SENTIMENT CLASSIFICATION USING AN ENSEMBLE SYSTEM PRESERVING THE PERFORMANCE BETWEEN DOMAINS

Anaid Villamizar
Lancaster University
a.villamizaralbornoz@lancaster.ac.uk

ABSTRACT

Machine learning paradigms and sentiment classification have become an attractive topic for academia, businesses and governments. In this paper, we propose a methodology using an ensemble system of convolutional neural networks (CNN) to classify sentiments of reviews from different domains, using only the new domain data to train the ensemble system. For this, CNNs are evaluated and some of them are re-trained using the weights from the previous domain as initial weights and using a specific proportion of the train dataset in each CNN. For the experiments, we used Amazon products reviews, movies (IMDb) reviews, restaurant reviews (Yelp) and Facebook comments. According to the experimental results, our approach shows that the performance of the previous domains is highly preserved.

1. INTRODUCTION

Ensemble systems have been successfully improving the performance of models, and confidence in the models' decisions. They have been applied to diverse problems such as: feature selection, missing feature, incremental learning among others. For example, incremental learning has been applied to real-world problems, such as when data is available in batches over a period of time and it needs to incorporate extra data into the knowledge base in an incremental way, avoiding access to previous data [1].

Incremental learning is a machine learning paradigm which is related to transfer, multitask, lifelong learning [2] and learning without forgetting. [4]. In general, examples of real world applications using these machine learning paradigms are: intelligent assistants, chatbots, and physical robots that interact with humans [3] and "require learning new visual capabilities while maintaining performance on existing ones". [4].

Lifelong learning allows the model to learn constantly, accumulate the knowledge learned in a previous domain, and use that knowledge to support any future learning. A specific application of lifelong learning is the sentiment classification, which has been studied in previous research projects, [3] and which can be very useful in the real world for marketing purposes and many others. However, lifelong learning needs to store some data to preserve performance on the previous domain [4].

In this paper, we propose a methodology using an ensemble system of convolutional neural networks (CNNs) to classify sentiments of reviews from different domains, where the knowledge is transferring from previous domains (keeping a similar performance in each domain). This uses only the new domain data to train the ensemble system, and the performance of the previous domains is highly preserved. With this methodology, the performance (accuracy) of the CNNs are evaluated in a new domain, and according to this evaluation some of the CNNs are selected to be re-trained:

- Using the weights from the previous domain as initial weights.

- Using a specific proportion of the train dataset in each CNN.

We performed the experiments using four datasets: Amazon products, movies (IMDb), restaurant reviews (Yelp) and Facebook comments. The label was a binary variable, 1 for positive sentiment and 0 for negative sentiment. We did 60 experiments and for each experiment we made some changes such as: the specific proportion of the training dataset in the CNNs, proportion of testing dataset, and the sequence of the domains.

This section will be followed by various sections:

Section 2: The Related Work.
Section 3: The methodology that we have proposed.
Section 4: The results and analysis of the experiments.
Section 5: Discussion about the results.
Section 6: Conclusion.

2. RELATED WORK

Ensemble systems have been used in many real-world applications to improve the performance of the models. In this section, we will describe researches that have been performed using ensemble systems, sentiment classification, and machine learning paradigms.

Adaptive mixtures of local experts. They use multiple networks, where each network learns to handle a subset of tasks, so the network becomes an expert on that subset. They use a weighted combination rule, where the weights are defined by a gating network. Therefore, the gating network decides the network that will work for each input [5]. However, all the data needs to be available with this approach, because the model needs to be retrained with new task [6]

Learning from Nonstationary Environments: Concept Drift. Ensemble systems have been used in incremental learning. An example of this is Learn++.NSE which was proposed for Nonstationary Environments: Concept Drift. Concept Drift is related to learning from a stream of data, where the data is available over a period of time and may be changed (cyclical or noncyclical, systematic or random, gradual or rapid and others). So, the learner should be able to learn, forget and remember important data according to the environment. The algorithm consists of training a new classifier for each new available data, and the ES makes the decision using a dynamically weighted majority voting. The voting weights are based on the classifiers' accuracy on current and past environments. If a classifier correctly predicts previously unknown instances, it receives a higher weight and if a classifier mis-classifies previously known data, it is penalized [1].

Additionally, ensemble systems have been used in combination of a lifelong learning. **Lifelong Machine Learning**, "considers systems that can learn many tasks over a lifetime from one or more domains. They efficiently and effectively retain the knowledge they have learned and use that knowledge to more efficiently and effectively learn new tasks" [7]. In the paper called **Expert Gate: Lifelong Learning with a Network of Experts,** the authors developed a lifelong learning system based on a Network of Experts that can deal with a sequence of new tasks without saving all previous data. They use different expert models that are responsible for different tasks and a gating mechanism which recognizes the task using an undercomplete autoencoder. The autoencoders are used to evaluate task relatedness at training time, to decide the most relevant prior model to be used for training a new expert, and if fine-tuning or learning without-forgetting can be selected. Moreover, the autoencoder uses a test sample to recognize the task and load the appropriate model automatically for the task. They evaluated their approach on image classification and video prediction

problems. This approach does not require storing all the training data like our methodology however it does need to capture the meta-knowledge of the task [6].

