International Journal of Information Technology & Decision Making

SOCIAL MEDIA CROSS-SOURCE AND CROSS-DOMAIN SENTIMENT CLASSIFICATION

Received Day Month Year
Revised Day Month Year

Due to the expansion of Internet and Web 2.0 phenomenon, there is a growing interest in sentiment analysis of freely opinionated text. In this paper, we propose a novel cross-source cross-domain sentiment classification, in which cross-domain labeled Web sources (Amazon and Tripadvisor) are used to train supervised learning models (including two deep learning algorithms) that are tested on typically non labeled social media reviews (Facebook and Twitter). We explored a three step methodology, in which distinct balanced training, text preprocessing and machine learning methods were tested, using two languages: English and Italian. The best results were achieved using undersampling training and a Convolutional Neural Network. Interesting cross-source classification performances were achieved, in particular when using Amazon and Tripadvisor reviews to train a model that is tested on Facebook data for both English and Italian.

Keywords: Convolutional neural network; Cross-domain data; Sentiment analysis; Social media; Facebook; Twitter.

1. Introduction

Technological advances, such as the Internet expansion, Web 2.0 phenomenon and massive mobile device adoption, have increased the availability of freely opinionated text (e.g., blog reviews, social network comments). This big data source of unstructured texts enriches the value of sentiment analysis, also termed opinion mining, which uses computational methods to automatically analyze human opinions, sentiments and evaluations towards entities (e.g., products, services, organizations).[1] Indeed, several studies have analyzed opinion dynamics in social networks and their potential impact in decision making.[2,3,4] Thus, sentiment analysis is a key tool of modern decision support systems, helping to support decisions in several real-world applications, such as involving hotels,[5] stock markets,[6,7] and traffic accidents.[8]

Given the importance of social media platforms (e.g., Facebook, Twitter), several works have proposed supervised machine learning algorithms for the sentiment analysis of social media texts (e.g., Naive Bayes, Support Vector Machines).[9] Yet, designing an accurate machine learning classifier for a particular sentiment domain and data source requires a substantial effort in terms of the data analyst time and execution of computational experiments. Moreover, some specific domains have less labeled data when compared with others (e.g., most Amazon reviews are about electronics). These two issues can be handled by using a cross-domain sentiment

analysis,[10,11] which is a recent transfer learning research trend that aims to reuse sentiment models, previously fitted to some domains (e.g., electronics), to predict the sentiment of texts from other domains (e.g., books).

Some modern Internet platforms commonly ask for user labeled inputs. For instance, Amazon and Tripadvisor promote the writing of reviews under a 5-star rating system. However, sentiment labeled data is much scarce in other social media platforms. For example, Facebook is a popular social network with around 2 billion monthly active users[a] but only a small fraction of Facebook pages allow labeled reviews. Moreover, Twitter is a another relevant social network, with 330 million monthly active users[b], and that is commonly used to spread opinions about a wide range of domains, such as products[12] or stock markets.[13] Yet, Twitter labeled data is much difficult to get, often requiring a laborious manual effort. In addition, there may be differences in the types of texts written in different Web platforms. For example, Twitter restricts the maximum size of text characters, while Facebook does not. As explained in Sect. 2, the majority of cross-domain studies consider a single Web data source (e.g., Amazon reviews). As shown in[14,15], the combination of multiple data sources if often valuable, allowing to augment information quality and reduce bias. Therefore, there is a potential gain and research interest in studying what we term here as "cross-source cross-domain" sentiment classification, in which cross-domain data, from one or more labeled sources, is used to create sentiment analysis models that are later applied to classify non labeled cross-domain texts from other sources. In this work, we propose such approach, under the following main contributions:

(1) We approach a cross-source cross-domain sentiment classification, using distinct data sources and domain for training and testing the models. We adopt cross-domain big data labeled sources from different Web platforms (Amazon and Tripadvisor) to train the sentiment classification models. Then, the learned models are used to predict the sentiment of cross-domain texts from two unlabeled social media sources (Facebook and Twitter). Moreover, we consider datasets written in two distinct languages (English and Italian). The analyzed datasets are made publicly available[c] and thus can be used in future cross-source or cross-domain research studies.
(2) We compare distinct data-driven approaches, in terms of: number of sentiment classes (2 or 3); feature engineering (stemming or part-of-speech tagging for the removal of nouns, pronouns and conjunctions); and balanced training methods (oversampling or undersampling). As for the learning algorithm, we propose a word embedded Convolutional Neural Network, which is compared with another deep learning model (Deep Feedforward Network) and two other classifiers (Support Vector Machines and Naive Bayes).

[a] https://sproutsocial.com/insights/facebook-stats-for-marketers
[b] https://blog.hootsuite.com/twitter-statistics/
[c] https://github.com/paolazola/Cross-source-cross-domain-sentiment-analysis

(3) The proposed cross-source cross-domain approach is compared with a recent sentiment lexicon[16] and a state of the art cross-domain method that is based on a autoencoder structural correspondence learning (AE-SCL) method.[17]

The paper is structured as follows. The next Sect. 2 reports a summary of previous work for sentiment analysis and domain adaptation. In Sect. 3, we describe the data, modeling approaches and evaluation procedure. The Sect. 4 reports a brief description of the models used and evaluation metrics. Then, we describe the conducted experiments and obtained results (Sect. 5). Finally, in Sect. 6, we discuss the main conclusions and future work.

2. Related works

The related works are summarized in Table 1, which assumes a chronological order. Each study is characterized in terms of the language used, if it is a cross-domain or cross-source approach, data source used and size, type of text preprocessing (**L** – lemmatization, **S** – stemming, **P** – part-of-speech tagging), sentiment analysis method and number of sentiment classes adopted.

Sentiment analysis studies typically focuses on one specific domain at a time, such as hotels,[5] movies[18] or stock markets.[6,19] Cross-domain sentiment analysis,[10,11,20,21] also known as domain adaptation sentiment analysis, is a recent form of transfer learning.[22] The goal is to learn a classification model from some domains (e.g., electronics, books) and then reuse the models to classify other domain texts (e.g., music reviews). This alleviates the need to collect and curate data for each new domain, and it is particularly relevant for accessing the sentiment of new product opinions for which scarce data are available.[23] The **Cross-domain** column of Table 1 signals the relevant works in this field.

The rationale for adopting a cross-domain sentiment analysis also translates into cross-source sentiment analysis. Developing an accurate model for one source is costly and several social media sources, such as Facebook or Twitter, contain a huge amount of unlabeled reviews. However, most cross-domain sentiment analysis works assume a single data source, as shown by the column **Cross-source** of Table 1. Often, this source consists in the popular Amazon platform,[22] with the analysis of distinct reviews of sold products.[10,32,38,11,51] Within our knowledge, there are only two cross-domain works that use distinct sources. Aue and Gamon[27] considered only traditional Web sites. More recently, Ziser and Reichart[17] used a single source, the Blitzer's Amazon dataset with reviews of products (e.g., books, electronics) to train binary sentiment classification models that were then tested on blog texts (from 16 nondisclosed domains).

There are two main sentiment classification methods: lexicon and machine learning based. A lexicon is a special dictionary in which words are assigned to sentiment scores.[39,46] The main advantage is that, once a lexicon is built, a fast unsupervised sentiment classification is achieved, by summing the overall word scores. Thus, there is no need for labeled data. However, lexicons tend to produce lower performances

Table 1. Summary of related work.

Study	Lang[a]	Cross-domain	Cross-source	Source[b]	Data Size[c]	L	S	P	Method[d]	Sentiment Classes
Pang et al.[24]	ENG			WS	2K			X	N-gram+NB, N-gram+SVM, N-gram+ME	2
Dave et al.[25]	ENG	X		WS			X	X	N-gram+NB, N-gram+SVM	2
Salvetti et al.[26]	ENG			WS	27K		X	X	L+NB, L+MM	2
Aue and Gamon[27]	ENG	X	X	WS	2K,5K,12K				N-grams+NB, N-grams +SVM	2
Cui et al.[28]				WS	200K				N-grams+PA, N-grams+LM, N-grams+Winnow	2
Ng et al.[29]	ENG			WS	4K			X	L+SVM	2
Blitzer et al.[10]	ENG	X		WS	8K				SCL, SCL-MI	2
Go et al.[12]				SM	1.6M			X	N-grams+NB, N-grams+ME, N-grams+SVM	2
Ohana and Tierney[30]	ENG			WS	2K				SW+SVM	2
Dang et al.[31]	ENG			WS	2K,8K			X	N-grams +SVM	2
Pan et al.[32]	ENG	X		WS	20K				SFA	2
Glorot et al.[33]	ENG	X		WS	340K				SDA	2
Shi and Li[5]	CHI			WS	4K				Fr+SVM, Tf-Idf+SVM	2
Jo and Oh[34]	ENG			WS	24K, 27K				S-LDA, ASUM	2
Yoshida et al. [35]	ENG	X		WS	10K			X	GmWdDinD	3
Gräbner et al.[36]	ENG			WS	80K			X	LDB	3 5
Neri et al. [37]	ITA			SM	1K				SKMs	-
Bolleaga et al.[38]	ENG	X		WS	8K,68K	X		X	FE+L1LR	2
Gosh and Kar[39]				WS	300			X	SLX	2
Ortigiosa et al.[40]	SPA			SM	3K			X	L,NB,J48,SVM	2
Dos Santos and Gatti[41]	ENG			WS,SM	12K,80K				We+CNN, Ce+CNN	2
Mensil et al.[18]	ENG			WS	50K				N-GM,NB, SVM,RNN	2
Pouransari and Ghili[42]	ENG			WS	60K				BOW/W2V+LR, BOW/W2V+RF, BOW/W2V+SVM RNN	2 5
Tang et al.[43]				WS	335K,5K				UWCVMC	4,5
Wallin[11]	ENG	X		WS	636K			X	LR+BOW	2,5
Fang and Zhan[44]	ENG	X		WS	5.1M			X	SVM,NB,RF	2
Lai et al.[45]	ENG,CHI			D	230K,20K				We+RNN, We +RCNN	4 to 20
Ganin et al.[20]	ENG	X		WS	8K				DANN	2
Kumar et al.[46]				WS					NB,LR,SW	2
Tripathy et al.[47]	ENG			WS	50K				N-grams+NB, N-grams+ ME, N-gram+SVM, N-gram+SGD	2
Conneau et al.[48]	ENG,CHI			WS	11M				Ce+VDCNN	2 to 14
Dragoni et al.[21]	ENG	X		WS	1M				We+NN	2
Radford et al.[49]				WS	82M				LSTM	2
Ziser and Reichart[17]	ENG	X	X	WS,B	78K,40K				AE-SCL	2
Dragoni and Petrucci[50]	ENG	X		WS	1M				FM+L	2
Zhang et al. [51]	ENG	X		WS	56K				IATN	2
This work	ENG,ITA	X	X	WS,SM	1.3M		X	X	We+NB, We+SVM, We+MLP, We+CNN	2,3

[a] **Language** – ENG (English), CHI (Chinese), SPA (Spanish), ITA (Italian).

