Genetic Programming for Domain Adaptation in Product Reviews

Iti Chaturvedi
Information Technology
James Cook University, Australia
iti.chaturvedi@jcu.edu.au

Erik Cambria, Sandro Cavallari
School of Computer Science and Engineering,
NTU, Singapore
cambria@ntu.edu.sg

Roy E. Welsch
Sloan School of Management
MIT, USA
rwelsch@mit.edu

***Abstract*—There is a large variety of products sold online and the websites are in several languages. Hence, it is desirable to train a model that can predict sentiments in different domains simultaneously. Previous authors have used deep learning to extract features from multiple domains. Here, each word is represented by a vector that is determined using co-occurrence data. Such a model requires that all sentences have the same length resulting in low accuracy. To overcome this challenge, we model the features in each sentence using a variable length tree called a Genetic Program. The polarity of clauses can be represented using mathematical operators such as '+' or '-' at internal nodes in the tree. The proposed model is evaluated on Amazon product reviews for different products and in different languages. We are able to outperform the accuracy of baseline multi-domain models in the range of 5–20%.**

***Index Terms*—Genetic Programming, Sentiment Analysis**

I. Introduction

In recent years, sentiment analysis has become increasingly popular for processing social media data on online communities, blogs, wikis, microblogging platforms, and other online collaborative media [1]. Sentiment analysis is a branch of affective computing research that aims to mine opinions from text (but sometimes also images [2] and videos [3]). Most of the literature is on English language but recently an increasing number of works are tackling the multilinguality issue [4], especially in booming online languages such as Chinese [5] and Spanish [6]. Besides traditional domains like marketing and financial forecasting, sentiment analysis applications also include many other areas like monitoring and detecting driver impairment, testing user experience for video games, cyber-issue detection, and helping medical professionals assess the wellbeing of patients, etc.

Sentiment analysis techniques can be broadly categorized into symbolic and sub-symbolic approaches: the former uses lexicons [7] to encode the polarity associated with words and multiword expressions; the latter consist of supervised [8], semi-supervised [9] and unsupervised [10] machine learning techniques that perform sentiment classification based on word co-occurrence frequencies. Among these, the most popular are algorithms based on deep neural networks [11], belief networks [12], randomized networks [13], generative adversarial networks [14], and capsule networks [15]. There are also some hybrid frameworks that leverage both symbolic and sub-symbolic approaches [16], [17].

While most works approach it as a simple categorization problem, sentiment analysis is actually a complex research problem that requires tackling many NLP tasks [18], including subjectivity detection [19], aspect extraction [20], aspect target sequence modeling [21], word polarity disambiguation [22], time expression recognition [23], intensity measure [24], and commonsense reasoning [25].

One of the main issues of sentiment analysis is that different words are used to express different sentiments in different domains, e.g., the word 'fast' is used to describe good electronics however it carries no polarity in book reviews [26], [27]. Previous authors have tried to overcome challenges such as out of vocabulary words using neural machine translation. This method aims to maximize the conditional probability of a parallel sentence pair (s_{en}, s_{fr}), where s_{en} belongs to the source language such as English and s_{fr} belongs to the target language such as French [27]. Such methods overlook the potential of using monolingual data that is available in abundance in a particular language. A few authors have looked at monolingual training however they only focus on decoding the results instead of improving the training of such models [28].

Domain adaptation is where a model adapts to new products or languages that were not seen during training but the task remains the same. Recently, genetic programming (GP) was used to extract domain independent features from images [29], [30]. GP aims to solve tasks by the natural evolution of computer programs via mutation and crossovers [31]. Inspired by their work, we use GP for domain adaptation of sentiment across products and languages. To our knowledge, this is the first time GP has been applied to multi-domain sentiment analysis. Hence, we refer to the our model as Genetic Opinion Adaptation Learning (GOAL).

Most classifiers require that all sentences have an equal number of features. A GP is able to model sentences of variable lengths. Another limitation is that sentences from different languages differ in the order of nouns and adjectives. The task of GP is to order the phrases in correct grammatical order using mathematical operators. Here, each leaf node in the GP tree is the word vector representation for a single word or a phrase in the sentence. The remaining internal and root nodes are mathematical operators. Solving the GP tree will provide the class label of the sentence. Pre-trained

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

word vectors capture linguistic patterns. For example, the following two operations are equivalent: (a) king – man (b) queen – woman. Hence, the operators such as '+' and '-' are meaningful in the word vector space.

In contrast to traditional GP where the input features are words, in our model the input features are extracted from sentences using deep learning [32]. The hidden neurons in the first layer of a deep model, will learn n-gram features or kernels from different languages. Hence, one kernel may learn features in French and another will learn features in German. The contrastive divergence (CD) approach will sample features with high frequency into the upper layers, resulting in the formation of complex phrases. The word vectors are syntactically and semantically connected. Hence, the mathematical operations such '+' and '-' can be translated into 'positive' and 'negative' sentiment in word vector space. Each feature input into a GP classifier can be interpreted as a phrase of two or three words in vector space. The corresponding tree structure will be a good approximation of the underlying grammar. Furthermore, in [33] the authors proposed multitask GP that can simultaneously perform sentiment detection, identify names, disambiguate meanings of words and also extract relations in a single model.

The remainder of the paper is organized as follows: Section II illustrates related work; Section III describes CDBN; Section IV introduces the proposed GOAL; next, Section V validates the proposed method on two benchmark datasets; finally, Section VI proposes concluding remarks.