There is previous research about Lifelong Learning in sentiment classification of product reviews (positive and negative polarities). For instance, **Lifelong Learning for Sentiment Classification** [3] adopts a Bayesian optimization framework based on stochastic gradient descent. In this paper, they use a balanced distribution dataset from 20 types of diverse products or domains crawled from Amazon.com. The knowledge is incorporated using penalty terms to effectively exploit the knowledge gained from past learning. The components of this approach are: Past Information Stores, Knowledge Base, Knowledge Miner and Knowledge-Based Learner. In the Past Information Store, they do not store the original data from previous tasks, but they store: "a) Pˆ t(w|+) and Pˆ t(w|–) for each word which are from taskˆ t's NB classifier; and b) the number of times that word appears in a positive (+) document and the number of times that word appears in negative documents" [3]. In contrast, the methodology that we propose does not store information of the dataset, we only store the weights of the previous task to re-train the ensemble system.

There is earlier research about learning a new task or domain without using training data from previous task. For example, **Learning without forgetting** paper [4], its approach is to train the model using only new task data, preserving the previous capabilities without using the training data for the previous tasks. They use convolutional neural networks, "which can be seen as a hybrid of knowledge distillation and fine-tuning, learning parameters that are discriminative for the new task while preserving outputs for the original tasks on the training data". They evaluated their method on a diversity of image classification problems [4].

3. METHODOLOGY

We created an ensemble system using CNNs to classify sentiments of reviews from different domains which preserves a high percentage of knowledge from previous domains (without saving training data) and also learns about new domain. We consider sequential learning where domains data come one after another.

The ES consists of four CNNs, three of which have different architectures and the fourth (we will call "Powerful CNN") is a copy of one of the others (the best). The CNNs work for binary classification; 1 for positive sentiment and 0 for negative sentiment. The ES uses as an aggregation rule, the average of all votes. If the average of the votes is greater than 0.5, the prediction is 1, and if the average is lower than 0.5 the prediction is 0, and if the average is 0.5, the prediction of the ES uses a random tie breaking.

The first layer of all CNNs have an efficient embedding layer which maps the vocabulary indices into embedding dimensions. The vocabulary indices consist of 400,000 words [8]. The CNNs are based on [9], the differences between the CNNs are: number of convolutional layers, hidden layers and hyperparameters.

3.1 Description of the methodology

The aim of the methodology is to highly preserve the knowledge of previous domain and learn about the new domain using an ensemble system. The methodology uses only the new domain data to train the ES and the knowledge of the previous domains is highly preserved. Every time when the data from a new domain is available, the performance (accuracy) of the CNNs is evaluated and according to this evaluation and the "selection rules", some of the CNNs are selected to be re-trained in the new domain. The re-training of the selected CNNs is done using the weights from the previous domain (transferring weights) as initial weights, and a proportion of training data from the new domain. Therefore, the weights of the selected CNNs are changing with the new domain.

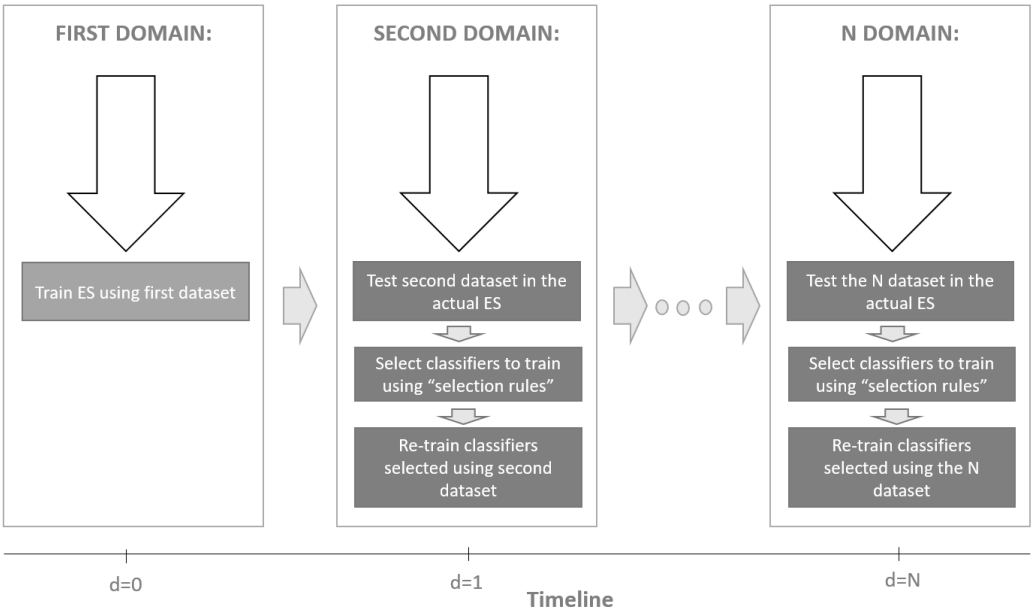

Figure. 1 Methodology

Figure 1 shows the methodology that we propose in this paper, the details of which are explained as following:

When the dataset from the first domain is available, it is split into a training set and a testing set. The training set is used to train the ES. At this point, all the CNNs are trained using the same training set and the weights are randomly set. After this, the training set is deleted. In this step, we call the ES, "ES-d0"

After that, the dataset from the second domain (new domain) is available. The dataset is split into a training set and a testing set. Then, the ES (ES-d0) and the CNNs are evaluated using the testing set (from the new domain) and according to the "selection rules" the CNNs are selected to be re-trained using the training dataset of the second domain (new domain). After this, the training set is deleted. This process is repeated for each new domain until n new domain. The "selection rules" take into consideration the testing accuracy of the ES and each CCN using the dataset of the new domain. During the selection of the CNNs, "Guilty CNN" and the "Second best CNN" are identified, which is going to help define how the CNNs will be trained. Each of them are described below:

- "Guilty CNN" has the lowest accuracy and it is going to be trained using a specific proportion of the training dataset. We will called this proportion: "proporGuilty".
- "Second Best CNN" has the second best accuracy and it is going to be trained using a specific proportion of the training dataset. We will called this proportion: "propor2ndBest".