[b] **Data source type** – B: blogs, D: documents (e.g., Stanford sentiment treebank, News database), SM: social media (Facebook and Twitter), WS: Web sites (Amazon, Citysearch, Electronics reviews, MyMovies and other movies reviews, Tripadvisor, Yelp).

[c] **Number of instances** – K: thousand, M: million.

[d] **Sentiment Analysis method** – AE-SCL: autoencoder structural correspondence learning, ASUM: aspect and sentiment unification model, BOW: bag of words, Ce: character embedding, CNN: convolutional neural network, DANN: domain-adversarial neural network, FE: feature extraction, FM: fuzzy model, Fr: frequency, GmWdDinD: generative Bayesian model of word with domain dependence or domain independence, IATN: interactive attention transfer network, J48: decision tree, L: lexicon information, L1LR: L1 regularized logistic regression, LDB: lexicon database, LM: language modeling, LR: logistic regression, LSTM: long-short term memory neural network, ME: maximum-entropy, MM: Markov model, MI: mutual information, NB: naive Bayes, PA: passive-aggressive algorithm, S-LDA: sentence latent Dirichlet allocation, SDA: stacked denoising auto-encoders, SFA: spectral feature alignment, SGD: stochastic gradient descent, SKMs: sentiment knowledge mining system, SLX: sentiment lexicon database, SVM: support vector machine, SCL: structural correspondence learning, SW: SentiWordNet [52], RCNN: recurrent convolutional neural network, RF: random forest, RNN: recursive neural network, UWCVMC: user word composition vector model, VD-CNN: very deep convolutional neural network, W2V: word to vec, We: word embedding.

when compared with supervised machine learning approaches.[53] Thus, machine learning is widely used for sentiment analysis.[24,26,42]

Sentiment classification studies initially explored simpler feature engineering (e.g., N-grams or Bag-of-Words) and machine learning algorithms (e.g., Naive Bayes, Support Vector Machines). After 2014, recent text classification advances, such as word embedding and deep learning,[54] were naturally incorporated into sentiment analysis works.[40,45,48] Focusing on transfer learning problems, Ganin et al.[20] proposed a domain adversarial neural network where the hyperparameter are determined by a reverse cross-validation approach. Recently, Zhang et al.[51] analyzed the jointly impact of sentence network attention and aspect network attention in the interactive attention transfer network (IATN).

The novelty of our work is highlighted in the last row of Table 1. We address a novel cross-source cross-domain sentiment analysis, in which Web sources that contain easy labeled reviews (Amazon and Tripadvisor) are used to fit a sentiment analysis model, which is then reused to predict the sentiment of two typically unlabeled social media platforms (Facebook and Twitter). Moreover, we propose a recent deep learning method, which is based on a word embedded Convolutional Neural Network and that is compared with three machine learning methods (a modern Deep Feedforward Network, a Support Vector Machine and Naive Bayes), a recent sentiment lexicon and state of the art cross-domain method. We also explore stemming or part-of-speech tagging, to reduce the word sparsity, and oversampling or undersampling methods, to deal with the unbalanced sentiment datasets. Finally, to enrich the experimental comparison analysis, we consider two languages (English and Italian) and two sentiment classification tasks ("negative","positive" and "negative", "neutral", "positive").

3. Materials and methods

3.1. *Sentiment analysis data*

In this work, we consider texts written in two languages, English and Italian. We also consider two sentiment output label sets, with 2 ("negative", "positive") and 3 ("negative", "neutral", "positive") classes. The datasets analyzed are made freely available at `https://github.com/paolazola/Cross-source-cross-domain-sentiment-analysis`. The texts come from four major sources of data:

(1) **Amazon**: we gathered the data directly from the *Amazon.com* Web site. The reviews regard different products, such as electronic devices, kitchen objects, clothes and house accessories. For the polarity classification, we consider two 5-star rating value transformations: $\{1, 2, 3\} \rightarrow$ "negative" and $\{4, 5\} \rightarrow$ "positive"; and $\{1, 2\} \rightarrow$ "negative", $3 \rightarrow$ "neutral" and $\{4, 5\} \rightarrow$ "positive". The data was collected from January to February 2018 and it includes 282,781 English and 161,443 Italian reviews.

(2) **Tripadvisor**: we collected reviews directly from the *Tripadvisor.com* Web site. The 5-star reviews are related with restaurants, hotels, monuments and interest points, cities and activities. The same Amazon label transform was adopted to create the 2 and 3 class outputs. The data was collected from January to February 2018 and the dataset is composed by 519,735 randomly sample reviews for the English language and 324,376 for the Italian one.
(3) **Facebook**: the data was retrieved directly from the *Facebook.com* social network. We considered only comments from specific public pages having a 5-start rating system, such that we could compute the same 2 and 3 class sentiment labels. The sampled reviews performed from January to February 2018 are about several topics, namely universities, events, famous people, locals, parties, shops and cities (total of 5,792 English and 1,077 Italian texts).
(4) **Twitter**: to reduce the manual labeling effort, we selected preferentially publicly labeled data. For English, we used the Sentiment140 labeled test set developed by Stanford University,[12] which has 497 reviews about companies, events, locations, movies, persons, etc. The data was collected in 2009. The data are structured in three label classification: "negative", "neutral" and "positive". As for Italian, we adopted the SENTIPOLC (SENTIment POLarity Classification) labeled dataset that was organized within Evalita 2014.[55] It includes a set of 4,513 twitter status IDs, with annotations concerning polarity classification and irony detection about politics, news and famous people. Since the dataset only includes two classes ("positive" and "negative") and two authors are Italians, we performed an extra manually 3 level classification ("negative", "neutral" and "positive") of 937 tweets, collected at April 2018 and regarding Italian television shows and other more general topics. To get binary versions of Sentiment140 and our Italian manually labeled data, we merge the original negative and neutral classes into the "negative" label.

Fig. 1 plots the data source percentage rating/sentiment class distributions. In all cases, the sentiment classes are unbalanced. Some sources (Amazon, Tripadvisor) present the common J-shaped distribution, with a much lesser number of negative reviews.[11] As reported in Li et al.,[56] this might be due to the following reasons: people tend to publish opinions about popular products, which are more likely positive; and there may exist many flaunt positive reviews from the product companies and dealers.

3.2. *Cross-source methodology*

We adopt four learning algorithms, as detailed in Sect. 4: Naive Bayes (NB), Support Vector Machine (SVM), Deep Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN). Also, the text reviews are firstly preprocessed in order to remove numbers, capitalized letters, whitespaces, punctuation, stopwords and urls. After this preprocessing, we further apply stemming (Stem) or part-of-speech (POS) tagging (Sect. 3.3).

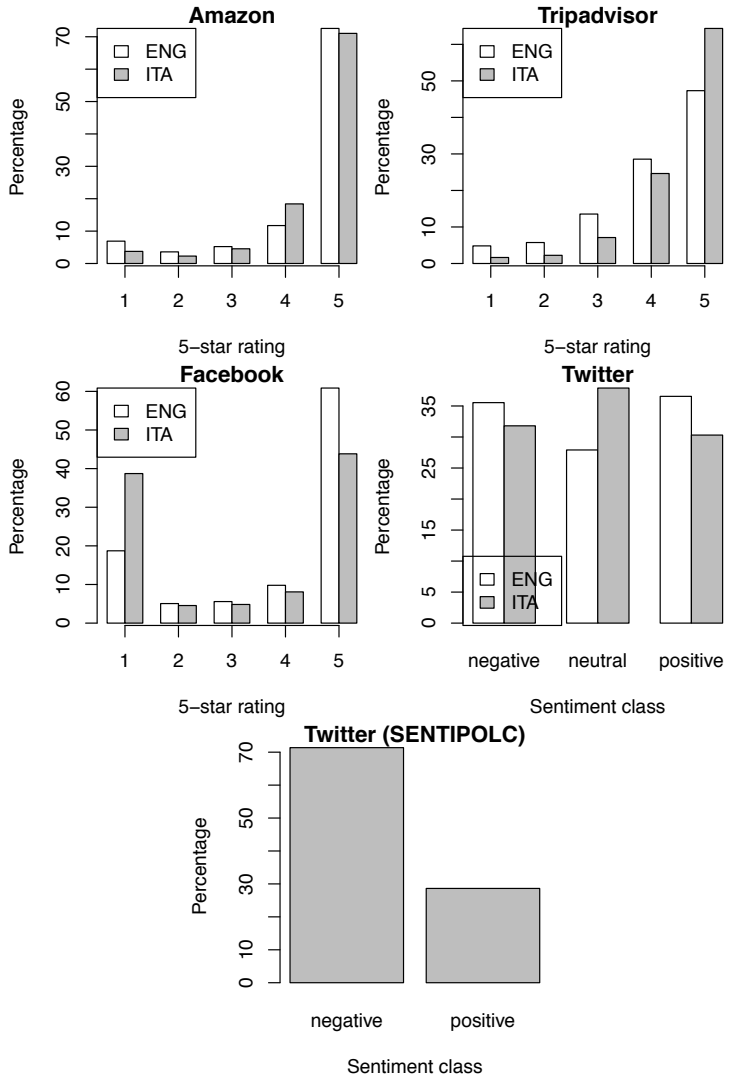

Fig. 1. Sentiment distribution values for the distinct data sources.