II. Related Work

Traditional models such as long short-term memory (LSTM) truncate all sentences to the same length leading to loss of information [34], [35]. Instead GP can evolve a population of variable-length trees. Each tree can model sentences of a specific length. The optimal tree will be able to model sentences of different lengths using sub-trees. Similarly, the order of phrases can be easily changed in a GP classifier by simply changing a branch in the tree. The optimal tree will then be able to model sentences of different languages by assigning suitable weights to different branches using mathematical operations.

Previously, [36] showed that GP can be used for text classification. In their model, the leaf nodes were features such as term frequency, max frequency term, information gain weights for term etc. that do not generalize well to new domains and languages. In this paper, instead, we consider a semi-supervised convolutional deep belief network (CDBN) to extract features from text. Deep learning is a type of semi-supervised learning. Here the pre-training is unsupervised Gibbs sampling and it is followed by supervised gradient descent. The low-dimensional features learned are used to train a GP classifier. We are also motivated by the work done in [29]. They showed that a model could adapt across images from different domains using genetic operators [30]. Similarly, in our paper we consider one-dimensional convolutional features instead of two dimensional image features and the deep model is simultaneously trained with product reviews in different domains and languages.

Another model for multi-lingual sentiment analysis leverages on machine translation [37]–[39] via parameter sharing between two LSTMs. However, machine translation results in loss of sentiment because it uses the lemma form of all words. Our proposed model does not perform any translation and every variant of a word has a distinct word vector representation. Our approach to classify sentences into different intensities of sentiment consists of two steps: (a) Learning features using CDBN trained on data from multiple source domains; (b) Constructing a GP algorithm based on the features learned using CDBN.

Figure 1(a) illustrates the state space of a GP for a book review. The convolutional features learned at different hidden neurons may belong to different languages. The word vector representation of positive bi-grams such as 'beautiful drawings' and the French translation 'beaux dessins' will be similar. In this way, the neural network uses word vectors to distinguish polarities and the GP is able to perform domain adaptation across different domains and languages. In addition, the mathematical functions linking words can capture the context between words far apart in a sentence. Validation of the proposed method is performed on three real-world benchmarks taken from Amazon.com.

III. Convolutional Deep Belief Network

In this section, we begin with a description of the unsupervised restricted Boltzmann machine (RBM) model. A hierarchy of RBM's where the hidden layer of one RBM serves as the visible layer of the next RBM results in a CDBN.

Each sentence is transformed to a word vector representation of dimension $d \times L$ where L is the length of the sentence and d is the dimension of pre-trained vectors for each word. The word vector input is used to train an RBM that is a bipartite graph consisting of two layers of neurons: a visible and a hidden layer; where the connections among neurons in the same layer are not allowed. To learn such weights and maximize the global energy function E, the approximate maximum likelihood CD approach can be used. This method employs each training sample to initialize the visible layer. Next, it uses the Gibbs sampling algorithm to update the hidden layer and then reconstruct the visible layer consecutively, until convergence. As an example, here we use a logistic regression model to learn the binary hidden and visible neurons.

The continuous state $\hat{h}_j$ of the hidden neuron j, with bias b_j, is a weighted sum over all continuous visible nodes $\boldsymbol{v}$ and is given by:

$$\hat{h}_j = b_j + \sum_i v_i w_{ij}, \tag{1}$$

where w_{ij} is the connection weight to hidden neuron j from visible node v_i. The binary state h_j of the hidden neuron can be defined by a sigmoid activation function:

$$h_j = \frac{1}{1 + e^{-\hat{h}_j}}, \tag{2}$$

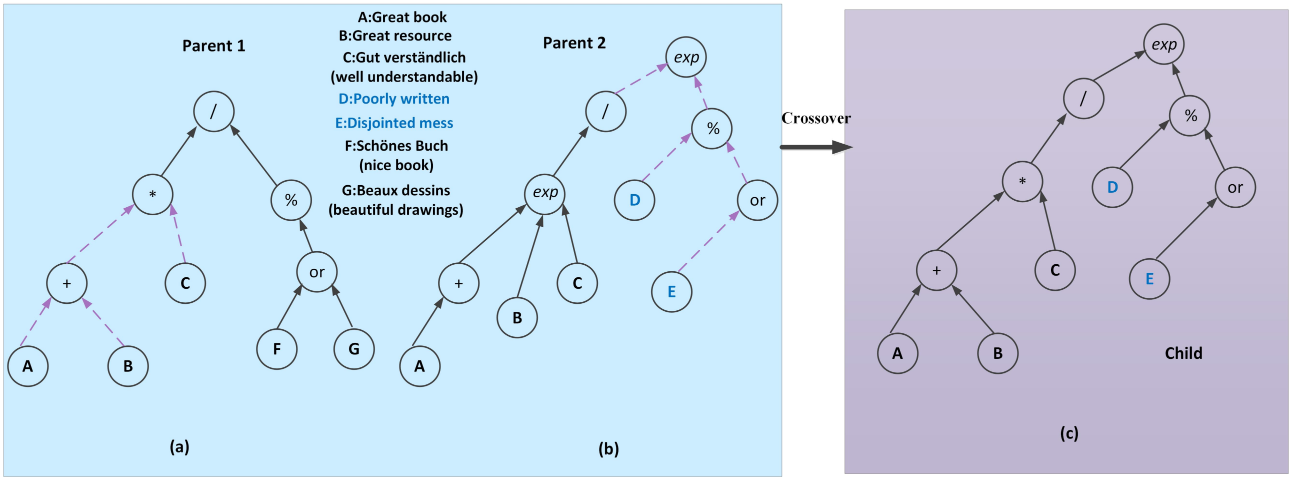

Fig. 1: Sub-tree crossover operator in GP for a book review. Negative reviews are in blue. The sub-tree crossover works by selecting two elite parent solutions/trees (a and b) and randomly selecting an internal node in each of the parent trees (purple dashed arrows). Crossover results in two new children (c)

Similarly, in the next iteration, the binary state of each visible node v_i is reconstructed.