There is a CNN called "Powerful CNN" in the ES, that is going to be trained every time when a new domain arrives, and it is going to be trained using a specific proportion of the training dataset. We will called this proportion: "proporPowerful".

"proporGuilty", "propor2ndBest" and "proporPowerful" are proportion of the training set and they should be defined before each experiment.

The “selection rules” algorithm is shown in Figure 2.

```
#Selection rules algorithm

if accuracyEnsembleSystem > percentage1 (79%)
   The classifier to be trained is the "Guilt CNN"
else:
   if accurracyOfCNNs < percentage2 (78%)
      The classifier to be trained is the "Guilt CNN"
      The classifier to be trained is the "Second Best CNN"
   else:
      A random classifier is trained using the "proporRandom"

Train "Powerful CNN"
```

Figure 2. Selection rules algorithm.

Figure 2: First, the testing accuracy (accuracyEnsembleSystem) of the ES is evaluated. If it is greater than percentage1 (79%), the “Guilty CNN” should be re-trained. If the accuracy of the ES is lower or equal to percentage1 (79%) and if one of the CNNs has an accuracy lower than percentage2 (78%), the “Guilty CNN” and the “Second Best CNN” should be re-trained. The “Powerful CNN” should always be trained.

3.2 Experiments

3.2.1 Datasets description

We performed the experiments using four datasets: Amazon products, movies (IMDb), restaurant reviews (Yelp) and Facebook comments. The reviews datasets were taken from UCI Machine Learning Repository which is used by the machine learning community [10]. The Facebook comments were taken from Center for Ultra-scale Computing and Information Security [11]. The characteristics of each dataset are explained as following:

- The datasets collected from UCI consist of 1000 reviews of products (cell phones and accessories category) sold on amazon.com, 1000 IMDb movie review and 1000 Restaurant reviews (Yelp). The label is a binary variable; 1 for positive sentiment and 0 for negative sentiment. Each type of reviews contains 50% of positives reviews and 50% of negatives reviews [10].
- The Facebook comments consist of 1000 comments. The labels are “P” for positive, “N” for negative and “O” for neutral sentiment. The dataset is imbalanced, because it is divided on 280 neutral comments, 641 positive comments, and 79 negative comments.

3.2.2 Pre-processing of the data

At the beginning, we had to make some changes to Facebook comments, because the characteristics of this dataset are different to UCI datasets. It was necessary to delete the neutral sentiments and the dataset was re-labelled: the positive reviews were labelled as 1 and the negative reviews as 0. After that, we pre-processed all the reviews and comments. We corrected the word spelling using spell library [12]. We used Natural Language Toolkit [13] to tokenize the comments and reviews. Then we convert each token using an index of words [8].

3.2.3 Description of experiments

Firstly, we trained and tested all the CNNs independently with each dataset before starting the experiments. The purpose of this was to know the performance of each CNN with each dataset. Then, we performed 60

experiments and we repeated each experiment 10 times. The duration of experiment and the 10 repetitions was around 2 hours. During the experiments, we made some changes on the following parameters: “proporGuilty”, “propor2ndBest”, “proporRandom” and “proporPowerful” which are the proportion of training set for the CNNs (Table 1). We also changed the sequence of the differences domains (Table 2) and changed the proportion of the testing set (0.1 and 0.2) of ES.

	proporGuilty	proporPowerful	proporRandom	propor2Best
1	0.7	0.8	0.6	0.5
2	0.5	0.8	0.6	0.7
3	0.7	0.8	0.5	0.5
4	0.8	0.9	0.5	0.7
5	0.6	0.8	0.7	0.6
6	0.6	0.8	0.6	0.7

Table 1. Proportions of training set using in the experiments

Difference sequences of datasets or domains			
D=0	D=1	D=2	D=3
amazon	yelp	imdb	facebook
yelp	amazon	imdb	facebook
imdb	yelp	amazon	facebook
amazon	facebook	imdb	yelp
amazon	imdb	facebook	yelp
facebook	amazon	imdb	yelp

Table 2. Difference sequences of datasets or domains during the experiments (D is domain)

