Studying the feasibility of Domain Adaptation for Emerging Yelp Business Domains

DKE Team Project Report

Rafi Trad, Imad Hajjar, and Ali Hasham

Otto-von-Guericke University
Magdeburg, Germany

Abstract. Not always is it possible to attain an adequate amount of labelled data in a specific domain for machine learning purposes, and manual annotations would be too arduous. In such cases, a promising solution would be to exploit data from other domains to address the lack of data in the domain of interest, aka. the target domain; such an approach is called domain adaptation, and it entails the challenge of the test data and training data being sampled from different distributions. Many domain adaptation approaches were devised to address this challenge in NLP settings, when the models are built in one domain (source domain) and utilised in a new one (target domain); However, to the best of our knowledge, the target domain's available data were to be exploited in some way in order to train the cross domain models.

In our work, we investigate to which extent the domain adaptation is profitable for new emerging domains, as we study the feasibility of building a cross domain classifier for the sake of sentiment polarity detection without exploiting any data from the target domain of interest throughout training. Two target domains were experimented with according to their similarity to the source domain, in order to account for the two extreme cases: the closest and farthest from the source domain. The domain adaptation attempt was evaluated, and its feasibility proved to be rather insignificant.

Keywords: Transfer learning · Domain Adaptation · Sentiment Analysis · Polarity Detection · Opinion Mining · Sentiment Classification.

1 Introduction

The myriads of generated data each day paved the way in front of a multitude of applications to emerge and gain accelerated interest. One kind of such data repositories with precious utility for businesses is online review databases. Learning from such databases, or from the crowds, can lead to significant boosts in the competitiveness of different businesses. Sentiment Analysis / Opinion Mining is an important application in such environments with many domains, whereby the contentment and satisfaction of customers in regard to products / services is measured automatically, since the humongous data size makes it prohibitive to rely on manual processing [3].

Practically, it is often the case that a specific business domain enjoys a flood of reviews whose polarities are predefined (labelled) while another one doesn't, viz., the latter domain either has only a limited amount of data, or a plentiful of them yet only a tiny proportion is labelled. That would hinder the performance of the typical supervised classifiers for sentiment analysis, and consequently it would be tempting to exploit the available data from one domain in favour of the other. Such a *Transfer Learning* attempt is more precisely known as *Domain Adaptation*, in which a model is trained on one or more domains and used in a new domain [4], but that is particularly problematic for opinion mining.

Different domains are underpinned by different data distributions, so the training data would be from one distribution and the test data/ actual data from another one. In such cases, the accuracy of trained models/classifiers deteriorate ([4], [6]), especially keeping in mind that Sentiment Analysis is domain dependent [3]. And systems, which have been developed with a specific domain borne in mind, yield poor performance on other domains. This issue starts as early as the features engineering stage, in which the relevant features are extracted from the dataset and sometimes moulded into useful forms [1].

We seek to perform domain adaptation and alleviate its aforementioned challenges by keeping the model somewhat general. Careful feature extraction augmented with lexical semantic processing shall be performed, alongside tackling common relevant problems like over-fitting in order to build a cross-domain polarity detector. Presumably, and as mentioned earlier, the abstraction of the model is not without detractors, because it probably is at the cost of a higher level of performance if the detector were domain-specific rather than cross-domain. But in scenarios where in-domain labelled data are scarce, or even absent, such an attempt is in order.

We will harness Yelp, a famous review database which spreads across a variety of business domains and has been used by many users to find good local businesses. Users write reviews in free-text format in addition to an integer rating system of 5 stars to denote satisfaction. Yelps intrinsic knowledge has a profound effect on businesses, For instance, "An extra half-star rating causes restaurants to sell out 19 percentage points more frequently, with larger impacts when alternate information is more scarce" [2].

2 Related Work

The infamous domain specificity property of Sentiment Analysis is well known among researchers. [3] recognised this limitation while trying to customise Sentiment Analysis classifiers to new domains whose labelled data were too few to allow for full supervised learning. Four approaches were followed to overcome the challenges of domain dependency: training one classifier on all available data, limiting the source domain's features to those observed in the target domain, assembling an ensemble of classifiers, and using the target domain's available data (the so called in-domain unlabelled data).

[7] sat a special focus on the fact that in order to customise sentiment analysis to new domains, the discriminative learning methods' assumption of the test and train data being from the same distribution should be discarded. They harnessed structural correspondence learning to link source domain's features (the resource-rich domain) to the target's (the resource-poor one), and formed correspondences between the two sets of features, outperforming supervised and semi-supervised learning approaches. As a sequel, [6] extended the structural correspondence learning (SCL) algorithm reducing its error due to adaptation, and correlated the similarity of domains with the practicality of adaptation among them. Pivot features were carefully selected according to their frequency and mutual information with source labels, and the $\mathcal{A}$-distance was established as a way of discerning how adaptable a domain is to another one.

To assess the feasibility of cross-domain adaptations under different circumstances, [5] investigated the need for a proper feature representation to empower domain adaptation. They formally defined a generalisation bound of a classifier trained on the source domain and was to be applied in the target domain, which depended on the structured representation for domain adaptation. In addition, realising the limitation of discriminative learning methods and the inability to initiate a full-fledged learning attempt on a domain whose labelled data were too few, [4] theorised about the conditions under which a cross domain classifier is expected to perform well on target data, and how to combine them during training with the large amount of labelled source data to attain the best performance in the new domain.