In this work, we assume a research methodology that contains three main steps (Fig. 2). Let $A \rightarrow B$ denote a sentiment classification model that was trained on A and tested on B, where A and B denote cross-domain corpus. In step 1, we execute single source experiments ($A = B$, $A \in$ {Amazon,Tripadvisor}). In the step 1 of Fig.2, the dashed boxes and arrows (e.g., - - →) denote the path followed by the Amazon dataset, while the dotted box and arrows (e.g., ····›) represent the Tripadvisor analysis path. The goal is to perform initial experiments to gather insights about balanced training (oversampling or undersampling) and hyperparameter (e.g., number

of neural network hidden nodes) selection. This selection is based on a grid search, which uses a range of grid values for several hyperparameters (Sect. 4). The best hyperparameter values, in terms of classification performance on the test data, are then fixed for steps 2 and 3. We note that step 1 test data is from the same training data source, while in step 3, we perform the target cross-source tests using external source data (Facebook and Twitter) that was never used in the modeling decisions defined in steps 1 and 2. Moreover, step 1 also provides the estimation of single source test classification performances, which can be used to evaluate the quality of the proposed transfer learning sentiment approach. In effect, any cross-source test classification measure (of steps 2 or 3) close to the single source performance (of step 1) would indicate a high quality sentiment analysis.

Next, in step 2 we conduct Amazon→Tripadvisor and Tripadvisor→Amazon cross-source experiments, aiming to select the best text processing and machine learning method. The solid arrows (⟶) in step 2 of Fig.2 represent the same paths that were followed by the Amazon→Tripadvisor and Tripadvisor→Amazon experiments. The learning models use fixed balanced training and hyperparameter values, as set in step 1. There are two main text processing options (Stem or POS) and four learning algorithms (NB, SVM, MLP and CNN).

Finally, in step 3 we use the labeled sentiment sources for training (input domain) and perform the testing on both non labeled sources (target domain). In step 3 of Fig.2, the dashed arrows (⇢) represent the path when the target test domain is Twitter, while the dotted arrows (⇢) refer to the Facebook target domain. A fixed text processing and machine learning model (set in step 2) is used. Only one training model is obtained for each language, allowing to obtain the final cross-source results: Amazon ∪ Tripadvisor→Facebook and Amazon ∪ Tripadvisor→Twitter.

In steps 1 and 2, we use the three main features: *date*, *review text* and *sentiment class*. The *date* is used to chronologically order the messages, such that a rolling window evaluation scheme can be applied.[13] The rolling window is a realistic and robust evaluation method that considers several training and test iterations through time. First, the texts are ordered by the *date* field and split into k distinct partitions of equal size. For a particular i iteration, the i-th partition is selected and further split into training (oldest data) and test sets (newest data). The training data are then balanced using undersampling or oversampling.[57] The former method decreases the dataset size by randomly sampling the majority examples in order to equal number of minority ones. The latter expands the data by sampling with repetition the minority examples in order to equal the number of majority ones. Next, the machine learning model is fit and evaluated using the test set, which keeps the original sentiment class distribution.

In the step 3, since the *date* feature is not available (at both Sentiment140 and SENTIPOLC), we execute a k-fold cross validation[21], which works as follows. First, the full training data is set by selecting all Amazon texts and a sample of Tripadvisor reviews, such that each source is similarly represented. We note that

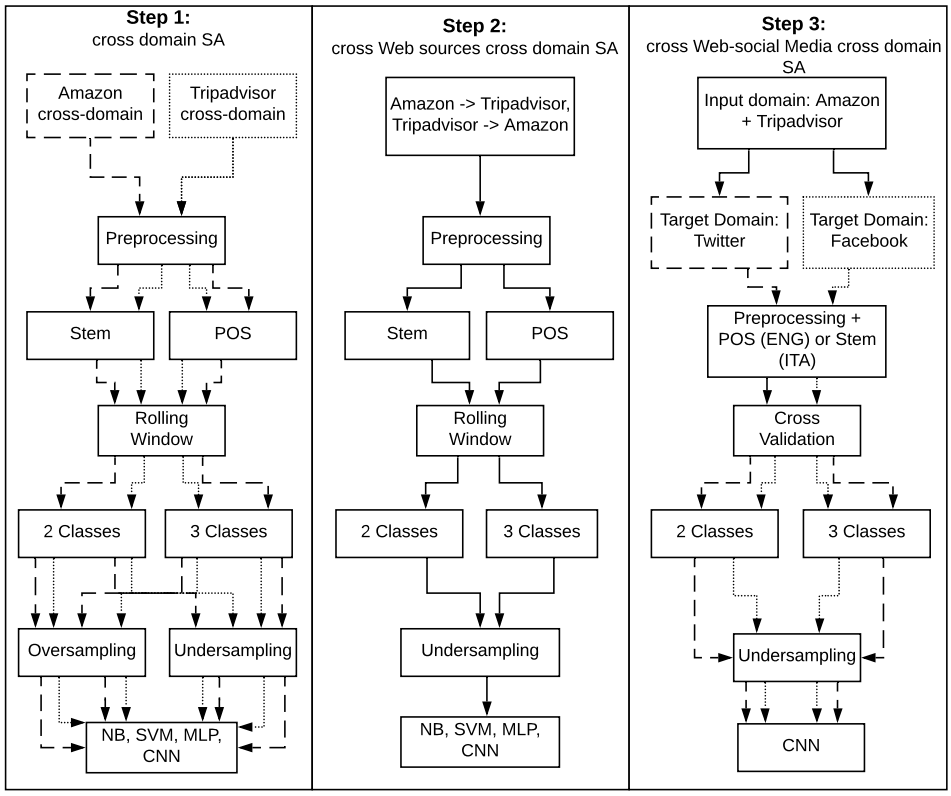

Fig. 2. Adopted three step methodology for the cross-source cross-domain sentiment analysis (SA).

Tripadvisor is in general twice the size of Amazon data, thus a 50% sampling is often adopted. Then, the merged training data is randomly divided into k-folds. For a particular i iteration, all data samples except the ones belonging to the i-th fold are used to train the sentiment model. The balancing method is applied only to the training data. After fitting the model, it is tested two times, using the whole Facebook and Twitter messages as the test sets and leading to two sets of classification performance measures.

3.3. *Stem and Part-of-Speech Text preprocessing*

To reduce the word embedding size and computational effort, we test two alternative Natural Language Processing (NLP) techniques to compress text: stemming (Stem) and Part-of-Speech (POS) tag removal. As an example, Table 2 presents the text sentence reduction that is achieved when using Stem or POS preprocessing with the two largest data sources (Amazon and Tripadvisor), showing that in certain cases a high compression rate is achieved (e.g., around 70% for English Tripadvisor data when using POS).

Table 2. Comparison of text sentence size before and after preprocessing for Amazon and Tripadvisor sources (values denote the average number of words per sentence).

Language	**Source**	Original	Stem	POS
English	Amazon	35.60	17.60	10.70
	Tripadvisor	125.06	61.03	34.97
Italian	Amazon	42.20	26.08	13.50
	Tripadvisor	70.94	41.03	21.14

In the literature the *stemming* procedure refers to the process of stripping off affixes (both suffix and prefix) from the word and maintaining only the root of the word.[58] Similar to the stemming is the *lemmatization*, where each word is re conducted to its lemma or lexeme. The benefit of stemming and lemmatization is in data sparseness reduction even if for some languages, such as English, the dictionary is characterized by a diminish morphology and therefore the stemming procedure might not show a considerable improvement in the performance. However for other languages, such as Latin ones (Italian in our research), the vocabulary is very rich of morphology and the stemming might help in reducing the number of features in the text, increasing the classification performance. To implement the data stem we used the *Snowball Stemmer*[59] available on *NLTK* module in Python. Two stem illustrative examples are: "affordable" → "afford" (English) and "bellissima" → "bell" (Italian).

Part-of-Speech (POS) tagging is a technique used to assign the appropriate parts of speech tag to each word in a text (it is also known as word classes, morphological classes or lexical tags).[60]

We use the POS tagging in order to exclude all *nouns*, *pronouns* and *conjunctions* from the text in order to remove potential "domain" terms from the reviews and thus maintain more useful words for the cross-domain sentiment extraction, such as *adjectives* and *adverbs*. It has been demonstrated that *adjectives* are good indicator for opinion classification.[61] Also, some literature works in sentiment classification did not consider *nouns*.[31,40] The POS tagging was performed by using the *RDRPOSTagger*[62] library developed in the R software. The *RDRPOSTagger* supports both English and Italian languages and it is more fast in tagging when compared with other POS taggers, such as *Treetagger*[63] available in Python. Two POS tag removal demonstrative examples are: "really worthy the money" → "really worthy" (English) and "Città meravigliosa in tutto" → "meravigliosa" (Italian).