Lastly, the weights w_{ij} are updated as the difference between the original and reconstructed visible layer labeled as the vector $\boldsymbol{v}_{recon}$, using:

$$\triangle w_{ij} = \alpha(< v_i h_j >_{data} - < v_i h_j >_{recon}), \tag{3}$$

where α is the learning rate and $< v_i h_j >$ is the expected frequency with which visible unit i and hidden unit j are active together when the visible vectors are sampled from the training set and the hidden units are determined by (1).

A hierarchy of RBM layers results in a deep belief network (DBN). In such a model, the lower layers learn abstract concepts and the higher layers learn complex features for sentences. To train such a multi-layer system, we must compute the gradient of the total energy function with respect to the weights in all the layers.

To extend the DBN to a CDBN, we simply partition the hidden layer into Z groups. Each of the Z groups is associated with a $n_x \times n_y$ filter where n_x is the width of the kernel and n_y is the height of the kernel. Let us assume that the input has dimension $\mathrm{L_x} \times \mathrm{L_y}$. Then the convolution will result in a hidden layer of Z groups each of dimension $(\mathrm{L_x} - n_x + 1) \times (\mathrm{L_y} - n_y + 1)$. These learned kernel weights are shared among all hidden units in a particular group. The energy function of layer l is now a sum over the energy of individual blocks given by:

$$\mathrm{E}^l = -\sum_{z=1}^{\mathrm{Z}} \sum_{i,j}^{(\mathrm{L_x}-n_x+1),(\mathrm{L_y}-n_y+1)} \tag{4}$$
$$\sum_{r,s}^{n_x,n_y} v_{i+r-1,j+s-1} h_{ij}^z w_{rs}^l .$$

For the case of sentences, we consider a one-dimensional convolution hence we set the height of the kernel n_y equal to the input word vector length d. Figure 2 illustrates the state diagram for the proposed deep genetic program. The training data is collectively trained using source language (English) and target language (French) samples. Frequently occurring subjectivity clues such as ‘poorly’ and ‘dejantee’ (see Figure 2) are used to select a sub-set (about 20%) of significant product reviews to pre-train the deep model. In order to model the underlying parse tree structure of the sentence, the features are used to train a GP classifier.

IV. Deep Genetic Programming

In this section, we introduce the GP model for classifying sentences. Next, we describe our proposed GOAL framework for evolving the neurons of a CDBN using GP.

A. Genetic Programming for Sentences

GP evolves a population of potential models, each structured in a tree-like fashion, with mathematical functions linking input nodes and constants. The probability of a given model surviving into the next generation depends on its classification accuracy on the training set. Fitness proportional selection, combined with these genetic operators such as crossover and mutation produces a new generation of offspring solutions.

Figure 1 describes the crossover operator during GP. The role of crossover is to take two promising solutions and combine their information to give rise to a new offspring, with the goal that the offspring have better performance than the parents. The sub-tree crossover works by selecting two elite parent solutions/trees (a and b) and randomly selecting an internal node in each of the parent trees (purple dashed arrows). This results in two offspring's that are created by interchanging the sub-trees below the identified nodes in the parent solutions. For example, in Figure 1 (c) the GP will try to use positive operators such as ‘+’ or ‘*’ for ‘Great Book’ and negative operators such ‘-’ or ‘%’ for ‘Poorly Written’. This is because the vector representation for each review in the

