Transductive transfer learning based Genetic Programming for balanced and unbalanced document classification using different types of features

Project report submitted in partial fulfilment of the requirement for the

degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

ABHIN SHARMA(191446)

Under the supervision of
Dr. Rakesh kanji, Assistant Professor (SG)

To

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Certificate

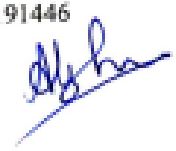
Candidate's Declaration

I hereby declare that the work presented in this report entitled "**Transductive transfer learning based Genetic Programming for balanced and unbalanced document classification using different types of features**" in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2023 to may 2023 under the supervision of Dr rakesh kanji, Assistant Professor (SG), Department of Computer Science and Engineering.

I also authenticate that I have carried out the above mentioned project work under the proficiency stream .....

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Abhin Sharma 191446



This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Rakesh kanji
Assistant Professor (SG)
Department of Computer Science and Engineering
Dated: 8th , May 2023

I

PLAGIARISM CERTIFICATE

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 10-05-2023

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: Rohini Sharma Department: CSE Enrolment No 19446

Contact No. 6230142186 E-mail: rohini@jiit.ac.in

Name of the Supervisor: Dr. Rakesh Kanji

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letter): TRANSDUCTIVE TRANSFER LEARNING BASED GENETIC PROGRAMMING FOR BALANCED AND UNBALANCED DATASET

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 55
- Total No. of Preliminary pages = 10
- Total No. of pages accommodate bibliography/references = 2

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found Similarity Index at 8 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Rakesh Kanji

(Signature of Guide/Supervisor)

(Signature of HOD)

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.jiit@gmail.com

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor Dr. Rakesh Kanji, Associate Professor (SG), Department of CSE Jaypee University of Information Technology,Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of "Computer Science" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Rakesh kanji, Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Abhin Sharma 191446

TABLE OF CONTENT

Content **Page No.**

Chapter 4: PERFORMANCE ANALYSIS

Chapter 5: CONCLUSIONS

List of Abbreviations

No.	Abbreviation	Full Form
1.	GP	genetic programming
2.	TF	Text frequency
3.	IDF	inverse document frequency
4.	NLP	Natural language processing
5.	SD	Standard deviation

List of Figures

List of Tables

ABSTRACT

Document classification is one of the predominant tasks in Natural Language Processing. However, some document classification tasks do not have ground truth while other similar datasets may have ground truth. Transfer learning can utilise similar datasets with ground truth to train effective classifiers on the dataset without ground truth. This paper introduces a transductive transfer learning method for document classification using two different text feature representations—the term frequency (TF) and the semantic feature doc2vec. It has three main contributions. First, it enables the sharing of knowledge in a dataset using TF and a dataset using doc2vec in transductive transfer learning for performance improvement. Second, it demonstrates that the partially learned programs from TFs and from doc2vecs can be alternatively used to ‘‘label then learn’’ and they improve each other. Lastly, it addresses the unbalanced dataset problem by considering the unbalanced distributions on categories for evolving proper Genetic Programming (GP) programs on the target domains. Our experimental results on two popular document datasets show that the proposed technique effectively transfers knowledge from the GP programs evolved from the source domains to the new GP programs on the target domains using TF or doc2vec. There are obviously more than 10 percentages improvements achieved by the GP programs evolved by the proposed method over the GP programs directly evolved from the source domains. Also, the proposed technique effectively utilises GP programs evolved from unbalanced datasets (on the source and target domains) to evolve new GP programs on the target domains, which balances predictions on different categories..

CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION:

Transfer learning is the process of reusing previously learnt problem models in machine learning. In a learning transition, machines use expertise acquired from prior tasks to boost predictions for future ones.

Move knowledge is a type of predictive technologies.starts with pre-existing models to perform new tasks. Models are first trained on big datasets, which are frequently used for diverse but related tasks. By utilising learning learnt from prior training models, transfer learning can reduce the amount of knowledge and learning time necessary to tackle new tasks.

The pre-trained model in transfer learning is often a deep neural network that has learned to recognise complicated patterns in data. Using a supervised learning approach, the model is trained on a large dataset, often including millions of samples. The generated model can then be utilised to solve a new, related job, such as image classification or natural language processing.

Because of its ability to leverage knowledge learned from pre-trained models and achieve good performance with less data and training time, transfer learning has become a popular technique in many areas of machine learning, including computer vision, speech recognition, and natural language processing.

Transfer learning includes striving using what we've obtained in one task to better understand the concepts in another. Weights automatically shift from a network conducting "task A" to a network performing "task B." Transfer learning is frequently employed in computer vision and natural language processing tasks such as sentiment estimation due to the enormous amount of CPU power required.

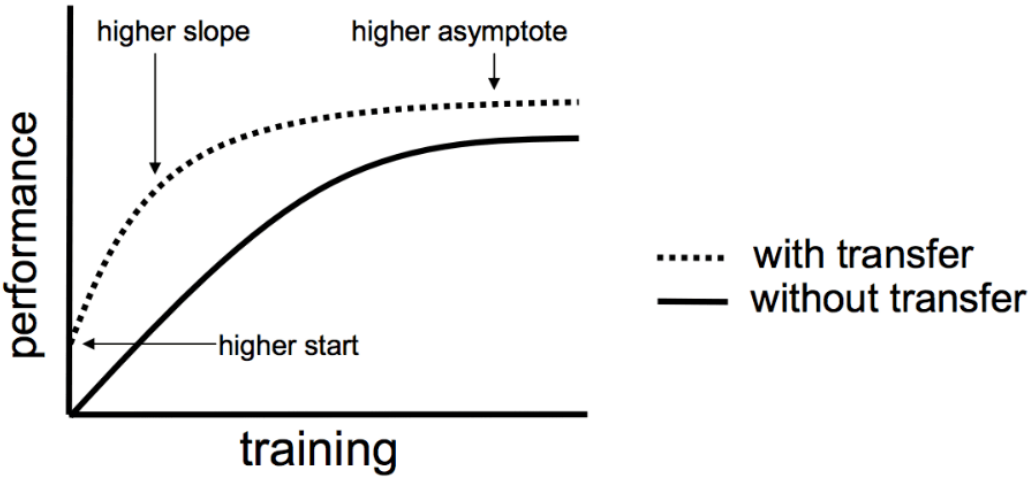


fig1- how transfer learning improves learning

NLP stands for Natural Language Processing, which is a subfield of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language. NLP involves techniques such as machine learning, deep learning, and linguistics to analyze, manipulate, and model natural language data. Some applications of NLP include language translation, sentiment analysis, text classification, speech recognition, and chatbots.

Transfer learning is a powerful technique that can be used to solve problems with limited data availability. Transfer learning can be applied by choosing a related task with a large amount of data and constructing a model for that activity. This previously trained model can then be repurposed and fine-tuned for the particular challenge at hand. Alternatively, you might be lucky enough to come across a pre-existing pre-trained model that you can use as a starting point for developing your own system.

By using transfer learning, it is possible to develop highly accurate models even when data is scarce. This is achieved by transferring the knowledge and insights from a related task with abundant data to the new problem domain. Ultimately, transfer learning can be an effective way to solve complex problems that may have been impossible to tackle otherwise.

Transfer learning is a technique that involves applying knowledge learned from solving one problem to a different but related one. This method varies from typical machine learning addresses in that it takes into account similarities

between the source and target domains.Transfer instruction is a technique that requires applying knowledge obtained from solving a given issue to a different but related one. This method varies from typical machine learning addresses in that it takes into account similarities among the source and target domains.Unsupervised transfer learning, inductive transfer learning, and transductive transfer learning are the three primary types of transfer learning methods depending on the availability of labelled data in the source and target domains. When there is no labelled data in the original or target domains, uncontrolled transfer learning is performed. When the original domain uses labelled data but the target domain does not, inductive transfer learning is employed. When both the initial and target fields have labelled data, yet the data are not identical, transductive transfer training is applied.

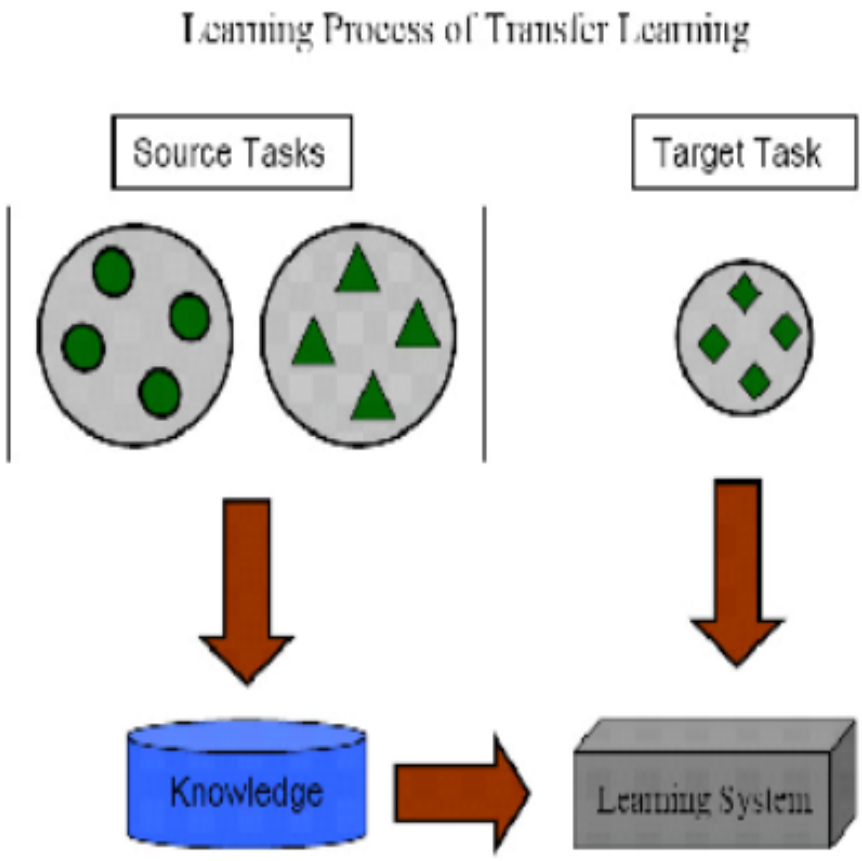


fig2- how transfer learning works

TYPES OF TRANSFER LEARNING:

A.) INDUCTIVE TRANSFER LEARNING:

The inducance transference learning is an instance in which a model is trained on a labelled source domain and then retrained on a labelled target domain. This method enables the model to learn features that are crucial to both the target and source areas, resulting in enhanced efficiency on the target domain.Inductive transfer learning entails fine-tuning the pre-trained model on the new task with labelled data from the target domain. The last layer of the pre-trained model may be changed throughout the fine-tuning process, or some of the layers may be frozen and just the weights of the remaining layers are updated.Inductive transfer learning can be particularly effective when the source and target domains have similar feature representations and data distributions. By leveraging the knowledge learned from the source domain, the model can generalize better to the target domain, even when the amount of labeled data in the target domain is limited.Inductive transfer learning can be demonstrated by utilising a pre-trained image classification model to recognise objects or a pre-trained natural language processing model to perform sentiment analysis on a new dataset. Overall, when labelled data is scarce in the target domain, inductive transfer learning is an effective strategy for enhancing model performance.