3.4. *Word Embedding*

Word Embedding is a distributed representation in which each word is represented as a vector in a continuous space and similar words are mapped to nearby points. The Vector Space Models (VSM) has been applied to text data since the 1960s and they assumed a greater interest in recent years. Among VSM it is possible to distinguish two main approaches:[64]

(1) count based method: it is based on word co-occurence in order to build dense vectors. An example of this approach is the Latent Semantic Allocation (LSA);
(2) predictive method: predict a word based on its neighbours. N-grams, Neural Probabilistic Language Models (NNLM)[65] and the Word2Vec[66] model are some examples of this approach.

The state of the art in VSM is associated to Mikolov et al.,[66] which proposed a Feedforward Neural Network with an input, projection and output layer under two versions: Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram Model (Skip-gram). In the CBOW model, the word w_t is predicted by considering the nearby words (context), while in the Skip-gram it tries to maximize the classification of a word based on another word in the same sentence. In this last case, the word w_t is used to classify the context.

In this paper, we performed a word level embedding by using the *Keras* library tool based on a Feedforward Neural Network. The input is a integer matrix called **I**, where each word is mapped to its absolute frequency given the dataset words' distribution. The matrix has n rows which denotes the different reviews in the dataset and c columns. Each review has a variable number of words and, in order to reduce the sparseness in the matrix **I** and ensure the same dimension to each review, we defined the column number c as follow:

$$c = \left\lfloor \frac{\sum_{i=1}^{n} length(r_i)}{n} \right\rceil \quad (1)$$

where $\lfloor \rceil$ represents the round function and $length(r_i)$ denotes the number of words in the i-th review. The matrix **I** is then passed to the embedding layer. The embedding layer maps a two-dimensional matrix in a sequence of e matrices. In this paper the number of matrices are e =128. The embedded matrix **O** is then composed by n rows (for each review) and c columns. Each element $\mathbf{O}(i, j \times 128)$ represents the numerical depiction (real number) of the $n-^{th}$ sentence.

A small demonstration example is provided, which considers three messages:

(1) "sicly beaches were fantastic and food amazing. What a super happy holiday";
(2) "The hotel is good, receptionist helpful in giving advises and the swimmingpool was wonderful"; and
(3) "A new car has been promoted by the company. It is fantastic, the best on the market with many new accessories.".

After preprocessing (e.g., with removal of punctuation, stop words and POS nouns), the sentences become:

(1) "fantastic amazing super happy";
(2) "good helpful wonderful"; and
(3) "new promoted fantastic best many new".

The demonstration assumes text data with the following term frequency values: {*good*=3245, *helpful*=1700, *new*=1200, *many*=2400, *great*=3000, *fantastic*=2500,

free=1400, *amazing*=1000, *super*=600, *happy*=1100, *wonderfull*=300, *best*=734, *promoted*=5}. Thus, the initial **I** integer matrix becomes:

$$\begin{bmatrix} 2500 & 1000 & 600 & 1100 & & \\ 3245 & 1700 & 300 & & & \\ 1200 & 5 & 2500 & 734 & 2400 & 1200 \end{bmatrix}$$

In this example, the average size is $c = 4$. Sentences with a length greater than 4 are truncated and sentences with less than 4 elements are padded with zeros[67], resulting in the final **I** matrix:

$$\begin{bmatrix} 2500 & 1000 & 600 & 1100 \\ 3245 & 1700 & 300 & 0 \\ 1200 & 5 & 2500 & 734 \end{bmatrix}$$

Since now the matrix **I** is composed by sentences with the same number of columns (tokens), it is possible to compute the word embedding via a Feedforward Neural Network, obtaining for each token a real numbers representation. In this example, it is denoted with a sequence of 128 real values. Thus, for each sentence we concatenate the single word embedding (1×128) obtaining a sentence embedding equal to ($1 \times (4 \times 128)$). Considering a flatted representation, the matrix **O** is then composed by 3 rows and 512 columns denoting the concatenated word embedding.[41]

4. Models

The models described here were used for both binary and multiclass classification. As reported in Sect. 3.4, the input of all the machine learning algorithms is the word embedding matrix **O**, with n rows corresponding to the n reviews in the data set, while the output is related with the rating vector V (with 2 or 3 classes). Three of the learning models (SVM, MLP and CNN) have hyperparameters that were tuned using a grid search. Using only single source data (step 1 of Sect. 3.2), a rolling window validation was executed, providing several training and test iterations thought time. For each learning model (SVM, MLP or CNN), we select the hyperparameter value that resulted in the best average classification performance (Area Under the Curve metric, see Sect. 4.5) on the rolling window single source test data. The details of the selected hyperparameters, fixed in step 1 and used in steps 2 and 3 of Sect. 3.2, are presented in Sect. 5.

4.1. *Naive Bayes*

The label l^* can be assigned to a review r using the formulation: $l^* = \arg\max_l P(l|r)$. The Naive Bayes (NB) method is based on the Bayes' rule and on the strong hypothesis that there is independence between every pair of input features.[68] The probability of label l based on r is computed as:

$$P(l|r) = \frac{P(r|l) * P(l)}{P(r)} \tag{2}$$

And, in case of binary classification (0,1), the label for the r-th review is based by on:

$$\frac{P(l_0|r)}{P(l_1|r)} = \frac{P(r|l_0) * P(l_0)}{P(r|l_1) * P(l_1)} \tag{3}$$

4.2. *Support Vector Machine*

Support Vector Machines (SVM) are widely used in text classification,[69] often outperforming the NB algorithm.[70] It can be used for both classification and regression tasks and the model is based on a maximized margin criterion.[71] For the binary classification, the SVM algorithm can compute the best separating hyperplane in a feature space (after the kernel transformation). Given $l_j \in 1, -1$, corresponding to negative (-1) or positive (1) classes, the solution of the SVM model for the review r_j is given by:

$$\overrightarrow{w} = \sum_j \alpha_j l_j \overrightarrow{r_j}, \alpha_j \geq 0; \tag{4}$$

where the α_j are obtained by solving a dual optimization problem. The support vectors are the $\overrightarrow{r_j}$ values such that $\alpha_j > 0$.[72] In this work, we selected the popular Gaussian kernel, also termed Radial Basis Function (RBF), which presents less hyperparameters when compared with other polynomial kernels. The model contains just two hyperparameters: the γ Gaussian kernel parameter and C, a penality parameter that indicates the sensibility of the model to misclassification. To set these hyperparameters, a grid-search was adopted in step 1 using the values $\gamma \in \{0.01, 0.1, 1.0\}, C \in \{0.1, 1.0, 10.0\}$. The best values were selected using test data (from step 1, using the same data source) and are reported in Sect. 3.1. The SVM model was implemented using the *sklearn* module in Python, which is based on the popular *libsvm* library.

4.3. *Multilayer Perceptron*

The adopted Multilayer Perceptron (MLP) model corresponds to a modern deep learning variant of the original feedforward neural network,[73] which includes three hidden layers (with H_1, H_2 and H_3 hidden nodes), usage of Dropout regularization, Adagrad gradient training and ReLU activation functions on the hidden nodes:[54]

$$\begin{aligned} \text{ReLU } f(z_i) &= max(0, z_i) \\ \text{Softmax } P(l|r) = f(z_l) &= \frac{\exp(z_l)}{\sum_{k \in K} \exp(z_k)} \end{aligned} \tag{5}$$

where z_i is the weighted sum of the i-th neural unit a, f is the activation function and K is the set of output nodes. ReLU is a popular activation function often used in deep learning experiments due to its good convergence property and faster training of deep layers.[74] The Softmax function allows the outputs to be interpreted

as class probabilities (where $\sum_{k \in K} f(z_k) = 1$). The weights of the MLP are typically estimated by using a gradient descent algorithm.[75] To fit the weights, we used the Adagrad gradient descent variant, which automatically adapts the learning rate η, performing smaller updates for more frequently used weights and larger updates for infrequent weights. This algorithm is particularly suitable for sparse data tasks, such as text classification, which often contains very frequent and infrequent words.[76,42] To prevent overfitting, we use a Dropout value of 20% as the regularization method. Dropout randomly ignores neural connections during training and this significantly reduces overfitting, often obtaining major improvements when compared with other regularization methods.[77] The grid search ranges for the MLP hyperparmeters were set to: $H_1 \in \{50, 60, ..., 90, 100, 150, 200, 250\}$, $H_2 \in \{10, 20, 30, ..., 50, 100, 125\}$ and $H_3 \in \{5, 10, 15, ..., 25\}$. The grid search was restricted to present a decreasing order in the number of hidden units per layer, such that $H_1 > H_2 > H_3$. Similarly to Prusa et al.[78] and Mahendhiran et al.[73] that used a fixed number of epochs (e.g., 100) for each experiment, this hyperparameter value was set to 100.

4.4. *Convolutional Neural Network*

Convolutional Neural Network (CNN) is a class of deep feedforward neural networks that exploits local connectivity patterns designed to process data that comes in the form of multiple arrays.[21] CNNs have obtained competitive state-of-the-art results in several classification tasks, including image classification and text classification.[79] The design of a CNN is composed by an input layer, M convolutional layers, H MLP hidden layers and an output layer. When compared with MLP, the main difference is the presence of the initial convolutional layers, each composed by a convolutional layer and a pooling layer.

