mathbf{q} = [1-p, ..., 1-p, 1]^T \in R^{N+1}$, where $\mathbf{q}_\alpha$ and $\mathbf{q}_\beta$ represent the probabilities of the α^{th} and β^{th} features surviving the corruption (which occurs with probability p). Then, two

features m_α and m_β survive corruption with the probability $(1-p)^2$. If we define $\mathbf{S} = \mathbf{X}\mathbf{X}^T$ as the scatter matrix of the uncorrupted input, we can then express the expected value of $\mathbf{Q}$ as:

$$E[\mathbf{Q}]_{\alpha,\beta} = \begin{cases} \mathbf{S}_{\alpha,\beta}\mathbf{q}_\alpha\mathbf{q}_\beta \text{ if } m_\alpha \neq m_\beta \\ \mathbf{S}_{\alpha,\beta}\mathbf{q}_\alpha \quad \text{if } \alpha = \beta \end{cases}, \tag{6}$$

and, $E[\mathbf{P}]_{\alpha,\beta} = \mathbf{S}_{\alpha,\beta}\mathbf{q}_\beta$. To embed non-linearity into the mSDA, after computing $\mathbf{W}$ for each layer, a non-linear 'squashing' function is applied to the output of the layer. The mSDA's advantages include: (i) only one pass through training data is required; (ii) a convex optimal solution is guaranteed; and (iii) optimisation is in a closed form. Additionally, mSDAs have shown a huge speedup ($\times$450) during training with comparable performance to SDAs [27,28]. Note that, the computation of $E[\mathbf{Q}]^{-1}$ is computationally expensive for data with high dimensionality (e. g., when representing text as a Bag of Words). To cope with this problem, Chen et al. reduced dimensionality of the input data to only 5 000 frequent terms [27].

Ganin et al. utilised an mSDA representation of features as the input to a *Domain Adversarial Neural Networks* (DANNs) [21]. DANNs are topologies similar to SDA-DSS [26], in which, on the top of a feature extractor network, $G_f(.)$, two other deep networks are augmented for predicting a class label, $G_y(.)$, and a domain label, $G_d(.)$, see Fig. 1(d). During the training phase, the objective function is designed to minimise the label prediction loss, and simultaneously, maximise the domain prediction loss. The latter guarantees that the two domains are mapped to each other. The authors showed that DANNs based on mSDA representation outperform the original representation of data on the Amazon Reviews dataset. Note that in this case, the mSDA representation is a concatenation of the abstract features of the all layers of mSDA.

Variants of mSDA

Clinchant et al. combined the domain prediction idea of Domain Adversarial Networks [21] with a mSDA through a closed-form solution (mSDA+Target Regularisation) [29]. The objective function, which includes a regularisation term, is defined as:

$$\mathcal{L}(\mathbf{W}) = \frac{1}{m}\sum_{i=1}^{m}\|\mathbf{x_i} - \mathbf{W}\check{\mathbf{x}}_i\|^2 + \lambda\|\mathbf{r_i} - \mathbf{c}\mathbf{W}\check{\mathbf{x}}_i\|^2, \tag{7}$$

where $\mathbf{c} \in R^N$ is a linear classifier trained to distinguish the source and target domains, and r_i indicates the regulation objective; if $r_i = 1$ the reconstructed features move toward target specific features, if $r_i = d_i$ (with $d_i \in [-1, +1]$ indicating domain of x_i) the model promotes domain invariant features, and if $r_i = 0$ for the source domain, and $r_i = 1$ for the target domain, the model penalises the source specific features. This approach achieved comparable results on the Amazon Reviews dataset with DANNs with less computational costs.

