

A novel Auto-ML Framework for Sarcasm Detection

A thesis submitted in partial fulfilment of the requirements for Degree of DProf Data Science

Sept 2021

The dissertation is written by
Sharjeel Imtiaz

Approved by

_____, Chair, Doctoral Dissertation Committee

_____, Members, Doctoral Dissertation Committee

Accepted by

_____, Chair, Department of Computer Science

_____, Dean, College of Arts and Sciences

TABLE OF CONTENTS

LIST OF TABLES	IX
DEDICATION.....	XI
ABSTRACT.....	xii
CHAPTER 1 INTRODUCTION	- 1 -
1.1 Introduction	- 1 -
1.2 Background	- 3 -
1.3 Introduction	- 7 -
1.3.1 Aim and Objectives of the Research	- 9 -
1.4 Irony and its types	- 12 -
1.5 Other verbal irony or pure irony.....	- 12 -
1.6 Machine Learning Techniques	- 13 -
1.7 Deep learning technique	- 13 -
1.7.1 Convolutional Neural Network	- 15 -
1.7.2 Long Short-Term Memory (LSTM).....	- 20 -
1.8 Research Questions	- 22 -
1.9 Research Contribution.....	- 24 -
1.10 Thesis outline	- 26 -
CHAPTER 2 LITERATURE REVIEW AND SYSTEMATIC REVIEW	- 27 -
2.1 Systematic Review	- 27 -
2.2 Criteria of inclusion and exclusion.....	- 27 -
2.3 Search Criteria - I	- 29 -

2.4	Search Criteria II	- 33 -
2.5	Literature Review	- 36 -
2.6	Pattern-based Techniques.....	- 37 -
2.7	Context-based Techniques	- 42 -
2.7.1	Feature Concatenation – Context-based Approach.....	- 48 -
2.8	Transfer learning	- 51 -
	Transudative Transfer Learning	- 52 -
2.9	Automated Machine Learning (AutoML).....	- 59 -
2.10	Gap Analysis and Findings	- 69 -
2.11	Problem Statement	- 77 -
2.12	Summary	- 78 -
	CHAPTER 3 RESEARCH METHODOLOGY.....	- 79 -
3.1	Existing Research Methodology	- 79 -
3.2	Adapted Methodology.....	- 83 -
3.2.1	Data Collection.....	- 85 -
3.2.2	Preprocessing	- 88 -
3.3	Exploring Features Methodology.....	- 90 -
3.3.1	Features Desired.....	- 90 -
3.3.2	Preprocessing Strategy	- 91 -
3.3.3	Cognitive Algorithm for Irony Detection.....	- 94 -
3.3.4	Features Extraction.....	- 94 -
3.4	Feature’s taxonomy	- 95 -
3.4.1	Verbal irony Detection.....	- 98 -

2)	“I love waiting forever for the doctor #sarcasm”	- 98 -
3.4.2	Incongruity Features.....	- 99 -
3.4.3	Syntactic Features	- 99 -
3.4.4	POS Tag	- 100 -
3.4.5	Lexical Feature.....	- 101 -
3.4.6	Incongruity Lexical Features.....	- 103 -
3.4.7	Incongruity features Methodology	- 103 -
3.4.8	Pragmatic Feature Methodology	- 105 -
3.4.9	Emoticons / Emojis	- 106 -
3.5	Existing Machine Learning Techniques.....	- 107 -
3.5.1	Regression Technique (LR)	- 107 -
3.5.2	SVM	- 108 -
3.5.3	Deep Learning Model for Big data	- 109 -
3.5.4	Deep Learning Model (LSTM)	- 111 -
3.6	Transfer Learning Strategies	- 112 -
3.7	Evaluation Matrices.....	- 113 -
CHAPTER 4 EXPERIMENTATION, ALGORITHM, MODELS, RESULTS, AND		
	EVALUATION	- 116 -
4.1	A novel comparison of core models vs. deep learning for sarcasm detection ..	- 116 -
4.2	Irony Detection Algorithm	- 117 -
4.2.1	Implicit Incongruity Algorithm (IIA).....	- 117 -
4.2.2	Explicit Incongruity Algorithm (EIA).....	- 125 -
4.3	Pragmatic Features	- 129 -

4.4	Methodology	129 -
4.4.1	SVM	130 -
4.4.2	Logistic Regression	130 -
4.4.3	KNN	131 -
4.4.4	Neural Network	131 -
4.4.5	Adapted hybrid model (LSTM-CNN)	132 -
4.4.6	Data Collection	135 -
4.5	Results and Discussion	136 -
4.5.1	Experiment	136 -
4.6	Evaluation	136 -
4.7	Baseline Results	137 -
4.7.1	Evaluation of Data Set (DS1)	137 -
4.7.2	Evaluation of Data Set (DS2)	139 -
4.8	Conclusion	140 -

CHAPTER 5 THE NEWLY DEVELOPED AUTOML FRAMEWORK FOR SARCASM

	DETECTION	142 -
5.1	Adapted DeepConcat Model	144 -
5.2	Adapted AutoML DeepConcat	149 -
5.3	Model Selection, hyperparameters optimization, and Architecture Search	149 -
5.4	Result and Discussion	154 -
5.4.1	Experiment Setup	155 -
5.4.2	Preprocessing over DS3	156 -
5.4.3	AutoML DeepConcat Evaluation	158 -

5.5	Conclusion.....	- 164 -
CHAPTER 6 TRANSFER LEARNING - DOMAINS ADAPTATION		- 167 -
6.1	Data Source	- 167 -
6.2	Methodology	- 169 -
6.2.1	Transfer Learning Strategies	- 171 -
6.3	Pretraining Supervision	- 173 -
6.4	Experiment	- 173 -
6.5	Existing Pre-train model.....	- 173 -
6.6	AutoML Pretrain Proposed Model	- 176 -
6.6.1	Last strategy	- 181 -
6.6.2	Chain-Thaw Strategy.....	- 183 -
6.6.3	Proposed Strategy Faded out.....	- 185 -
6.7	Results and Discussion (Faded-out).....	- 187 -
6.8	Conclusion.....	- 191 -
CHAPTER 7 DISCUSSION, CONCLUSION, AND RECOMMENDATION ...		- 193 -
7.1	Discussion	- 194 -
7.2	Recall of Research Questions.....	- 194 -
7.3	Discussion	- 201 -
7.4	Conclusion and Recommendation.....	- 202 -
7.4.1	Thesis Summary	- 202 -
7.5	Conclusion.....	- 204 -
7.5.1	Research Limitation	- 211 -
7.5.2	Future Direction	- 212 -

BIBLIOGRAPHY	- 214 -
APPENDIX.....	- 232 -

LIST OF FIGURES

Table 3.1: Few existing methodology overviews	- 81 -
Figure 3.2: Adapted Research Methodology	- 84 -
Table 3.2.1b: Irony Dataset.....	- 86 -
Figure 3.3.4: Feature Taxonomy.....	- 95 -
Table 4.2.1d: 2-Gram Pattern-based intended phrases	- 121 -
Figure 4.4.5c: Baseline CNN-BiLSTM Models.	- 135 -
Figure 5.4a: AutoML DeepConcat Framework.....	- 150 -
Figure 5.4b: Fully Connected Network	- 152 -
Figure 5.4c: LSTM feed-forward and backward	- 154 -
Figure 6.2.1: Chain-thaw strategy to freeze all layers sequentially.....	- 172 -
Figure 6.6: Training of the formal and informal dataset using Proposed Framework.....	- 177 -
Figure 6.6.1: Last layer strategy from pretrain BiLSTM with 512 units	- 182 -
Figure 6.6.3a: Removing dense layers sequentially and evaluating the performance.....	- 186 -
Figure 6.6.3b: Removing dense layer at CNN with 512 units.....	- 187 -

LIST OF TABLES

Table 2.2: Inclusion/Exclusion Criteria	- 29 -
Table 2.3a: Systematic Review.....	- 30 -
Table 2.3b: Selected Articles	- 32 -
Table 2.4a: Systematic Review.....	- 34 -
Table 2.4b: Selected Articles	- 35 -
Table 2.8: Transfer Learning Review Matrix	- 58 -
Table 2.9: Automated Machine Learning Models (AutoML) Review Matrix	- 68 -
Table: 3.2.1a: Data Collection.....	- 85 -
Table 3.4.4: POS Tag.....	- 101 -
Table 3.4.7: Incongruity Features verbal types with bigram patterns.....	- 104 -
Table: 3.4.9: Emoticon features by former researcher.....	- 107 -
Table 4.2.1e: Polarities contrast Incongruity SUBSET	- 123 -
Table 4.2.1f: Pattern incongruity Subset.....	- 124 -
Table 4.2.1g: Polarity contrast incongruity 3-gram and 1-gram subset.....	- 125 -
Table 4.4.2b: Polarities contrast incongruity subset	- 128 -
Table 4.4.2c: Explicit incongruity features	- 129 -
Table 4.5.1: Description of benchmarking dataset split.....	- 136 -
Table 4.7.1a: Comparisons of benchmarking dataset DS1 with our baseline	- 138 -

Table 4.7.2a: Comparisons of benchmarking dataset DS2 with our baseline	- 140 -
Table 4.7.2b: Scaling effect on SVM	- 140 -
Table 5.5.1a: Description of benchmarking dataset	- 156 -
Table 5.5.2: Preprocessing Level F1 results on DS3 dataset	- 157 -
Table 6.2b: Distribution of stars assigned to reviews.....	- 169 -
Table 6.6: Pretrain Models.....	- 180 -
Table 6.6.1: Benchmark imbalance dataset performance on AutoML-Deep pretrain framework .	- 183 -
Table 6.7: Imbalance dataset performance on proposed AutoML-DeepConcat pretrain Framework.....	- 190 -

DEDICATION

I dedicated this thesis to my late mother. She was a major part of my life. She made my life with her care, love, and full dedication. She always encouraged me and nourished me throughout her life. I hope my dedication to her, God will accept my best wishes and prayers for her.