The contribution of the convolutional layer in the CNN regards the convolution operation itself, which is a kind of sliding window function that performs a matrix product between the input and a filter matrix or vector, called also kernel or feature detector, and that is smaller than the input matrix size. This convolution operation leads to a sparser interaction in CNN, thus fewer parameters are estimated, improving the computational efficiency. Another feature that distinguishes CNN from the other neural networks is the parameter sharing, which refers to the use of the same parameter for more than one function in a model, since each member of the kernel is used at every position of the input. By adopting this parameter sharing, the layers also assume a equivariance in translation property.[54] Another important element in a CNN is the pooling layer which further modifies the convolutional layer output, replacing the values in some location by the summary statistics of the nearby outputs. Two famous pooling functions are the max polling and the average pooling. For example, if the convolutional vector output c is divided into v rectangular areas, each composed by $e = \frac{c}{v}$ elements, then the pooling output is a vector of length v such that each element corresponds to the maximum or average of the e-th rectangular. In this paper, we adopt a CNN with a convolution

layer with its max pooling layer followed by another convolutional layer and an average pooling layer. Then, as described in Sect. 4.3, the same MLP procedure is added. The CNN hyperparameters were searched using the ranges: first filter $\in \{1, 3, 6, 12, 24, 32, 64, 96, 128\}$, first kernel $\in \{1, 5, 9, 14, 20\}$ max pooling $\in \{1, 2\}$, second filter $\in \{2, 6, 12, 24, 36, 48, 64, 128, 184, 256\}$ and second kernel $\in \{2, 3, ..., 6\}$.

4.5. *Evaluation*

The classification performance is based on three metrics: Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve, the macro-averaging F1-score and Accuracy.

Each classifier outputs a probability for a particular class label (l) and review (r): $P(l|r)$. For the decision parameter $D \in [0, 1]$, it can be assumed that class l is positive if $P(l|r) > D$. The ROC curve plots all the trade-offs (distinct D values) between correctly predicting the positive or negative l class values, showing one minus the specificity (x-axis) versus the sensitivity (y-axis). ROC curves can be applied to unbalanced tasks and without knowing *a priori* the false positive and false negative costs.[80] To obtain a single metric, the $AUC = \int_0^1 ROCdD$ is often used. A random classifier presents an AUC of 0.5, while the ideal classifier should present an AUC of 1.0. For the multiclass models, we compute the global AUC, which weights each class AUC according to the most frequent classes.

In classification, it is often assumed that the predicted class label l is the one with the highest probability. The confusion matrix maps the predicted versus the desired labels, allowing to compute several metrics, such as Accuracy, Precision, Recall and F1-score:[81]

$$\begin{array}{ll} \text{Accuracy} & = \frac{TP+TN}{TP+FP+TN+FN} \\ \text{Precision}_l & = \frac{TP_l}{TP_l+FP_l} \\ \text{Recall}_l & = \frac{TP_l}{TP_l+FN_l} \\ \text{F1-score}_l & = 2 \times \frac{\text{Precision}_l \times \text{Recall}_l}{\text{Precision}_l + \text{Recall}_l} \end{array} \quad (6)$$

where TP_l, FP_l, FN_l denote the number of true positives, false positives and false negatives for class l. To combine the F1-score multiclass results into a single measure, we use the macro-averaging F1-score, which first computes the F1-score$_l$ for all l labels and then averages the overall result. The classification metrics were implemented using the *rminer* R package.[82]

To evaluate the overall performance of the sentiment models, we use the same procedure adopted by Oliveira et al.:[6] first, we compute the metric (AUC, macro-averaging F1-score or accuracy) for each iteration of the rolling window (steps 1 and 2) or k-fold cross validation (step 3); then, we average the k distinct results. Statistical significance is obtained by applying the non-parametric Wilcoxon signed rank test.[83] The model selection decision (e.g., best hyperparameter value, best balancing method) is mainly based on AUC values as the single metric. In fact, on one hand the macro-averaging F1-score corresponds to just one specificity versus

sensitivity trade-off, while the AUC is computed over all possible D trade-offs. On the other hand, accuracy is sensitive to unbalance data as our test sets and it might be misleading to performance evaluation. However, accuracy is a common metric often used in sentiment classification, thus, we deemed appropriate to include it in the results.

5. Results and Discussion

We conducted the computational experiments using code written in the Python language and executed using two different multi-core servers (e.g., Intel Xeon E5 at 2.30 GHz). In both steps 1 and 2, we used k=20 iterations of the rolling window evaluation scheme. In step 3, we used the same $k = 5$-fold cross validation employed in the recent work of Dragoni and Petrucci.[50]

5.1. *Step 1 results*

In each rolling window iteration of step1, the reviews were sorted, such that 60% of the oldest data was used for training and 40% for testing. Also due to computational requirements, we conducted the step 1 hyperparameter grid selection only for the undersampling and binary classification case. Hyperparameters are then fixed with the best searched values used for the oversampling and multiclass models. The selected values are shown in Table 3.

Table 3. List of selected hyperparameters.

	Stem				POS			
	English		Italian		English		Italian	
	Tripadvisor	Amazon	Tripadvisor	Amazon	Tripadvisor	Amazon	Tripadvisor	Amazon
SVM:								
C	0.1	1	0.1	1	0.1	1	0.1	0.1
γ	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
MLP:								
H_1	200	100	200	90	100	100	200	200
H_2	125	30	125	50	50	30	125	125
H_3	25	10	25	20	15	10	25	25
CNN:								
first filter	12	32	12	6	12	32	12	24
first kernel	9	5	9	5	9	5	9	9
max pooling	2	1	2	2	2	1	2	1
second filter	24	64	24	12	24	64	24	48
second kernel	6	2	6	3	3	2	4	2
H_1	200	100	200	90	100	100	200	200
H_2	125	30	125	50	50	30	125	125
H_3	25	10	25	20	15	10	25	25

The sentiment classification results for step 1 are presented in Table 4, which shows interesting AUC results for Tripadvisor and Amazon data sources, in both languages. The best AUC values were obtained for the English Tripadvisor data: 81% AUC for binary task and 78% for the three sentiment classification.

Table 4. AUC (macro-average F1-score, accuracy) results for sentiment classification in step 1 (best AUC values per dataset and same number of classes are in **bold**).

Balance	Class	Model	English				Italian			
			Stem		POS		Stem		POS	
			Amazon	Tripadvisor	Amazon	Tripadvisor	Amazon	Tripadvisor	Amazon	Tripadvisor
Under	2	NB	0.52 (0.48, 0.59)	0.53 (0.49, 0.54)	0.65 (0.54, 0.64)	0.61 (0.56, 0.62)	0.57 (0.48, 0.61)	0.59 (0.46, 0.58)	0.59 (0.47, 0.59)	0.59 (0.46, 0.58)
		SVM	**0.54** (0.45, 0.58)	0.52 (0.46, 0.67)	0.67 (0.54, 0.63)	0.64 (0.55, 0.61)	0.55 (0.46, 0.62)	0.53 (0.47, 0.76)	0.42 (0.39, 0.71)	0.50 (0.47, 0.87)
		MLP	0.50 (0.40, 0.53)	0.51 (0.47, 0.56)	0.64 (0.59, 0.75)	0.66 (0.61, 0.68)	0.62 (0.49, 0.62)	0.55 (0.49, 0.65)	0.63 (0.50, 0.64)	0.50 (0.25, 0.41)
		CNN	0.51 (0.44, 0.54)	0.56 (0.50, 0.55)	**0.74*** (0.64, 0.74)	**0.81** (0.72, 0.76)	0.70 (0.53, 0.65)	**0.75*** (0.57, 0.70)	**0.73** (0.56, 0.69)	0.61 (0.44, 0.61)
	3	NB	0.52 (0.28, 0.47)	0.52 (0.28, 0.44)	0.63 (0.35, 0.61)	0.61 (0.34, 0.54)	0.57 (0.30, 0.56)	0.61 (0.32, 0.49)	0.59 (0.30, 0.55)	0.56 (0.27, 0.51)
		SVM	0.51 (0.20, 0.35)	0.50 (0.18, 0.29)	0.64 (0.32, 0.57)	0.62 (0.31, 0.51)	0.46 (0.16, 0.28)	0.46 (0.23, 0.44)	0.41 (0.12, 0.31)	0.43 (0.14, 0.37)
		MLP	0.52 (0.20, 0.32)	0.52 (0.30, 0.44)	0.55 (0.31, 0.51)	0.65 (0.40, 0.53)	0.60 (0.31, 0.48)	0.60 (0.33, 0.46)	0.61 (0.30, 0.48)	0.50 (0.10, 0.26)
		CNN	0.52 (0.26, 0.46)	0.55 (0.29, 0.36)	**0.69** (0.38, 0.58)	**0.78*** (0.50, 0.62)	0.66 (0.32, 0.47)	**0.74*** (0.40, 0.52)	**0.70** (0.33, 0.53)	0.55 (0.16, 0.33)
Over	2	NB	0.53 (0.49, 0.62)	0.53 (0.50, 0.55)	0.65 (0.54, 0.64)	0.61 (0.54, 0.60)	0.57 (0.47, 0.61)	0.60 (0.47, 0.59)	0.59 (0.47, 0.60)	0.59 (0.46, 0.58)
		SVM	**0.54** (0.49, 0.62)	0.51 (0.47, 0.68)	0.70 (0.55, 0.64)	0.65 (0.54, 0.60)	0.61 (0.49, 0.64)	0.61 (0.47, 0.59)	0.55 (0.47, 0.74)	0.60 (0.46, 0.57)
		MLP	0.49 (0.49, 0.79)	0.51 (0.50, 0.59)	0.65 (0.59, 0.75)	0.58 (0.50, 0.64)	0.60 (0.53, 0.85)	0.51(0.53, 0.84)	0.61 (0.55, 0.84)	0.50 (0.26, 0.43)
		CNN	0.50 (0.51, 0.70)	0.54 (0.50, 0.54)	0.70 (0.69, 0.83)	0.81 (0.75, 0.82)	0.67 (0.59, 0.86)	0.68(0.62, 0.87)	0.72 (0.58, 0.87)	0.65 (0.57, 0.81)
	3	NB	0.53 (0.29, 0.54)	0.53 (0.29, 0.43)	0.63 (0.34, 0.60)	0.50 (0.26, 0.38)	0.56 (0.29, 0.58)	0.59 (0.28, 0.54)	0.59 (0.31, 0.57)	0.50 (0.22, 0.37)
		SVM	0.53 (0.29, 0.46)	0.53 (0.23, 0.35)	**0.70** (0.37, 0.57)	0.51 (0.11, 0.15)	0.62 (0.33, 0.64)	0.61(0.27, 0.49)	0.58 (0.24, 0.42)	0.50 (0.09, 0.16)
		MLP	0.48 (0.29, 0.61)	0.49 (0.31, 0.65)	0.59 (0.37, 0.70)	0.50 (0.21, 0.39)	0.58 (0.36, 0.86)	0.51(0.35, 0.83)	0.58 (0.36, 0.84)	0.50 (0.31, 0.75)
		CNN	0.50 (0.32, 0.61)	0.53 (0.24, 0.32)	0.68 (0.46, 0.77)	0.50 (0.21, 0.32)	0.65 (0.40, 0.86)	0.65 (0.42, 0.86)	**0.70** (0.40, 0.86)	0.50 (0.33, 0.77)

* Statistically significant under a pairwise comparison when compared with the respective oversampling approach (p-value < 0.05).