In pattern recognition, the geometrical information of the data has repeatedly been shown to be quite important [30,31]. Therefore, keeping this information within the abstract representation of the data should help in improving recognition performance. To preserve the local structure of data in the latent structure, a graph regularisation has been proposed within the mSDA objective function (GmSDA) [32]. This regularisation term is given by:

$$Tr(\mathbf{W}\check{\mathbf{X}}\mathbf{L}\check{\mathbf{X}}^\mathbf{T}\mathbf{W}^\mathbf{T}), \tag{8}$$

where L denotes the graph Laplacian, obtained by $L = diag(\sum_j B_{ij}) - \mathbf{B}$, where $\mathbf{B}$ is the edge weight matrix:

$$B_{ij} = \begin{cases} 1, \text{if } \breve{x}_i \text{ is in the neighbor of } \breve{x}_j \\ 0, \text{otherwise} \end{cases}. \tag{9}$$

This brand of mSDA has been shown to be able to learn a more robust feature representation for sentiment recognition from reviews, and has achieved a higher performance on the Amazon Reviews dataset with respect to the original mSDA paradigm.

A further technique, mSDA++, adds a DA on top of the mSDA structure to reduce the dimensionality of the data [33]. Additionally, in the same study, the authors applied a domain adaptation technique (named EASYADAPT [34]) before applying mSDA to reduce the effect of unmatched domains. On average, on a subset of the Amazon Reviews dataset, mSDA++ has been shown to outperform the original SDA.

To increase the robustness of Stacked Autoencoders (SAE) to outliers, $\ell_{2,1}$-norm has been proposed as the objective function instead of Frobenius norm [35]. This approach (named $\ell_{2,1}$-norm Stacked Robust Autoencoders: $\ell_{2,1}$SRA) is different from mSDA as it uses regularisation instead of marginalised corruptions. The objective function is defined as follows:

$$\mathcal{L}(\mathbf{W}) = \sum_i ||\mathbf{x}_i - \mathbf{W}\mathbf{x}_i||_{2,1} + \lambda ||\mathbf{W}||_F^2. \tag{10}$$

Similar to mSDA, to add non-linearity, a squashing function is used after weight estimation. On the Amazon Reviews dataset, this regularisation approach outperforms mSDAs.

From the literature, it can be seen that mSDAs have two main limitations: (i) minimising the reconstruction error does not take care of divergence related issues between the source and target domains; and (ii) mSDAs learn a linear mapping and then a non-linearity is embedded by applying a nonlinear squashing function – leading to an inadequate modelling of any nonlinear relationships presented in the data. Wei et al. [36] proposed *Deep Nonlinear Feature Coding* (DNFC) to target these limitations by the introduction of two elements into the mSDA paradigm: (i) domain divergence minimisation by *Maximum Mean Discrepancy* (MMD); and (ii) nonlinear coding by kernelisation. MMD quantifies the domain divergence and by incorporating it into each layer of a mSDA, the two domains get closer to each other. A term is added in the objective function to reduce the domain divergence:

$$tr(\mathbf{W}\breve{\mathbf{X}}\mathbf{M}\breve{\mathbf{X}}^{\mathbf{T}}\mathbf{W}^{\mathbf{T}}). \tag{11}$$

This is similar to the regularisation term in (8), but instead of a graph Laplacian, $\mathbf{M} = [M_{i,j}]_{N\times N} = \frac{1}{N^2}$. Furthermore, by applying nonlinear kernelisation and mapping data onto a *Reproducing Kernel Hilbert Space* (RKHS), nonlinearities in the data can be learnt. On the Amazon Reviews dataset, this approach yielded a notable improvement on sentiment classification with respect to mSDA.

Similar to the previous approach, *Hybrid Heterogeneous Transfer Learning* (HHTL) applies mSDA to learn both the deep learning structure and the feature mappings between cross-domain heterogeneous features to reduce bias issues caused by the cross-domain correspondences [37]. On each iteration, k, after building a mSDA layer, the abstract representations of the features, $\mathbf{H}_k$, are mapped to each other through the following objective

function:

$$\min_{\mathbf{G}_k} ||\mathbf{H}_k^{Source} - \mathbf{G}_k \mathbf{H}_k^{Target}||_F^2 + \lambda ||\mathbf{G}_k||_F^2. \quad (12)$$

A significant improvement has also been achieved with respect to the combination of mSDA and using Canonical Correlation Analysis (as an alternative to the aforementioned regularisation) on the multi-lingual Amazon Reviews dataset.