B.TRAINING THROUGH TRANSDUCTIVE CHANGE:

A kind of learning transferred methodology .in which On whom a predictive algorithm is taught labelled data in an area of birth and then applied to labelled data in a target domain, the spread of information in the desired field may differ from that in the desired field.Transductive transfer learning, as opposed to inductive transfer learning, which requires fine-tuning the pre-trained model on the new task, only employs the pre-trained model to produce predictions on the target domain. This is due to the fact that the labelled data in the target domain is already available, eliminating the requirement to retrain the model.The key

concern with transductive transfer learning is that the source and target domains may have different feature representations or data distributions, resulting in poor performance. To address this, several adaption approaches such as instance re-weighting, feature adaptation, and domain adaptation may be used.Using a pre-trained model for image classification to predict the properties of a new picture dataset is an example of transductive transfer learning, as is using a pre-trained natural language processing model to identify the sentiment of fresh text data.

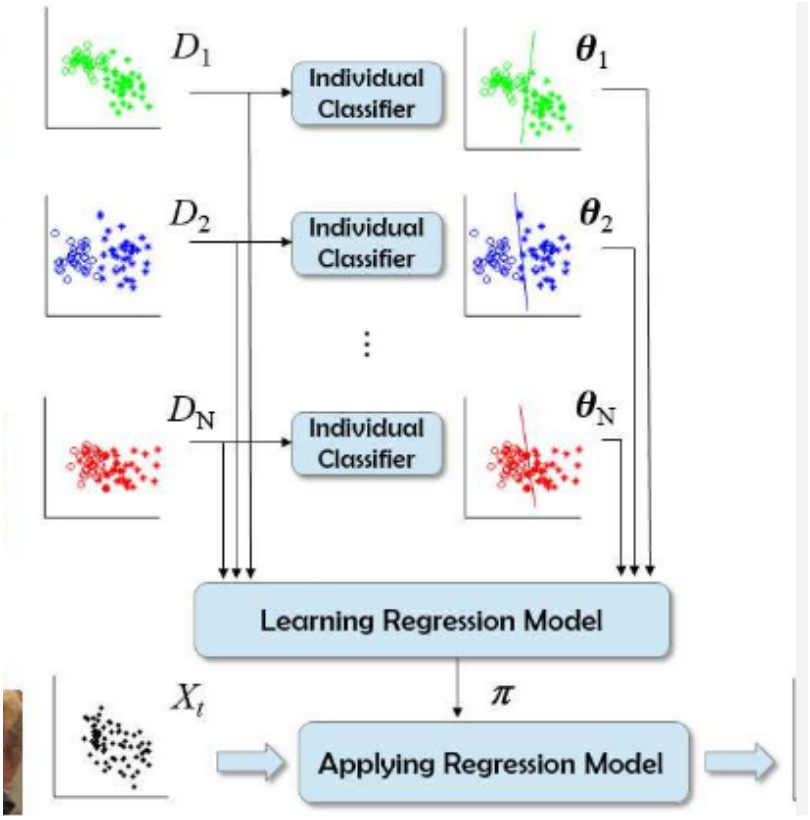


Fig3- visual representation of transductive transfer learning

C.) UNSUPERVISED TRANSFER LEARNING:

Unsupervised transfer teaching is an instance of send method of learning that entails training a model on unstructured information from a source domain and then modifying it to untreated data from a target domain. The aim is to make use

of the expertise obtained in the source domain to enhance the model's performance in the target field, even when labelled data is missing in either area.The main approach in unsupervised transfer learning is to learn a representation of the data that is transferable between domains. This is typically done using unsupervised learning techniques such as autoencoders or generative models, which can learn meaningful representations of the data without requiring labels.Unsupervised transfer learning has applications in a wide range of fields, including computer vision, natural language processing, and speech recognition. For example, by fine-tuning the model using unsupervised techniques, a pre-trained model for image classification on one dataset can be transferred to a new dataset without labelled data. Similarly, with limited labelled data, a pre-trained language model can be fine-tuned on a new domain.Overall, unsupervised transfer learning is a powerful technique for transferring knowledge from one domain to another, even when labelled data is few or unavailable.

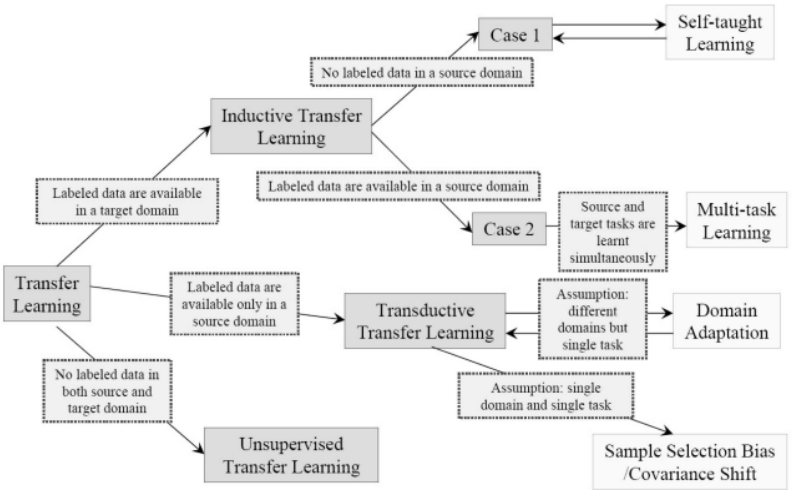


Fig4- types of transfer learning

GENETIC PROGRAMMING:

Genetic programming is a machine learning technique inspired by the process of natural selection and evolution. It is a form of evolutionary computation that aims to create computer programs that can solve a given problem.

It is based on the notion that a population of programmes can evolve through natural selection in a manner similar to how biological species evolve over time. In GP, a population of computer programmes is randomly initialised, and the programmes are assessed based on how successfully they address the problem at hand. Successful programmes are chosen to "reproduce" by merging their genetic material via crossover and mutation operations. This results in a new generation of programmes that are better suited to the challenge than the prior generation.The evolutionary process continues for several generations, with the finest programmes from one generation passing on to the next. Over time, the population converges on a set of programmes that satisfactorily tackle the problem at hand. The evolutionary process continues for several generations, with the finest programmes from one generation passing on to the next. Over time, the population merges on a set of programmes that satisfactorily tackle the problem at hand.

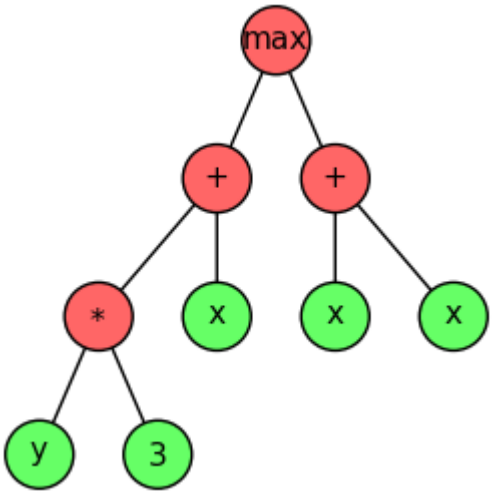


fig5- genetic programming-DEEP

1.2 PROBLEM STATEMENT

Transfer learning provides several advantages when building machine learning models. Among its key benefits are resource savings and increased efficiency while developing new models. Furthermore, because a significant portion of the model has already been pre-trained, it can be useful when training models with unlabeled datasets.

Transfer learning for machine learning has the following primary advantages:
removing the requirement for each new model to have a significant collection of labelled training data.
enhancing the deployment and development of machine learning for a variety of models.a broader strategy for computer problem solving that uses many techniques to address new problems.Instead of training in real-world settings, models can be trained in simulations.

Transfer learning for the study of NLP:

Natural language processing (NLP) is a critical technology that allows machines too understtand when analyze mankind dialogues, whether in's spoken or written. NLP plays a significant role in improving human-system interaction, and it has many practical applications, such as voice assistants, speech recognition software, automatic captioning, translations, and language contextualization tools. In addition, transfer learning can be employed to adapt models to different languages. By fine-tuning components of a model that have been trained on English language data, it's possible to create effective models for other languages or tasks. Since digital English language resources are widely available, NLP models can be trained on a large dataset and then transferred to a new language model with similar features or functions to improve its performance.