[13] wielded a deep learning approach (based on Stacked Denoising Auto-Encoders) to abstract meaningful representations for reviews, and utilised these high-level feature representations throughout training, topping the then state-of-the-art approaches. Thenceforth, neural approaches became the state-of-the-art instead of SCL, but SCL remained appealing to [23] who married it to auto-encoders neural networks in order to encode information from pre-trained word embeddings, which improved the performance of the cross domain classifier.

A new representation learning approach was proposed by [11], positing that competent domain adaptation relies on features whose domain cannot be distinguished easily. The idea was implemented in a neural network architecture trained on labelled data from the source domain and unlabelled ones from the target, i.e. there was no need for any labelled data from the target domain, and the distributions of features across the two domains were aligned via back-propagation.

3 Background and Problem Formalisation

3.1 Sentiment Polarity Analysis

Sentiment Analysis -SA- (also called Opinion Mining, review mining or appraisal extraction, attitude analysis) is the task of detecting, extracting and classifying opinions, sentiments and attitudes concerning different topics, as expressed in textual inputs[21]. It is the study of subjectivity and polarity, the former being how emotional an opinion/review is, and the latter is about affection (negative or positive) towards the discussed aspects [8]. In our setting, we will concentrate on the sentiment polarity analysis only.

3.2 Transfer Learning and Domain Adaptation

It is not clear how to distinguish between *Transfer Learning* and *Domain Adaptation*, at least not yet; some literature even use the two terms synonymously[14], but we would like to consider *Domain Adaptation* to be a special case of transfer learning. Abiding by the definition provided by [19], *transfer learning* can be formalised as: "Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function $f_T(.)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$."

Three categories of transfer learning exist, Fig. 1: *inductive*, *unsupervised*, and *transductive.* The *inductive transfer learning* is the case where the learning tasks $\mathcal{T}_S$ and $\mathcal{T}_T$ are different, i.e. $\mathcal{T}_S \neq \mathcal{T}_T$ regardless of the source and target domains ($\mathcal{D}_S$ and $\mathcal{D}_T$ respectively); If the target task $\mathcal{T}_T$ is different from but still related to the source task $\mathcal{T}_S$, we arrive at the *unsupervised transfer learning.* Finally, the *transductive transfer learning* setting has $\mathcal{T}_S = \mathcal{T}_T$ but $\mathcal{D}_S \neq \mathcal{D}_T$. Consequently, *Domain adaptation* can be defined as a special case of the *transductive transfer learning* where the feature spaces between domains $\mathcal{X}s$ are similar; viz., $\mathcal{T}_S = \mathcal{T}_T \wedge \mathcal{D}_S \neq \mathcal{D}_T \wedge \mathcal{X}_S = \mathcal{X}_T$ ([19],[14]).

4 Domain Adaptation Methodology

Domain adaptation can be carried out in many ways, which we believe are well structured in [14] for Natural Language Processing (NLP) settings, Tab. 1. The approach we plan to adopt is similar to the *feature space transformation - generalisable feature selection*, in which a set of source domains $\{d_k\}_{k=1}^{K}$ with their labelled data are used to train a model for a target domain, with which they share some features called *generalised features*, a subset of the original feature set.

Reflecting on our planned approach, we adapt the two rough steps of *generalisable feature selection* approach thus:

- Domain Selection: identify a source domain $\mathcal{D}_S$ which has an ample amount of labelled reviews, and perform features engineering steps that would keep the

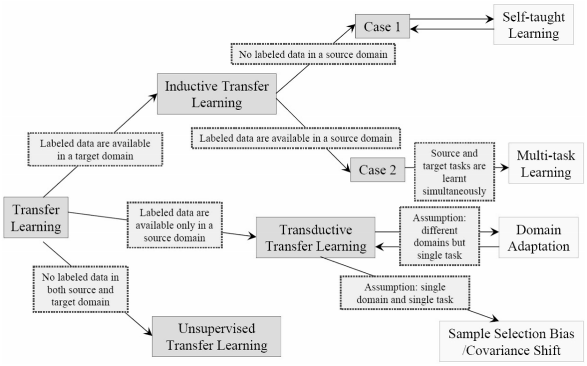

Case 1
Self-taught Learning
No labeled data in a source domain
Inductive Transfer Learning
Labeled data are available in a target domain
Labeled data are available in a source domain
Case 2
Source and target tasks are learnt simultaneously
Multi-task Learning
Transfer Learning
Labeled data are available only in a source domain
Transductive Transfer Learning
Assumption: different domains but single task
Domain Adaptation
No labeled data in both source and target domain
Assumption: single domain and single task
Unsupervised Transfer Learning
Sample Selection Bias /Covariance Shift

Fig. 1: An overview of different types of Transfer Learning, reprinted from [19].

Table 1: Domain Adaptation different approaches' classes for NLP tasks.

Feature Space Transformation	Simple Feature Augmentation
	Structural Correspondence Learning (SCL)
	Generalizable Feature Selection
	Distributional Representation
Prior Based Adaptation	MAP Adaptation
	Hierarchical Bayesian Framework
	Estimating Priors in Target Domain
Instance Selection and Weighting	Pseudo In-Domain Data Selection
	Self-training and TrAdaBoost
	Instance Weighting Framework

features $\mathcal{X}_S$ at a generalisable level and shareable with the target domain $\mathcal{D}_T$ -accounting for the relation between the domains-, using lexical and semantic processing. Two $\mathcal{D}_T$ domains will be chosen according to their similarity to $\mathcal{D}_S$ (the most and least similar); Sec. 5 details this procedure.