Stacked Instance-wise Denoising Autoencoder is a variation of HHTL, which selects only the instances which have a non-empty set for the intersection between a randomly generated Boolean vector $r \in R^N$ and non-zero values [38]. This process of initialising r repeats t times, with one mSDA being generated at each time step. Finally, all the weights of the t mSDAs are combined together to form the final network. This helps to handle high-dimensional data and reduces the size of the data for training each mSDA. This approach has been shown to outperform HHTL in the multilingual analysis of the Amazon Reviews dataset.

An alternative approach to handle large amounts of data, is through online learning. Budiman et al. introduced the *online Marginalized Linear Stacked Denoising Autoencoder*, in which the network parameters (Scatter matrix $\mathbf{S}$) get updated sample by sample, therefore enabling the network to handle a larger amount of high dimensional data [39]. The authors show this approach can achieve an equivalent performance to mSDA on the Amazons Reviews dataset, whilst using less memory and CPU time consumption. Additionally, for high-dimensional data, it is possible to use 'pivot features'[2] [40] as well as 'most-frequent terms'[3] [24], when applying SDAs.

Stacked Denoising Autoencoders in an ensemble of classifiers

In this subsection, we review a selection of studies, in which, SDAs were considered as part of a classifier combination schema for sentiment analysis. Rong et al. [41] proposed a bagging architecture consisting of multiple SAE trained using bootstrapped training data as well as a classifier on top of the abstract layers. The final decision is set through a majority voting scheme to predict the sentiment. Yang et al. proposed a similar approach *Boosted Multi-Feature Learning* (BMFL), but also incorporated instance weighting (similar to AdaBoost) [42]. This weighting considers both misclassification as well as a domain similarity criterion.

In another study, Yang et al. [43] applied a co-training algorithm to create a SDA (representing corpus-based model), and *Latent Dirichlet Allocation* [44] (to represent a lexicon-based sentiment recogniser). This approach outperformed a linear SVM on Movie Review and SemEval2013. Lastly, Baecci et al. proposed a fusion framework for combining information from textual content (features represented by Continuous Bag of Words) and image content (feature represented by DAs) through a linear regressor to classify sentiment on Twitter [45].

Model Comparison

Since most of the authors of the aforementioned methods did not provide analysis on similar benchmarks and the metrics differ between the papers, it is hard to draw conclusive remarks

[2]Pivot features are features which occur frequently in both the source and target domains and behave similarly (noun, adjective, ...) in both.

[3]Most frequent terms of the vocabulary of unigrams and bigrams.

on the performance of the methods. Moreover, each of the studies used different topologies and parameters of mSDA and therefore, the same accuracy is not comparable even when the same model is used in different papers. However, in Fig. 2, we provide a comparative performance analysis for the studies which have provided results on the domain adaptation task for the Amazon Reviews dataset. As can be seen in this figure, the variants of mSDA are promising approaches for future development and extension.

$$
DA \lesssim \begin{cases} ScheDA^{15} \\ SDA \lesssim \begin{cases} SDA-DSS^{26} \\ mSDA^{27} \lesssim \begin{cases} EASYADAPT+mSDA^{33} \\ GmSDA^{32} \\ BMFL^{42} \\ DNFC^{36} \\ \ell_{2,1}\ SRA^{35} \\ HHTL^{37} \lesssim Instance-wise\ DA^{38} \\ DANN\ with\ mSDA\ representation^{21} \\ mSDA+Target\ Regularisation^{29} \end{cases} \end{cases} \end{cases}
$$

Figure 2: Comparative performance of domain adaptation on the Amazon Reviews dataset.