I acknowledge Dr. Allan J Brimicombe, who helped and supported me to finish this dissertation/thesis. He always promptly advised and supervised me with useful tips that helped me to define my research methodology and polished my research skills.

I am thankful to Dr. Yang Li for helping me to complete the thesis at a crucial time and Dr. Julie Wall for her valuable comments that helped me to make proper correction in the thesis. My family supported me in completing the tasks and encouraged me to do the research alone with full dedication, especially my wife and kids. During my mother's demise, it was a difficult time in the pandemic, but my family's support and encouragement helped me stay on the path to finish the goal of my life. I had a tough time during the last stage of my thesis. I thank all of those who helped me in UEL and continually helped me to accomplish the given task.

ABSTRACT

Many domains have sarcasm or verbal irony presented in the text of reviews, tweets, comments, and dialog discussions. The purpose of this research is to classify sarcasm for multiple domains using the deep learning based AutoML framework. The proposed AutoML framework has five models in the model search pipeline, these five models are the combination of convolutional neural network (CNN), Long Short-Term Memory (LSTM), deep neural network (DNN), and Bidirectional Long Short-Term Memory (BiLSTM). The hybrid combination of CNN, LSTM, and DNN models are presented as CNN-LSTM-DNN, LSTM-DNN, BiLSTM-DNN, and CNN-BiLSTM-DNN. This work has proposed the algorithms that contrast polarities between terms and phrases, which are categorized into implicit and explicit incongruity categories. The incongruity and pragmatic features like punctuation, exclamation marks, and others integrated into the AutoML DeepConcat framework models. That integration was possible when the DeepConcat AutoML framework initiate a model search pipeline for five models to achieve better performance. Conceptually, DeepConcat means that model will integrate with generalized features. It was evident that the pretrain model BiLSTM achieved a better performance of 0.98 F1 when compared with the other five model performances. Similarly, the AutoML based BiLSTM-DNN model achieved the best performance of 0.98 F1, which is better than core approaches and existing state-of-the-art Tweeter tweet dataset, Amazon reviews, and dialog discussion comments. The proposed AutoML framework has compared performance metrics F1 and AUC and discovered that F1 is better than AUC. The integration of all feature categories achieved a better performance than the individual category of pragmatic and incongruity features. This research also evaluated the performance of the dropout layer hyperparameter and it achieved better performance than the fixed percentage like 10% of dropout parameter of the AutoML based Bayesian optimization. Proposed

AutoML framework DeepConcat evaluated best pretrain models BiLSTM-DNN and CNN-CNN-DNN to transfer knowledge across domains like Amazon reviews and Dialog discussion comments (text) using the last strategy, full layer, and our fade-out freezing strategies. In the transfer learning fade-out strategy outperformed the existing state-of-the-art model BiLSTM-DNN, the performance is 0.98 F1 on tweets, 0.85 F1 on Amazon reviews, and 0.87 F1 on the dialog discussion SCV2-Gen dataset. Further, all strategies with various domains can be compared for the best model selection.

CHAPTER 1

Introduction

1.1 Introduction

Sarcasm or verbal irony is present in many social domains, such as Amazon reviews, Twitter tweets, and dialog discussions. Sarcasm detection using deep learning approaches is a conventional classifier proposed by (Felbo, et al., 2017) that classifies multiple tasks like sentiment, sarcasm, and emotions using a single model. Verbal irony is a type of irony that is a synonym for sarcasm. (Riloff, et al., 2013; Maynard & Greenwood, 2014; Joshi, et al., 2015; Poria, et al., 2016; Van Hee, et al., 2018) few authors particularly worked on contextual features of sarcasm. The contextual features are categorized into incongruity and pragmatic features. Incongruity features are the concept of positive phrase contrasting in polarity with the negative term in a tweet. For instance, in the sentence “I love being ignored”, it can be observed that the term “love” has a positive polarity which contrasting with the negative phrase “being ignored”. It is an example of implicit incongruity with contrasting polarities.

The novelty of this research is the development of a model that overcomes the gap of sarcasm detection classification for multiple social domains like Twitter, Amazon reviews, and dialog comments using a single automated model. The primary aim of this work is sarcasm detection with the significant performance by building, training, and evaluating the AutoML based pretrained models. Automation Machine Learning (AutoML) is the

process of automating the Machine Learning (ML) tasks that apply to real-world problems, the process automates the pipeline or steps like preprocessing, feature engineering, model search, and hyperparameters optimization. Feature engineering is the concept of extracting incongruity features using semi-supervised algorithms which further integrate these features into the model. The incongruity features are the contrasting polarity features among terms and phrases which are vital clues for sarcasm detection. The model is trained for each preprocessing level or cleaning step and further evaluated to ensure that clues are necessary or not for example, punctuation clue. The research proposed the explicit incongruity and implicit incongruity algorithms in Chapter 4. These algorithms are the main contribution of this research, which extracted the incongruity features in the form of polarity contrast among terms and phrases (further details are in Chapter 4). The second outcome of this research aim is to finalize the model which is best for sarcasm detection evaluated over multiple datasets. The experiment was performed using the core and deep learning models. The core models are support vector machine (SVM), k-nearest neighbors (KNN), and logistic. However, after deemed the existing literature the focus of this research is the deep learning models: Long Short-Term Memory (LSTM), Convolution Neural Network (CNN), Bidirectional Long Short-Term Memory (BiLSMT), and hybrid combinations CNN-LSTM-DNN, CNN-BiLSTM-DNN. The main hypothesis is that AutoML based pretrain model with algorithmic-based feature extraction produces more efficient results. Finally, the research outcome is the contribution of a new AutoML framework that automates sarcasm detection for social media domains like Amazon

reviews, Twitter, and dialog discussion comments. The proposed AutoML framework has the following steps or pipelines: model search, hyperparameters optimization, and model architecture. During each iteration of the model search pipeline, the model integrates the incongruity features extracted from the proposed semi-supervised algorithm. The AutoML framework is comprised of five models, where each model is integrated with contextual and pragmatic features.

1.2 Background

One of the early works of irony was on the irony echoic where it is expressed as a reaction to thought with a critical or mocking attitude (Sperber, 1981; Sperber & Wilson, 1986; Kreuz & Caucci, 2007). The other meaning of echoic irony is the literal meaning defined by words in the form of critical or mocking phrases and terms rather than expressions or moods. (Sperber & Wilson, 1986) according to the echoic reminder theory, irony in its Greek term is defined as “eironeia”, which means unprincipled trickery. On the other hand, sarcasm is described as speaking bitterly and originates from the Greek word “Sarazein”. These two terms “eironeia” and “Sarazein” differ slightly in meaning from word origin as described in previous sentences. Besides the literal and echoic meaning of irony, it is also defined as echoic reminder theory. (Sperber, 1981; Wilson, 1986; Keruz, 1989) the former authors defined sarcasm as speaking utterance but unable to experiment using model over irony dataset.

Another theory proposed echoic irony, which will depict the irony or non-irony expression in the form of communication expression. (Wilson, et al., 2005) the definition of irony is expressed into two categories; according to the first one, it is echoic irony where the utterance of ironic expression depends on the tune of the speaker while communicating. For example, “He is a fine friend”. The second category is written as an interpretive and attribute declaration, which is about ironic overtones but lacks the intended meaning. For example, “a tall man is a man”. These concepts of irony do not cover the true meaning of contextual clues, such as positivity or negativity. (Wilson, 2006) there are some examples reflected without alluding the utterances; for instance, “punctuality is the thief of time”, the person always late deliberately for any of appointment but the sentence is not expressing the emotion positively or negatively. (Gibbs & Colston, 2007) verbal irony is one type of irony, which is also the concept of sarcasm.

The theory of irony implies that positive and negative terms are present in the sentence. However, these positive and negative terms may not be present in the sentence. For example, “That's just what I needed today!”. But such cases are rare as most social platforms like Amazon’s reviews have noisy factors in the form of sarcasm.

The linguistic computation work of irony was mostly found in social media in the form of verbal irony. Verbal irony is often called sarcasm, it is the concept where a positive phrase

is followed by a negative term. For example, “A sister walks into her brother's messy apartment and says, "I see you're still the king of clean!"”.

(González-Ibáñez, et al., 2011) the experiment conducted for sarcasm classification is one of the original works of sarcasm or irony detection in a long text like in an Amazon review. The drawback of the research was to extract the content-based contextual patterns, which is static due to fixed patterns which will be searched entirely into all documents. For instance, it extracts the fixed tags or patterns that are formed with words, for example, “Company <xyz>”. Another point is that the content-based patterns extraction is not a recommended methodology like “Company <name>”, the word company is a fixed pattern will be occurred while searching in the document. (Davidov, et al., 2010) this work was on the Twitter domain, the sarcasm was detected from the tweets based on the hashtag “#sarcasm”. (Tsur, et al., 2010) here author proposed the algorithm Semi-supervised Sarcasm Identification Algorithm (SASI) that extracts those sentences which are tagged by humans as sarcastic (hence the “semi-supervised” part) and will identify the sarcastic sentence in Amazon reviews. (Reyes, et al., 2012) proposed another hashtag-based technique in the domain of education, humor, and politics. The work of this author was on hashtags rather than on contextual-based recognition of sarcasm. Therefore, the limitation of the research was that technique unable to identify sarcasm using tags. The Part of speech (POS) tagging is the grammatical annotation of sentence that generate tag of each word in

a corpus. The features or patterns ought to be generalized so that it can be applied over any domain dataset.