- Domain Adaptation: The generalised set of features $\mathcal{X}_S$ alongside the labels are harnessed to train a polarity detector ξ_{ST} that will be deployed in the target domains of interest $\mathcal{D}_T$. In our work, we assume that $\mathcal{D}_T$ is a new emerging domain, so no instances from $\mathcal{X}_T$ shall be used in the training of the classifier.

4.1 Source In-domain Modelling

It is intuitive to think that a cross-domain classifier which will perform in another domain should at first perform adequately in its own domain, i.e., achieve a good in-domain performance. Subsequently, we will assess the in-domain performance of the classifier ξ_{ST}.

For that sake, a baseline model ξ_{bS} shall be built to define a sort of ground truth in order to compare the in-domain performance of ξ_{ST} with. An important aspect to mention is that ξ_{bS} will be trained and tested using the original 5-star labels as well as the 3-label polarity calibre, hence we distinguish between the two using ξ_{bS5} notation for the former, and ξ_{bS3} for the latter. For other classifiers, only the 3-label scale will be used.

5 Choosing Yelp Source and Target Domains

In the context of Domain Adaptation, not all domains are adaptable to each other to the same extent; for instance, one would expect the laptops domain to be adaptable to mobile phones more than software services. Such domain divergences may be too acute that the adaptive classifier's performance worsens with the transfer of knowledge rather than improves (*negative transfer*) [19]. Additionally, the domain may be itself intractable, i.e. with a maximum achievable accuracy that is notably less than that of another more amenable domain [3].

5.1 Choosing $\mathcal{D}_S$

To choose the source domain we opted for the one whose labelled data are the most abundant, especially that the supervised sentiment polarity analysis techniques do require a considerable amount of labelled reviews, hence we chose the *Restaurants* domain as our source domain.

5.2 Choosing $\mathcal{D}_T$

Choosing the target domain should be painstaking as $\mathcal{D}_T$ shouldn't diverge a lot from $\mathcal{D}_S$, lest the adaptability be too poor due to a negative transfer attempt; subsequently, to measure the transferability and adaptability across domains is in order. Common measures of divergence (L_1, $Kullback - Leibler$, ..) aren't accurate if applied in the *transductive distribution-free* settings [4], but a usable metric in that context is the $\mathcal{A}$-distance.

The $\mathcal{A}$-distance. Originally, the $\mathcal{A}$-distance assesses the similarity between two probability distributions, and it has been shown that it upper-bounds the generalisability for the sake of domain adaptation [13]. The domains are characterised by their induced distributions on the instance space, which means the more distinctive the domains, the more divergent their distributions. In our setting, we are particularly concerned in those differences that have direct impact on classification accuracy [6]. Intuitively, if two domains are adaptable to each other, they share an amount of similitude with which it becomes hard to distinguish between them. This idea can be utilised in order to approximate the intractable $\mathcal{A}$-distance between them, the so called proxy $\mathcal{A}$-distance (PAD or $\hat{d}_A$), Eq. 1. One way is to use a linear SVM proxy trained to discriminate between the two domains, then computing its error ϵ, which will be used in the final approximation as [13]:

$$\hat{d}_A = 2(1 - 2\epsilon) \ . \tag{1}$$

The pairwise proxy $\mathcal{A}$-distances between the restaurants domain and all other domains were measured harnessing a linear bag-of-words SVM classifier, following the steps:

1. Generating two sets of sentences belonging to the domains we want to measure the distance between. For each domain, we elicited all sentences and subsequently selected a random sample of 10,000 sentences, which served in balancing the mixed training dataset afterwards and increased the efficiency considerably.
2. Common vocabulary between the domains was assembled using the sentences sets.
3. A training dataset was compiled through mixing two sets of domain's sentences: $U = \{(x_i, 0)\}_{i=1}^{n} \cup \{(x_i, 1)\}_{i=n+1}^{N}$[11].
4. A linear SVM classifier was trained to distinguish between domains' sentences.
5. The generalisation error ϵ on the problem of discriminating between source and target sentences was eventually used in Eq. 1 to get the $Proxy\ \mathcal{A} - distance$ (PAD) approximation [11], which we averaged over 10 runs for the sake of a more plausible approximation.

The latest 3 steps were mainly carried out using an available PAD implementation[1].’It is noticeable that as ϵ increases, PAD decreases Fig. 2a, indicating that the two domains are more and more similar, and it reaches 0 when they are identical. Additionally, the source risk becomes indicative of the target's if $\mathcal{D}_T$ is akin to $\mathcal{D}_S$[11].

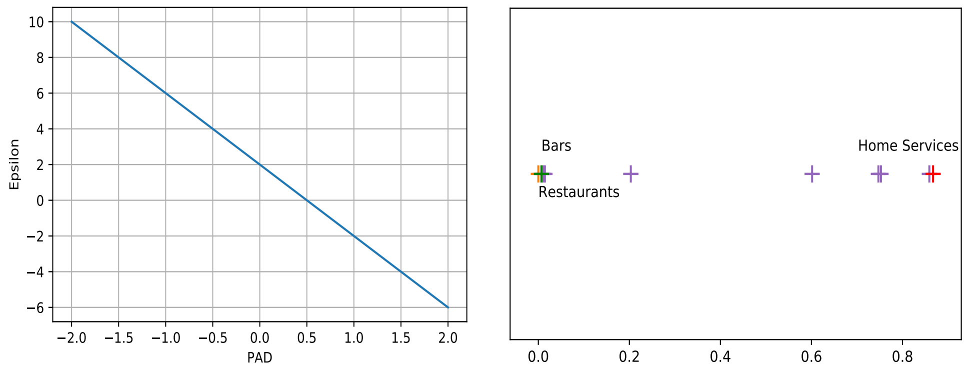
According to the results we obtained Fig. 2b the closest domain to *Restaurants* was *Bars*, while the remotest was the *Home Services*; consequently, thus the two will serve as $\mathcal{D}_T$ in our domain adaptation experiments.