CONCLUSIONS

Stacked Denoising Autoencoders are widely used in sentiment recognition from (multi-domain, multi-lingual) textual content, due to their capability of creating a useful latent subspace which excludes non-informative contents from the feature representation. This review presented a selection of advances in this field, in terms of the network topologies as well as theoretical background. Approaches such as injecting various levels of noise, incorporating lexicon and sentiment labels, keeping geometrical information, regularisation, domain adaptation, and combining different classifiers can all enhance sentiment recognition within the SDA framework.

Nevertheless, there exist a variety of SDAs, proposed for other domains and learning tasks such as: SDA with dropout to avoid overfitting [46] and sparsification to improve robustness to noise [47] for image classification; iterative SDA for face recognition [48]; and, relational SDA for tag recommendation [49]. These methodologies could be transferred to sentiment recognition with slight adjustments leading to further improvements to system robustness and performance.

Further, despite considerable progress in sentiment analysis, there is still a lack of studies on cross-lingual and cross-cultural analysis. This is mainly due to the lack of large databases in different languages. Although automatic translation could leverage this unavailability, the performance of the translation system may not be sufficiently high enough to transfer sentiments across languages. Moreover, the effect of cultural differences on the presentation

of emotions and sentiments deserves greater research attention.

Similarly, SDAs should benefit the recognition of broader range of affects (such as love, patriotism, hate) rather than the pure positive/negative sentiments nominally performed from textual contents. However, to the best of our knowledge, this paradigm has not been investigated due to a lack of relevant annotated data. Finally, whilst recommendation systems remain the main pillar application for sentiment analysis, it could be also useful in a range of other fields. The benefits of sentiment analysis in dialogue management, brand reputation, health care and mental health state recognition (e. g., depression state), to name but a few applications, have yet to be fully realised.

Acknowledgement

This research has received funding from the European Unions Horizon 2020 Programme research and innovation programme under grant agreements No. 644632 (MixedEmotions) and No. 338164 (ERC StG iHEARu). This research has also received funding from the Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 115902. This Joint Undertaking receives support from the European Unions Horizon 2020 research and innovation programme and EFPIA.

References

1. R. B. Cattell, "Sentiment or Attitude? the core of a terminology problem in personality research," Journal of Personality **9**, 6–17 (1940).

2. C. D. Broad, "Emotion and sentiment," Journal of Aesthetics and Art Criticism **13**, 203–214 (1971).

3. M. Munezero, C. S. Montero, E. Sutinen, and J. Pajunen, "Are they different? affect, feeling, emotion, sentiment, and opinion detection in text," IEEE Transactions on Affective Computing **5**, 101–111 (2014).

4. P. A. Thoits, "The sociology of emotions," Annual Review of Sociology **15**, 317–342 (1989).

5. B. Schuller, A. E.-D. Mousa, and V. Vryniotis, "Sentiment analysis and opinion mining: on optimal parameters and performances," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **5**, 255–263 (2015).

6. P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th International Conference on Machine Learning* (ACM, Helsinki, Finland, 2008) pp. 1096–1103.

7. J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annual Meeting of the Association of Computational Linguistics*, Vol. 7 (Prague, Czech Republic, 2007) pp. 440–447.

8. B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 43rd Annual Meeting of the Association of Computational Linguistics* (Michigan, USA, 2005) pp. 115–124.

9. A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (Association for Computational Linguistics, Portland, Oregon, USA, 2011) pp. 142–150.

10. P. Nakov, S. Rosenthal, A. Ritter, and T. Wilson, "SemEval-2013 task 2: Sentiment analysis in twitter," in *Proc. 7th International Workshop on Semantic Evaluation*, Vol. 2 (Atlanta, USA, 2013) pp. 312–320.