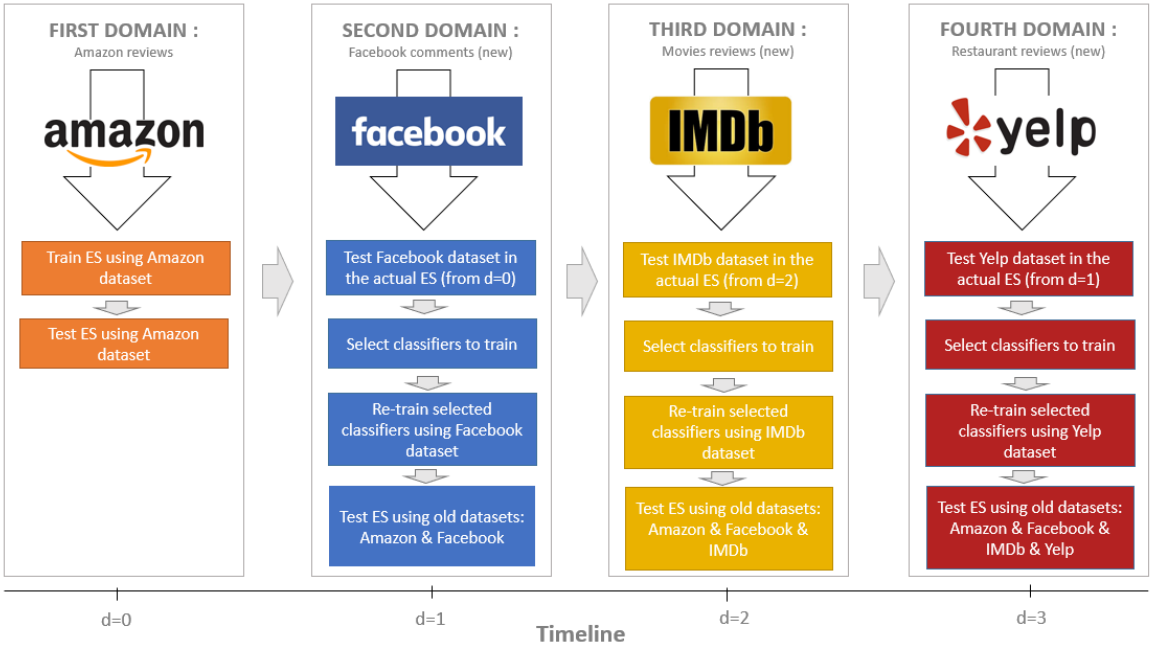

Figure 3. Example of experiment

Figure 3 shows an example of an experiment when the sequence of the domains was: Amazon, Facebook, IMDb and Yelp. The process followed during the experiment is described by domain below:

- **First domain:** The Amazon reviews dataset were split into a training set and a testing set. The ES was trained, all the CNNs were trained using the same training set and the weights are randomly set. After this, the training set was deleted. Then the ES was tested using Amazon reviews and the accuracy, precision and recall were saved as metrics of "old domain" to analyse the experiment.

- **Second domain:** The Facebook comments dataset were split into a training set and a testing set. The ES was tested using Facebook comments (accuracy, precision and recall were saved as metrics of "new datasets"). The CNNs are evaluated using the testing set and according to the "selection rules", the CNNs are selected to be re-trained using the Facebook training set. After this, the training set was deleted. Then the ES was tested using Amazon reviews and Facebook comments, and the accuracy, precision and recall were saved as metrics of "old domain" to analyse the experiment.

- **Third domain:** The IMDb reviews dataset was split into a training set and a testing set. The ES was tested using IMDb reviews (accuracy, precision and recall were saved as metrics of "new datasets"). The CNNs were evaluated using the testing set and according to the "selection rules" the CNNs are selected to be re-trained using the IMDb training dataset. After this, the training set was deleted. Then the ES was tested using Amazon reviews, Facebook comments, IMDb reviews, and the accuracy, precision and recall were saved as metrics of "old domain" to analyse the experiment.

- **Fourth domain:** The Yelp reviews dataset were split in training set and testing set. The ES was tested using Yelp reviews (accuracy, precision and recall were saved as metrics of "new datasets"). The CNNs are evaluated using the testing set and according to the "selection rules" the CNNs are selected to be re-trained using the Yelp training dataset. After this, the training set was deleted. Then the ES was tested using Amazon reviews, Facebook comments, IMDb reviews and Yelp reviews, and the accuracy, precision and recall were saved as metrics of "old domain" to analyse the experiment.

During the experiments, the accuracy, precision and recall were recorded during the testing for new domains and old domains. Afterwards, we did an exploratory analysis in R using line graphs and tables to look simple relationships between: sequence of domains and accuracy. Then, to evaluate the experiment, a multiple linear regression was used to analyse the data collected from the experiments and response variable (accuracy, recall and precision) using a critical value of p=0.05.

The form of the statistical model used is:

$$y_i \sim \beta_0 + \sum_j \beta_j X_{ij} + \epsilon$$

In the statistical model, y is the response variable, X_{ij} are the values explanatory variables. A Normal distribution was assumed. Then, a backwards elimination was carried out using Anova() to obtain the least significant variable at each stage. In each stage the least significant variable was removed, using a critical value of p=0.05.

4. RESULTS/ANALYSIS

We trained and tested all the CNNs independently with each dataset before starting the experiments. The purpose of this was to know the performance of each CNN with each dataset independently and the results are in Table 3. The table shows that the dataset with lowest accuracy are IMDb for all classifiers, while

Amazon and Yelp have a better performance and their accuracies are very similar. The Facebook dataset has the better performance, but this dataset has different characteristics compared to the other datasets (this dataset is imbalanced and very small).