(Filatova, 2012) the Amazon MTurkers platform allows workers (“**turkeys**”) to accomplish small tasks that computers are unable to perform, such as recording audio or flagging unacceptable photos for social networks. It is the tool that generates standard rating, the scale from 1 to 5, but it is unable to mark ironic sentences in the document. (Maynard & Greenwood, 2014) the author collected the tweets based on the hashtag “#ironic, #sarcasm, and #sarcastic” but analyzed the sentence polarities using dictionaries. The polarities of words are filtered from public dictionaries which can be accessed from the programming tools.

Further, work on sarcasm has limitations, which was hashtag-based tweets collection. (Riloff, et al., 2013; Joshi, et al., 2015; Poria, et al., 2016) the work was on contextual patterns in the form of pragmatic and incongruity features as accomplished by the former researcher. The contextual patterns are 2-gram patterns, these are verbal combinations “verb” and “adverb” and “verb present” and “verb past participle” in the sentences. These POS features are patterns and strong clues for sarcasm as compared to hashtags. Because hashtag #sarcasm is not a true indicator for sarcasm in the tweet. Similarly, one of the strong works on contextual patterns was done by (Felbo, et al., 2017), the work collected

emojis and contextual clues to detect sarcasm among multiple domains like tweets, YouTube comments, and dialog discussion comments.

(Van Hee, et al., 2018) the work was on the comparisons of various authors' proposed models which performed well on challenging dataset SemEval-2018 Task 3. There were two tasks, first task was about Task A which was on the detection of verbal irony. According to the author, the competitive dataset collected irony using the hashtags #irony, #sarcasm, #not. This annotation scheme will mark each tweet as sarcastic based on a hashtag. The hashtag is not a true indicator of a sarcastic tweet therefore, required context features to detect. (Rangwani, et al., 2018; Wu, et al., 2018; Baziotis, et al., 2018) developed deep learning models to detect the context sarcasm features like incongruity (phrase and term polarity contrast) and pragmatic (e.g., emoticons) for domain Twitter. This research will investigate the knowledge gap among former authors who proposed deep learning techniques to recognize verbal irony or sarcasm for multiple domains (see Chapter 2 gap analysis with detailed synthesis on research). Furthermore, this research will provide the research questions, aim, and problem statement in the upcoming sections.

1.3 Introduction

Sarcasm is present in various domains like Twitter tweets, Amazon reviews, discussion dialogs, YouTube comments, and Forbes news. Sarcasm extraction methods are categorized into lexical, deep learning, and statistical-based methods. The features are split

into contextual features like implicit and explicit to detect sarcasm. Initially, the pattern-based features detect verbal irony (Reyes, et al., 2012). The former work of sarcasm detection was in the domain of politics and was effective in detecting the contextual features of sarcasm (Kreuz & Caucci, 2007). The contextual clues are a true indicator for the detection of sarcasm. These contextual clues are pragmatic features such as punctuations and exclamation symbols. The generalized features extraction concept was proposed by the author (Joshi, et al., 2015). The task of feature extraction is classified into explicit incongruity and implicit incongruity algorithms. The explicit incongruity is defined among terms of opposite polarities in a tweet. Implicit incongruity is described as the polarity contrast between words and phrases. The definition implies that polarity contrast occurred between the positive term and negative phrase in the sentence. For example, “it is love being ignored”, the term "love" is positive and the 2-gram phrase "being ignored." is negative.

This research focuses on exploring the gap of feature extraction algorithms for sarcasm. The feature extraction algorithm was applied to a small sample taken from the large datasets in the domain of tweets. The features are categorized into two classes explicit and implicit incongruity. The explicit incongruity is divided into many sub-features like token sequence, total positive terms, total negative terms, the overall sentiment of the tweet, and the count of polarity contrast between positive and negative words. (Joshi, et al., 2015) the former state-of-the-art core technique logistic detected sarcasm with the support of

pragmatic and incongruity types of features. On the opposite side, the deep learning model produced better results as compared to SVM (Ghosh & Veale, 2017).

It will examine the baseline method with the support of lexical features that are represented with word embeddings. AutoML is categorized into generalized models and methods: evolutionary, tree-based, and deep learning-based. Therefore, it is desired to observe the deep learning neural networks performance with comparison to the core models after inclusion of these incongruity and pragmatic features.

1.3.1 Aim and Objectives of the Research

The primary aim of the study is to detect sarcasm after the integration of features into the Machine Learning for long and short sentences. Generalized features are a requirement for the detection of irony or sarcasm in various domains. Thus, objectives of primary aim are as follows:

- It was observed that not all the contrasting words or phrases in the text are strong clues for sarcasm, however weak clues are evident mostly in informal text like in tweets. Thus, objective is to propose an algorithm that deal with contrasting features in the context of tweet. Contextual features are important clues for sarcasm detection, that is polarity contrast among phrases and terms. These patterns are extracted based on verbal pairs like noun/verb and verb/verb.

- Another form of feature is important to extract is called as pragmatic features like punctuation, capital letters, and emotions such as laugh expressions “hahaha”. The concatenation of features into deep learning model is real challenge thus need to observe various integration options with baseline feature. For instance, these pragmatic and contextual features can concatenate together with baseline features at the hidden layer of the deep learning model.

The secondary aim of this research is to classify sarcasm with a novel AutoML framework that automates the task of Machine Learning using model search and parameters optimization pipelines.

- It is novel to propose AutoML framework to initiate a model search pipeline for deep learning-model however, the real objective is to pretrain model that further fine-tune to other datasets. The primary task of AutoML is to pretrain the model over formal text like news and informal text like tweets. Then pretrain model will be saved and loaded to transfer the weights to other domains, this concept is called as transfer learning.
- There are lot of ways or strategies which fine-tune other dataset that will optimize other domains with different strategies like ‘full’, ‘last’, and ‘chain-thaw’, however, devising new strategy is real aim.

AutoML framework pipelines are model search, parameter optimization, and model architecture found in the existing frameworks: AutoML-Sklearn (Feurer, et al., 2019), TPOT (Olson, et al., 2016), AutoML-Keras (Kotthoff, et al., 2017), and AutoML-Zero (Jin, et al., 2019). These incongruity and pragmatic features are not limited to a particular dataset; these are generalized to extract from many different domains' datasets. The integration of these features in the AutoML framework models required architectural demands and transparent integration that is irrespective to details of model parameters. However, there is some limitation of existing AutoML frameworks AutoML-Keras and AutoML-Sklearn that do not have capacity to extract features from NLP social media domains. The core AutoML like TPOT is a Python based tool that optimized Machine Learning pipelines using genetic algorithms. It is applied to the data to find the best possible model for the data. AutoML-Keras and Auto-Sklearn are also python-based tool kits that are part of the Sklearn library to optimize the data to find out the best possible model. These extracted features are generalized with the POS tags, incongruity features, and pragmatic occurrences in the text which can be adapted to multiple domains.

Secondly, this research will identify the best model during the AutoML process that classifies the text into the sarcasm and non-sarcasm categories. The AutoML architecture will incorporate preprocessing data, feature engineering, automate model search and hypermeter optimization. The outcome of the AutoML search model and hyperparameter optimization pipelines is the best model that will be adapted to the various domains.

1.4 Irony and its types

Verbal irony or sarcasm are split into many types. It is classified into verbal irony, situational irony, and pure irony for multiple social media domains. In this section, these irony types are further elaborated with examples. Verbal irony is identified using polarity contrast among terms in the tweet. The polarity contrast of verbal irony contains an evaluative expression where polarity (positive, negative) is inverted between the term and phrase. For example, *“I love waking up with migraines”*, depicts incongruity contrast between terms. In this example, the first term *“love”* is a positive polarity term contrasting with the second negative term *“migraines”*. In the second example, *“I love this year’s summer; weeks and weeks of awful weather”*, love is a positive term in the literal phrase but inverted in polarity with the negative polarity phrase *“awful weather”*.

1.5 Other verbal irony or pure irony

The words in the tweets have opposite meanings however, it is not true for every case. In this example, *“Human brains disappear every day. Some of them have never even appeared #brain #human brain #Sarcasm”*. The real issue is that when recognizing sarcasm in the tweet with opposite polarity, the negative connotation makes the verb or noun negative.

Another type of irony is situational irony, the definition explains that situations that fail to meet some expectations are called situational irony, for example, *“firefighters who have a fire in their kitchen while they are out to answer a fire alarm”*. (Shelley, 2001) this sentence

is a typical ironic situation because it is an unexpected situation that fails to meet by firefighters. A sentence contains a negative polarity situation with an unexpected situation such as overcoming the fire, but it failed the expectation.

1.6 Machine Learning Techniques

The Machine Learning techniques are categorized into core techniques and deep learning techniques. Core techniques are SVM, Logistic, KNN, and ANN. The deep learning techniques are of various forms like Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and the Convolutional Neural Network (CNN). These techniques are discussed in existing Literature Review in Chapter 2. NLP is a subfield of computer science and artificial intelligence concerned with interactions between computers and human (natural) languages. It is used to apply Machine Learning algorithms to text and speech. Deep learning techniques like BERT and ELMO are the most popular due to trendy advanced topics like sentiment, emotional analysis, and sarcasm detection. Further, like to elaborate the deep learning technique in detail.