6 Dataset

Upon scrutiny, the Yelp dataset (round 10) consists of around 7.27M reviews, separated into 11 businesses. Figure 4a illustrates the separation of reviews into positive, negative and neutral for each of the 11 businesses. It is evident that the positive label pervades the label space regardless of the domain.

Since textual reviews play an integral role in our research, we have also generated the associated word clouds for positive, negative and neutral reviews of the 3 businesses we tackled (Restaurants, Bars and Home Services) Fig. 3, in addition to the average review length per domain Fig. 4b.

[1] `https://github.com/rpryzant/proxy-a-distance`



(a) (b)

Fig. 2: (a): PAD relationship to ϵ. (b): The measurements, where pairwise PAD distances between restaurants (extreme left, in yellow), and other businesses are depicted. Bars domain is the closest (in green), while home services is the remotest (in red).

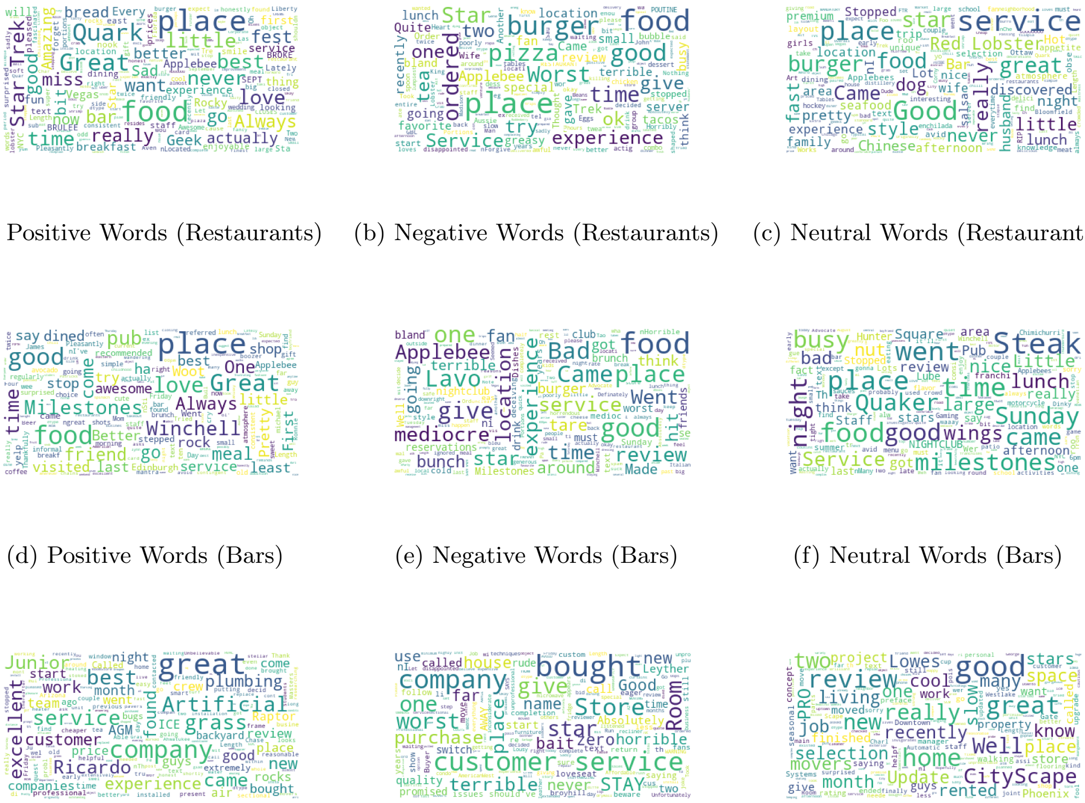
7 Preprocessing

Two preprocessing pipelines were implemented, one for each row of Tab. 2. For the baseline, we have applied standard preprocessing steps in following sequence:

- Dropping records with null values (2 such records).
- Replacing new line characters, slashes (/) and punctuation.
- Case-folding.
- Tokenisation.
- Removing stop-words.
- Stemming.

To preprocess the data for the gold-standard models, we have applied the steps in following sequence:

- Dropping records with null values (2 records).
- Replacing foreign accents with most likely letters by using Python's Unicode library, which provides the ASCII of transliterations Unicode words.
- Case-folding.
- Removing newlines, tabs, replacing ” with ’.
- Expanding the abbreviations before removing the punctuation, in order to pay attention to the linguistic abbreviations. Expansion of abbreviation is achieved by means of a manual dictionary which holds the most likely expansions of an abbreviation.
- Removing punctuations.
- Tokenisation.