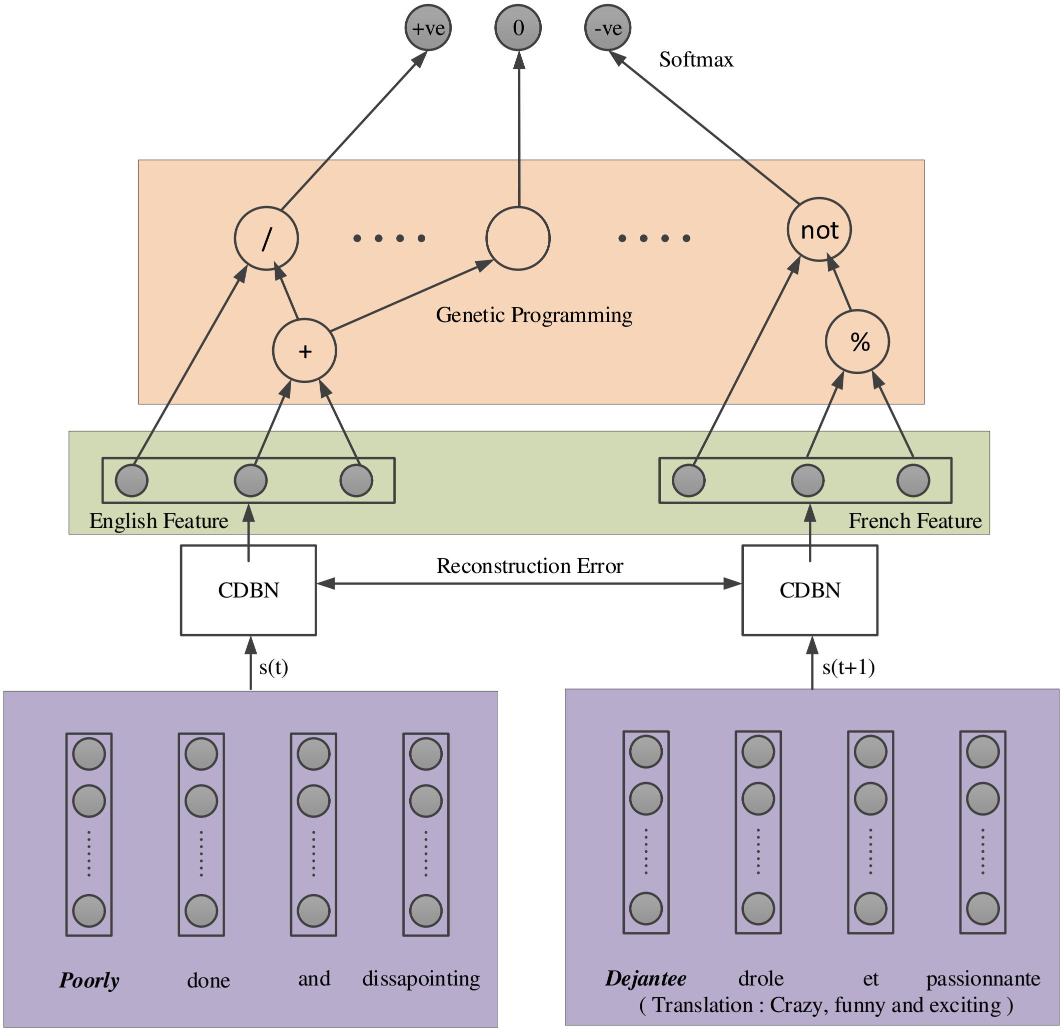

Fig. 2: State diagram for the proposed GOAL. The training data is collectively trained using source language (English) and target language (French) samples. In order to model the underlying parse tree structure of the sentence, the features are used to train a GP classifier.

test set is used to compute the GP tree, if the value computed is less than 0.5 then the review is predicted as negative and if it is greater than 0.5 then the review is positive.

Algorithm 1 describes a simple steady state GP classifier for sentences. Here, a GP algorithm in the form of a tree is used for classification where leaf nodes are words in a sentence and all other nodes are functions such as the set $\boldsymbol{F} = \{+, -, *, /, sin, cos, exp, <, >, sqrt, cube\}$. Here $/$ denotes protected division that returns 1 if the denominator is 0. We first initialize a large population of GP's randomly. We start with a root p and n_p children for the root. Next, for each child we randomly generate a new sub-tree until the maximum number of nodes is reached. The next generation is reproduced through the crossover operation. Here, we selectively rank all the GP based on training error. Next, two elite parent GP's are selected, and we randomly replace a sub-tree in one parent GP with a sub-tree from the second parent GP resulting in two new children GP for the next generation. This process of evolution continues until convergence when the accuracy of elite GP in each generation does not improve any further. The elite program in the last generation is used to determine the class label of test sentences.

A GP classifier for sentences would have leaf nodes equal to the sentence length. In this paper, however, we train the GP on concepts and hence fewer nodes are needed. A GP can get stuck in local optimum solution if the population similarity in each generation is low (<0.6%). Hence, we keep the maximum number of nodes to 20, so that the similarity between trees is high and it can easily converge to the global optimum solution. Irrespective of the number of input features, each GP tree will have 20 nodes and discard the remaining features. It is interesting to note that each feature learned by deep learning is a phrase of two or more words. The length of phrases increases with each layer of deep learning. Hence, we see that 20 nodes are sufficient to represent even long sentences.

Algorithm 1 GP classification

```
Input 1 : Training and test data $(x_{ij})_{n \times T}$ for $n$ neurons and $T$ sentences
Input 2 : Corresponding class labels $(n_y)_{1 \times T} \in \{P+, P, 0, N, N+\}$
Output 1 : GP classifier
Output 2 : Class labels of Test Sentences
% Initialize a population of random GP's
Initialize GP root $p$ and children p.children with length $n_p$
repeat
    $p.children[k] \leftarrow$ randomly generated subtree
until $k < n_p$
% Crossover Operations to generate new population
repeat
    Select Two Elite GP $p1$ and $p2$ based on Accuracy
    Select subtree1 = $p1.children[1 : k1]$ for any $k1$
    Select subtree2 = $p2.children[k1 + 1 : n_p]$
    Merge subtree1 and subtree2 resulting in two new children
    $l = l + 2$
until convergence
Each test sample is classified using predicted GP
Accuracy : % of correctly classified test samples
```

B. Evolving CDBN using Genetic Programming

We first construct a minimal CDBN with visible layer of L$\times d$ nodes, where L is length of the sentence and d is the word vector length; there are several hidden convolution layers of k-gram neurons, then there is a penultimate hidden logistic layer of n_h neurons and the last layer is output neurons each class $n_y \in \{P+, P, 0, N, N+\}$ where '$P+$' is strongly positive, 'O' is neutral and '$N+$' is strongly negative review.

The n_h features expressed at the hidden neurons after training form the new input data of T samples. Next, we build a GP classifier over the hidden neurons. Each test sample is then used to generate an embedding of dimension n_h features from CDBN and then classified by computing the GP. Figure 2 provides the flowchart of the deep GP classifier. CDBN assumes that each input word is represented as a d dimensional vector. Each grey circle, is a single feature in the vector. In this paper, we used pre-trained word vectors. For two consecutive training samples $s(t)$ and $s(t+1)$ reconstruction error is used to update the weights.