Another application of transfer learning is adapting models to different languages. Components of models that have been trained and refined using the English language can be repurposed for similar languages or tasks. Because digital resources for the English language are widely available, models can be

trained on extensive datasets before transferring components to a model for a new language.Transfer learning has various uses in improving machine learning models that involve natural language processing. For example, models can be trained to recognize different aspects of language simultaneously, or pre-trained layers that understand specific vocabularies or dialects can be incorporated.

Here are some other benefits knowledge transfer：

The potential of machiine learnning is dependent if different organisations and enterprises having extensive access to sophisticated models. Machine learning must be accessible and adaptive to the particular local needs and requirements of organisations in order to revolutionise businesses and processes. Only a small percentage of businesses will have the capacity to sort information and build an algorithm.

A major challenge facing supervised machine learning is the need for large amounts of labeled data. Labeling data can be a time-consuming task, particularly when dealing with big data. The need for enormous quantities of labeled data makes it difficult to develop the most powerful models on a large scale. It is likely that the development of these algorithms will be concentrated in organizations that have access to and the resources necessary for generating large amounts of labeled data.

Keyword extraction is one of the best methods for removing significant information from unstructured text. The amount of time that organisations must spend analysing data may be considerably reduced. It might help in gathering company intelligence, analysing consumer comments, keeping an eye on social media, and enhancing customer service. RAKE breaks down a document's content into a list of candidate keywords before starting to extract keywords from it. The text of the document is initially divided inside an array of sentences using

they designated them comes word delimiter.phrase delimiters and stop words are used to separate this array into groups of connected words. A candidate keyword is a group of words that are assigned the same place in a textual sequence.

1.3 OBJECTIVES

The primary objective of the research is to develop a transductive transfer learning-based GP which is a profitable (genetic programming) function that allows multiple feature extraction algorithms to share knowledge for document classification in a highly efficient manner. The paper employs TF (term frequency) and doc2vec (a word embedding-based feature) for this purpose. To achieve this goal, GP TF-based programs developed in source domains are directly applied to target domains. In contrast to current transfer learning methods, the newly proposed GP system is designed to transfer knowledge between GP-evolved programs based on both TF and doc2vec techniques.Furthermore, this novel approach has been applied to unbalanced datasets.

We'd want to look at the subsequent study goals:

The study's main goal is to answer two key questions: first, whether GP programmmes developed with TF in one domain can be effectively discussed in order to develop models with doc2vec in another domain, and second, whether GP-evolved projects established with doc2vec in one field may be utilised to develop effective techniques in TF in the same field.

Inductive transfer learning utilizes the same beginning and ending points as the source task but performs different tasks within them, like distinguishing between animals and mammals in an image classification scenario. The target domain for this approach involves labeled training examples. Self-taught learning is an inductive transfer learning algorithm that uses unlabeled images from the source domain to train effective classifiers for the target images. On the other hand,

transductive transfer learning is utilized in scenarios where there is a lack of labeled data in the target domain, as seen in unsupervised domain adaptation. This technique is helpful when obtaining labels for training data is challenging, but there exist similar labeled training data that can be employed for knowledge transfer to the target domain.

GP programs evolved from different runs are typically unique due to the stochastic search process and the existence of multiple optimal solutions. GPTF-based programs are flexible and can be applied directly to the target domain,while word embedding-based features are effective for document prediction.Combining the knowledge from both models can enhance the performance of document classification. Additionally, there are no reports of GP transfer learning algorithms being used for unbalanced document classification problems.Therefore, it is important to investigate how to effectively transfer GP program se volved from unbalanced datasets in the source domain to unbalanced datasets in the target domain. Transductive transfer learning is a type of transfer learning where the objective is to improve the performance of a machine learning model.

To reduce the amount of labeled data needed to achieve high performance on a target task by leveraging labeled data from related but different source domains.To improve the generalization ability of a machine learning model by transferring knowledge from a related but different domain.Overall, the goal of transductive transfer learning is to make machine learning models more efficient, robust, and adaptable by transferring knowledge learned from related but different domains.Genetic programming is a machine learning technique that uses a population-based approach to evolve computer programs that solve a given problem. To automatically generate computer programs that can solve complex problems without human intervention.To optimize the performance of computer programs by iteratively evolving better versions of them over time.Overall, the main goal of genetic programming is to develop intelligent systems that can solve complex problems in a more efficient and effective way than traditional approaches, by leveraging the power of evolution and natural selection.To find novel and creative solutions to problems that may not be obvious or easily solvable by human experts text classification is a supervised machine learning technique that involves assigning predefined categories or

labels to text documents based on their content,To automate the process of classifying large volumes of unstructured text data into meaningful categories or labels, which can help in information retrieval and analysis.To improve the accuracy and efficiency of text processing and analysis by automating the classification of text data, which can be time-consuming and error-prone if done manually.

1.4 METHODOLOGY:

This section provides an overview of the theoretical framework that underp ins the thesis. It is divided into eight subsections: natural language processing, data cleaning and preprocessing, text representation, distance measurement, deep learning and NLP, K-mean, keyword extraction, and summarisation. Each subsection introduces fundamental concepts and theories that are essential to understanding the topic.

1.4.1 Natural language processing:

NLP involves creating algorithms and technologies to enable computers to comprehend, interpret, and generate human language. This involves various techniques such as text analysis, sentiment analysis, language translation, and speech recognition. The applications of NLP include virtual assistants, chatbots, text summarization, and language translation tools. The ultimate aim of NLP is to help computers understand both written and spoken human language by converting it into numerical computational input. NLP encompasses different statistical theories, algorithms, and techniques that allow computers to extract, categorize, label, and comprehend human language. Preparing text data for analysis through data cleaning and preprocessing is a critical step in NLP. The accuracy and efficiency of NLP models can be significantly impacted by the quality of the data and how well it is preprocessed. These techniques are aimed at standardizing and cleaning text data to make it easier to analyze and enhance the accuracy of NLP models. However, specific techniques employed are often

domain and application-specific, and there is usually a trade-off between computational complexity and accuracy.

1.4.2 Data Cleaning and Pre-processing

Data cleaning and preprocessing are crucial steps in natural language processing (NLP) that involve preparing text data for analysis. The quality of the data and how well it is preprocessed can significantly impact the accuracy and effectiveness of NLP models. These steps involve removing irrelevant information such as stop words, special characters, and punctuation, correcting spelling and grammar errors, and converting the text to lowercase. Tokenisation, which includes breaking down the text into individual words or phrases, and stemmming or lemmmatization, which involves reducing words to their basic form t, are two more approaches to preprocesssing reduce the complexity of the data. Overall, these techniques are aimed at cleaning and standardizing text data to enable easier analysis and improve the accuracy of NLP models. However, it's important to keep in mind that the specific techniques used will depend on the specific application and domain, and that there is often a trade-off between accuracy and computational complexity.

1.4.3 Transfer learning:

Transfer learning is a machine learning technique that allows a model trained on one task to be reused or adapted on a different task. This approach is particularly useful when the new task has limited labeled data, as it leverages the knowledge gained from the original task. Transfer learning has many applications in natural language processing, computer vision, and speech recognition, among others. In language modeling, pre-trained models can be used as a starting point to fine-tune on a new dataset or to extract features for downstream tasks. Transfer learning has the potential to reduce the amount of labeled data needed to train effective models, making it a valuable tool for machine learning practitioners.

However, it is important to carefully select the source task and ensure that the features learned in the pre-training stage are relevant to the target task.

In transductive transfer learning, the goal is to transfer knowledge from a source domain to a target domain where the target domain does not have labeled data. This is especially useful when it is difficult or expensive to obtain labeled data for the target domain, but there is labeled data available for a similar source domain. By leveraging the knowledge learned from the labeled source domain, the model can improve its performance on the target domain.

1.4.4 Document classification:

The methodology of document classification involves several steps, including data collection and preprocessing, feature extraction, model selection and training, and evaluation.Firstly, the data collection process involves gathering a large dataset of documents to be classified. The dataset may contain text from various sources, such as articles, research papers, news articles, and social media posts.Once the data has been collected, it must be preprocessed to prepare it for analysis. This step typically involves removing irrelevant information, such as stop words and punctuation, and converting the text to a standard format, such as lowercase.The next step is feature extraction, where relevant features are identified and extracted from the preprocessed text data. Common techniques used for feature extraction include bag-of-words, TF-IDF, and word embeddings.Once the features have been extracted, a machine learning model is selected and trained using the labelled data. Common models used for document classification include Naive Bayes, SVM, and neural networks.Finally, the performance of the model is evaluated using metrics such as accuracy, precision, recall, and F1-score. The evaluation process helps identify any weaknesses in the model and suggests ways to improve its performance.Finally, the model can be used to classify new, unseen documents. The text is preprocessed, represented numerically, and then fed into the trained machine learning algorithm. The algorithm predicts the category of the document based on the learned features and the model's classification rules.

1.4.5 TF-IDF:

Term Frequency-Inverse Document Frequency is abbreviated as TF-IDF. It is a method used in information retrieval and text mining to assess the significance of a word or phrase inside a document or corpus. The frequency with which a term appears in a document is measured by TF. It is determined by dividing the total number of terms in the document by the number of occurrences of a term.IDFdetermines how uncommon a term is across the entire corpus. It is calculated by dividing the total number of documents in the corpus by the number of documents containing the phrase by the logarithm of that number. The TF-IDFscore for a phrase is calculated by multiplying its TF and IDF values. Thegreatera term's TF-IDF score, the Text classification, information retrieval, and keyword extraction are just a few of the natural language processing tasks thatTF-IDF may help with. It is also a frequent search engine strategy for ranking search results depending on their relevance to a query.TF-IDF (Term Frequency-Inverse Document Frequency) is a widely used statistical technique in Natural Language Processing (NLP) for document analysis and classification. It is a numerical representation of the importance of a word in a document relative to a collection of documents. The technique is based on two factors: term frequency and inverse document frequency.Term frequency(TF) refers to the number of times a word appears in a document. It is calculated by dividing the number of times a word appears in a document by the total number of words in that document. The more frequent a word appears in a document, the higher its TF value. However, a high RF value doesn't necessarily mean that the word is important, as it may also appear frequently in other documents. The combination of TF and IDF values creates a score that represents the importance of a word in a document relative to the collection of documents.This score is known as the TF-IDF score. Documents with similar TF-IDF scores are likely to contain similar content and can be grouped together for classification purposes. In conclusion, TF-IDF is a powerful technique for document analysis and classification that takes into account both the frequency of words in a document and their rarity in the entire corpus. It provides a numerical representation of the

importance of words in documents, which is useful for tasks such as document classification, sentiment analysis, and topic modeling.