As previously mentioned, the first layer of all CNNs have an efficient embedding layer which maps the vocabulary indices into embedding dimensions. We used vocabulary indices for these experiments which have 400,000 words because we need the CNNs to work for any dataset. However, some experiments were carried out using only the vocabulary of each dataset and the accuracy was higher, but for the purpose of this paper, this method was limited and not useful.

Dataset	Accuracy Classifier 1	Accuracy Classifier 2	Accuracy Classifier 3
Amazon	79.1	79.8	77
Facebook	88.8	88.8	88.8
IMDb	71.6	77.5	70.8
yelp	79.1	82.2	76.2

Table 3. Accuracies of each classifier with a test proportion of 0.1

Then we carried out the experiments of the ES using difference sequences of domains (Table 2). For this we divided the experiments into groups according to the parameters that we were testing (Table 4). Each group consist of 5 experiments.

# Group	propor2Best	proporGuilty	proporPowerful	proporRandom	test_size
1	0.5	0.7	0.8	0.6	0.1
2	0.7	0.5	0.8	0.6	0.1
3	0.5	0.7	0.8	0.5	0.1
4	0.7	0.8	0.9	0.5	0.1
5	0.6	0.6	0.8	0.7	0.1
6	0.7	0.6	0.8	0.6	0.1
7	0.5	0.7	0.8	0.6	0.2
8	0.7	0.5	0.8	0.6	0.2
9	0.5	0.7	0.8	0.5	0.2
10	0.7	0.8	0.9	0.5	0.2
11	0.6	0.6	0.8	0.7	0.2
12	0.7	0.6	0.8	0.6	0.2

Table 4. Groups of experiments

Table 4 shows that we created 6 groups of experiments for the proportion of the testing dataset (test size) 0.1 and 0.2. The results of all the experiments are in the Appendix. In this paper, we will show the better results that we found. In general, the results for groups that used test size 0.1 were more stable than groups with test size 0.2. The group that got better results were 1 and 6. The results for those groups are shown in the following figures:

dataset	yelp	imdb	facebook
Acc. new domain	71.7	69.3	61.1

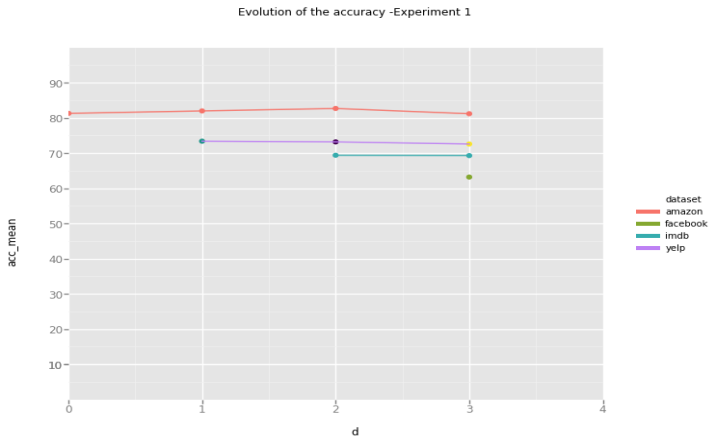

Figure 4. Group 1 - Experiment 1. Sequence of domains: ['amazon', 'yelp', 'imdb', 'facebook']

dataset	amazon	imdb	facebook
Acc. new domain	73.4	68.4	57.6

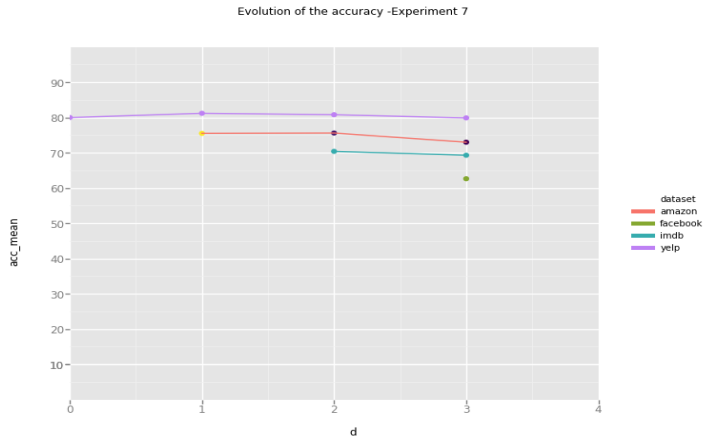

Figure 5. Group 1 - Experiment 7. Sequence of domains: ['yelp', 'amazon', 'imdb', 'facebook']

dataset	yelp	amazon	facebook
Acc. new domain	71	73.5	60.8

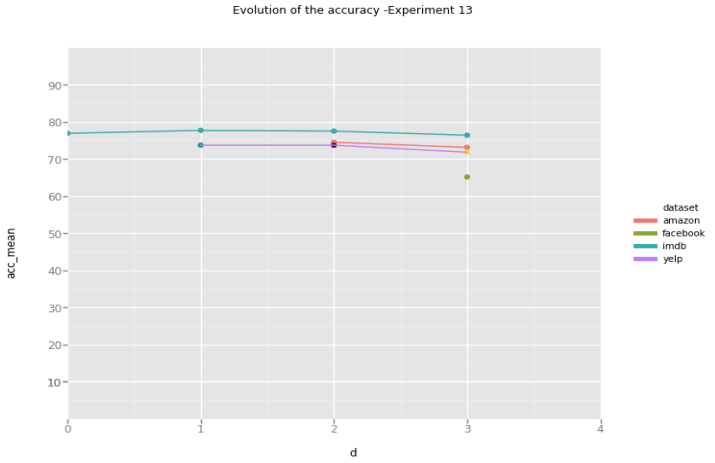

Figure 6. Group 1 – Experiment 13. Sequence of domains: ['imdb', 'yelp', 'amazon', 'facebook']