To determine the number of hidden layers in the CDBN, we compute the change in error $\triangle\epsilon$ on the validation samples. This is the root mean square error between input training sample and reconstructed sample at each visible node. If there is a significant change in the error $\triangle\epsilon$, a new hidden layer is added. The contrastive divergence approach will sample features with high frequency into the upper layers resulting in the formation of n-grams at hidden neurons in the first layer, bigger phrases at hidden neurons in the second hidden layer and so on.

V. Experiments

In this section, the proposed GOAL (available on GitHub[1]) was applied to three real world sentiment classification problems in order to assess its efficacy. All three datasets consist of review text and rating labels (1-5). A rating of 1 is strongly negative, 2 is weakly negative, 4 is weakly positive, and 5 is strongly positive. The reviews with rating label 3 are removed as they are deemed as ambiguous and hence are indecisive about a product. In order to have a fair comparison for each benchmark dataset, we have reported results by previous authors on the same dataset.

A. Parameter Setting

We used pre-trained word vectors for different languages (provided by Facebook[2]). Following previous authors, the word vector length was empirically set to 300, and unknown words were randomly initialized to vectors from Gaussian distributions. In each iteration an individual undergoes either crossover (with probability 0.8) or mutation (with probability 0.19) or is selected as elite (with probability 0.01) and passed to the next generation. There is a population of 2000 GP's in each generation where the maximum tree size for each GP is 20 nodes [40]. Training stops when the mean square error (MSE) of the elite GP in a generation is less than 0.02. Our best results are obtained with an ensemble of GOAL 10-fold cross-validations that differ in their random initialization and mini-batches of 100 samples. Lastly, for the CDBN to determine the number of hidden layers and the number of neurons in each layer we consider the validation error on training data. The training was done using stochastic gradient descent in an unsupervised manner. We found a model with four layers and 50 neurons in each layer optimal. The width of the kernels was progressively increased from 3 to 7 words in the higher layers.

B. Multi-domain Sentiment Dataset

In this section, we verify the effectiveness of GOAL in classifying sentences using the multi-domain sentiment analysis dataset [41]. Following previous authors, we first report the results on the binary problem of classifying reviews as positive (4 or 5) and negative (1 or 2). The four domains consist of 'Books' (B), 'DVD' (D), 'Electronics' (E), and 'Kitchen' (K) reviews, where each domain contains 2000 reviews. Hence, as an illustration training data in the form of 1000 positive and 1000 negative reviews were taken.

1) Unique Source Domain - Binary Labels: We construct 12 cross-domain tasks of sentiment classification on this dataset. Here, 2000 reviews in one domain are the training data and 2000 reviews in a different domain are the test data. Table I shows the comparison for different methods. In all tasks, the training set is from one unique source (S) domain and the test set is from another target domain. For example, when 'Books' (B) is the target domain then 'Dvd' (D) is the

[1] http://github.com/senticnet/genetic-programming-for-domain-adaptation

[2] http://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

source domain. We can see that by only using a CDBN we get a very low accuracy (<60%). Lastly we see that by training a GP using the features learned by CDBN (GOAL) the accuracy improves by over 10%.

The proposed GOAL outperforms Transfer Deep Network (TDN) by over 5% and R3 [28] by over 20%. In TDN [42], the authors considered two parallel deep auto-encoders to learn transferable features and classification features. However they do not use convolutional neural networks, hence they are unable to capture the context of words. In R3 [28], the authors proposed three rules that must be satisfied for cross-domain classification. They considered handcrafted features, instead in our method we automatically learn cross-domain features.

TABLE I: Accuracy of the 12 English tasks with binary labels. In all tasks, the training set is from one unique source domain and the test set is from another domain. For example, when 'Books' (B) is the target domain then 'Dvd' (D) is the source (S) domain.

Target	S	TDN	R3	CDBN	GOAL
Books	D	86.5	70.0	52.2	89.5
	E	83.3	62.0	51.5	90.2
	K	82.5	66.0	54.8	90.2
Dvd	B	86.2	72.0	53.1	92.2
	E	83.2	68.0	51.3	91.0
	K	83.2	70.0	53.8	89.5
Electronics	B	85.1	70.0	52.2	91.7
	D	86.2	70.0	52.0	89.5
	K	87.8	74.0	54.1	91.7
Kitchen	B	87.9	72.0	52.9	87.7
	D	88.1	73.0	51.6	89.7
	E	90.0	80.0	51.8	91.0

2) Multiple Source Domains - Binary Labels: Next, we considered the problem of polarity detection in the target domain 'Books' when the model is trained on the source data from the other three domains. To evaluate cross-domain transfer we follow the experiment defined in [43]. Here, 2000 samples are taken from each of the three source domains and the model is tested on the 2000 samples in the target domain. The 2000 samples in each source domain are divided equally among the 4 ratings.

TABLE II: Classification accuracy by different models in the multi-domain Amazon dataset. In all tasks, the training set is from three domains and the test set is from the fourth domain.

Target	PDM	GOAL
Books	71	86.9
Dvd	71	88.9
Electronics	76	89.6
Kitchen	75	89.9

3) Visualization of features: Our experiment also show that the classification performance seems to benefit from adaptation between semantically close domains such as 'Books→Dvd(92.2%)', however the performance is less when the source and the target domains are dissimilar such as 'Books→Kitchen(87.7%)'. Lastly, Table III illustrates the predicted mathematical expressions for four hidden neurons on the Books review dataset. The terminal nodes A1:A8 correspond to n-gram features learned by CDBN. We also visualized some positive and negative 4-grams in the first layer of CDBN. Positive bi-grams are 'trademark humor' and 'communicating wisdom'. An example of a negative bi-gram learned is 'peculiar biblical'.