1.7 Deep learning technique

One of the primary techniques of deep learning technique is Multilayer Perceptron (MLP), it is a type of feedforward neural network. An MLP comprises of two hidden layers and perceptron with multiple hidden layers, however, networks have an input layer, a hidden

layer, and an output layer. Except for the input nodes, each node is a neuron that process nonlinear activation function. There is a forward propagation concept where MP is having multiple input initialized with weights. The network processes the information through multiple neurons from multiple hidden layers and one output layer. The output is processed using the feedforward propagation. MLP utilizes a supervised learning technique, which is called backpropagation for training. The multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable using chain-rule and weights are updated accordingly. Another concept is called learning rate which will decide how quickly or slowly weights will be updated. A deep learning model is presented with many dense layers in below given Figure 1.7.

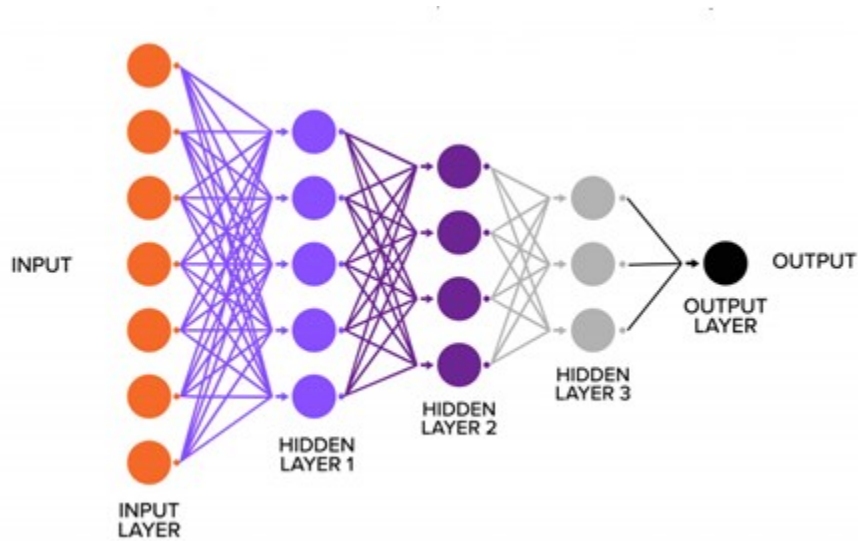


Fig 1.7: Deep Learning Network

1.7.1 Convolutional Neural Network

Convolutional Neural Networks (CNN) are biologically inspired networks used in computer vision for image and text classification (Aggarwal, 2018). The CNN architecture consists of four main layers, including convolution, pooling, fully connected, and prediction layers. The convolution layer maps each region of the given text into the smaller matrix or kernel/filter. These small matrices or filters are convolved with the image using the sliding concept, the whole process is the result of the linear operation that is the sum of product between image vector and filter. The features are blurring, sharpening, embossing, edge detection, and more features of the given image. For example, the filter size is 3×3 which is used for feature extraction from text and images vector matrix. Each filter applied Rectified Linear Activation Unit (ReLU) with the text or image matrix. Thereafter, the

filter applied and produced a smaller dimension of feature matrix-like 3×3 which further output max pool to get the strong feature. The whole concept is called dimensionality reduction because the whole image is transformed into strong vector features after applying convolution and max pool operation. Additionally, the pooling concept is used to get the strong features and removed weak features from the 3×3 feature matrix. The pooling is divided into three types of pools like max pool, average pool, and sum pool.

These baseline features are outcome of dense layer and then it performs a classification task using the SoftMax layer. In this research, it will classify a tweet into positive or negative sarcasm categories. The last neuron in the fully connected layer that will take the weight and input linear combination to the sigmoid function (returns a value between (0,1)). The output layer is fully connected and maps function in an application-specific way, that is function will be executed according to problem of interest such as prediction or classification problem. Additionally, the classification problem, if it is single output then sigmoid will be applied otherwise SoftMax activation. The SoftMax activation function transforms values between 0 and 1 for multiple outputs. If one of the inputs is small or negative, then SoftMax will produce small probabilities around 0.5. If the input is large, then it produces a large probability close to 1. The SoftMax-probability will decide whether a tweet or review is sarcastic or not. Equation 1.7.1 explains that Y is maximum after applying SoftMax function that is near to 1 that categorized as sarcastic otherwise it is 0 or non-sarcastic tweet or review.

$$y = \text{soft max}(W_0 * C_{ij} + b_0)$$

(Rodriguez-Serrano, et al., 2013) a word embedding is a learned representation of text where words may have the same meaning or different. A class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector. Here words translate into the embedding vector using embedding dictionaries. The embedding in Machine Learning or NLP is a technique that maps words to vectors, which is for better analysis, for example, "Toyota" or "Honda" are related. But in vector space, it is set close to each other according to some measure, it can strengthen the relationship between the words by calculating the vectors “king” minus “man” plus women in two-dimensional space thus getting the result “queen”. Embedding dictionaries are publicly available like Wikipedia and Glove that have millions/billions of vectors that map to words and words in pairs. It sustains to the problem that two given words always exhibit more intricate relationships. For example, the “*man*” may be regarded as the “*woman*” that both words described the human beings; on the other hand, the two words are considered opposites. The convolution operation is explained, such that the grid 2*2 convolves with a 3*3 filter to get features. These features are produced by a simple sliding window concept, as shown in below Figure 1.7.1a.

	Input	*	Kernel														
	<table><tr><td>0.5</td><td>0.2</td><td>0.5</td></tr><tr><td>0.3</td><td>.4</td><td>.7</td></tr><tr><td>0.2</td><td>0.3</td><td>0.4</td></tr></table>	0.5	0.2	0.5	0.3	.4	.7	0.2	0.3	0.4		<table><tr><td>0.5</td><td>0.2</td></tr><tr><td>0.3</td><td>.4</td></tr></table>	0.5	0.2	0.3	.4	
0.5	0.2	0.5															
0.3	.4	.7															
0.2	0.3	0.4															
0.5	0.2																
0.3	.4																
	Convolution (the sliding window linear operation)																
Sliding 1	<table><tr><td>$0.5*0.5+0.2*0.2+0.3*0.3+0.4*0.4$</td></tr></table>		$0.5*0.5+0.2*0.2+0.3*0.3+0.4*0.4$	<table><tr><td>$0.2*0.5+0.5*0.2+0.4*0.3+0.4*0.7$</td></tr></table>		$0.2*0.5+0.5*0.2+0.4*0.3+0.4*0.7$	Sliding 2										
$0.5*0.5+0.2*0.2+0.3*0.3+0.4*0.4$																	
$0.2*0.5+0.5*0.2+0.4*0.3+0.4*0.7$																	
Sliding 3	<table><tr><td>$0.3*0.5+0.4*0.2+0.2*0.3+0.3*0.4$</td></tr></table>		$0.3*0.5+0.4*0.2+0.2*0.3+0.3*0.4$	<table><tr><td>$0.4*0.5+0.7*0.2+0.3*0.3+0.4*0.4$</td></tr></table>		$0.4*0.5+0.7*0.2+0.3*0.3+0.4*0.4$	Sliding 4										
$0.3*0.5+0.4*0.2+0.2*0.3+0.3*0.4$																	
$0.4*0.5+0.7*0.2+0.3*0.3+0.4*0.4$																	

Figure 1.7.1.a: Convolution Operation

The convolution operation is performed when the image/text is represented as an input vector. The operation is a convolution operation as presented above performed by sliding window concept where it produced four operations of linear multiplication. Finally, the output feature matrix is the result of a convolution operation that is the linear operation with filter, for example, the sliding window is the linear operation of the first four elements “ $0.5*0.5+0.2*0.2+0.3*0.3+0.4*0.4$ ”, here input multiplies with the kernel of $2*2$ and added together to produce slides as given above in Figure 1.7.1a. Finally, max pooling will get the strong feature that notion is called dimensionality reduction. The max pool technique attenuates weak features but keeps only strong features with the concept of maximum pool. Further, discrete features concatenated with user embedding features as highlighted in Figure 1.7.1.b with red color that is inspired by (Hazarika, et al., 2018). This diagram explains the architecture of a CNN that input the sentence in the form of sequence of embedding or vector representation. The basic architecture of CNN is illustrated in Figure 1.7.1.b.

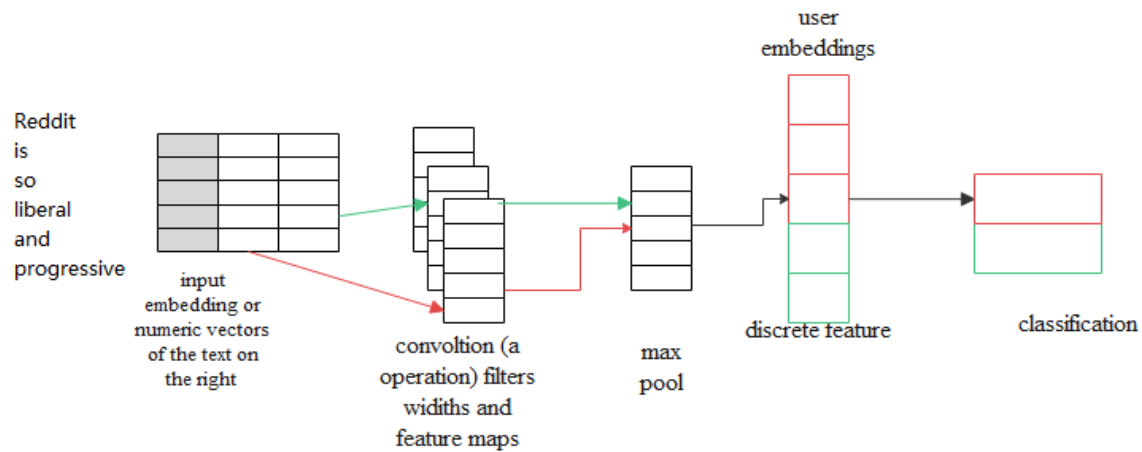


Fig 1.7.1.b: Convolutional Neural Network (CNN)

The above example sentence, “Reddit is so liberal and progressive” that first maps each word to vector from dictionary Glove and Wikipedia, then, it will apply the convolution operation. After the convolution operation, the max pool will select the strong vector. Finally, the output layer will apply the SoftMax operation to classify the sentence into the sarcastic and non-sarcastic classes. The red box represents the sarcastic and the green box represents the non-sarcastic category.