5.2. *Step 2 results*

Step 2 aims to select the best text processing (Stem or POS) and machine learning methods (NB, SVM, MLP, CNN). The respective results are presented in Table 5 and in terms of two cross-source types of results: Amazon→Tripadvisor and Tripadvisor→Amazon. The table highlights in **bold** the best AUC result per test target source (Tripadvisor or Amazon), language (English or Italian) and number of classes (2 or 3).

For the Italian language, quality results were achieved, with all AUC values higher or equal to 0.70, thus similar to the single source experiments (Table 4). However, the English results are much lower than the ones obtained in step 1, being closer to the random classification (AUC of 0.50). To better understand this behavior, we analyzed the sentiment data source distributions (Table 6). Table 6 shows that the Amazon English (ENG) reviews are related to a reduced number of products (45). Moreover, this dataset presents a much higher standard deviation when compared with other data sources. To check if this difference is affecting the English results, we created a new dataset, termed Amazon ENG2, by removing the most reviewed product from Amazon ENG. This new dataset has a standard deviation that is more similar to the other sources (Table 6). We tested the new dataset in step 2 (Table 7). The obtained results show a substantial improvement in the classification performances (with statistical significance), with the best models obtaining AUC values that range from 0.74 to 0.81.

Analyzing the best step 2 results (Table 5 and Table 7), we conclude that the

Table 5. AUC (macro-average F1-score, accuracy) results for cross-source sentiment classification in step 2 (best AUC values per test source, language and same number of classes are in **bold**).

Classes	Algorithm	Amazon→Tripadvisor				Tripadvisor→Amazon			
		Stem		POS		Stem		POS	
		English	Italian	English	Italian	English	Italian	English	Italian
2	NB	**0.54** (0.52, 0.72)	0.59 (0.51, 0.70)	0.49 (0.45, 0.49)	0.60 (0.52, 0.72)	0.53 (0.47, 0.50)	0.57 (0.44, 0.53)	0.50 (0.45, 0.48)	0.58 (0.43, 0.52)
	SVM	**0.54** (0.49, 0.65)	0.59 (0.49, 0.67)	0.51 (0.41, 0.49)	0.44 (0.40, 0.64)	0.51 (0.46, 0.71)	0.60 (0.44, 0.53)	0.51 (0.44, 0.72)	0.49 (0.47, 0.87)
	MLP	0.52 (0.50, 0.77)	0.58 (0.47, 0.62)	0.51 (0.48, 0.51)	0.61 (0.51, 0.68)	0.52 (0.49, 0.55)	0.59 (0.49, 0.62)	0.50 (0.45, 0.49)	0.58 (0.47, 0.59)
	CNN	0.53 (0.44, 0.50)	0.72 (0.53, 0.67)	0.51 (0.48, 0.51)	**0.75** (0.56, 0.69)	**0.55*** (0.50, 0.54)	**0.76*** (0.58, 0.71)	0.49 (0.46, 0.50)	0.70 (0.54, 0.65)
3	NB	**0.52** (0.31, 0.57)	0.58 (0.34, 0.68)	0.50 (0.27, 0.34)	0.55 (0.32, 0.65)	0.52 (0.28, 0.39)	0.58 (0.26, 0.44)	0.51 (0.26, 0.32)	0.56 (0.27, 0.47)
	SVM	0.52 (0.21, 0.32)	0.48 (0.22, 0.37)	0.50 (0.18, 0.28)	0.44 (0.11, 0.19)	0.48 (0.15, 0.23)	0.45 (0.14, 0.25)	0.45 (0.15, 0.23)	0.46 (0.19, 0.35)
	MLP	0.51 (0.25, 0.48)	0.56 (0.29, 0.52)	0.50 (0.28, 0.37)	0.59 (0.28, 0.44)	0.51 (0.30, 0.40)	0.58 (0.29, 0.46)	0.50 (0.27, 0.33)	0.57 (0.27, 0.42)
	CNN	0.52 (0.26, 0.36)	0.69 (0.30, 0.46)	0.50 (0.27, 0.33)	**0.70** (0.35, 0.52)	**0.54*** (0.28, 0.35)	**0.72*** (0.36, 0.52)	0.50 (0.28, 0.35)	0.69 (0.31, 0.46)

* Statistically significant under a pairwise comparison when compared with other approaches for the same test source and language (p-value < 0.05).

Table 6. Statistics of the data source reviews.

	Amazon ENG	Amazon ENG2	Tripadvisor ENG	Amazon ITA	Tripadvisor ITA
Number of items	45	44	96	123	116
Number reviews	282,781	207,898	519,735	161,443	324,376
Mean (reviews/items)	6,289	4,730	5,413	1,312	2,816
Median	1,803	1,716	2,377	1,123	1,162
Standard Deviation	12,042	6,039	6,707	1,337	7,031
Minimum	10	10	71	20	219
Maximum	74,883	28,888	27,141	7,475	57,864

deep CNN model is the best machine learning algorithm, presenting the best overall AUC performances. Moreover, the POS tag processing method is the best option for the English language (using Amazon ENG2). For the Italian language, stemming leads to better results when Tripadvisor is used as the training source, while POS tag outperforms stemming when Amazon training data is used. Since the performance differences are slight (ranging from 1 to 6 percentage points), we opted to select stemming for the Italian language, since it provides the highest AUC values (0.76 for 2 classes and 0.72 for 3 classes).

5.3. *Step 3 results*

Using the sentiment models selected in step 2 (undersampling, usage of CNN, POS tag for the English language, stemming for the Italian language), we executed the final step 3 (Sect. 3.2), aiming to measure the value of using easy labeled sources (Amazon and Tripadvisor) to train sentiment models that are evaluated on typically non labeled sources (Facebook and Twitter). Table 8 shows the obtained performances for the proposed cross-source cross-domain CNN (CS-CD CNN). This approach is compared with two methods: a sentiment lexicon and a cross-domain

Table 7. AUC (macro-average F1-score, accuracy) results for cross-source sentiment classification in step 2 and using Amazon ENG2 (best AUC values per number of classes are in **bold**).

Classes	Algorithm	Amazon ENG2 → Trip. ENG		Trip. ENG→ Amazon ENG2	
		Stem	POS	Stem	POS
2	NB	0.50 (0.45, 0.59)	0.63 (0.59, 0.77)	0.50 (0.46, 0.50)	0.62 (0.55, 0.59)
	SVM	0.50 (0.44, 0.59)	0.67 (0.55, 0.66)	0.50 (0.45, 0.72)	0.62 (0.54, 0.59)
	MLP	0.50 (0.42, 0.61)	0.64 (0.59, 0.76)	0.50 (0.44, 0.52)	0.66 (0.61, 0.68)
	CNN	0.49 (0.39, 0.53)	**0.78*** (0.66, 0.76)	0.50 (0.44, 0.49)	**0.81*** (0.70, 0.75)
3	NB	0.49 (0.23, 0.37)	0.61 (0.39, 0.74)	0.50 (0.25, 0.32)	0.62 (0.33, 0.51)
	SVM	0.52 (0.22, 0.37)	0.66 (0.36, 0.65)	0.51 (0.16, 0.22)	0.61 (0.30, 0.48)
	MLP	0.49 (0.19, 0.37)	0.62 (0.35, 0.58)	0.50 (0.23, 0.37)	0.63 (0.41, 0.58)
	CNN	0.51 (0.19, 0.36)	**0.74*** (0.41, 0.60)	0.51 (0.26, 0.34)	**0.76*** (0.48, 0.60)

* Statistically significant under a pairwise comparison when compared with other approaches using the same number of classes (p-value < 0.05).

sentiment classification method. We selected the crowdsourcing lexicon proposed by Mohammad and Turney,[16] since it supports both English and Italian languages. As for the cross-domain method, we used the AE-SCL version whose code is freely available in GitHub[d]. The AE-SCL was trained using Blitzer's Amazon product reviews and tested on Twitter and Facebook data. We note that the AE-SCL code only supports the English language and a binary sentiment classification, thus the Italian and three class results are omitted for this method in Table 8.

The best results are achieved by the CS-CD CNN method for Facebook (English and Italian). When compared with the lexicon[16] and AE-SCL,[17] the proposed CS-CD CNN is competitive for the Facebook data, producing the best AUC values (with statistical significance). For Twitter, CS-CD CNN compares favourably in terms of AUC values for the Italian binary classification and English three class, obtaining the same AUC values as the crowdsourcing lexicon for the English binary classification. The AE-SCL produces the second best Facebook English AUC values. The generic crowdsourcing lexicon achieves the worst Facebook English AUC results but obtains the best AUC value for the Twitter Italian three class case, although the 0.55 value is close to the random AUC discrimination of 0.50.