dataset	facebook	imdb	yelp
Acc. new domain	62.9	64.4	72.3

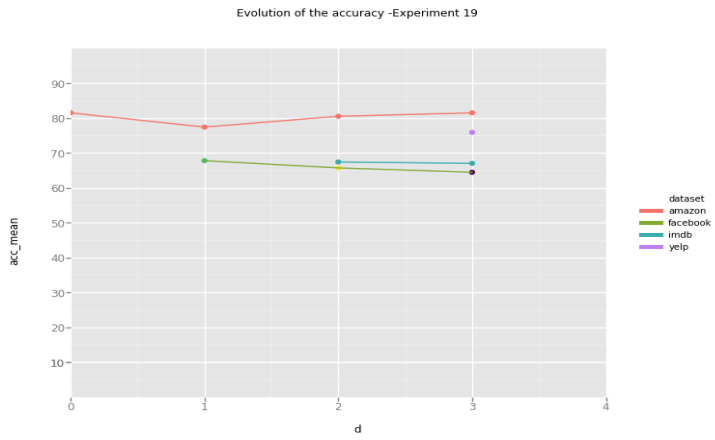

Figure 7. Group 1 – Experiment 19. Sequence of domains: ['amazon', 'facebook', 'imdb', 'yelp']

dataset	imdb	facebook	yelp
Acc. new domain	67.8	61.6	74.2

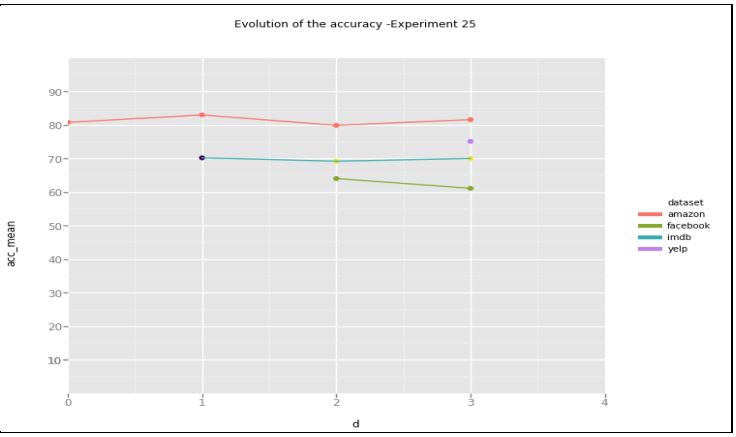

Figure 8. Group 1 – Experiment 25. Sequence of domains: ['amazon', 'imdb', 'facebook', 'yelp']

dataset	yelp	imdb	facebook
Acc. new domain	74.2	67.9	66.11

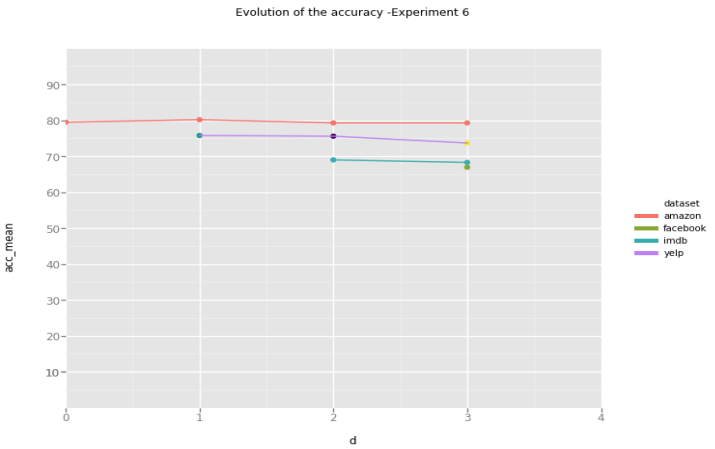

Figure 9. Group 6 – Experiment 6. Sequence of domains: ['amazon', 'yelp', 'imdb', 'facebook']

dataset	amazon	imdb	facebook
Acc. new domain	71.1	70.9	59.7

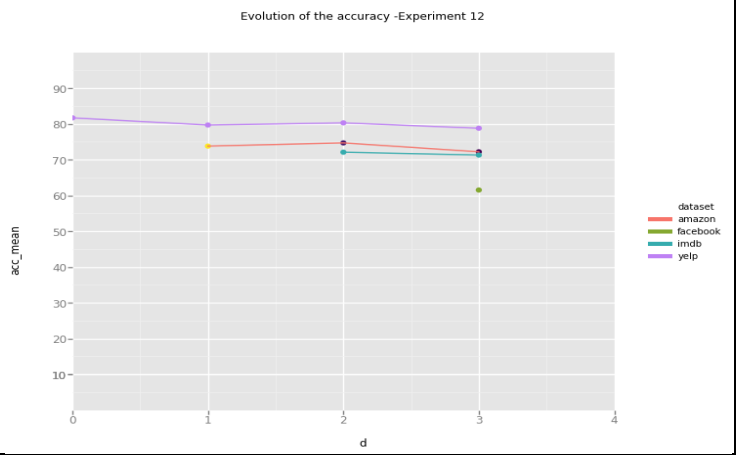

Figure 10. Group 6 – Experiment 12. Sequence of domains: ['yelp', 'amazon', 'imdb', 'facebook']