1.7.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory– commonly called “LSTM” – It is a unique sort of Recurrent Neural Network (RNN), equipped for LSTM, the dependencies in the network are the capability to remember previous vector representation and concatenated it with the next available word vector if it is required. (Hochreiter & Schmidhuber, 1997) the author worked on the problems like automatic term recognition, figurative language translation, sentiment analysis, and sarcasm detection. Automatic term recognition is the task of identifying domain-specific terms. The main advantage of LSTM is its ability to remember information for previous terms. Hence, LSTM is a model for words that has dependencies in a sequence of text, because the meaning of a phrase only depends on the words that preceded it. This is the way; it will perform context analysis to get the term dependencies in the network.

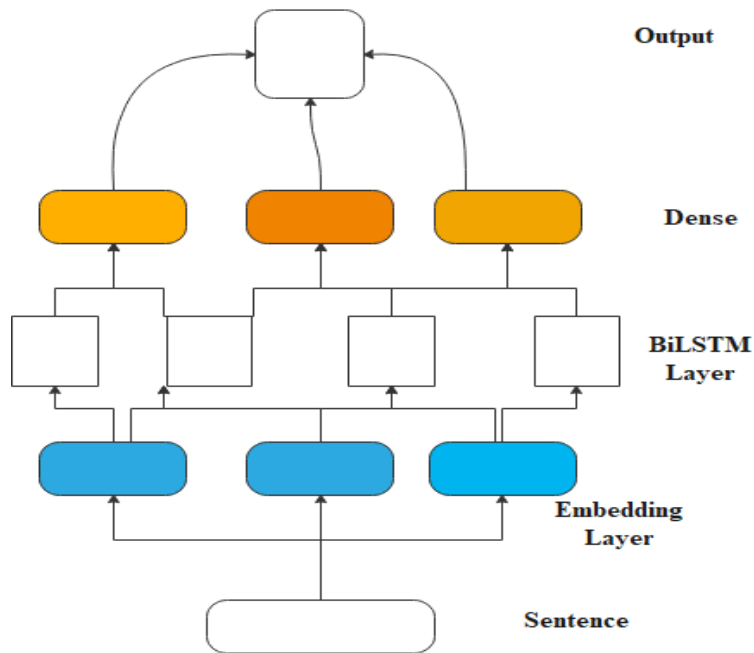


Figure 1.7.2: BiLSTM architecture for classification

BiLSTM architecture worked on the principle of LSTM, however, it recognized the dependencies among word vectors in both directions reverse and forward. The first layer is the embedding layer that takes a vector or representation of the words. This architecture takes input as a vector for each word in the sentence and further feeds into embedding layers. The embedding is the vector representation of the words w_0, \dots, w_n , here w_0 is the weight and w_n is the last weight in the network. These representation vectors are numeric vectors that feed to the input nodes. These vectors map words into two-dimension spaces x and y . Like the word “king” x and the y coordinate is $(0,3)$, on the other hand, $(3,0)$ is the opposite word “Queen” vector. This representation of words is illustrated in the x and y plane. These words and vector mappings are extracted from publicly

available large dictionaries like Glove and Wikipedia. Similarly, the “man” and “women” relationship are represented in the form of vectors and drawn on the x-axis and y-axis. The antonyms word vector is drawn into x and y planes, for instance, vectors “women” to the “man”. In this architecture, these vectors are given to “ReLU” functions to process the vector linearly in the BiLSTM model. The ReLU for short text is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. The ReLU function overcomes the vanishing gradient problem, allowing models to learn faster. BiLSTM can remember the input sequence information and its order. The hidden layers $h_{t-1} \dots h_t \dots h_{t+1}$ represented by the BiLSTM feedforward and feed backward that remember previous information of the words vector. The output unit classifies into one unit using the sigmoid function or tangent function $1/(1+e^{-x})$. The sigmoid activation function is also called the logistic function, it is traditionally a very popular activation function for neural networks. The input to the function is transformed into a value between 0 and 1. The difference between sigmoid and tangent function difference is projection between 0 to 1 and tangent projection is between -1 to 1.

1.8 Research Questions

Crowdsourcing is the practice of engaging a crowd or group for a common goal for innovation, problem-solving, or efficiency. It is powered by new technologies, social media, and web 2.0. Crowdsourcing can take place on many different levels and across various industries. Crowdsourcing is powered by new technologies and social media in the form of comments and

reviews. These reviews expressed people's opinions, sentiments, and emotions. These sentiments have been expressed in the form of reviews on Amazon, which have sarcasm. The organization needs to understand the customer behavior towards the product of interest so the companies can do marketing and distribution of an interesting category. (Davidov, et al., 2010; Tsur, et al., 2010; Reyes, et al., 2012; Filatova, 2012) few authors worked in similar domains like Twitter and Amazon to understand people's opinions, but these opinions are vital for the product if it is not noisy or trickery then it mark it as positive. The former authors observed the bitter opinion or deceitful opinion, and unprincipled trickery opinion (these are deceitful thoughts in the form of opinions of the audience) in the Amazon reviews that suspect to sarcasm or verbal irony. The author worked on multiple sarcasm domains like tweets, dialogs, and YouTube comments (Felbo, et al., 2017). It claimed that sarcasm or bitter opinion required intrinsic features to be extracted.

The research scope is to classify sarcasm into multiple domains like Amazon reviews, Twitter tweets, and dialog discussion comments. To achieve the goal, this research will raise research questions as mentioned below:

- What are the existing methodologies and techniques to detect sarcasm?
- What is the technique and algorithm which used to extract the features from various social media domains?

- Which Machine Learning technique is the best to evaluate the best results over benchmarking compared to the core approach? How to select the best model as the baseline model?
- What is the best methodology to extract the features related to various categories like pragmatic and incongruity features? Which category of features are more suitable for the sarcasm detection?
- What are the strategies to transfer the knowledge from the existing domain to a new domain?
- What is the new strategy of transfer learning that improve the performance?
- How the transfer learning will be implemented using the AutoML automation?
- What are the standard rating criteria for sarcasm detection related to social domains?
- What are standard rating criteria that developed for other domains like Twitter tweets, Reddit News and Amazon reviews?

1.9 Research Contribution

This research one of the contributions is the incongruity algorithm that will extract and detect contextual features. The main contribution of AutoML framework which automate the model search with best parameters and feature integration at architecture pipeline. The outcomes of these algorithms are sub-features that is integrated into the AutoML framework models. The features integration will increase the performance of the model which will also be input to the core

techniques (Chapter 4). There are few other factors that enhanced the performance of the model such as preprocessing of different levels and scaling techniques such as lambda, max-min, and range. Different scaling techniques have significant effect on the core ML techniques for example, lambda might effect the performance of KNN over other techniques.

To accomplish research objective, the AutoML framework is proposed that searches the best model with the integration of extracted features. The search pipeline of the AutoML framework is initiated with grid search to search the best model. It was found that best model is LSTM-DNN among all other combinations such as CNN-DNN, CNN-LSTM-DNN, LSTM-DNN, CNN-BiLSTM-DNN, and BiLSTM-DNN. The models are pretrained over the formal and informal datasets like Twitter tweets and Forbes News. The limitation is training the models for longer hours thus rigorous training which supported by Bayesian optimization required solution. Therefore, it is required that model must pretrain over large datasets using AutoML framework model, however, with restricted training cycles.

The pretrain model fine-tunes its performance over other domain datasets using novel strategy and former strategies. The other domains' fine-tuning with pretrain model and novel strategy faded-out is another contribution which fine-tunes over multiple domains such as Amazon reviews, Twitter tweets, and dialog discussion comments.

The pretrain models BiLSTM and CNN fine-tuned over the other domains' datasets using transfer learning strategies like “last”, “full”, “chain-thaw”, and newly proposed “faded-out”. The plan is to observe that which model pretrained with better accuracy over the formal data Forbes news or Twitter informal text like tweets, however, the model pretrained effectively over formal text. (Joshi, et al., 2015; Felbo, et al., 2017) the proposed framework AutoML DeepConcat pretrain models outperformed existing models over the fine-tuning of dialog discussion comments and Twitter tweet datasets. Further, this work will be extended in future for many other domains' fine-tuning like informal datasets like Instagram, Facebook, and YouTube comments, and formal dataset News dataset Forbes.

1.10 Thesis outline

Chapter 2 will like to elaborate comprehensively the review of the literature that belong to different categories like context-based, transfer learning, and AutoML. **Chapter 3** is about the synthesis to find out adapted methodology in the existing research. **Chapter 4** will discuss the features, incongruity (contrasting) polarities of the sentence's phrases, pragmatic markers. **Chapter 5** is about the existing model, its comparisons, and results. **Chapter 6**, this chapter the author proposed the novel model, framework, and its benchmark performance over different domains datasets. **Chapter 7** is about discussion, conclusion, and recommendation.