1.4.6 Doc2vec:

Doc2Vec is a neural network-based technique that generates vector representations (or embeddings) of whole documents, such as words, phrases, and sentences. Doc2Vec, also known as paragraph vectors, allows you to express each document in a high-dimensional space as a fixed-length vector.

Doc2Vec is a Word2Vec technique extension that generates vector representations of specific words. While Word2Vec generates vector representations of individual words, Doc2Vec generates vector representations of complete texts, which can then be used for text classification, clustering, and similarity detection.

Doc2Vec works by training a neural network on a corpus of documents, where each document is represented as a sequence of words or tokens. During training, the algorithm learns to predict the next word in a given sequence of words, as well as a special document-level vector, which represents the context of the entire document. The resulting document-level vector can then be used to compare and measure the similarity between different documents.

1.4.7 Stemming:

Stemming is a linguistic processing (NLP) approach for condensing phrases to their core or roots form. It involves removing prefixes and suffixes from words so that words that have the same base form are treated as identical. For example, the words "jump", "jumps", "jumping" would all be reduced to the base form "jump" using stemming. This method is crucial in NLP applications as text mining and information retrieval since it may help decrease the size of the feature space and improve the validity of the model.There are several algorithms for stemming, including Porter stemmer, Snowball stemmer, and Lancaster stemmer.

These algorithms use different sets of rules and heuristics to determine the root form of a word. While stemming can improve the accuracy of some NLP tasks, it can also lead to the loss of important information, such as nuances in meaning conveyed by different forms of a word. As such, it is important to carefully consider the use of stemming in any NLP application and to evaluate its impact on the accuracy and effectiveness of the model.

1.4.8 Tokenization

Tokenization refers to the process of breaking down a text into smaller units, known as tokens. This process involves separating documents into sentences, which is known as sentence tokenization, and further breaking down those sentences into individual words, which is known as word tokennization.

Tokenization is often one of the first steps in many NLP applications, such as machine translation, sentiment analysis, and information retrieval. By breaking down text into tokens, it becomes possible to analyze the text in a more structured way and to apply various algorithms and techniques to the individual tokens.Another approach is to use more sophisticated algorithms to identify and extract tokens from the text. For example, one common technique is to use regular expressions to match patterns of characters that represent meaningful tokens, such as email addresses or URLs.In addition to standard tokenization techniques, there are also specialized techniques that are used for specific applications. For example, in biomedical text mining, there are specialized tokenization techniques for identifying specific entities, such as genes or proteins.Overall, tokenization is a fundamental technique in NLP that is used in a wide range of applications. By breaking down text into tokens, it becomes possible to analyze and understand the text in a more structured and meaningful way.

1.4.9 Text representation

Text representation, also known as feature extraction, is a crucial aspect of natural language processing (NLP). It involves converting text data into a numerical format that can be easily processed by machine learning algorithms. The goal of text representation is to capture the essential information contained within the text, such as the meaning and context, while removing noise and irrelevant details. There are various techniques for text representation, including bag-of-words (BoW), term frequency-inverse document frequency (TF-IDF), and word embeddings. BoW represents text as a collection of unordered words without considering the order or context, while TF-IDF assigns weights to words based on their frequency in the document and inverse frequency in the corpus. Word embeddings, on the other hand, represent words as dense vectors in a high-dimensional space based on their co-occurrence patterns. The choice of text representation technique depends on the specific application and the nature of the text data. Effective text representation is crucial for accurate and efficient NLP models.

1.4.10 One-of-K encoding:

One-of-k is a text representation technique used to convert category data into a numerical format that machine learning algorithms can use. It is a simple and effective method for representing categorical data as binary vectors, such as words or labels. Each unique category is assigned An integer array of size equal to then total number of categories in one-hot encoding. The vector has a value of one in the category index and zero in all other indices. One-hot encoding is useful because it preserves the categorical information of the data while allowing machine learning models to process the data numerically. However, one potential drawback of one-hot encoding is that it can lead to high-dimensional data, especially when there are many categories. This can make it difficult to use certain machine learning algorithms, and other text representation techniques may be more appropriate in these cases.

1.4.11 Word embedding

Word embeddings are a form of natural language processing technology that uses numerical vectors to represent words in a high-dimensional environment. These numerical vectors contain the semantic and syntactic links between words, making them valuable for text classification, information retrieval, and machine translation, among other NLP tasks.

Word embeds work by binding each word in a vocabulary to a unique linear in a continuous linear area, such that phrases with equivalent meanings are assigned to vectors in that space that are near together. To develop this mapping from massive volumes of text data, unsupervised machine learning algorithms such as Word2Vec, GloVe, or FastText are typically utilised.

Once the word embeddings are trained, they can be used to represent words in a way that is more computationally efficient and semantically meaningful thar traditional bag-of-words or one-hot encoding techniques. This makes them a powerful tool for a wide range of NLP applications.

1.4.12 FLDA:

FLDA stands for "Fisher Linear Discriminant Analysis". It is a statistical technique used for feature extraction and dimensionality reduction in machine learning and pattern recognition.FLDA is similar to principal component analysis (PCA) in that it seeks to project high-dimensional data onto a lower-dimensional space while preserving the most important discriminatory information. However, unlike PCA, which seeks to maximize variance in the data, FLDA seeks to maximize the separation between classes in the data. In FLDA, the data is first projected onto a subspace that is most discriminative, by maximizing the ratio of the between-class variance to the within-class variance. This is achieved by finding the linear transformation that maximizes the Fisher criterion, which is a measure of the separation between the classes.FLDA is commonly used in applications such as image recognition, speech recognition, and bioinformatics. By reducing the dimensionality of the data. FLDA can help to improve dimension.

CHAPTER 2

LITERATURE SURVEY

2.1 COMPARISON OF LITERATURE SURVEY:

Wenlonfg Fuu , Biing Xuue, Xiaaoying Gao, Menngjie Zhaang :

In recent years, transfer learning has become an active research area in the field of machine learning and natural language processing (NLP). Transfer learning is a type of machine learning where knowledge gained from one task is applied to another related task. It has been shown to be effective in solving a wide range of NLP problems, including document classification.The paper compares the performance of the proposed approach to several other state-of-the-art methods on both balanced and unbalanced datasets. The experimental results show that the proposed approach outperforms the other methods on most of the datasets, especially on the unbalanced datasets. The authors also conducted several experiments to analyze the impact of different factors on the performance of the proposed approach, such as the amount of labeled and unlabeled data, the number of selected features, and the type of features used.Overall, the paper contributes to the field of NLP by proposing a new approach to document classification that combines transfer learning and GP with different types of features. The experimental results demonstrate the effectiveness of the proposed approach and provide insights into the impact of different factors on its performance.

Karl Weisss, Tagghi M. Khhoshgoftaar and DinggDing Waang:

That sounds like a useful survey paper for anyone interested in heterogeneous transfer learning. The discussion of asymmetric and symmetric transformations is

particularly interesting, as these approaches can have different advantages and disadvantages depending on the specific application. It's also helpful that the paper includes a list of software downloads, as this can save researchers time and effort in implementing transfer learning solutions for their own projects.. having a single open-source software repository for published transfer learning solutions would definitely benefit the research community. It would make it easier for researchers to access and compare different transfer learning algorithms, and would promote more efficient and reliable experiments.Regarding the focus on either correcting marginal or conditional distribution differences, this is an important consideration in transfer learning. Correcting marginal distribution differences involves aligning the overall statistical properties of the source and target domains, while correcting conditional distribution differences involves aligning the relationships between the input features and output labels in the source and target domains. The choice of which approach to focus on may depend on the specific characteristics of the source and target domains and the learning task at hand.Correcting both the marginal and conditional distribution disparities can be difficult, but it has been shown in some circumstances to increase transfer learning performance. Adversarial training is one method for correcting both distributions, in which a domain discriminator is trained to discriminate between the source and target domains, and a generator is educated to provide features that are indistinguishable between the two domains. Another method is to employ multi-kernel learning, which combines numerous kernel functions to represent the various distributions in the source and target domains. It is crucial to highlight that the efficacy of these approaches is dependent on the specific characteristics of the source and target domains, as well as the task at hand. As a result, future study should concentrate on improving methodologies.

Donghwa Kim, Deok Seong Seo, Suhyoun Cho, Pilsung Kang:

The study offers a method for document classification called multi-co-training (MCT), which employs three alternative document representation approaches to expand the variety of feature sets for classification. The three approaches are term frequency-inverse document frequency (TF-IDF) based on the bag-of-words scheme,

topic distribution based on latent Dirichlet allocation (LDA), and document to vector (Doc2Vec) neural-network-based document embedding. MCT outperforms benchmark approaches under a variety of scenarios and is resistant to parameter changes, according to the experimental results. As a result, MCT can be a useful method for document categorization, particularly when there is insufficient label information and the content is in an unstructured sparse format.

Thi Thu Huong Dinh, Thi Huong Chu, Quang Uy Nguyen:

The authors provide a comprehensive review of the literature on transfer learning in GP, including both theoretical and empirical work. They discuss various approaches to transfer learning in GP, including indirect encoding, coevolution, and hybrid methods. They also explore different types of transfer learning, such as instance-based, feature-based, and model-based transfer.Overall, the paper provides a valuable overview of transfer learning in GP and highlights its potential for improving the performance of GP in a wide range of domains. It also identifies several open research questions and challenges, such as determining the optimal level of transfer, designing effective transfer mechanisms, and developing methods for handling domain differences.