[d]`https://github.com/yftah89/structural-correspondence-learning-SCL`

Table 8. AUC (macro-average F1-score, accuracy) results for cross-source sentiment classification in step 3 (includes a comparison with two other methods; best AUC values in **bold**).

Classes	Algorithm	Target: Facebook		Target: Twitter	
		English	Italian	English	Italian
2	CS-CD CNN	**0.81*** (0.72, 0.81)	**0.78*** (0.73, 0.60)	**0.68** (0.60, 0.61)	**0.60** (0.56, 0.56)
	Lexicon	0.67 (0.64, 0.70)	0.56 (0.58, 0.58)	**0.68** (0.68, 0.70)	0.56 (0.56, 0.62)
	AE-SCL	0.74 (0.25, 0.28)	-	0.50 (0.50, 0.56)	-
3	CS-CD CNN	**0.76*** (0.49, 0.60)	**0.80*** (0.55, 0.51)	**0.65** (0.37, 0.46)	0.50 (0.35, 0.35)
	Lexicon	0.59 (0.46, 0.63)	0.54 (0.37, 0.49)	0.62 (0.51, 0.51)	**0.55** (0.36, 0.47)

* Statistically significant under a pairwise comparison when compared with other approaches using the same number of classes (p-value < 0.05).

5.4. *Discussion*

Table 9 summarizes the main AUC results achieved by the proposed CNN method in all three steps. It is interesting to notice that step 2 (cross Web sources and cross domain SA) improves the test classification performance for Amazon when Tripadvisor is used as training domain. Specifically, Amazon English AUC in step 2 raises by 7 percentage points (p.p.) for both classification tasks (with 2 and 3 classes) when compared with the step 1 results. Similarly, the Amazon Italian AUC increases by 3 p.p. for the binary classification and 2 p.p. for the three-class task. In contrast, there is slight decrease in the AUC performance (from 2 to 4 p.p.) for Tripadvisor when using Amazon training data. The exception is the binary Italian case, which results in the same AUC (75%).

Table 9. Summary of the main CNN sentiment classification results (AUC values).

Classes	Target domain	English		Italian		Target domain	English	Italian
		Step 1	Step 2	Step 1	Step 2		Step 3	Step 3
2	Amazon	0.74	0.81	0.73	0.76	Facebook	0.81	0.78
3		0.69	0.76	0.70	0.72		0.76	0.80
2	Tripadvisor	0.81	0.78	0.75	0.75	Twitter	0.68	0.60
3		0.78	0.74	0.74	0.70		0.65	0.55

More important are the CS-CD CNN step 3 results for Facebook, which correspond to high quality AUC values: 0.81 for the binary and 0.76 for the three class English classification. Similar quality results were reached for the Italian language,

with 0.78 and 0.80 for the binary and three class classifications. As shown in Table 9, these AUC results compare well with the single source (step 1) and cross Web sources (step2) test performances. In effect, the AUC values range from: step1 – 0.69 to 0.81; and step2 – 0.70 to 0.81. This comparison confirms that the proposed CS-CD CNN method is valuable when using Facebook as a target test source. For demonstration purposes, Fig. 3 presents the word clouds, after the POS tag removal, for the most frequent 100 words when using the Amazon, Tripadvisor and Facebook English data. The word clouds denote some similarity among the text sources (e.g., high frequency of *great*, *good*, *best*, *nice* and *bad* terms), helping to explain why the CS-CD approach provides good results for Facebook.

Fig. 3. Example of word clouds (first 100 words) for preprocessed English data.

For Twitter, a reasonable discrimination was achieved in three cases (AUC>0.60). Better results were obtained for the English language, while a poor performance (similar to a random classifier) was achieved for the Italian three class classification. The best performance on Facebook target source was expected, since Facebook comments are not restricted to the character size limit of Twitter, thus the used sentiment words should be more similar to the Amazon and Tripadvisor source reviews. Also, the language differences might be explained by higher complexity of the Italian Latin language in terms of the type of tenses and adjectives used. Indeed, we note that in step 3 the average number of words is: English – 10.2 for Facebook and 4.0 for Twitter; and Italian – 27.0 for Facebook and 13.0 for Twitter. These values denote differences between the text sources, especially for Twitter. For example, the POS tag average sentence size is 10.7 for English Amazon (Table 2), which is much closer to the 10.2 value of Facebook than the 4.0 of Twitter. Moreover, users tend to write tweets with slang and abbreviations, which typically are sparse and thus are not easily visible when analyzing word clouds. Two real examples of such tweets are:

- "i luv the book'da vinci code"; and
- "omgg i ohhdee want mcdonalds damn i wonder if its open lol".

Since slang and abbreviations (e.g., *luv*, *omgg*) are not often used in Amazon or Tripadvisor reviews, the CS-CD model would produce poor results when tested with these type of tweets. For demonstration purposes, Table 10 shows 10 examples of the binary CS-CD CNN probability for the positive class. In this example, the model

correctly identifies the sentiment of 4 Facebook posts and 3 tweets. In particular, the last two rows of Table 10 exemplify that the CS-CD CNN does not correctly detect the sentiment polarity for tweets with the *argghhhh* slang and *omgg* abbreviation.

Table 10. Examples of binary CS-CD CNN positive sentiment classification (correct values using a 0.5 classification threshold are in **bold**).

Text[a]	**Source**[b]	**Target**[c] **class**	**CS-CD CNN probability**
Just back from a superb few days in Liverpool, much of which was spent in this wonderful club. The staff and musicians were excellent[...]	FB	1	**1.00**
First time in with hen party and must say barmaid was sooo rude n sharp wen asked for some merchendise even tho[...]	FB	0	**0.00**
Absolutely fabulous want to go again went with my three girls. Next time I would stay alot longer and want to write my name on the wall[...]	FB	1	**0.93**
Needed at least 3 full days... going back, absolute!! Fantastic 'premium' exclusive collections.	FB	1	**0.74**
A incredible journey back in time. You can feel the history surrounding you[...]	FB	1	0.27
Jquery is my new best friend.	TW	1	**0.74**
I'm itchy and miserable!	TW	0	**0.02**
Obama's speech was pretty awesome last night!	TW	1	**0.74**
argghhhh why won't my jquery appear in safari bad safari!!!	TW	0	0.78
omgg i ohhdee want mcdonalds damn i wonder if its open lol	TW	1	0.39

[a] **Text** – [...]: truncated text. The complete data are available at `https://github.com/paolazola/Cross-source-cross-domain-sentiment-analysis`.
[b] **Source** – FB: Facebook, TW: Twitter.
[c] **Target class** – 0: negative sentiment, 1: positive sentiment.

The obtained results for the proposed CNN model confirm that the combination of freely available labeled Web sources, such as Amazon and Tripadvisor, can help to train generic sentiment analysis models that provide valuable predictions when applied to unlabeled social media texts, particularly for Facebook. The proposed CS-CD approach alleviates the need for arduous human labeling of these social media texts and thus it can be a key element of modern decision support systems. For instance, to perform social media analytics in the areas of Marketing and Finance (e.g., brand monitoring, customer support, analysis of commodity price opinions).[84]

6. Conclusions

In this work we explored a novel cross-source cross-domain sentiment analysis. Our goal is to easily classify the sentiment of distinct items (e.g., restaurant, hotel, book, music) by first fitting a sentiment classifier to easy-to-collect labeled Web sources (from Amazon and Tripadvisor) and then reusing such model to predict the sentiment of typically unlabeled social media reviews (from Facebook and Twitter).

Thus, our cross-source transfer learning approach allieviates the need to construct sentiment models for each single data source and does not require any human effort to classify unlabeled texts.

We adopted a three step experimental methodology, in which distinct modeling methods were tested: balancing training methods – undersampling and oversampling; text preprocessing – stemming and Part-of-Speech (POS) tagging; and learning algorithms – Naive Bayes (NB), Support Vector Machine (SVM), deep Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN). We also considered two different languages (English and Italian) and two types of sentiment classification ({"negative", "positive"} and {"negative", "neutral", "positive"}). The first two steps confirmed the undersampling and CNN learning algorithm as the best modeling approach. Also, the selection of adverbs and adjectives via POS tagging resulted in the best English results, while stemming led to slight better Italian classification performances. In the last step, we applied the selected models under the proposed cross-source cross-domain approach. When using both Amazon and Tripadvisor training sources, the most important results are the high quality classification performances that were obtained using the Facebook source as the target domain. Indeed, the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve was 81% for the polarity classification and 76% for three class classification on the English language. Similar results were reached for the Italian language: 78% AUC for polarity and 80% for three classes. As for Twitter, a reasonable discrimination was achieved for the English language (AUC from 65% to 68%).

To the best of our knowledge, we believe that this is the first work that considered a social media cross-source cross-domain sentiment classification, which is valuable to reduce the laborious human labeling of texts in modern social network platforms, in particular when using Facebook as test target source. In the future, we expect to extend the proposed methodology to other languages (e.g., German, Portuguese), aiming to discover patterns among the language families (e.g., Germanic, Latin). Moreover, other improvements could be achieved by adopting a deep contextualized word representation based on attention networks.[85] We also intend to experiment with other Web opinion platforms, such as Foursquare (`https://foursquare.com/`) or StockTwits (`https://stocktwits.com/`).

Acknowledgments

Research carried out with the support of resources of Big&Open Data Innovation Laboratory (BODaI-Lab), University of Brescia, granted by Fondazione Cariplo and Regione Lombardia. The work of P. Cortez was supported by FCT - Fundação para a Ciência e Tecnologia within the Project Scope UID/CEC/00319/2019. We would also like to thank the three anonymous reviewers for their helpful suggestions.