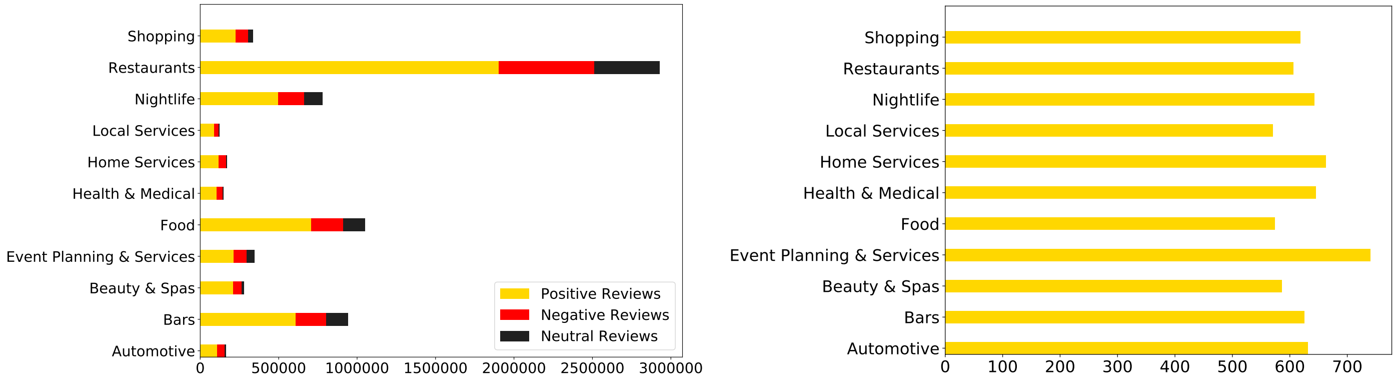


(a) Positive Words (Restaurants) (b) Negative Words (Restaurants) (c) Neutral Words (Restaurants)

(d) Positive Words (Bars) (e) Negative Words (Bars) (f) Neutral Words (Bars)

(g) Positive Words (Home Services) (h) Negative Words (Home Services) (i) Neutral Words (Home Services)

Fig. 3: Word Clouds for the 3 domains under scrutiny (Restaurants, Bars and Home Services).



(a)

(b)

Fig. 4: (a): Number of reviews per domain alongside the distribution of the class labels. (b): The average review length per domain.

- Adding PoS tags to tokens.
- Lemmatising PoS tagged tokens.
- Detecting multi-word phrases inside a sentence (converting the common bi-grams into a single word).
- Feature Selection using ANalysis Of VAriance (ANOVA).

Polarisation. Yelp uses a scale of 5 stars to measure contentment of reviewers. However, for all of our cross-domain experiments, we will use a cruder scale, which is the tripartite polarity scale of negative, positive and neutral. Converting the 5-star labels to the 3-polarity ones is the process we call *polarisation*, and the effect of adopting the new polarity scale is demonstrated in the difference of performance between ξ_{bS5} (5-star labels) and ξ_{bS3} (3-polarity labels) Sec. 9.1.

Vectorisation. The data textual representation should be converted to a quantified one, because this numeric quantification of the text is necessary for the machine learning algorithms. The datasets were transformed into a $n \times m$ matrix form where we have n documents (reviews) and m terms. The matrix element $c_{i,j}$ represents the frequency of the term j in document i. This way of representing words and documents is commonly known in the Information Retrieval community as *Bag of Words* [16].

$$\begin{bmatrix} c_{00} & c_{01} & \cdots & \cdots & c_{0m} \\ c_{10} & c_{11} & \cdots & \cdots & c_{1m} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{n0} & c_{n1} & \cdots & \cdots & c_{nm} \end{bmatrix} \tag{2}$$

8 Modelling

To carry out the experiments of domain adaptation, we have built six types of models, Tab. 2. The rows of the table represent whether the model is based on the standard (baseline) or the more-advanced preprocessing steps. The columns represent whether the models have been applied locally (in-domain), or in another domain (cross-domain). We refrain from detailing the process of building every single model for the sake of brevity.

Table 2: The Models required for our Experiments.

	Used In-Domain	Used Cross-Domain
Baseline	ξ_{bS3}, ξ_{bS5}, ξ_{bT}	ξ_{bST}
Advanced	ξ_{T}	ξ_{ST}

8.1 Models Used

Usually, the data of text classification problems conform to a multinomial distribution, where multiple events (terms) happen in n number of trials (the number of reviews in our dataset) and multiple outputs are observed (1-5 stars) in those n reviews. In order to construct the ground truth model ξ_{bS} in $\mathcal{D}_S$, besides other baseline models, we opted for a widely used algorithm for text classification, which is *Multinomial Naive Bayes* (MNB) [17].

For the advanced and gold-standard models, Linear Support Vector Machines (SVM) with a specific learning algorithm called Stochastic Gradient Descend (SGD) was harnessed. The main reason for choosing SVM is that it's one of the state-of-the-art methods for text classification [9]. An additional reason is that the data is practically polarised into 2 categories (positive and negative), so a linear classifier is expected to perform adequately.

8.2 Software Used

Two Python libraries that are well known in Machine Learning and Natural Language Processing fields were utilised:

- nltk [15]: The Natural Language Toolkit – is a suite of open source Python modules, data sets, and tutorials supporting research and development in Natural Language Processing.
- scikit-learn [20]: Python libraries for data mining and analysis.
- Gensim [22]: which is a Python toolkit for topic modelling (used only to detect common multi-word expressions).

Configuration of MNB Parameters. The objective function of MNB is shown in Eq. 3.

$$s_{map} = \arg\max_{s \in S} P(s|r) = \arg\max_{s \in S} P(s) \prod_{t \in r} P(t|s) \ . \tag{3}$$

$P(s)$ is the prior probability of a star-rating/polarity, and $P(t|s)$ is the conditional probability that term t occurred knowing that the star-rating/polarity was s. $P(t|s)$ has a smoothing parameter α to address the problem where the conditional probability might be zero. In order to choose a reasonable value for α, GridSearchCV class of sklearn was used. A 3-fold cross-validation process for each α_i was applied, where $\alpha_i \in [0.1, 1]$ with step=0.1 (30 runs). After that, we chose the model with the highest F1 mean score.