11. P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," Neural networks **2**, 53–58 (1989).

12. S. Zhai and Z. Zhang, "Semisupervised autoencoder for sentiment analysis," in *Proc. 30th Conference on Artificial Intelligence* (Phoenix, Arizona USA, 2016) pp. 1394–1400.

13. Y. Le Cun, *Modèles connexionnistes de l'apprentissage*, Ph.D. thesis, Paris 6 (1987).

14. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research **11**, 3371–3408 (2010).

15. K. J. Geras and C. Sutton, "Scheduled denoising autoencoders," in *Proc. 5th International Conference on Learning Representations* (Toulon, France, 2015) pp. 1–11.

16. K. J. Geras and C. Sutton, "Composite denoising autoencoders," in *Machine Learning and Knowledge Discovery in Databases: European Conference, Part I*, edited by P. Frasconi, N. Landwehr, G. Manco, and J. Vreeken (Springer, Riva del Garda, Italy, 2016) pp. 681–696.

17. H. Zhou, L. Chen, and D. Huang, "Cross-lingual sentiment classification based on denoising autoencoder," in *Natural Language Processing and Chinese Computing* (Springer, Shen Zhen, China, 2014) pp. 181–192.

18. H. Zhou, L. Chen, F. Shi, and D. Huang, "Learning bilingual sentiment word embeddings for cross-language sentiment classification," in *Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (Beijing, China, 2015) pp. 430–440.

19. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436–444 (2015).

20. H. Zuo, G. Zhang, V. Behbood, J. Lu, and X. Meng, "Transfer learning in hierarchical feature spaces," in *Proc. 10th International Conference on Intelligent Systems and Knowledge Engineering* (Taipei Taiwan, 2015) pp. 183–188.

21. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," Journal of Machine Learning Research **17**, 1–35 (2016).

22. M. Kleć, "Sparse autoencoders in sentiment analysis," in *Proc. 9th International Conference on Natural Language Processing* (Warsaw, Poland, 2014).

23. B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. Conference on Empirical Methods in Natural Language Processing, Volume 10* (Association for Computational Linguistics, Stroudsburg, PA, USA, 2002) pp. 79–86.

24. X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. 28th International Conference on Machine Learning* (Washington, USA, 2011) pp. 513–520.

25. C. Scheible and H. Schütze, "Multi-domain sentiment relevance classification with automatic representation learning," in *Proc. 14th Conference of the European Chapter of the Association for Computational Linguistics* (Buenos Aires, Argentina, 2014) pp. 200–204.

26. B. Liu, M. Huang, J. Sun, and X. Zhu, "Incorporating domain and sentiment supervision in representation learning for domain adaptation," in *Proc. 24th International Joint Conference on Artificial Intelligence* (Buenos Aires, Argentina, 2015) pp. 1277–1283.

27. M. Chen, K. Q. Weinberger, Z. Xu, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proc. 29th International Conference on Machine Learning* (Edinburgh, Scotland, 2012) pp. 767–774.

28. M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized stacked denoising autoencoders," in *Proc. Learning Workshop*, Vol. 36 (Utah, USA, 2012) p. 2.

29. S. Clinchant, G. Csurka, and B. Chidlovskii, "A domain adaptation regularization for denoising autoencoders," in *Proc. 54th Annual Meeting of the Association for Computational Linguistics* (Berlin, Germany, 2016) pp. 26–31.

30. P. Vincent and Y. Bengio, "Manifold parzen windows," in *Advances in neural information processing systems* (MIT, Whistler, Canada, 2003) pp. 849–856.

31. G. Hinton and S. Roweis, "Stochastic neighbor embedding," in *Advances in neural information processing systems*, Vol. 15 (Vancouver, Canada, 2002) pp. 833–840.

32. Y. Peng, S. Wang, and B.-L. Lu, "Marginalized denoising autoencoder via graph regularization for domain adaptation," in *Proc. 20th International Conference on Neural Information Processing, Part II*, edited by M. Lee, A. Hirose, Z.-G. Hou, and R. M. Kil (Springer, Daegu, Korea, 2013) pp. 156–163.