C. Cross-Language Sentiment Dataset

In this section, we verify the portability of GOAL for classifying sentences across other languages. In particular, we used a cross-language sentiment analysis dataset available in [44]. Similar to the previous experiment, there are three domains (namely, 'Books', 'Music' and 'DVD') and four languages (English, French, German and Japanese). We tokenized the Japanese sentences in order to split phrases into individual words (available on GitHub[3]).

We construct 18 cross-domain cross-language tasks of sentiment classification on this dataset. Following previous authors, we first report the results on classifying reviews as positive (4 or 5) and negative (1 or 2). Here a balanced dataset of 2000 reviews (1000 positive and 1000 negative) in one domain and in English language are the training data and 2000 reviews in a different domain and in a different language is the test data. In addition, we consider an unbalanced set of 20,000 reviews (unequal number of positive and negative reviews) in the target domain as training data. This makes it difficult for the classifier to learn both positive and negative features in the target domain simultaneously. Domain adaptation from the source domain is then needed to learn the features accurately. Table V shows the comparison for different methods. The proposed GOAL outperforms the baseline distributional correspondence function (DCF) [26] by over 10%. In DCF, the authors represent terms in vectorial space based on their distributional correspondence with respect to a fixed set of terms. Hence, their method relies on human effort for selecting suitable 'bilingual pivots'. Instead in our model can automatically learn a dictionary of features that is portable across languages.

TABLE III: Predicted Mathematical Expression for Book reviews. The terminal set corresponds to 4-gram features learned by CDBN

GP	Books→Dvd	$\exp(\sqrt(A4))$
	Books→Electronics	$exp((A5 - A2))$
	Books→Kitchen	$\gamma((\sqrt(A2) * (\exp(A2))$
+ve	A1	trademark humor sideways religion
	A2	wordy numerous metaphors make
	A3	exercises communicating wisdom effective
	A4	preacher meditate ability respond
-ve	A5	peculiar biblical misinterpretations editors
	A6	witches almanac elizabeth pepper
	A7	man battles fray sword
	A8	necessarily machines today review

Next, we consider the problem of polarity detection in the target language when the model is trained on product reviews in other languages. Here we consider 600 product reviews each from Books, DVD, and Electronics, resulting

[3] http://github.com/gpeterson2/Japanese-Tokenizer

in 1800 training reviews in each language. Similarly, the test data is 1800 reviews from the target domain. It can be seen in Table IV that the proposed GOAL outperforms the accuracy of baselines by 5–10%. For the case of the cross-language sentiment dataset, almost 10% improvement is observed over Distribution Matching based Matrix Completion (DMMC) [27]. This is because they consider active learning to include human annotation into the prediction. Their model is not practical for diverse languages such as English and Japanese. On the other hand, GOAL is able to automatically learn features from different languages simultaneously.

We also observed a 5% improvement over hybrid heterogeneous transfer learning (HHTL) [45], where the authors have introduced a new bias matrix to improve heterogeneous transfer from English to other languages in a deep auto-encoder framework. Auto-encoders try to reconstruct the inputs and hence are not scalable to a large number of layers. English performs slightly lower on the binary task compared to other languages. This is because word vectors in other languages are accurate due to small training samples. It can also be seen that by using convolution we are able to outperform baselines by a big margin in Japanese, for example, DCF shows 72.1% accuracy for ’English_Music’→’Japanese_Books’, however GOAL shows 89%.

TABLE IV: Classification accuracy by different models in the cross-language Amazon dataset. In all tasks, the training set is from three domains and three languages and the target test set is from the three domains in the fourth language.

Target	DMMC	HHTL	GOAL
English	-	-	83.7
French	72	82.5	89.7
German	75	82.7	93.6
Japanese	68	75.5	90.1

TABLE V: Accuracy of the 18 cross-domain and cross-language tasks with binary labels. In all tasks, the training set is from one unique source domain and in English and the test set is from another domain and another language. For example, when ’English_Music’ is the source domain and ’Japanese_Books’ is the target domain.

Target Language	Source→Target	DCF	GOAL
German	English_Dvd→Books	82.4	89.0
	English_Music→Books	81.2	90.0
	English_Books→Dvd	82.7	89.0
	English_Music→Dvd	83.4	90
	English_Books→Music	84.3	89
	English_Dvd→Music	81.6	90
Japanese	English_Dvd→Books	76.1	90.0
	English_Music→Books	72.1	89.0
	English_Books→Dvd	80.5	90.0
	English_Music→Dvd	79.0	90.0
	English_Books→Music	83.1	90
	English_Dvd→Music	81.6	90
French	English_Dvd→Books	84.8	89.0
	English_Music→Books	84.5	90.0
	English_Books→Dvd	82.3	89.0
	English_Music→Dvd	84.1	90
	English_Books→Music	84.3	89
	English_Dvd→Music	84.7	90

D. Semeval 2017 Arabic dataset

In this section, we verify the portability of GOAL for classifying short tweets across languages. In particular, we used the SemEval 2017 Task 4 dataset [46]. This dataset contains tweets in ‘English’ and ‘Arabic’. Arabic is written right to left, hence we reversed the sentences before processing. Following previous authors, we report the results on classifying reviews as positive (4 or 5) and negative (1 or 2). The dataset contains 20,510 English tweets and 1,656 Arabic tweets. We used all the English and 80% of the Arabic tweets to train the classifier and the remaining 20% of Arabic tweets as the test dataset. Table VI shows that our method outperforms baseline NileTMRG [46] and ELiRF-UPV [47] by over 8%. This is because both baselines use a traditional convolutional neural network that is unable to model the variable length features in sentences.