Configuration of SGD Parameters. The objective function of SGD is shown in Eq. 4, and the learning phase involves minimising this function.

$$E(w, b) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \alpha R(w) \ . \tag{4}$$

We sat two parameters manually for SGD:

- random_state = 0: The seed of the pseudo random number generator to use when shuffling the data. We sat this parameter for a practical reason so we can get the same results over multiple runs.
- class_weight = ‘balanced’: to adjust the weights inversely proportional to class frequencies in the input data.
- All other parameters were left at their default values except for α which was tuned in a similar manner to the α of MNB but with a range of [0.00005, 0.00015]. It is worth mentioning that L is by default the Hinge Loss function, and R is the L2 norm regularization function.

Training and Testing the models. The dataset was split into training set (70%) and testing set (30%). As formerly mentioned, all of our models were built using the 3-polarity scale (except for ξ_{bS5}). In that context, and inasmuch as the neutral reviews are ambiguous, we trained the models using only positive and negative labels, and considered the reviews that the model was not confident about as neutral (for MNB: using an uncertainty interval of $]20\% - 80\%[$ to denote the neutral label, while the Hinge Loss function was utilized in SGD). Regarding testing, it was carried out against a test set that has the three labels.

On the other hand, only one model (ξ_{bS5}) was built using the original 5-star scale, both in the training and testing phases. In this model, an MNB classifier was utilised.

9 Evaluation

9.1 Source In-Domain Evaluation

Evaluation Criteria. The evaluation of the in-domain performance was executed using Accuracy, F1-score [18] (Higher values of these measures usually mean that the model performs well) and Cohen's Kappa [10]. Cohen's Kappa measures the degree of agreement for two classifiers. $\kappa = 1$ means total agreement between the two classifiers. $\kappa \leq 0$ means it's more likely the agreement happened by chance. Cohen's Kappa statistic would be computed for ξ_{ST} and two random classifiers, one of which predicts the most frequent label, while the other's predictions are randomly stratified, i.e., they adhere to the class distribution. The purpose of such a comparison is to measure to which extent the cross-domain classifier agrees with a random dummy one. The formula of Cohen's-kappa measure is:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad . \tag{5}$$

Where:

- p_o is the probability of observed agreement between the two classifiers (similar to the Accuracy measure).
- p_e is the probability of expected agreement (hypothetical agreement), and is calculated as:

  $$p_e = \frac{1}{N^2} \sum_{k \in K} n_{k1} n_{k2} \quad . \tag{6}$$

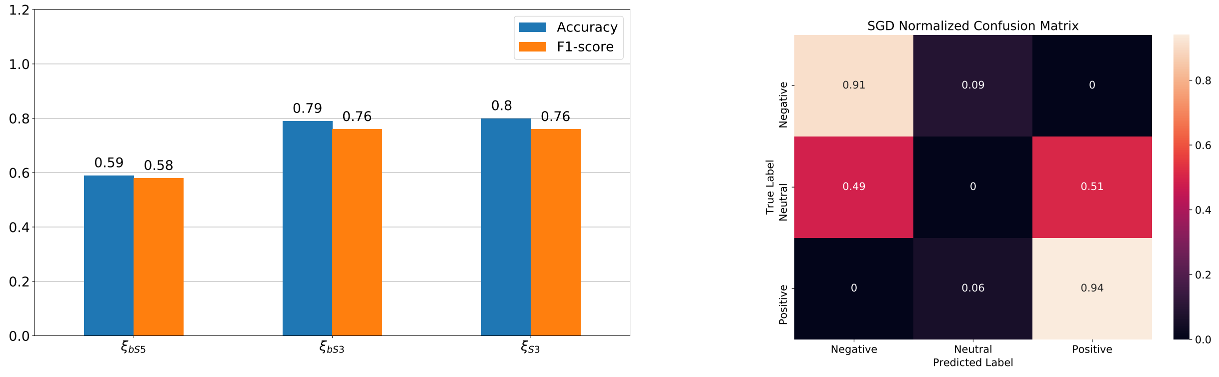
  Where:

  - N: The total number of reviews.
  - K: The labels (negative, positive and neutral).
  - n_{ki}: The number of times classifier i predicted label k.

In-Domain Evaluation Results. Figure 5 shows the results of the in-domain evaluation. Regarding comparing the ξ_{ST} with the random classifiers, we got $\kappa = 0$ when using a 'most-frequent' random classifier and $\kappa \approx -0.02$ when using a stratified random classifier. The interpretation for both of these numbers is that the classifier in question agrees with both of the random classifiers only by chance.

9.2 Cross-Domain Evaluation

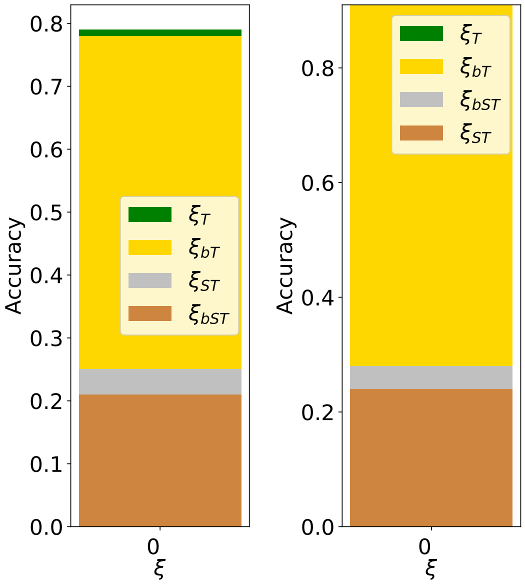
Evaluation Criteria



(a)

(b)