Prafull Sharma, Yingbo Li:

In this work, we present the community's beginning opened-object word-levels corpues within tagged keywordss an important phrases. This combination of words corpues comes form Wikipediaa, within randomized papers added with do it more general. Our innovative self-labelling technique is then used to label the data based on contextual word properties. As the findings show, the keywords and key phrases recovered using the proposed self-labelling technique are extremely similar to human-labelling (ground truth). We trained the bidirectional LSTM as a keyword and keyphrase extraction using our self-labeled corpus.The trained model beats previous algorithms for retrieving Preprints keywords and key phrases.

Guangming Lu, Yule Xia:

This work extends text classification technology research by combining the Text Rank algorithm with then naiive Bayees method off then Hadoop clouds computiing platformed. The suggested weight method is refined, and key phrases are employed as text features. Experiment findings demonstrate that when extraction keywords are utilised as features, word

Wen Zhang, Taketoshi Yoshida, Xijin Tang:

The paper discusses the evaluation of three different text representation methods, namely TF*IDF, LSI, and multi-word, for text-based information processing. The performance of these methods is compared for both English and Chinese document collections, with LSI being found to outperform the other two methods in text categorization and information retrieval for English documents. The paper also explores transfer learning methods for GP in two families of symbolic regression problems. The results show that transfer learning techniques can improve GP performance on unseen data and reduce code bloat in GP by limiting the size of transferred individuals. Overall, the paper highlights the importance of choosing the right text representation method for indexing and weighting text and the potential benefits of transfer learning in GP.

2.2 LITERATURE REVIEW OF DIFFERENT METHODOLOGY:

Serial Number	Author	Name	Advantages	Disadvantages
1.	Wenlong Fu , Bing Xue, Xiaoying Gao, Mengjie Zhang	Transductive transfer learning based Genetic Programming for balanced and	allows us to effectively transfer knowledge from the source domain to the target domain	The deep learning algorithm used is more time consuming

		unbalanced document classification using different types of features		
2.	Karl Weiss, Taghi M. Khoshgoftaar and DingDing Wang	A survey of transfer learning	Helps us to understand current trends in transfer learning	One significant limitation in this project has been the number of validation data sets available.
3.	Donghwa Kim, Deok Seong Seo, Suhyoun	transductive transfer learning in text classification technies	helps us understand the role of transfer learning in text classification technques .	It can cause tool deflection and vibration, which are undesirable during machining.

4.	Thi Thu Huong Dinh,Thi HuongChuQuang UyNguyen	Department of Computer Science University of North Texas	results show that LSI has both good semantic and statistical quality,	This not useful for performing on large datasets. Fails in case of repetitive words
5.	Guangming Lu, Yule Xia	School of Electrical Information Engineering, Yunnan Minzu University, Kunming, Yunnan, 650500, China	Experimental results show that compared with the traditional algorithm, the text classification efficiency and accuracy are greatly improved when the extraction keywords are used as features	One significant limitation in this project has been the number of validation data sets available

TABLE-1

LITERATURE SURVEY

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 Methods:

Transductive transfer learning involves transferring knowledge from a source domain to a target domain where the target domain has labeled data available only for a subset of the data. Genetic programming can be used as a machine learning technique for transductive transfer learning.

The proposed transductive transfer learning approach using different types of features is illustrated in Figures 3. The approach consists of three main sections. Firstly, the basic GP algorithm is employed to evolve GP TF-based programs on the source domain as shown in Figure. Secondly, Figuredepicts the creation of new doc2vec-based GP programs from training data that has been labeled by GP programs (classifiers) in the target domain. Lastly, Figure 3(c) shows the generation of GP TF-based programs using training data labeled by GP programs in the target domain. It is noteworthy that each vector dimension is associated with a document.

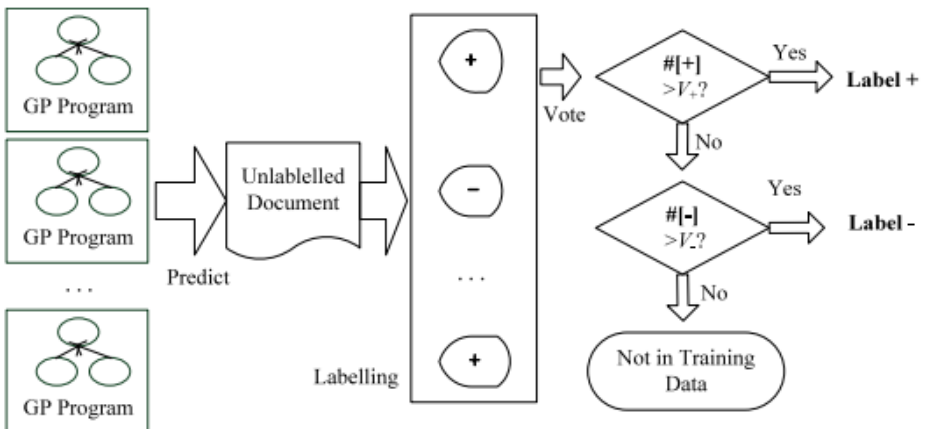


fig6- GP Programs voting unlabelled data into source data

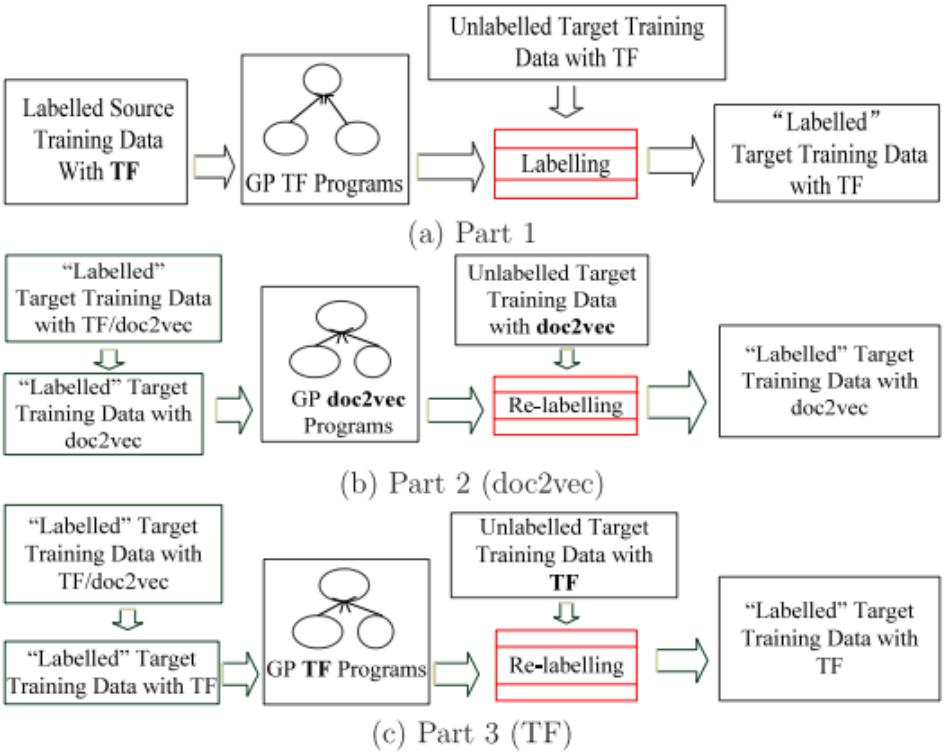


fig7- The proposed transductive GP system could include three key workflows:

• To begin with, the process entails the conversion of Part 1 GP TF-based classifiers to Part 2 GP doc2vec-based classifiers. The GP TF-based techniques are utilized to produce target training data that is "labeled" with TF. At the same time, doc2vec is employed to generate objective training data that is "labeled." It's important to note that the raw target training documents used to create "labeled" target data for training with TF can also be used to extract doc2vec features.

• In Part 2, the GP doc2vec-based classifiers are used to generate labeled training data, which is then used to train new GP doc2vec-based classifiers or to convert the existing GP TF-based classifiers into GP doc2vec-based classifiers. Then, in Part 3, the labeled training data generated using TF in Part 1 is used to train new GP TF-based classifiers, which incorporate the knowledge learned from the Part 2 GP doc2vec-based classifiers. So, while the labels are the same in Part 2 and Part 3, the features used for labeling are different (doc2vec vs. TF).

• The final phase involves using the new GP TF-based classifiers developed in Part 3 to annotate unlabeled data in the target domain, and then using this newly annotated data to generate new GP doc2vec-based classifiers in Part 2. This process is iterative, with the new classifiers being used to annotate more unlabeled data and generate even better classifiers. The final result is a set of high-performing GP classifiers that can effectively classify documents in the target domain using both TF-based and doc2vec-based representations.

Figure 8 depicts an example of the recommended approach for evolving GP TF-based programmes. To begin, the GP TF-based programmes derived from the source domain are used to generate "labelled" target training data with TF. The labels are shared by the target training data and doc2vec, and the training data is used to evolve GP doc2vec-based programmes. These GP doc2vec-based evolved programmes are also utilised to produce new "labelled" target training data with TF. Finally, from the new "labelled" training data with TF, GP TF-based programmes are developed.GPdoctf,n means evolving programmes using TF based on the target training data labelled by the GP programmes evolved using doc2vec at iteration n, GPtfdoc,n for evolving programmes using doc2vec based on the target training data labelled by the GP programmes evolved using TF at iteration n. The techniques of relabeling target training data are intended to yield more useful target training data for the development of more effective GP programmes.