References

1. B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.
2. Y. Dong, Q. Zha, H. Zhang, G. Kou, H. Fujita, F. Chiclana, and E. Herrera-Viedma. Consensus reaching in social network group decision making: Research paradigms and challenges. *Knowledge-Based Systems*, 162:3–13, 2018.
3. Y. Dong, M. Zhan, G. Kou, Z. Ding, and H. Liang. A survey on the fusion process in opinion dynamics. *Information Fusion*, 43:57–65, 2018.
4. R. Ureña, G. Kou, Y. Dong, F. Chiclana, and E. Herrera-Viedma. A review on trust propagation and opinion dynamics in social networks and group decision making frameworks. *Information Sciences*, 478:461–475, 2019.
5. H. Shi and X. Li. A sentiment analysis model for hotel reviews based on supervised learning. In *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, volume 3, pages 950–954. IEEE, 2011.
6. N. Oliveira, P. Cortez, and N. Areal. Stock market sentiment lexicon acquisition using microblogging data and statistical measures. *Decision Support Systems*, 85:62–73, 2016.
7. N. Wang, S. Ke, Y. Chen, T. Yan, and A. Lim. Textual sentiment of chinese microblog toward the stock market. *International Journal of Information Technology & Decision Making*, pages 1–23, 2018.
8. X. Fu, J. Lee, C. Yan, and L. Gao. Mining newsworthy events in the traffic accident domain from chinese microblog. *International Journal of Information Technology & Decision Making*, 2018.
9. B. Liu. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 2015.
10. J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447, 2007.
11. A. Wallin. *Sentiment analysis of Amazon reviews and perception of product features*. PhD thesis, Master's thesis, Lund University, 2014.
12. A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12), 2009.
13. N. Oliveira, P. Cortez, and N. Areal. The impact of microblogging data for stock market prediction: using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, 73:125–144, 2017.
14. L. Dalla Valle and R. S. Kenett. Official statistics data integration for enhanced information quality. *Quality and Reliability Engineering International*, 31(7):1281–1300, 2015.
15. L. Dalla Valle and R. Kenett. Social media big data integration: A new approach based on calibration. *Expert Systems with Applications*, 111:76–90, 2018.
16. S. M Mohammad and P. D. Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.
17. Y. Ziser and R. Reichart. Neural structural correspondence learning for domain adaptation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 400–410, 2017.
18. G. Mesnil, T. Mikolov, M. Ranzato, and Y. Bengio. Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *arXiv preprint arXiv:1412.5335*, 2014.
19. J. Li, Z. Xu, H. Xu, L. Tang, and L. Yu. Forecasting oil price trends with sentiment of online news articles. *Asia-Pacific Journal of Operational Research*, 34(02):1740019,

2017.

20. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.
21. M. Dragoni and G. Petrucci. A neural word embeddings approach for multi-domain sentiment analysis. *IEEE Transactions on Affective Computing*, 8(4):457–470, 2017.
22. S.J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.
23. X. Fang and J. Zhan. Sentiment analysis using product review data. *Journal of Big Data*, 2(1):5, 2015.
24. B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.
25. K. Dave, S. Lawrence, and D.M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.
26. F. Salvetti, S. Lewis, and C. Reichenbach. Automatic opinion polarity classification of movie. *Colorado research in linguistics*, 17(1):2, 2004.
27. A. Aue and M. Gamon. Customizing sentiment classifiers to new domains: A case study. In *Proceedings of recent advances in natural language processing (RANLP)*, volume 1, pages 2–1. Citeseer, 2005.
28. H. Cui, V. Mittal, and M. Datar. Comparative experiments on sentiment classification for online product reviews. In *AAAI*, volume 6, pages 1265–1270, 2006.
29. V. Ng, S. Dasgupta, and SM Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 611–618. Association for Computational Linguistics, 2006.
30. B. Ohana and B. Tierney. Sentiment classification of reviews using sentiwordnet. In *9th. IT & T Conference*, page 13, 2009.
31. Y. Dang, Y. Zhang, and H. Chen. A lexicon-enhanced method for sentiment classification: An experiment on online product reviews. *IEEE Intelligent Systems*, 25(4):46–53, 2010.
32. S. J. Pan, X. Ni, J. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM, 2010.
33. X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.
34. Y. Jo and A.H Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM, 2011.
35. Y. Yoshida, T. Hirao, T. Iwata, M. Nagata, and Y. Matsumoto. Transfer learning for multiple-domain sentiment analysis—identifying domain dependent/independent word polarity. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.
36. D. Gräbner, M. Zanker, G. Fliedl, and M. Fuchs. Classification of customer reviews based on sentiment analysis. In *Information and Communication Technologies in Tourism 2012*, pages 460–470. Springer, 2012.
37. F. Neri, C. Aliprandi, F. Capeci, M. Cuadros, and T. By. Sentiment analysis on social media. In *2012 IEEE/ACM International Conference on Advances in Social Networks*

Analysis and Mining, pages 919–926. IEEE, 2012.
38. D. Bollegala, D. Weir, and J. Carroll. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 132–141. Association for Computational Linguistics, 2011.
39. M. Ghosh and A. Kar. Unsupervised linguistic approach for sentiment classification from online reviews using sentiwordnet 3.0. *Int J Eng Res Technol*, 2(9), 2013.
40. A. Ortigosa, J. M Martín, and R. M Carro. Sentiment analysis in facebook and its application to e-learning. *Computers in Human Behavior*, 31:527–541, 2014.
41. C. Dos Santos and M. Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.
42. H. Pouransari and S. Ghili. Deep learning for sentiment analysis of movie reviews. Technical report, Stanford University, 2014.
43. D. Tang, B. Qin, T. Liu, and Y. Yang. User modeling with neural network for review rating prediction. In *IJCAI*, pages 1340–1346, 2015.
44. X. Fang and J. Zhan. Sentiment analysis using product review data. *Journal of Big Data*, 2(1):5, 2015.
45. S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273, 2015.
46. KL S. Kumar, J. Desai, and J. Majumdar. Opinion mining and sentiment analysis on online customer review. In *Computational Intelligence and Computing Research (ICCIC), 2016 IEEE International Conference on*, pages 1–4. IEEE, 2016.
47. A. Tripathy, A. Agrawal, and S. K. Rath. Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, 57:117–126, 2016.
48. A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1107–1116, 2017.
49. A. Radford, R. Jozefowicz, and I. Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.
50. M. Dragoni and G. Petrucci. A fuzzy-based strategy for multi-domain sentiment analysis. *International Journal of Approximate Reasoning*, 93:59–73, 2018.
51. K. Zhang, H. Zhang, Q. Liu, H. Zhao, H. Zhu, and E. Chen. Interactive attention transfer network for cross-domain sentiment classification. In *The 33rd AAAI Conference on Artificial Intelligence (AAAI-2019)*, Honolulu, Hawaii, USA, 2019.
52. S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.
53. Y. Li, Q. Ye, Z. Zhang, and T. Wang. Snippet-based unsupervised approach for sentiment classification of chinese online reviews. *International Journal of Information Technology & Decision Making*, 10(06):1097–1110, 2011.
54. I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press Cambridge, 2016.
55. SENTIPOLC. `http://www.di.unito.it/~tutreeb/sentipolc-evalita14/index.html`. In *Evalita*, 2014.
56. S. Li, Z. Wang, G. Zhou, and S.Y.M. Lee. Semi-supervised learning for imbalanced

sentiment classification. In *IJCAI proceedings-international joint conference on artificial intelligence*, volume 22, page 1826, 2011.
57. G.E. Batista, C. P. Ronaldo, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.
58. M. Litvak and N. Vanetik. *Multilingual Text Analysis Challenges, Models, and Approaches*. World Scientific, 2019.
59. M.F. Porter. Snowball: A language for stemming algorithms, 2001.
60. C. D Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.
61. J. Wiebe. Learning subjective adjectives from corpora. *Aaai/iaai*, 20, 2000.
62. D.Q. Nguyen, D.D. Pham, and S.B. Pham. Rdrpostagger: A ripple down rules-based part-of-speech tagger. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 17–20, 2014.
63. H. Schmid. Probabilistic part-ofispeech tagging using decision trees. In *New methods in language processing*, page 154, 2013.
64. M. Baroni, G. Dinu, and G. Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247, 2014.
65. Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.
66. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
67. Z. Wood Doughty, N. Andrews, and M. Dredze. Convolutions are all you need (for classifying character sequences). In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 208–213, 2018.
68. A.Y Ng and M.I Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848, 2002.
69. Y. Liu, J.W. Bi, and Z.P. Fan. A method for ranking products through online reviews based on sentiment classification and interval-valued intuitionistic fuzzy topsis. *International Journal of Information Technology & Decision Making*, 16(06):1497–1522, 2017.
70. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.
71. Z. Wang and X. Xue. Multi-class support vector machine. In *Support Vector Machines Applications*, pages 23–48. Springer, 2014.
72. N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
73. P.D. Mahendhiran and S. Kannimuthu. Deep learning techniques for polarity classification in multimodal sentiment analysis. *International Journal of Information Technology & Decision Making*, 17(03):883–910, 2018.
74. Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.
75. S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
76. J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language*

processing (EMNLP), pages 1532–1543, 2014.
77. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
78. J.D. Prusa and T.M. Khoshgoftaar. Improving deep neural network design with new text data representations. *Journal of Big Data*, 4(1):7, 2017.
79. Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.
80. T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.
81. I. Witten, E. Frank, M. Hall, and C. Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Franscico, USA, San Francisco, CA, 4th edition, 2017.
82. P. Cortez. Data mining with neural networks and support vector machines using the r/rminer tool. In *Industrial Conference on Data Mining*, pages 572–583. Springer, 2010.
83. M. Hollander and D.A Wolfe. *Nonparametric statistical methods*. Wiley-Interscience, 1999.
84. P. Zola, P. Cortez, and M. Carpita. Twitter user geolocation using web country noun searches. *Decision Support Systems*, 120:50–59, 2019.
85. Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018.