TABLE VI: Accuracy of cross-language twitter task with binary labels. The training data is in English and the test data is in Arabic

NileTMRG [46]	ELiRF-UPV [47]	GOAL
77	73	85

VI. Conclusion

In this paper, we have proposed a sentiment classifier that can be trained in one domain or language and may be used to classify sentences in a new domain or language. This is achieved using deep convolutional belief networks to automatically extract n-grams from product reviews. The deep model is trained on one or more source languages with abundant data and tested on the target language that has few training samples.

Next, in order to mimic variable length sentences structures, we use a previously proposed GP classifier to evolve the features extracted using CDBN. We show that our model is able to accurately classify positive, negative and neutral reviews in languages such as French and Japanese. Our simulation and experimental study show that the proposed method outperforms baseline approaches in terms of prediction accuracy by over 5–20%. Last but not least, the mathematical functions linking words in GP provide valuable clues towards polarity of the sentence and capture the context between words that are far apart in a sentence. One limitation of the proposed model is that the variance is high during heuristic search. We can also target the domain adaptation problem using a multi-task evolutionary framework in the future.

VII. Acknowledgement

This work is partially supported by the Computational Intelligence Lab (CIL) at the Nanyang Technological University. This work is also partially supported by the College of Science and Engineering at James Cook University.

References

[1] E. Cambria, N. Howard, Y. Xia, and T.-S. Chua, "Computational intelligence for big social data analysis," *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 8–9, 2016.

[2] E. Ragusa, E. Cambria, R. Zunino, and P. Gastaldo, "A survey on deep learning in image polarity detection: Balancing generalization performances and computational costs," *Electronics 8 (7), 783*, vol. 8, no. 7, p. 783, 2019.

[3] I. Chaturvedi, R. Satapathy, S. Cavallari, and E. Cambria, "Fuzzy commonsense reasoning for multimodal sentiment analysis," *Pattern Recognition Letters*, vol. 125, no. 264-270, 2019.

[4] S. L. Lo, E. Cambria, R. Chiong, and D. Cornforth, "Multilingual sentiment analysis: From formal to informal and scarce resource languages," *Artificial Intelligence Review*, vol. 48, no. 4, pp. 499–527, 2017.

[5] H. Peng, E. Cambria, and A. Hussain, "A review of sentiment analysis research in chinese language," *Cognitive Computation*, vol. 9, no. 4, pp. 423–435, 2017.

[6] I. Chaturvedi, E. Cambria, and D. Vilares, "Lyapunov filtering of objectivity for Spanish sentiment model," in *IJCNN*, 2016, pp. 4474–4481.

[7] F. Xing, F. Pallucchini, and E. Cambria, "Cognitive-inspired domain adaptation of sentiment lexicons," *Information Processing and Management*, vol. 56, no. 3, pp. 554–564, 2019.

[8] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, and E. Cambria, "Bayesian network based extreme learning machine for subjectivity detection," *Journal of The Franklin Institute*, vol. 355, no. 4, pp. 1780–1797, 2018.

[9] A. Hussain and E. Cambria, "Semi-supervised learning for big social data analysis," *Neurocomputing*, vol. 275, pp. 1662–1673, 2018.

[10] E. Cambria, T. Mazzocco, A. Hussain, and C. Eckl, "Sentic medoids: Organizing affective common sense knowledge in a multi-dimensional vector space," ser. Lecture Notes in Computer Science. Berlin Heidelberg: Springer-Verlag, 2011, vol. 6677, pp. 601–610.

[11] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, and E. Cambria, "Learning word representations for sentiment analysis," *Cognitive Computation*, vol. 9, no. 6, pp. 843–851, 2017.

[12] I. Chaturvedi, Y. S. Ong, I. Tsang, R. Welsch, and E. Cambria, "Learning word dependencies in text by means of a deep recurrent belief network," *Knowledge-Based Systems*, vol. 108, pp. 144–154, 2016.

[13] G.-B. Huang, E. Cambria, K.-A. Toh, B. Widrow, and Z. Xu, "New trends of learning in computational intelligence," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 16–17, 2015.

[14] Y. Li, Q. Pan, S. Wang, T. Yang, and E. Cambria, "A generative model for category text generation," *Information Sciences*, vol. 450, pp. 301–315, 2018.

[15] W. Zhao, H. Peng, S. Eger, E. Cambria, and M. Yang, "Towards scalable and reliable capsule networks for challenging NLP applications," in *ACL*, 2019, pp. 1549–1559.

[16] E. Cambria, A. Hussain, C. Havasi, and C. Eckl, "Sentic computing: Exploitation of common sense for the development of emotion-sensitive systems," ser. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2010, vol. 5967, pp. 148–156.

[17] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, "Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis," *Cognitive Computation*, vol. 10, no. 4, pp. 639–650, 2018.

[18] E. Cambria, S. Poria, F. Bisio, R. Bajpai, and I. Chaturvedi, "The CLSA model: A novel framework for concept-level sentiment analysis," in *LNCS*. Springer, 2015, vol. 9042, pp. 3–22.