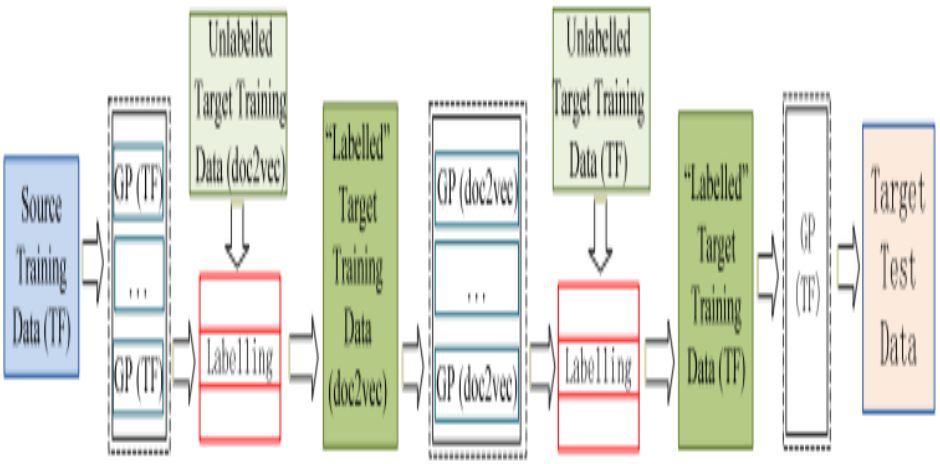


fig8- Gp TF based transfer learning

3.2 Working of the project

Proposed algorithm 1:

Modified GP Search Algorithm (MGP):

Input: Ngen generation, Nrestart time, GP gpoutput: a single GP programme

1: configure the injected GP programmes gpi=gpinput,i

2: for r = 1 to Nrestart, do 3: initialization with gpi injection

4: from generation g = 1 to generation Ngen, do

5: typical GP changing operations

6: finish for 7: set injected GP gpr best is the best solution discovered.

8: finish for

9: return gpN restart best

The workflow of evolving GP programmes for the target data always begins with Part 1. GPdoctf,n then repeats the procedures from Part 2 to Part 3, stopping at Part 3 (when GP TF-based programmes are evolved). GPtfdoc,n also repeats the procedures from Part 2 to Part 3, but it terminates at Part 2 (when GP doc2vec-based programmes are evolved). Figure 4 shows an example of GPdoctfBased on our experiments, the typical GP search algorithm utilised in can gradually evolve a better answer over a long generation. Algorithm 1 describes a modified GP search algorithm (MGP) for effectively searching for appropriate GP programmes. Existing GP programmes from the source domains were used to assist in the search for good GP programmes in the target countries. As a result, current GP programmes are utilised in this paper to compete with the worst initialised programmes. The GP search method resumes initialization after a restricted generation Ngen in order to avoid the local optimal trap. When the training data is imbalanced, a fitness function should incorporate diverse metrics to balance the test accuracy across all categories . A balance fitness function is provided in this paper.

Proposed algorithm 2:

Labelling and generating training data:

Nvote voting GP programmes, Ntotal training documents, threshold ratio rthresh, and category c are all input.

Dc training material labelled as output

1: Using Nvote GP programmes, obtain the vote number for each document Vd in category c.

2: get the histogram Hi=0,...,Nvote of the voting numbers for the category c training documents,

3: make Nsum = 0 and Vthreshold = NVote, and Dc =

4: do for r = Nvote to Nvote 2 5: Nsum = Nsum + Hr

6: If rthresh Ntotal Nsum, then

7: Vthreshold = r proceed to step 9.

8: if end

9: if end

10: for each document d do

11: if Vthreshold Vd then

12: Dc = Dc+document d with category c

13: stop if 14: stop for

15: return marked D

Algorithm 2 is proposed for labelling and training selection.Dc documents as training data in a target domain on category c.The final training data will be created by combining training data Dc.for all categories. In general, if more GP programmes vote in a category for a document, the prediction for the document will be more accurate. When the number of GP programmes voting category c for a document is close to half of the total number of GP programmes voting, the vote outcome is considered questionable. To appropriately pick documents in unbalanced datasets, the voting threshold should be no lower than half of the voting GP programmes (Note 2). Because the document is ambiguous, it is not chosen for training.

DISSIMILARITY BETWEEN GP AND TRADITIONAL SYSTEMS:

There are three significant distinctions between the proposed GP and Existing approaches and the transductive transfer learning methodology. To begin, unlike techniques that use classifiers with distinct features to vote on labels of source domain training texts. The proposed GP transfer learning system directly applies GP programmes to target training documents by leveraging TF from source domains to target domains. Second, not all of the target training documents are chosen to create new training data to evolve GP programmes. Algorithm 2 labels and selects target training documents to generate new training data. Third, the proposed GP system employs various characteristics (TF and doc2vec) at various levels. To create rich training data, GP programmes utilising TF are utilised, and GP programmes using doc2vec are used.for acquiring more appropriately marked training documents. Iterations of TF and doc2vec are used to generate training data and evolve GP programmes.

3.3 DATASETS USED:

The authors used several publicly available datasets for their experiments, including the 20 Newsgroups dataset, the Reuters dataset, the Ohsumed dataset, and the ACM Digital Library dataset. These datasets are commonly used in the field of document classification and should be easily accessible online.

Data	Source domain	Target domain
comp vs sci ($data_1$)	comp.os.ms-windows.misc comp.graphics sci.crypt sci.electronics	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x sci.space sci.med
rec vs talk ($data_2$)	rec.autos rec.motorcycles talk.politics.misc talk.politics.guns	rec.sport.baseball rec.sport.hockey talk.politics.mideast talk.religion.misc
rec vs sci ($data_3$)	rec.autos rec.sport.baseball sci.med sci.space	rec.motorcycles sci.crypt rec.sport.hockey sci.electronics
sci vs talk ($data_4$)	sci.electronics sci.med talk.politics.misc talk.religion.misc	talk.politics.mideast sci.crypt talk.politics.guns sci.space
comp vs rec ($data_5$)	comp.graphics comp.sys.ibm.pc.hardware comp.sys.mac.hardware rec.motorcycles rec.sport.hockey	comp.os.ms-windows.misc comp.windows.x rec.autos rec.sport.baseball
comp vs talk ($data_6$)	comp.graphics comp.windows.x comp.sys.mac.hardware talk.religion.misc talk.politics.mideast	comp.sys.ibm.pc.hardware comp.os.ms-windows.misc talk.politics.guns talk.politics.misc

Table2-twenty news group source and target domain

The twenty newsgroup. The first dataset comes from the 20 newsgroup dataset, which has been widely employed. This dataset includes 20 categories in total. Some categories are very closely related to each other, such as "talk.politics. misc" and "talk.politics.misc". In “talk” group, there are four subcategories, namely "talk.politics.misc", "talk.religion.misc", "talk.politics.mideast", and "talk.politics.guns". Being the same with the datasets used in [40], four major groups ("comp", "rec", "sci" and "talk") are used to split documents into source domains and target domains. Each group is considered as a class, such as "talk.politics.misc" and "talk.politics.guns" both in the category "talk". Table 1 provides the details of the source domains and target domains. There are six binary classification tasks. Training and test documents are separately obtained from the repository "20news-bydate" of the 20 newsgroup dataset. A predefined vocabulary is available in the 20 newsgroup dataset. The number of words used in the training data, the number of training data, and the number of test data in target domains are listed. There are 61188 words in the predefined vocabulary.

Data	Words (training data)	Train	Test
$data_1$	23569	2948	1962
$data_2$	22035	2138	1423
$data_3$	20754	2384	1586
$data_4$	24098	2298	1530
$data_5$	18870	2375	1582
$data_6$	19933	2192	1460

table3-:total number of words of source and test

In this work, the dataset Reuters-21578 Text Categorization Collection is employed as an unbalanced dataset. The raw dataset was obtained from the UCI machine learning repository.3 "Places," "Orgs," and "People" are the three categories. Each group is treated as a separate category. "places vs orgs" (data7), "people vs orgs" (data8), and "places vs people" (data9) are three small and unbalanced datasets. The number of papers in the category "places" outnumbers those in the categories "orgs" and "people". This paper does not use all documents in "places". Table 3 shows the quantity of training materials in each task type.

Data	Label		SD (train)		TD (train)		TD (test)	
	+	–	+	–	+	–	+	–
$data_7$	places	orgs	988	404	1263	223	459	86
$data_8$	people	orgs	407	404	380	223	182	86
$data_9$	places	people	988	407	1263	380	459	182

table4- documents in every category for source data in standard deviation

In the source domains, the working for generation GP programmes are the same as in [14]. if when source input of balanced, the fitnesss functionn facc employs the training accuracy. Because voting is employed to generate tagged training data, 100 GP evolving programmes are chosen as new classifiers in each run. 15 GP evolving programmes are chosen at random to generate the labellled source data. MGP is used to develop GP programmes in they destination domains, with the mutation probability set to 0.15, the crossover probability set to 0.80, the elitism (reproduction) probability set to 0.05, the population size set to 200, and the maximum depth of a programme set to 8. For evolving GP programmes using TF, their is 2 empiriical settinggs: threshold ratio rthresh = 0.2.

CHAPTER 4

PERFORMANCE ANALYSIS

4.1 Results on balanced datasets:

The proposed transfer learning method is compared to two recent transductive transfer learning systems, Subspace Alignment Domain Adaptation (SADA) and Feature-Level Domain Adaptation (FLDA). SADA and FLDA both use linear SVM models with doc2vec. SADA and FLDA are supported by the free library libTLDA5. For the results comparisons, multiple comparisons with Holm's method and a significant level of 0.05 are utilised. Bold figures indicate that the relevant findings are much better than the others. We present the outcomes of GP programmes utilising TF and doc2vec, together with remarks, after comparing them to the existing methods.