CHAPTER 2

Literature Review and Systematic Review

2.1 Systematic Review

A scoping exercise was done at the review where the key journals were likely to contain sarcasm-related literature. Further, the review was conducted by searching Elsevier's Scopus database and Google Scholar to extend the search. There were several reasons for choosing the Scopus database to complete the online review. Firstly, it is deemed to "cover approximately 22,000 plus titles and numerous international papers, amongst which there is coverage of 16,500 peer-reviewed journals in different areas" (Elsevier, 2016). Additionally, it integrates with various reputable online research journal databases. Therefore, it is selected as a vital search engine where a significant proportion of published journals and other material exist.

2.2 Criteria of inclusion and exclusion

Initially, sarcasm detection was initiated by (Kreuz & Caucci, 2007), this research worked on pragmatic clues like punctuation and capital letters to detect sarcasm. The inclusion and exclusion criteria are likely to exclude the previous years' articles. The search will initiate from 2007 and onwards.

This search excludes the book series, reviews of the conference, book chapters, and books. The reason is to exclude review papers because these papers are not having experimental details.

Inclusion criteria prioritized the search of journal-published articles along with conference papers. The inclusion criteria included articles and conference papers because many articles found are having the data aspects and experiment details. Despite the general inclusion and exclusion criteria, it searches term constitute of the patterns and contextual clues of sarcasm in Table 2.2.

Table 2.2: Inclusion/Exclusion Criteria

Inclusion	Exclusion
1. Further, investigate papers that included the pragmatic and incongruity features.	1. Exclude the systematic review of papers.
2. If multiple domain paper then adds to list with Machine Learning techniques SVM, Neural network, CNN, LSTM.	2. Remove those papers which are not cover methodology, experiment, and features in detail.
3. Give priority to highly cited papers.	3. Remove all those papers related to fugitive language. The languages like French and Hindi, the language papers can translate one language to another, these models are about classification.
4. Include all those papers which compared to the existing context and pattern-based part of speech work.	4. Exclude theoretical papers related to sarcasm and discussion papers because these papers are not about the experiment.

2.3 Search Criteria - I

To identify the relevant material, a strategy was found to build the search terms and list out keywords as follows:

1. Initially, keywords derived from the research questions in the form of search terms i.e., ‘Sarcasm’ and ‘sarcastic’ and ‘context’, ‘pattern’, ‘deep’, ‘CNN’, ‘SVM’, ‘LSTM’, and ‘neural network’.
2. Identification of synonyms or other terms will widen the search results. Use the Boolean operators AND & OR to construct the research string to incorporate synonyms and significant terms.

Following several redundant search attempts that discovered a total of 1315 research papers, as outlined in Table 2.3a.

Table 2.3a: Systematic Review

Search Terms	Attempts	Returned Documents
("sarcasm" OR "sarcastic*")	Attempt 1	1315
("sarcasm" OR "sarcastic*") AND ("Context")	Attempt 2	230
("sarcasm" OR "sarcastic*") AND ("Context") OR ("Pattern*")	Attempt 3	288
("sarcasm" OR "sarcastic*") AND ("Context") OR ("Pattern*") AND (EXCLUDE (DOCTYPE, "re") OR EXCLUDE (DOCTYPE, "cr") OR EXCLUDE (DOCTYPE, "bk") OR EXCLUDE (DOCTYPE, "Undefined"))	Attempt 4	273

2.0 Literature Review and Systematic Review

("sarcasm" OR "sarcastic*") AND ("Context") OR ("Pattern*")) AND (EXCLUDE (DOCTYPE, "re") OR EXCLUDE (DOCTYPE, "cry") OR EXCLUDE (DOCTYPE, "bk") OR EXCLUDE (DOCTYPE, "Undefined") AND PUBYEAR AFT 2006	ATTEMPT 5 YEAR >2006	242
("SVM") OR ("CNN") OR ("LSTM") OR ("KNN") OR "neural network"	ATTEMPT 6 Include ML Techniques	251
Total number of papers		251

Initially, selected 1351 articles using the “sarcasm” keyword. Further, filtered 230 articles using “sarcasm” and “context” keywords. Afterwards, at attempt 3 the search expanded with 288 articles using the “pattern” keyword. Attempt 4, selected 273 articles after excluding articles based on general exclusion criteria. After that, the year criteria excluded articles due to the usefulness of the topic that emerged from 2007 and selected 242 articles. In the end, ML techniques included the effectiveness of state-of-the-art and deep learning techniques at attempt 5 and finally selected 251 articles.

Further, reviewed the systematic review that each article in detail and selected useful articles that are closed to inclusion criteria. After a detailed review, finally filtered out 17 top articles that are part of review list as given in below Table 2.3b.

Table 2.3b: Selected Articles

Author	Article
(Erhan, et al., 2010)	This paper proposed the “last” layer strategy to transfer knowledge to other domains using pretrain model.
(Davidov, et al., 2010)	The algorithm will help to recognize the sarcastic sentences in multiple domains like Twitter and Amazon.
(Filatova, 2012)	Sarcasm was analyzed using crowdsourcing. This paper analysis is based on Amazon reviews.
(Riloff, et al., 2013)	The paper was on feature extraction of sarcasm which is presented in the sentences in the form of positive and negative situations.
(Donahue, et al., 2014)	This paper presented a full layer freezing strategy to transfer knowledge to another domain.
(Joshi, et al., 2015)	This paper was on incongruity features that were expressed as the implicit and explicit features.
(Bamman, 2015)	The research was on contextualized sarcasm detection.
(Poria, et al., 2016)	This paper proposed a deep learning-based pretrain model that works with the integration of other pretrain models-based features to classify sarcasm.
(Amir, et al., 2016)	It is using embedding to fulfill context requirements to detect sarcasm.
(Ghosh & Veale, 2017)	Magnets for sarcasm: Making sarcasm detection timely, contextual, and very personal.
(Felbo, et al., 2017)	Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion, and sarcasm. This paper proposed the concept of the pretrain model on a larger corpus and for multiple tasks using the knowledge transfer strategy ‘chain-thaw’.
(Ren & Ren, 2018)	Proposed the context based convolutional neural networks for Twitter sarcasm detection.
(Kolchinski & Potts, 2018)	Representing social media users for sarcasm detection.

(Oprea & Magdy, 2019)	Exploring author context for detecting intended vs perceived sarcasm.
(Kumar, et al., 2019)	Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. This paper was on the hybrid deep learning-based model to detect the sarcasm
(Kumar, et al., 2020)	Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM.

2.4 Search Criteria II

To identify the relevant material, a strategy was found to build the search terms and list out keywords as follows:

1. Initially, keywords were derived from the research questions in the form of the search terms i.e., ‘AutoML’ and ‘Automated Machine Learning’.
2. Identification of synonyms or other terms that will help to widen the search results. Use of Boolean operators AND & OR construct the research search string to incorporate synonyms and significant terms.

Following several redundant search attempts, discovered a total of 777 research papers, as outlined in Table 2.4a.

Table 2.4a: Systematic Review

Search Terms	Attempts	Returned Documents
TITLE-ABS-KEY ("automated machine learning" OR "AutoML")	Attempt 1	777
TITLE-ABS-KEY ("automated machine learning" or "AutoML") AND (EXCLUDE (DOCTYPE, "cr") OR EXCLUDE (DOCTYPE, "re") OR EXCLUDE (DOCTYPE, "ch") OR EXCLUDE (DOCTYPE, "ed") OR EXCLUDE (DOCTYPE, "no") OR EXCLUDE (DOCTYPE, "Undefined"))	Attempt 2	739
AND NOT ("computer vision")	Attempt 3	534
AND NOT ("crime") AND NOT ("forensic") AND NOT ("water")	Attempt 4	506
("healthcare") AND NOT ("speech")	Attempt 5	429
AND NOT ("competition")	Attempt 6	414
AND NOT ("biomedical")	Attempt 7	324
AND NOT ("bank")	Attempt 8	314
AND NOT ("traffic") AND NOT ("waste") AND NOT ("disease") AND NOT ("brain") AND NOT ("image") AND NOT ("ecommerce") AND NOT ("site") AND NOT ("fuel") AND NOT ("heart") AND NOT ("travel") AND NOT ("basketball")	Attempt 9	177
TITLE-ABS-KEY ("automated machine learning" OR "Atom" OR "tree-based pipeline optimization" OR "Auto-WEKA 2.0")	Attempt 10	183
Total number of papers		183

Using the search term from above Table 2.4a, it has collected a total of 777 and out of which 183 were relevant papers. The search excluded the papers related to different domains like ‘traffic’, ‘waste’, ‘healthcare’, ‘speech’, ‘competition’, ‘biomedical’, ‘bank’, ‘disease’, ‘brain’, ‘image’, ‘ecommerce’, ‘site’, ‘fuel’, ‘heart’ and ‘travel’. Finally selected 183 articles from the filtered articles and reviewed them. Further selected most relevant 9 articles that are belonged to the scope.