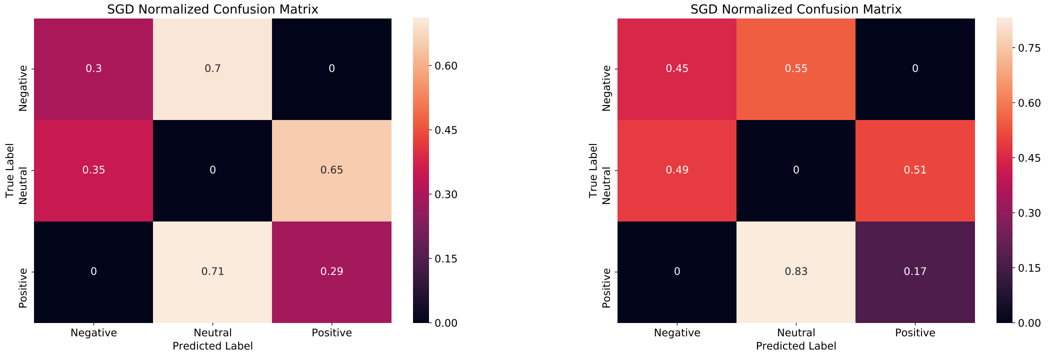
Fig. 5: (a): The accuracy and f1-score of in-domain performances. (b): The confusion matrix of ξ_{ST} when applied in Restaurants.



(a) Bars

(b) Home Services

Fig. 6: The models' accuracy in the two target domains, (a) Bars and (b) Home Services. The maximum attainable accuracy in our scenario is that of the gold-standard in $\mathcal{D}_T$, i.e. ξ_T, and other accuracies are plotted upon it to see to which degree we drew close to the performance of ξ_T. In (b), the accuracy of ξ_T and ξ_{bT} are identical.



(a) Bars

(b) Home Services

Fig. 7: The confusion matrices of ξ_{ST} when applied in (a) Bars and (b) Homes Services.

Transfer Loss [13]. It is important to be able to measure how the cross-domain polarity detector ξ_{ST} will perform within the target domain $\mathcal{D}_T$, and in that regard we hereafter define the **transfer loss**. As a preface, we denote the cross-domain transfer error as $e(S,T)$, where the train data are from the source domain $\mathcal{D}_S$ and the test data are from $\mathcal{D}_T$; clearly the classical in-domain error is then $e(T,T)$. The error which a baseline model ξ_{bT} produces in the target domain $\mathcal{D}_T$ is denoted as $e_b(T,T)$. Having the basic definitions written, we thereupon state the transfer loss $Loss(S,T)$ when we adapt from domain $\mathcal{D}_S$ to $\mathcal{D}_T$ as the difference between the transfer error and the in-domain baseline error Eq. 7.

$$Loss(S,T) = e(S,T) - e_b(T,T) = e(\xi_{ST}) - e(\xi_{bT}) \ . \tag{7}$$

The smaller the transfer loss, the better the transfer, and it can happen that $Loss(S,T) < 0$ if ξ_{ST} outperforms ξ_{bT}. In our scenario, we will define the error a classifier produces as $e(\xi) = 1 - acc(\xi)$, where acc denotes the accuracy.

Adaptation Loss [6]. Let the *classification accuracy* of a cross-domain polarity detector ξ_{ST} be 76% for instance, and the accuracy of a gold standard model ξ_T in $\mathcal{D}_T$ is 82%, then the *adaptation loss* of ξ_{ST} would be 6%, according to Eq. 8. Additionally, let it be that a baseline transfer model ξ_{bST} achieves a score of 70%, hence we write that *the relative reduction of error due to adaptation* is 50%, Eq. 9.

$$Adaptation\ Loss \ = acc(\xi_T) - acc(\xi_{ST}) \ . \tag{8}$$

$$Relative\ Reduction\ of\ Error = \frac{acc(\xi_{ST}) - acc(\xi_{bST})}{acc(\xi_T) - acc(\xi_{bST})} \ . \tag{9}$$

McNemar's Test. One often disregarded issue when measuring different classifier's performances is to statistically substantiate the significance of such observed differences, and to take into account the variability and uncertainty. For that purpose, we will exploit *McNemar's test*, which shall judge how statistically significant the differences we observe are. McNemar's test works well when the errors two algorithms (or classifiers) produce are independent, which fits the scenario where these two algorithms are isolated.

The p-value can be directly computed as in equations 10 and 11 to denote the percentage of cases in which the observed difference is a result of chance, hence providing an evidence in favour of or against the genuineness of differences[12]. It is worth noting that the joint performance of the two classifiers can be summarised by a 2 × 2 table, see Tab. 3, where $k = N_{10} + N_{01}$, i.e. the number of cases in which only one of the two classifiers makes an error.

$$P = 2 \sum_{m=n_{10}}^{k} \binom{k}{m} \left(\frac{1}{2}\right)^{k} : n_{10} > \frac{k}{2} \ . \tag{10}$$

$$P = 2 \sum_{m=0}^{n_{10}} \binom{k}{m} \left(\frac{1}{2}\right)^{k} : n_{10} < \frac{k}{2} \ . \tag{11}$$

Table 3: 2 × 2 Joint Performance Table of Two Classifiers.

	Correct	Incorrect
Correct	N_{00}	N_{01}
Incorrect	N_{10}	N_{11}

Results of Cross-Domain Evaluation (Domain Adaptation). Figures 6, 7 and 8 summarise the results of our domain adaptation attempt. After applying McNemar's test and verifying the results[2], all the differences in performance among classifiers proved to be statistically significant.