33. M. Sun, Q. Tan, R. Ding, and H. Liu, "Cross-domain sentiment classification using deep learning approach," in *Proc. IEEE 3rd International Conference on Cloud Computing and Intelligence Systems* (Shenzhen, China, 2014) pp. 60–64.

34. H. Daumé III, A. Kumar, and A. Saha, "Frustratingly easy semi-supervised domain adaptation," in *Proc. Workshop on Domain Adaptation for Natural Language Processing* (Association for Computational Linguistics, Uppsala, Sweden, 2010) pp. 53–59.

35. W. Jiang, H. Gao, F.-l. Chung, and H. Huang, "The $L_{2,1}$-Norm stacked robust autoencoders for domain adaptation," in *Proc. 30th Conference on Artificial Intelligence* (Phoenix, Arizona USA, 2016) pp. 1723–1729.

36. P. Wei, Y. Ke, and C. K. Goh, "Deep nonlinear feature coding for unsupervised domain adaptation," in *Proc. 25th International Joint Conference on Artificial Intelligence* (New York City, USA, 2016) pp. 2189–2195.

37. J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, "Hybrid heterogeneous transfer learning through deep learning," in *Proc. 28th Conference on Artificial Intelligence* (Québec, Canada, 2014) pp. 2213–2219.

38. L. Chen and W.-y. Deng, "Instance-wise denoising autoencoder for high dimensional data," Mathematical Problems in Engineering **2016**, 13 (2016).

39. A. Budiman, M. I. Fanany, and C. Basaruddin, "Online marginalized linear stacked denoising autoencoders for learning from big data stream," in *Proc. International Conference on Advanced Computer Science and Information Systems* (West Java, Indonesia, 2015) pp. 227–235.

40. J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Sydney, Australia, 2006) pp. 120–128.

41. W. Rong, Y. Nie, Y. Ouyang, B. Peng, and Z. Xiong, "Auto-encoder based bagging architecture for sentiment analysis," Journal of Visual Languages & Computing **25**, 840–849 (2014).

42. X. Yang, T. Zhang, C. Xu, and M.-H. Yang, "Boosted multifeature learning for cross-domain transfer," ACM Transactions on Multimedia Computing, Communications and Applications **11**, 35:1–35:18 (2015).

43. M. Yang, W. Tu, Z. Lu, W. Yin, and K.-P. Chow, "LCCT: a semisupervised model for sentiment classification," in *Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Denver, Colorado, 2015) pp. 546–555.

44. D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," Journal of Machine Learning Research **3**, 993–1022 (2003).

45. C. Baecchi, T. Uricchio, M. Bertini, and A. Del Bimbo, "A multimodal feature learning approach for sentiment analysis of social network multimedia," Multimedia Tools and Applications **75**, 2507–2525 (2016).

46. J. Liang and R. Liu, "Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network," in *Proc. 8th International Congress on Image and Signal Processing* (Trieste, Italy, 2015) pp. 697–701.

47. K. Cho, "Simple sparsification improves sparse denoising autoencoders in denoising highly corrupted images," in *Proc. 30th International Conference on Machine Learning*, Vol. 28 (Atlanta, Georgia, USA, 2013) pp. 432–440.

48. Y. Zhang, R. Liu, S. Zhang, and M. Zhu, "Occlusion-robust face recognition using iterative stacked denoising autoencoder," in *Proc. 20th International Conference on Neural Information Processing, Part III*, edited by M. Lee, A. Hirose, Z.-G. Hou, and R. M. Kil (Springer, Daegu, Korea, 2013) pp. 352–359.

49. H. Wang, X. Shi, and D.-Y. Yeung, "Relational stacked denoising autoencoder for tag recommendation," in *Proc. 29th AAAI Conference on Artificial Intelligence*, AAAI (AAAI Press, 2015) pp. 3052–3058.