Compared the new GP method to existing algorithms using doc2vec.:

It shows then averages and SD of SADA, FLAD, and GPtfdoc,2 test accuracies on Articles in the intended sites are checked. In GPtfdoc, first, Genetic TF-based programmes from the relevant domain of origin are used to obtain labelled target training data; second, the labelled target training data is used to train GP doc2vec-based programmes; third, the GP doc2vec-based programs are used to obtain relabelled target training data; fourth, the relabelled target training data is used to evolve GP TF-based programmes; and fifth, the newly evolved GP programs. There are two notable observations. First, in terms of test accuracy, GPtfdoc,2 clearly outperforms SADA and FLDA across the six datasets. Second, SADA and FLDA fail to successfully transfer models from the source domains to the target domains on data2, data4, and data5. SADA virtually always forecasts erroneously all test documents, particularly on data2.

It is possible that the vectors of documents for class "+" in the source domains are similar to the vectors of documents for class "-" in the destination domains. Because doc2vec is acquired from a black box, it is difficult to explain why the vectors for class"+" in The vectors from class "-" in the source domain are more comparable to the vectors from class "+" in the target domain. The vectors from class "-" in the source domain are more comparable to the vectors from class "+" in the target domain.

Data	SADA	FLDA	$GP_{tf\rightarrow doc,2}$
$data_1$	0.727 ± 0.000	0.630 ± 0.025	$\mathbf{0.867 \pm 0.019}$ ↑
$data_2$	0.031 ± 0.000	0.453 ± 0.063	$\mathbf{0.914 \pm 0.021}$ ↑
$data_3$	0.861 ± 0.000	0.505 ± 0.010	$\mathbf{0.899 \pm 0.011}$ ↑
$data_4$	0.108 ± 0.000	0.495 ± 0.039	$\mathbf{0.839 \pm 0.021}$ ↑
$data_5$	0.351 ± 0.000	0.468 ± 0.047	$\mathbf{0.895 \pm 0.020}$ ↑
$data_6$	0.801 ± 0.000	0.708 ± 0.056	$\mathbf{0.888 \pm 0.024}$ ↑

table5- test accuracies on sada flda on doc2vec

4.2 Outcomes on GP machines that only employ TF:

The vectors in the original area are more similar relative to the indexes in the intended domain from class "-" in comparison to the indexes in the destination domain from class "+".the target domains for the GP programmmes utilissing TF on datasets 1 .There are three interesting observations. First, although the test precision rates are not high, the genetically evolved programmme from the original disciplines can be put into practise to the areas of target.. Second, GPdoc→tf ,2 has best test accuracies on the six datasets. Third, the test accuracy expands significantly when moving from SGP (going from the input regions) to GPdoctf,2. It should be noted that the exercises in this table solely evolve GP programmmes using TF.In regards to the test's precision on the target domain, the GP programmmes evolved from the target training data labelled by the GP programmes from the source domains (SGP) surpass the GP programmes derived However, when these newly evolved GP programmes (using TF) are used to vote labels of the target training documents, the GP programmes (using TF) evolved from the relabelled training documents do not differ significantly from the GP programmes evolved from SGP. mFurthermore, all of those GP programmmes

(that only use TF) perform significantly worse than the GP programmmes from GPdoctf,2 (which evolved from GP programmmes that used doc2vec, see Table 4). As a result, incorporating TF with doc2vec to increase the performance of established GP programmmes benefiting from TF is profitable.

Data	*SGP*	$GP_{doc\rightarrow tf,1}$	$GP_{doc\rightarrow tf,2}$
$data_1$	0.655 ± 0.025	0.735 ± 0.025	$\mathbf{0.797 \pm 0.015}$ ↑
$data_2$	0.686 ± 0.030	0.776 ± 0.024	$\mathbf{0.831 \pm 0.016}$ ↑
$data_3$	0.634 ± 0.037	0.767 ± 0.048	$\mathbf{0.826 \pm 0.084}$ ↑
$data_4$	0.634 ± 0.036	0.726 ± 0.042	$\mathbf{0.807 \pm 0.021}$ ↑
$data_5$	0.639 ± 0.036	0.787 ± 0.037	$\mathbf{0.864 \pm 0.020}$ ↑
$data_6$	0.701 ± 0.038	0.799 ± 0.032	$\mathbf{0.859 \pm 0.017}$ ↑

table6-Test Accuracies on using TF only

Data	*SGP*	$GP_{tf\rightarrow tf,1}$	$GP_{tf\rightarrow tf,2}$
$data_1$	0.655 ± 0.025	0.668 ± 0.025	0.639 ± 0.018 ↓
$data_2$	0.686 ± 0.030	0.707 ± 0.033	0.700 ± 0.021 ↓
$data_3$	0.634 ± 0.037	0.715 ± 0.051	0.715 ± 0.108 ↓
$data_4$	0.634 ± 0.036	0.679 ± 0.035	0.698 ± 0.037 ↓
$data_5$	0.639 ± 0.036	0.712 ± 0.039	0.739 ± 0.036 ↓
$data_6$	0.701 ± 0.038	0.742 ± 0.024	0.735 ± 0.011 ↓

table7-Test accuracies (means and standard deviations) on the target domains from GP using TF (tf → tf) only

4.3 Results on GP systems using Doc2vec only:

The target domains for the GP programs using TF on the datasets from datal to data6. There are three interesting observations.First, the GP evolved programs from the source domains can be directly applied to the target domains although the test accuracies are not high. Second, GPdoc-tf,2 has the best test accuracies on the six datasets. Third, from SGP (from the source domains) to GPdoc→tf,2, the test accuracy improvements are obvious. Note that the experiments in this table evolve GP programs using TF only. Table 7 displays the outcomes of the GP programmes that used doc2vec to develop from the target training data tagged by the GP programmes that used TF. According to the table, the test performances improve from GPtfdoc,1 to GPtfdoc,2 on the six datasets. Table shows that GPtfdoc,2 has the significantly best performance.It demonstrates that employing GPtf and GPdoc in tandem can significantly improve the test performances of the evolved GP programmes. Furthermore, the results of

GPtfdoc,2 are compared to the results of GPdoctf,2. In this case, "" shows that the GPtfdoc,2 results are considerably better than the relevant GPdoctf,2 results in terms of t-tests with a significance threshold of 0.05. In comparison to GPdoctf,2 (see Table 5), GPdoctf2 has much higher test results.

Data	SGP	$GP_{tf\rightarrow doc,1}$	$GP_{tf\rightarrow doc,2}$
$data_1$	0.655 ± 0.025	0.799 ± 0.030	$\mathbf{0.867 \pm 0.019}$ ↑
$data_2$	0.686 ± 0.030	0.861 ± 0.033	$\mathbf{0.914 \pm 0.021}$ ↑
$data_3$	0.634 ± 0.037	0.859 ± 0.027	$\mathbf{0.899 \pm 0.011}$ ↑
$data_4$	0.634 ± 0.036	0.770 ± 0.039	$\mathbf{0.839 \pm 0.021}$ ↑
$data_5$	0.639 ± 0.036	0.873 ± 0.032	$\mathbf{0.895 \pm 0.020}$ ↑
$data_6$	0.701 ± 0.038	0.846 ± 0.034	$\mathbf{0.888 \pm 0.024}$ ↑

table8-test accuracies on target data using doc2vec

To further improve test performances on the predicted results on the six datasets, Tables 8 and 9 provide the voting results from GP programs using TF and doc2vec respectively. In Table 9, "↑" means that the results from VGPtf→doc,2 are significantly better than the relevant results from VGPdoc→tf ,2 .

Data	V_{SGP}	$V_{GP_{doc\rightarrow tf,1}}$	$V_{GP_{doc\rightarrow tf,2}}$
$data_1$	0.651 ± 0.011	0.784 ± 0.030	$\mathbf{0.839 \pm 0.006}$
$data_2$	0.727 ± 0.014	0.835 ± 0.007	$\mathbf{0.883 \pm 0.006}$
$data_3$	0.784 ± 0.042	0.889 ± 0.029	$\mathbf{0.906 \pm 0.004}$
$data_4$	0.710 ± 0.024	0.798 ± 0.010	$\mathbf{0.862 \pm 0.008}$
$data_5$	0.801 ± 0.040	0.835 ± 0.021	$\mathbf{0.900 \pm 0.007}$
$data_6$	0.784 ± 0.030	0.851 ± 0.009	$\mathbf{0.886 \pm 0.009}$

Table9-test accuracies on target domain using voting

From Table 9, all test average accuracies of VGPtf→doc,2 are higher than 0.909, except 0.869 on data4. Also, single GP programs from GPtf→doc,2 (see Table 7) can compete with the combination results from VGPtf-doc,1 on datal, data2, and data4. It shows that our proposed GP system effectively and further improves the test accuracies of GP programs by sharing knowledge between TF and doc2vec.

Data	V_{SGP}	$V_{GP_{tf\rightarrow doc,1}}$	$V_{GP_{tf\rightarrow doc,2}}$
$data_1$	0.651 ± 0.011	0.829 ± 0.011	$\mathbf{0.910 \pm 0.005}$ ↑
$data_2$	0.727 ± 0.014	0.902 ± 0.009	$\mathbf{0.938 \pm 0.011}$ ↑
$data_3$	0.784 ± 0.042	0.906 ± 0.005	$\mathbf{0.910 \pm 0.003}$ ↑
$data_4$	0.710 ± 0.024	0.830 ± 0.012	$\mathbf{0.869 \pm 0.005}$ ↑
$data_5$	0.801 ± 0.040	0.921 ± 0.009	0.921 ± 0.006 ↑
$data_6$	0.784 ± 0.030	0.899 ± 0.012	$\mathbf{0.912 \pm 0.008}$ ↑

table10-test accuracies (means and standard deviations) on the target domains from voting of GP programs using doc2vec from TF.

Table 10 offers the required percentage improvement to further assess the test performance improvement from GPdoctf,2 to VGPdoctf,2 and from GPtfdoc,2 to VGPdocdoc,2. Overall, the improvement from GPdoctf,2 to VGPdoctf,2 is greater than the improvement from GPtfdoc,2 to VGPdoctf,2. According to data3, GP programmes from GPdoctf,2 may be substantially more diverse than GP programmes from GPtfdoc,2.