dataset	yelp	amazon	facebook
Acc. new domain	70.6	74.4	56.2

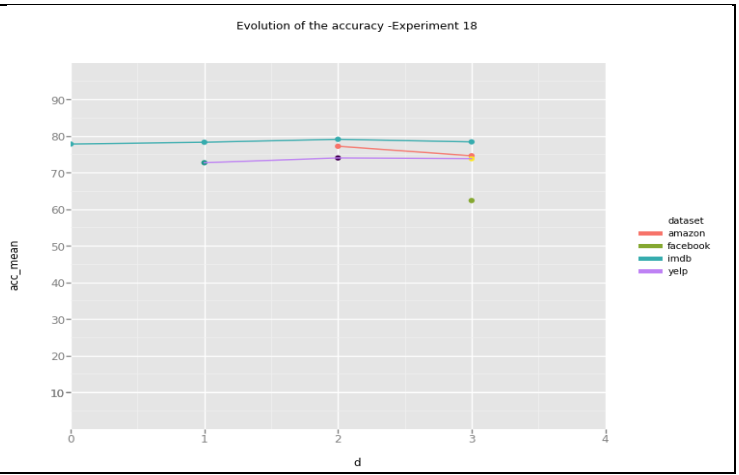

Figure 11. Group 6 – Experiment 18. Sequence of domains: ['imdb', 'yelp', 'amazon', 'facebook']

dataset	facebook	imdb	yelp
Acc. new domain	62.2	68.1	72.9

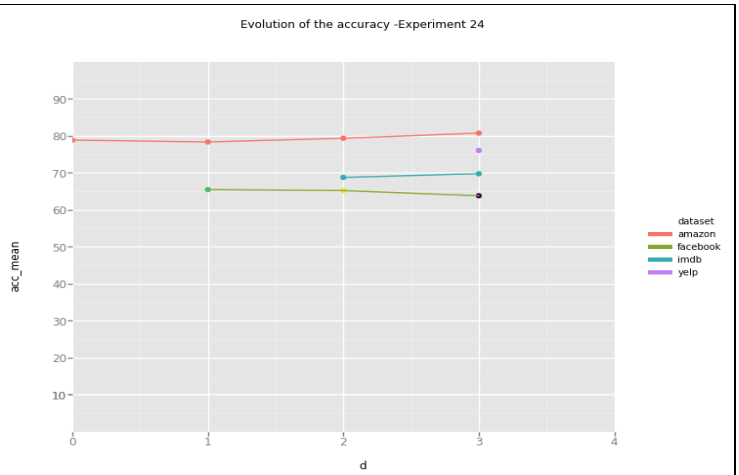

Figure 12. Group 6 – Experiment 24. Sequence of domains: ['amazon', 'facebook', 'imdb', 'yelp']

dataset	imdb	facebook	yelp
Acc. new domain	64.3	64.4	74.2

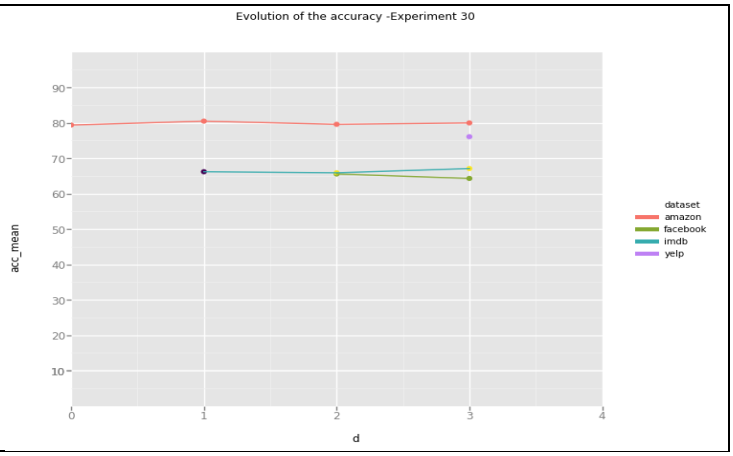

Figure 13. Group 6 – Experiment 30. Sequence of domains: ['amazon', 'imdb', 'facebook', 'yelp']

Comparing Group 1 with other groups, it can be seen that the accuracy is very stable when the ES is re-trained using other datasets or domains. Moreover, the ES improves the accuracy when the new domain is converted to the old domain. In other words, when the ES is re-trained using the new dataset, the accuracy improves by between 1% and 6%. However, this group is affected by Facebook dataset; the plots show that after re-training the ES, there is a decline in the accuracy that is recovered when the ES is re-trained using the next dataset. Although, for the Facebook domain, the accuracy tends to decrease.

Group 6 is the group of experiments that keeps the steadier accuracy after each re-training. This group is less affected by the Facebook dataset. However, if we compare the accuracy before re-training and after re-training a domain, the accuracy increases very slightly by between 0.5% and 4%. Therefore, the ES learns less.