10 Discussion and Conclusion

Pertaining to the in-domain experiments, it was evident that coarsening the label space enhanced the accuracy and F1 score considerably. On the other hand, using advanced modelling (advanced preprocessing pipeline and SVM) boosted the accuracy only by a minute amount, if ever (by 1% in restaurants Fig. 5a, and almost similarly in Bars and Home Services Fig. 6 - compare ξ_{bT} and ξ_{T}).

[2] Verification was carried out via `http://vassarstats.net/propcorr.html`

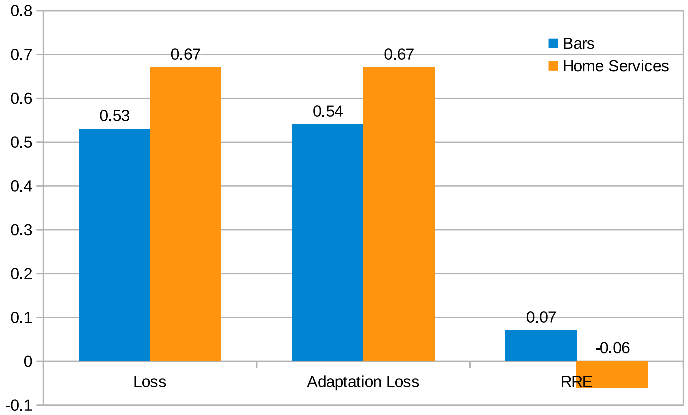


Fig. 8: Transfer Loss, Adaptation Loss and RRE of ξ_{ST} Bars and Home Services.

Cross domain experiments demonstrated a characterless performance when the ξ_{ST} was applied in the new domains Fig. 6. One noticeable difference we observe when shifting from Bars, the closest target domain, to Home Services, the farthest one, is that the advanced modelling approach showed an improvement in Bars, but a deterioration in Home Services. Moreover, Neither the transfer loss nor adaptation loss varied by much across the two target domains we have, as they were severe in the two cases. However, the relative reduction of error due to adaptation 8 varied when changing $\mathcal{D}_{\mathcal{T}}$: while it improved in Bars, it decreased in Home Services.

The overall feasibility of the domain adaptation for emerging domains was insubstantial, even for the closest domain to our source domain, abiding by the theory of learning [4], i.e. using a set of observations whose domain of origin cannot be easily discriminated from $\mathcal{D}_{\mathcal{S}}$ [11].

Should some data from $\mathcal{D}_{\mathcal{T}}$ were used in training ξ_{ST}, the efficacy of domain adaptation is expected to boost significantly, but since we are considering the target domains to be brand new (emerging), that option wasn't available. A prospective solution we speculate would be utilising a set of source domains instead of only one, so that we build a shareable generalisable representation of features among the source domains, assuming that these domains do share some features, in order to train a cross domain classifier that is intended to be deployed in the emerging target domain we desire [14].

References

1. Akhtar, M.S., Gupta, D., Ekbal, A., Bhattacharyya, P.: Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis. Knowledge-Based Systems **125**, 116–135 (2017)
2. Anderson, M., Magruder, J.: Learning from the crowd: Regression discontinuity estimates of the effects of an online review database. The Economic Journal **122**(563), 957–989 (2012)
3. Aue, A., Gamon, M.: Customizing sentiment classifiers to new domains: A case study. In: Proceedings of recent advances in natural language processing (RANLP). vol. 1, pp. 2–1. Citeseer (2005)
4. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine learning **79**(1-2), 151–175 (2010)
5. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: Advances in neural information processing systems. pp. 137–144 (2007)
6. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th annual meeting of the association of computational linguistics. pp. 440–447 (2007)
7. Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 conference on empirical methods in natural language processing. pp. 120–128. Association for Computational Linguistics (2006)
8. Calefato, F., Lanubile, F., Maiorano, F., Novielli, N.: Sentiment polarity detection for software development. Empirical Software Engineering pp. 1–31 (2017)
9. Chrisopher H. Manning, Prabhakar Raghavan, H.S.: Support vector machines and machine learning on documents, chap. 15, pp. 319–348. Cambridge University Press
10. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement **20**(1), 37–46 (1960). https://doi.org/10.1177/001316446002000104, `https://doi.org/10.1177/001316446002000104`
11. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research **17**(1), 2096–2030 (2016)
12. Gillick, L., Cox, S.J.: Some statistical issues in the comparison of speech recognition algorithms. In: Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on. pp. 532–535. IEEE (1989)
13. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 513–520 (2011)
14. Li, Q.: Literature survey: domain adaptation algorithms for natural language processing. Department of Computer Science The Graduate Center, The City University of New York pp. 8–10 (2012)
15. Loper, E., Bird, S.: Nltk: The natural language toolkit. In: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1. pp. 63–70. ETMTNLP '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002). https://doi.org/10.3115/1118108.1118117, `https://doi.org/10.3115/1118108.1118117`

16. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, chap. 6. Cambridge University Press, New York, NY, USA (2008)
17. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, chap. 13, pp. 253–261. Cambridge University Press, New York, NY, USA (2008)
18. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, chap. 8, pp. 155–156. Cambridge University Press, New York, NY, USA (2008)
19. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2010)
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
21. Ravi, K., Ravi, V.: A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. Knowledge-Based Systems (2015). https://doi.org/10.1016/j.knosys.2015.06.015
22. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), `http://is.muni.cz/publication/884893/en`
23. Ziser, Y., Reichart, R.: Neural structural correspondence learning for domain adaptation. arXiv preprint arXiv:1610.01588 (2016)