[19] I. Chaturvedi, E. Cambria, R. Welsch, and F. Herrera, "Distinguishing between facts and opinions for sentiment analysis: Survey and challenges," *Information Fusion*, vol. 44, pp. 65–77, 2018.

[20] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *AAAI*, 2018, pp. 5876–5883.

[21] H. Peng, Y. Ma, Y. Li, and E. Cambria, "Learning multi-grained aspect target sequence for chinese sentiment analysis," *Knowledge-Based Systems*, vol. 148, pp. 167–176, 2018.

[22] Y. Xia, E. Cambria, A. Hussain, and H. Zhao, "Word polarity disambiguation using bayesian model and opinion-level features," *Cognitive Computation*, vol. 7, no. 3, pp. 369–380, 2015.

[23] X. Zhong, A. Sun, and E. Cambria, "Time expression analysis and recognition using syntactic token types and general heuristic rules," in *ACL*, 2017, pp. 420–429.

[24] M. S. Akhtar, A. Ekbal, and E. Cambria, "How intense are you? predicting intensities of emotions and sentiments using stacked ensemble," *IEEE Computational Intelligence Magazine*, vol. 15, no. 1, pp. 64–75, 2020.

[25] E. Cambria, A. Hussain, C. Havasi, and C. Eckl, "Common sense computing: From the society of mind to digital intuition and beyond," in *Biometric ID Management and Multimodal Communication*, ser. Lecture Notes in Computer Science, J. Fierrez, J. Ortega, A. Esposito, A. Drygajlo, and M. Faundez-Zanuy, Eds. Berlin Heidelberg: Springer, 2009, vol. 5707, pp. 252–259.

[26] A. M. Fernández, A. Esuli, and F. Sebastiani, "Distributional correspondence indexing for cross-lingual and cross-domain sentiment classification," *Journal of Artificial Intelligence Research*, vol. 55, no. 1, pp. 131–163, 2016.

[27] J. T. Zhou, S. J. Pan, I. W. Tsang, and S.-S. Ho, "Transfer learning for cross-language text categorization through active correspondences construction," in *AAAI*, 2016, pp. 2400–2406.

[28] D. Bollegala, T. Mu, and J. Y. Goulermas, "Cross-domain sentiment classification using sentiment sensitive embeddings," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 398–410, 2016.

[29] M. Zhang and W. Smart, "Using gaussian distribution to construct fitness functions in genetic programming for multiclass object classification," *Pattern Recognition Letters*, vol. 27, no. 11, pp. 1266 – 1274, 2006.

[30] H. A.-S. Muhammad Iqbal, Bing Xue and M. Zhang, "Cross-domain reuse of extracted knowledge in genetic programming for image classification," *IEEE Transaction on Evolutionary Computation*, vol. 27, 2017.

[31] J. Zhong, L. Feng, and Y. S. Ong, "Gene expression programming: A survey [review article]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 3, pp. 54–72, 2017.

[32] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *ACL*, 2014, pp. 655–665.

[33] J. Zhong, L. Feng, W. Cai, and Y. S. Ong, "Multifactorial genetic programming for symbolic regression problems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 2018.

[34] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification." in *EMNLP*, 2015, pp. 1422–1432.

[35] A. Sil, G. Kundu, R. Florian, and W. Hamza, "Neural cross-lingual entity linking," in *AAAI*, 2018, pp. 5464–5472.

[36] H. J. Escalante, M. A. García-Limón, A. Morales-Reyes, M. Graff, M. M. y Gómez, E. F. Morales, and J. Martínez-Carranza, "Term-weighting learning via genetic programming for text classification," *Knowledge-Based Systems*, vol. 83, pp. 176 – 189, 2015.

[37] A. Balahur and M. Turchi, "Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis," *Computer Speech and Language*, vol. 28, no. 1, pp. 56–75, 2014.

[38] E. A. Platanios, M. Sachan, G. Neubig, and T. M. Mitchell, "Contextual parameter generation for universal neural machine translation," in *EMNLP*, 2018, pp. 425–435.

[39] X. Meng, F. Wei, X. Liu, M. Zhou, G. Xu, and H. Wang, "Cross-lingual mixture model for sentiment classification," in *ACL*, 2012, pp. 572–581.

[40] H. Al-Sahaf, M. Zhang, M. Johnston, and B. Verma, "Image descriptor: A genetic programming approach to multiclass texture classification," in *CEC*, 2015, pp. 2460–2467.

[41] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *ICML*, 2008, pp. 264–271.

[42] M. Long, J. Wang, Y. Cao, J. Sun, and P. S. Yu, "Deep learning of transferable representation for scalable domain adaptation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2027–2040, 2016.

[43] C. Seah, Y. S. Ong, and I. W. Tsang, "Combating negative transfer from predictive distribution differences," *IEEE Transactions on Cybernetics*, vol. 43, no. 4, pp. 1153–1165, 2013.

[44] P. Prettenhofer and B. Stein, "Cross-language text classification using structural correspondence learning," in *ACL*, 2010, pp. 1118–1127.

[45] T. Zhou, S. J. Pan, I. Tsang, and Y. Yan, "Hybrid heterogeneous transfer learning through deep learning," in *AAAI*, 2014, pp. 2213–2220.

[46] S. R. El-Beltagy, M. E. Kalamawy, and A. B. Soliman, "Niletmrg at semeval-2017 task 4: Arabic sentiment analysis," in *SemEval2017*, 2017.

[47] J.-À. González, F. Pla, and L.-F. Hurtado, "Elirf-upv at semeval-2017 task 4: Sentiment analysis using deep learning," in *SemEval2017*, 2017, pp. 723–727.