In both groups, it can be observed that datasets like Amazon, IMDb and Yelp have a similar accuracy when we compare to the accuracy of each classifiers (Table 3). Conversely, the accuracy of Facebook is very different when we compare with Table 3.

4.1 Results of the statistical evaluation

To evaluate the experiments, a multiple linear regression was performed to analyse the data collected from the experiments using a critical value of p=0.05. The results of all models created are in the Appendix.

4.1.1 Accuracy for the old domains

We created a multiple linear regression model where the variables are in Table 5.

Explanatory variables	proporGuilty, propor2ndBest, proporRandom, proporPowerful, dataset, proportion of the test dataset, sequence of the domains
Response variable	accuracy

Table 5. Explanatory and response variable

We found that the significant (p<0.05) explanatory variables are: datasets, proportion of the test size and sequence of the domains. The model shows that the estimated regression coefficient of proportion of the test dataset is -16.3 which indicates that if all other independent variables are held constant, then a decrease of one on the proportion of the test dataset is associated with a decrease of 16.3 accuracy. Afterwards, a backwards elimination was done using Anova(), and after proporGuilty, propor2ndBest, proporRandom and

proporPowerful were removed, dataset, proportion of the test dataset and sequence of the domains were significant again.

4.1.2 Precision for the old domains

For precision, we created a multiple linear regression model similar to the accuracy model but using the precision as a response variable. We found that the significant ($p<0.05$) explanatory variables are: datasets, proportion of the test dataset and sequence of the domains. The model indicates that the estimated regression coefficient of proportion of the test dataset is -25.4 which indicates that if all other independent variables are held constant, then a decrease of one in the proportion of the test dataset is associated with a decrease of 25.4 precision. Then, a backwards elimination was carried out using Anova(), and after proporGuilty, propor2ndBest, proporRandom and proporPowerful were removed. Dataset, proportion of the test dataset and sequence of the domains were significant again.

4.1.3 Recall for the old domains

For the metric recall, we created a multiple linear regression model similar to the precision model but using the recall as a response variable. We found that the significant ($p<0.05$) explanatory variables are: datasets and the sequence of the domains. Then, a backwards elimination was carried out using Anova(), and after the proportion of the test size , proporGuilty, propor2ndBest, proporRandom, and proporPowerful were removed. The dataset and the sequence of the domains were significant again.

5 DISCUSSIONS

The experiments show that the performance of the previous domain is highly preserved. However, the performance of the ES was affected by the Facebook dataset when it was learning, and it is less affected when the difference between the accuracy before re-training and after re-training a domain is small. The reason for this is that the Facebook dataset has other characteristics; it is an imbalanced and a small dataset. Previous work like Lifelong Learning for Sentiment Classification [2] was performed using balanced datasets, and the conditions in this paper were different.

Another outcome of this study indicates that the dataset, proportion of the test dataset, and sequence of the domains for testing accuracy and precision are the most significant according to multiple linear regression models. This means that there is a big impact for those metrics.

6 CONCLUSION AND FUTURE WORK

In this paper, we proposed a methodology using an ensemble system of convolutional neural networks (CNNs) to classify sentiments of reviews from different domains, where the knowledge is transferring from previous domains (keeping a similar performance in each domain). This was done transferring weights of some CNNs from a previous domain and using a specific proportion of the training dataset in each CNN. We demonstrated that the performance in each domain was very stable and highly preserved.

However, we used balanced and imbalanced datasets that affected the performance. To get results comparable to other researches, the methodology proposed should use balanced datasets or the same dataset as Lifelong Learning for Sentiment Classification paper or similar research. However, sometimes it is very difficult to get datasets from previous research because they are not available anymore.

Additionally, the ES could be improved by assigning the proporGuilty, propor2ndBest, proporRandom, and proporPowerful dynamically during the learning process.

7 REFERENCES

[1] Polikar, R. (2012) Ensemble Learning. doi: 10.1007/978-1-4419-9326-7.

[2] Chen, Z. and Liu, B. (2016) 'Lifelong Machine Learning'.

[3] Chen, Z., Ma, N. and Liu, B. (2013) 'Lifelong Learning for Sentiment Classification'.

[4] Li, Z. and Hoiem, D. (2016) 'Learning without Forgetting', pp. 1–13.

[5] Jacobs, R. A. et al. (1991) 'Adaptive Mixtures of Local Experts', (February). doi: 10.1162/neco.1991.3.1.79.

[6] Aljundi, R. (no date) 'Expert Gate: Lifelong Learning with a Network of Experts'.

[7] Silver, D. L. (2013) 'Lifelong Machine Learning Systems : Beyond Learning Algorithms Prior Work on LML', (Solomonoff 1989), pp. 49–55.

[8] Perform sentiment analysis with LSTMs, using TensorFlow. (2017)

https://www.oreilly.com/learning/perform-sentiment-analysis-with-lstms-using-tensorflow

[9] Keras examples. https://github.com/keras-team/keras/blob/master/examples/imdb_cnn.py

[10] Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[11] Social Media Data for Sentiment Analysis. Center for Ultra-scale Computing and Information Security. http://cucis.ece.northwestern.edu/projects/Social/sentiment_data.html

[12] Python 3 Spelling Corrector. https://pypi.org/project/autocorrect/

[13] Natural Language Toolkit. https://www.nltk.org/#natural-language-toolkit

ACKNOWLEDGEMENTS

We thank Leandro Soriano for helpful discussions.