Data	$GP_{doc\rightarrow tf,2}$ to $V_{GP_{doc\rightarrow tf,2}}$	$GP_{tf\rightarrow doc,2}$ to $V_{GP_{tf\rightarrow doc,2}}$
$data_1$	5.3	5.0
$data_2$	6.3	2.6
$data_3$	9.7	1.2
$data_4$	6.8	3.6
$data_5$	4.2	2.9
$data_6$	3.1	2.7

table11-percentage improvement from doc2vec to tf idf and vice versa

4.4 DISCUSSIONS ON THE RESULTS OF BALANCED DATASETS:

Because of the stochastic learning process and various alternative optimal solutions, the final result in each run is generally different in each run.To accurately forecast documents, the proposed GP system relies heavily on the diversity of predictions from GP programmes. First, programmes that use TF can successfully differentiate incomplete training documents in the target domains. There are three advantages to the suggested GP system. First, the TF-based GP programmes incorporate shared words from the source and target domains, allowing these GP programmes from the source domains to forecast some texts from the target domains. Second, doc2vec, a high-level feature, can be utilised to evolve GP programmes with excellent test accuracy. The GP doc2vec-based programmes can label training data with high accuracy, allowing the labelled training data to be effectively used to evolve new GP TF-based programmes for label prediction in the target domain.Third, there are forecasts from GP programmes that use doc2vec and GP programmes that use TF.

Rich diversity is beneficial when combining several GP programmes to improve test performance. Multiple GP programmes can be used to generate more effective new GP programmes from labelled training data.As a result, the variety of forecasts from GP programmes, as well as the methods/directions for efficiently evolving strong GP programmes, are critical in the suggested GP system. To enhance test performance even further, one option is to utillise more effective high-level features that outperform doc2vec.

Data	SADA	FLDA	$GP_{tf\rightarrow doc,2}$
$data_7$ (+)	**0.932 ± 0.000**	0.836 ± 0.023	0.727 ± 0.048
$data_7$ (-)	0.047 ± 0.000	0.230 ± 0.043	0.490 ± 0.079
$data_8$ (+)	0.385 ± 0.000	0.890 ± 0.031	0.868 ± 0.039
$data_8$ (-)	**0.826 ± 0.000**	0.378 ± 0.057	0.191 ± 0.049
$data_9$ (+)	0.231 ± 0.000	0.780 ± 0.046	0.796 ± 0.039
$data_9$ (-)	**0.791 ± 0.000**	0.579 ± 0.047	0.490 ± 0.079
Data	$V_{GP_{tf\rightarrow doc,2}}$	$GP_{doc\rightarrow tf,2}$	$V_{GP_{doc\rightarrow tf,2}}$
$data_7$ (+)	0.748 ± 0.016	0.620 ± 0.028	0.642 ± 0.012
$data_7$ (-)	0.488 ± 0.030	0.720 ± 0.057	**0.784 ± 0.023**
$data_8$ (+)	**0.911 ± 0.003**	0.784 ± 0.053	0.871 ± 0.022
$data_8$ (-)	0.144 ± 0.007	0.356 ± 0.083	0.254 ± 0.053
$data_9$ (+)	**0.846 ± 0.012**	0.681 ± 0.023	0.693 ± 0.007
$data_9$ (-)	0.428 ± 0.031	0.646 ± 0.055	0.662 ± 0.024

table12-test accuracies on each category on target domains from SADA FLDA AND GP.

4.4 Output on imbalanced datasets:

Geometric mean is used as a view of combination measurement to verify the performance of the outcomes on unbalanced datasets. Table 12 displays the geometric means and standard deviations of the results from SADA, FLDA, and GP using TF and doc2vec. VGPdoctf 2 has the considerably best results on data7 and data9, while FLDA has the significantly best results on data8. In summary, the suggested GP system evolving GP programmes are reasonably balanced predictions on each category on the target domains, as evidenced by the results on the unbalanced datasets.

Data	SADA	FLDA	$GP_{tf \to doc,2}$
$data_7$	0.208 ± 0.000	0.437 ± 0.033	0.594 ± 0.040
$data_8$	0.563 ± 0.000	$\mathbf{0.578 \pm 0.039}$ ↑	0.403 ± 0.045
$data_9$	0.427 ± 0.000	0.671 ± 0.015	0.622 ± 0.054
Data	$V_{GP_{tf \to doc,2}}$	$GP_{doc \to tf,2}$	$V_{GP_{doc \to tf,2}}$
$data_7$	0.603 ± 0.016	0.667 ± 0.028	$\mathbf{0.709 \pm 0.013}$
$data_8$	0.362 ± 0.008	0.524 ± 0.061	0.467 ± 0.051
$data_9$	0.601 ± 0.021	0.662 ± 0.028	$\mathbf{0.677 \pm 0.012}$

table13- output of imbalance dataset

4.5 Debates on unbalanced dataset results:

When the proposed GP system is applied to unbalanced datasets, the geometric mean fitness function can balance the test accuracy on each category. However, when the unbalanced training data (data8) in the target domain is small, it is hard for the proposed GP system to get a high test accuracy on label "-". From the number of training examples shown in Table 3, the number of selected "certain" training documents with label "-" in data8 is possibly less than 120. Because Algorithm 2 confines the voting threshold to being greater than half of the voting GP programmes on each category, the number of documents with the label "-" on each dataset will be less than the predicted quantity. The actual number of training documents with label "-" created from data8 is less than 60, resulting in poor training information on label "-" and the difficulties of evolving effective programmes on properly predicting documents with label "-". The number of training documents with the label "-" is greater than 100 for data7 and data9, and GP can evolve programmes that can appropriately distinguish documents with the label "-". In our future work, we will propose instance-based algorithms to increase the test accuracy of documents with the label "-" on data

CHAPTER 5

CONCLUSION

Overall, this study proposes a transductive transfer learning approach using genetic programming (GP) to transfer knowledge from a source domain to a target domain where labeled data is only available for a subset of the data. The proposed approach involves evolvving GP prograams using both term frequency (TF) and doc2vec features in a itearative process, and using the GP programs to vote on and select documents for training data. Experimental results show that the proposed GP system outperforms two contemporary transfer learning methods on a mix of balanced and unbalanced datasets. The study also suggests that GP can effectively transfer knowledge from evolving programs that use different types of features. Future work includes exploring ways to leverage GP for more effective embedding-based features and incorporating instance-based learning techniques for better performance on unbalanced small datasets.

5.1 FUTURE SCOPE

In the last several years, the field of NLP and deep learning has broken down many obstacles, and new cutting-edge methodologies and models have been revealed. With this in mind, the future of NLP is bright, and we urge that we continue to examine new embedding models and extraction approaches to enhance the outcome of our thesis.

REFERENCES

All references must be in IEEE Format as per following:

[1] S. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.

[2] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big Data 3 (1) (2016) 9.

[3] Sean Owen, Robin Anil, Ted Dunning. Mahout in Action, 2010:274-280.

[4] Yang J, Ji D, Cai DF. Keyword Extraction In Mu1ti-Document Based on TextRank Technology. NCIRCS,2008

[5] Sergey Brin Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In Proceedings of WWW pages 107-117, 1998.

[6] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In Proceedings of ACM-SIAM symposium on Discrete algorithms, 1998.

[7] Litvak M and Last M. Graph-based keyword extraction for single document summarization. In Proceedings of Workshop Multi-source Multilingual Information Extraction and Summarization,2008.

[8] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In Proceedings of EMNLP, 2004.

[9] Xiaojun Wan and Jiangguo Xiao. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In Proceedings of International Conference on Computational Linguistics(COLIN 2008) pages 969-976, 2008

[11] Rose S, Engel D, Cramer N, et al. Automatic keyword extraction from individual documents[J]. Text mining: applications and theory, 2010: 1-20.

[12] Lahiri S, Choudhury S R, Caragea C. Keyword and keyphrase extraction using centrality measures on collocation networks[J]. arXiv preprint arXiv:1401.6571, 2014.

[13] Page L, Brin S, Motwani R, et al. The PageRank citation ranking: Bringing order to the web[R]. Stanford InfoLab, 1999.

[14] Mihalcea R, Tarau P. Textrank: Bringing order into text[C]//Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.

[15] R. Barzilay, M. Elhadad, "Using lexical chains for text summarization," Advances in automatic text summarization, 1999, pp. 111-121.

[16] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in: Proceedings of the 2003 conference on Empirical methods in natural language processing, ACL, 2003, pp. 216-223.
G. Salton, A. Singhal, M. Mitra, C. Buckley, "Automatic text structuring and summarization," Information Processing & Management, vol. 33 (2), 1997, pp. 193-207

[18] Beluga S, Meštrović A, Martinčić-Ipšić S. An overview of graph-based keyword extraction methods and approaches[J]. Journal of information and organizational sciences, 2015, 39(1): 1-20.

[19] Zhang Q, Wang Y, Gong Y, et al. Keyphrase extraction using deep recurrent neural networks on Twitter[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016: 836-845.

[20] Meng R, Zhao S, Han S, et al. Deep keyphrase generation[J]. arXiv preprint arXiv:1704.06879, 2017.

[21] Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, Chris Dyer, “Problems With Evaluation of Word Embeddings Using Word Similarity Tasks”, Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP.

[22] Wang Y, Zhang J. Keyword extraction from online product reviews based on bi-directional LSTM recurrent neural network[C]//2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, 2017: 2241-2245.

[23] Humphreys J B K. PhraseRate: An HTML Keyphrase Extractor[J]. Technical report, 2002.

[24] Ritter A, Clark S, Etzioni O. Named entity recognition in tweets: an experimental study[C]//Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, 2011: 1524-1534.

[25] Loper E, Bird S. NLTK: the natural language toolkit[J]. arXiv preprint cs/0205028, 2002