Table 2.4b: Selected Articles

Author	Article
(Feurer, et al., 2015)	Efficient and robust automated Machine Learning, the AutoML with optimization and validation techniques.
(Olson, et al., 2016)	Automating biomedical data science through the tree-based pipeline optimization, the tree-based optimization technique.
(Kotthoff, et al., 2017)	Automatic model selection and hyperparameter optimization in WEKA, the ensemble-based model selection and hyperparameters using tool library.
(Jin, et al., 2019)	This paper is proposed the famous AutoML-Keras which are used for model search and hyperparameter optimization.
(Takahashi, 2019)	It proposed parallel nodes based on AutoML which run on multiple nodes at the same time to perform Conventional pipeline steps of automation.
(Li, et al., 2019)	Towards automated semi-supervised learning, a semi-supervised meta feature extraction approach compared with state-of-the-art AutoML - Keras
(Howard, et al., 2020)	Transfer Learning for Risk Classification of Social Media Posts- proposed automated Machine Learning for risk classification using pretrain model
(Anton, 2020)	Automated Machine Learning using Evolutionary Algorithms- proposed the evolutionary-based AutoML model search which is a fast approach.
(Giovannelli, et al., 2021)	Effective data pre-processing for AutoML will automate the preprocessing transformation using AutoML

Finally, searched 17 relevant literature search criteria-I, search criteria- II and searched 9 articles as given in above Table 2.4b. In total there are 26 articles relevant to the problem scope which are related to contextual-based sarcasm detection, transfer learning, and AutoML.

2.5 Literature Review

The trend towards sentiment classification is that negative, positive, and neutral because of two articles (Pang, et al., 2002; Pang & Lee, 2008). It was starting point considering sentiment classification using Machine learning approaches. (Ivanko & Pexman, 2003) one of the early works was conducted on sentiment. It was argued that a strong positive sentence with a negative phrase makes a vaguer statement. Thus, positive phrase contrast with negative phrases marked as negative. It is important to consider both statements while experimenting like strong positive statements and moderately positive. The study reveals that the extremely negative phrase was evident in a sarcastic statement.

Sarcasm's statement is mocking and disdainful. More negative phrases are a stronger indicator of sarcasm, rather than positive phrases. The negative phrases are contradicting with any positive phrase which is a strong indicator of sarcasm. On the opposite side, positive phrases in the text are simple statements rather than mocking. Sarcasm literature is classified into patterns and context-based methods using multiple domain classification. The literature on sarcasm detection is classified into four categories: hashtag-based, context-based techniques, transfer learning, and

AutoML. Following Section 2.6 that is focused on pattern-based techniques, Section 2.7 is about context-based techniques, and Section 2.8 presented a detailed gap analysis.

2.6 Pattern-based Techniques

Sarcasm is an elegant way for the speaker or writer to convey his/her message implicitly (Davidov, et al., 2010). The search tweets are collected based on hashtag #sarcasm, which implies that the tweet has sarcasm. The training was done on imbalance data of Amazon reviews where 471 reviews are positives and 5020 are negative, the negative ratio is more in the sample. This ratio is expected because non-sarcastic outnumber sarcastic sentences as it was collected based on #sarcasm. After all, most online reviews are tagged by #sarcasm (Liu, et al., 2014). The positive tendency is more reflected in the data, the average number of stars is 4.12. The other dataset involving 1,500 tweets tag marked with the #sarcasm hashtag (Tsur, et al., 2010). The author developed the SASI (semi-supervised algorithm for sarcasm identification) algorithm. Further, it extended the idea of previous research to multiple domains. It selected a small sample dataset of 80 positive and 550 negative datasets of Amazon reviews to determine sarcasm. It selected imbalanced data because more negative reviews are stronger clues of sarcasm than positive reviews. The SASI algorithm determined the content-based pattern features like in this example “[COMPANY] CW does not CW much”, “does not CW much about CW”. Here CW represents [company] which is the high-frequency word. It extracted the features in fixed patterns because it occurred frequently like the “[COMPANY]” tag.

The classification model KNN was applied over the data based on content-based patterns and punctuation features to classify sarcasm. The performance matrix F1 was 0.83 on Amazon reviews and 0.55 on Twitter tweets even combined features like punctuation marks and verbal patterns. But the baseline results were not significant as compared to the combined feature method like baseline + discrete features have 0.83 F1 therefore, combined features were best for sarcasm detection. The limitation of the research was the small dataset for training and insignificant results on the tweets. The tweets are collected using the #politics tag and extracted based on the SASI algorithm. These tweets will feed in the form of a vector to the model for sarcasm detection. The large feature space is desirable which must be generalized to detect sarcasm based on Part of Speech (POS) tags. The POS tag represents word grammatically in a sentence like verbal features are verb-verb and verb-noun. These are extracted to observe the 2-gram phrases polarity. The polarity with positive or negative contrast with other verbs or terms in a sentence than it is called sarcasm.

(González-Ibáñez, et al., 2011) the author collected sarcastic tweets using Twitter API which were filtered based on hashtag #sarcasm. The positive sentiment emotion was collected with hashtag #happy, #joy, and #lucky, and negative emotion by hashtag #sadness, #angry, and #frustrated. In total collected 900 tweets for each hashtag category. To identify sarcasm, it extracted lexical features and pragmatic features. Lexical features are individual positive words (Chung & Pennebaker, 2007), which taken from dictionary of four categories where 64 words for each positive category. These four categories are linguistic process (adverb, pronouns), psychological

process (positive and negative emotions), personal concern (work, achievement), and spoken category (assent and non-fluences). Another type of feature is pragmatic features which are emoticons and negative emotions. The former author proposed four types of feature classes: sarcastic positive (S-P), sarcastic negative (S-N), sarcastic positive-negative (S-P-N), and sarcastic non-sarcastic (S-NS). If the tweets are sarcastic positive tweets, it was denoted by S-P. Similarly, sarcastic tweets and negative-positive were denoted by S-N, and sarcastic tweets with both polarities positive and negative were denoted by S-P-N. Here, the negative tweets depicted the polarity of the tweet and positive tweets refer to positive polarity. The polarity depends on emotions. These emotions are sarcastic negative emotion (Nemo), positive emotion (Posemo), negation (Negate), emoticons (Smiley, Frown), and Auxiliary verb (Auxvb) like is, are, am. Similarly, punctuations are the clues which are identified in the sentence pragmatically. Sarcastic tweets usually embed more positive emotion words. Similarly, more negative words embed in the negative tweets (Negate is an important feature for S-P). The author utilized core techniques: sequential minimal optimization, SVM, and logistic regression to classify these four types and unigram features, but bigram and trigram were not extracted. The unigram is referred to as one word/term, bigram terms are two terms or words, and trigrams are three words/terms. Sequential minimal optimization (SMO) is an algorithm for solving the quadratic programming (QP) problem arise during the training of SVM. SMO is widely used for training SVM and is implemented by the popular LIBSVM tool. SMO performed better as compared to other techniques on each class of features data such as F1 of S-P was 0.71, S-P-N was 0.57, S-NS was 0.65., and S-N was 0.69. This research applied the validation criteria 5-fold (Davidov, et al., 2010). K-fold cross-validation

(CV) where a given data set is split into a k-number of sections/folds where each fold represents test set. Let's take a scenario of 5-folds cross-validation ($K=5$). This process is repeated until each fold will be part of the test set. The words in sentences were tokenized from available dictionaries. The criteria are based on dictionary-based lexica such as linguistic tags and verbal tokens formulated together with word tokens of each tweet. But the issue is that most dictionaries do not cover the verbal patterns because the integrated tool-based dictionaries have limited words. The other issue is that these dictionaries are small to cover all word tokens, therefore, required a big dictionary and coverage of all verbal patterns.

(Reyes, et al., 2012) another researcher was focused on irony, humor, and politics-related tweets. The dataset constitutes of 50,000 tweets where 10,000 tweets belong to each category using four hashtags #humor, #irony, #technology, and #humor. High-frequency word vectors (the word which appears in larger number) collected from all documents following features: feature ambiguity, perplexity polarity, emotional scenarios (imaginary and pleasant), and sentence complexity. The test was conducted using a Decision Tree (DT) with ambiguity features combining the categories Humor vs. Irony with F1 0.83, Humor vs. Politics with F1 0.78, Humor vs. Technology with 0.75, Humor vs. General with 0.72 respectively. The performance was measured with polarity, which was not enough to identify the humor, but emotional features combination like Humor vs. Irony, Humor vs. Politics, Humor vs. Technology, Humor vs. General because F1 was 0.62, 0.68, 0.61, and 0.56 respectively. All features and combinations gained F1 as 0.93, 0.86, 0.86, and 0.92. The other Machine Learning techniques are SVM and logistic

regression, but no comparison was given in the experiment with DT which can be given in the future.

(Maynard & Greenwood, 2014) work was on cross-language sarcasm classification in English and Czech languages. The tweet features extracted with 2-gram (refers to two consecutive words) and 3-gram (refers to three consecutive words) were identified with content, frequency, and pragmatic features. The 2-gram and 3-gram are two words, and three consecutive words based on successive occurrences with verbal patterns. It was observed in the former work that English tweets classification performance outperformed Czech tweets. The baseline features (lexical tokens or words) with patterns (pragmatic features like punctuations) performed well. However, not all the context-based patterns found in the tweets. The pragmatic-based features of sarcasm are also playing a significant contributing role in the latest trends. The research will detect sarcasm with the support of polarities of opposite meaning such as negative and positive terms in a tweet. The disadvantage of this approach is that the tweet doesn't need to always have negative and positive terms. Some of the hashtags do not represent the true meaning of irony. Here, the author focused on verbal irony or sarcasm (Reyes, et al., 2012). For example, “I am not happy that I woke up at 5:15 this morning #greatstart #sarcasm”, here in this tweet, it is not sarcastic but mistakenly hash tagged. Sometimes, sarcasm is not present even when the hashtag indicated it. For instance, the sentence “It is not like I wanted to eat breakfast anyway #sarcasm”, it indicated negative sentiment in the phrase “not like”. The limitation of this research was manual annotation; therefore, it is necessary to propose a supervised or semi-supervised approach.

2.7 Context-based Techniques

Former research identified the contextual clues work annotated in the tweets (Kreuz & Caucci, 2007). It has experimented with human intervention on two aspects: firstly, contextual clues are in the form of verbal patterns that formed with the combination of adjectives and adverbs POS tag. Secondly, the detection of punctuation clues and interjections are important pragmatic features. Both cases were examined by human annotation of the NLP task in binary classes 0 or 1, which indicate the presence and absence of features. This work has experimented with manually extracted features that were extracted by human annotation. These features were stronger clues like punctuations, that detect the sarcasm in non-literal sentences, for instance, “I slept very well last night!”. The author claims that words with a one-to-one mapping may involve challenges like the sequence of words or order of words mapping into other languages. The author focused on punctuation marks that were identified in some sentences. Therefore, both categories of features like POS features (adjectives and adverbs) and pragmatic are essential for the recognition of sarcasm.

Edwin (2013) research evaluated word sentiment in sentences where sarcasm is evident. The methodology was initiated by tokenizing the words using unigram patterns from SentiWordNet. It is a lexical resource for words that assigns three levels of sentiment scores low, medium, and high. After that, a translator translates the English language into the Indonesian language using Google

translator. It was noted that negative words were located within the tweet where the next word is two or three words away. The word context may change the meaning of the word in the Indonesian language such as “mahasiswa” (student) when proceeded by the word Harga (price),” which makes the word “harga mahasiswa” which change the polarity from neutral to the positive word.

Similarly, the affix will change the positive words into the negative word; therefore, it applied affix connotation with a word list, for instance, if remove the affix from the “untouchable” it converts to the “touchable”, the meaning of the word will change, the contrast between positive and negative word. However, initially, collected the 980 tweets for the experiment from Twitter, it was examined that unigram lexicons and sentiment words whereas sentiment words achieved high accuracy than the lexicon single word. (Lunando & Purwarianti, 2013) there are a few other features that occurred contextually like interjections and affixes that were not effective in comparison to sentiment features. The other clues are pragmatic however, there are other clues as well like sentiment features. (Pang, et al., 2002) the current research utilized the classifiers SVM, naive Bayesian, and maximum entropy model, however, the classifiers were applied to the small sample of the dataset.

Particularly, one of the important pieces of work was on context-based sarcasm that was initiated by (Riloff, et al., 2013). The focus of the study was on sarcasm rather than on multi-tasking sentiment and emotion. It collected 35,000 tweets with #sarcasm and 140,000 random tweets. (Gimpel, et al., 2010) the Carnegie Mellon University (CMU) tagger was designed for POS

tagging so that each word tag with verbs in the sentence, which was developed by (Owoputi, et al., 2013). Further, it extracted 1-gram (one word), 2-gram (two words), 3-gram (three words), and n-gram (more words) verb phrases occurring right after a positive phrase. For example, “I love waiting forever for the doctor #sarcasm.”, in this sentence “love” term is a positive polarity whereas the “waiting forever” phrase is extracted with negative polarity. It extracted verbal features based on bigram verbal patterns like V+V i.e., verb and verb occurred in the tweet. These verbal pair patterns are defined as connotations to extract useful phrases which are part of polarity contrast with verbal terms. The phrase “being ignored” is extracted based on verbal patterns V+V. In this, “being” is a present participle, and “ignored” is the past participle. Methodology opted seven verbal patterns of a bigram, the 7 POS bigram patterns are Verb and Verb (V+V), Verb and Adverb (V+ADV), Adverb and Verb (ADV + V), “to” +Verb (V), Verb and Noun (V+NOUN), Verb and Pronoun (V+PRO), Verb and Adjective (V+ADJ). Note that CMU POS taggers produced detailed verbal POS tags than more traditional POS taggers. For example, there is just a single V tag that covers all types of verbs participles. It applied SVM from the library LIBSVM. Further, three publicly available dictionaries are used for the sentiments: Liu05, MPQA05, and AFFIN11. These dictionaries are available as a library in R however, coverage of all phrases and words through these dictionaries is questionable because these dictionaries are useful for long and short sentences. It collected 35,000 tweets with #sarcasm and 140,000 random tweets. The result was evaluated using the Twitter dataset with the F1 score of 0.51 when applied with predicate positive and negative phrases. The results were not convincing but verbal features are the main contribution of this research, thus deep learning is better in performance. This research’s limitation was the

seed word “love” which is searched by bootstrapping/seed word approach, which will search the seed word “love” in the tweets, if it is found then it will extract. It is the limitation of searching the tweet with seed words however, it can be generalized based on the verbal pattern. Therefore, this research needs to overcome these limitations to make research more generalized for tweet extraction with the support of verbal patterns.

(Bamman, 2015) this piece of work was proposed for sarcasm detection using contextual information. The contextual information is author interactions, relationships with the audience, and the immediate communicative context. The research opted few features like author features that are part of the tweet address and audience features which is the interaction between the tweet being predicted and response features of the tweet. It shown good classification results of 0.85 F1 over Twitter’s tweets dataset. Effective results would be achieved when classifying tweets based on combined features with the technique logistic regression. The sarcasm was detected when more strong features were evident in the form of contextual clues in the tweets, for example, punctuation and negative/positive polarity phrases contrast. It was also observed that pragmatic markers like exclamation marks and punctuation are not strong clues as compared to contextual-based features. The examples of explicit markers are author interactions, relation with the audience, and the immediate communicative context. LR performed better on the medium size datasets to get better efficiency.

(Rajadesingan, 2015) another author worked on detecting the sarcasm polarity based on dictionary approaches like Liu05, MPQA05, and AFFIN11. These dictionaries are available in the R tool. (Warriner, et al., 2013) the sentiment score methodology was proposed named as SentiStrength. The author incorporated the SCUBA framework, which has wide feature categories which contrast polarity between phrases and pragmatic features that gain accuracy. The limitation of the research is the single dictionary which has a limited polarity of verbal phrases and candidate terms. The author also worked on features that are shared with any other social domain, but that must be validated appropriately.

(Joshi, et al., 2015) this work is the continuation of one effective approach of integration of incongruity or contrasting features into the model. The negative and positive words/phrases are expressed as an implicit and explicit contextual incongruity. The work was inspired by the author (Riloff, et al., 2013), which claims that explicit incongruity is expressed as a contrasting polarity between negative terms and positive terms. For example, “I wake up daily relaxing hell early in the morning”. Here, “relaxing” is the positive term whereas “hell” is a negative term, and both are contrasting. Implicit incongruity is expressed as a contrasting polarity between a positive term and a negative phrase. Consider the example of implicit incongruity, “I love the color” and “But bad battery life ruins it”, the positive verb “love” in the first sentence is thwarted by the negative phrase in the second sentence “bad battery”. There are features categorized into lexical, pragmatic, explicit, and implicit incongruity. The collected dataset of 18141 tweets was prospective. The method of detecting sarcasm in the tweet was mainly done by seed word/bootstrapping

methodology. The process will search tweets with these seed words/terms. It will learn the contrasting situations that occurred among seed words and phrases in the sentences. For example, the seed word like “love” was searched into the entire tweet document, if it is found then that sentence will extract and compare for polarity contrast with the term “love”. Furthermore, the bootstrapping methodology can be explored with the other domains like Amazon reviews and YouTube comments. The bootstrapping methodology was bound to a fixed term searching mechanism rather than extracting any term polarity contrast with phrases.

(Ghosh & Veale, 2016) this work combined a dense neural network and a combination of Long Short-Term-Model (LSTM). The performance was compared with a conventional core model SVM. The data collection was based on the hashtag indicator in a tweet. However, if the tweet does not have a hashtag marked, it is still complex to understand that tweet is non-sarcastic. A total of 39K tweets were considered, 19K sarcastic, and 21K non-sarcastic. Recursive-SVM will classify tweets based on social markers such as hashtags, retweets, and references. The positional dependencies depend on the temporal (time-based or sequence-based) dependencies between the terms, that is two terms that occurred in the tweet with some distance between them, these terms are dependent on each other. For example, one word may occur at a different position which is a negative contributor, and another is a positive word, so the model recognized it. Hashtag-based tweets were collected and observed using SVM, which has shown a better F-score. The convergence of the hybrid model CNN-LSTM was faster as compared to LSTM due to the dimensionality reduction feature of the CNN model, here convergence means training time. It was

also observed that if a sentence or text lacks contextual information like incongruity features then the performance of the model CNN-LSTM is persuasive. It was also observed that increasing the filter size of the model will boost the performance. The model CNN-LSTM-DNN produced 0.92 F1 scores better than the 0.82 F1 scores of CNN.

2.7.1 Feature Concatenation – Context-based Approach

(Ghosh & Veale, 2016; Poria, et al., 2016) the concatenation-based approaches are the direction for deep learning-based sarcasm classification. Therefore, it is possible to combine various features at the hidden layer of the deep learning model. Former researchers have shown remarkable performance for sarcasm detection. This research like to explore various kinds of features in the similar direction, where context information would be integrated with the deep learning model. The primary work is the integration of features into layers, which was investigated by (Poria, et al., 2016). This work was on detecting sarcasm in the tweets with the model deep learning. CNN incorporated the sentiments, emotions, and personality features into layers of deep learning model. The input is given to the model in the form of word vectors mapped from a dictionary Wikipedia (Wang, et al., 2009). The input vectors perform the sum of the product with each element of pre-determined kernel matrix. The output is the sum of the product operation between image and filter size in the form of strong and weak features. The detail of the feature categorized is as follows: sentiment and emotion features are based on anger, disgust, surprise, sadness, joy, and fear emotions. Similarly, personality features are based on openness, conscientiousness, extraversion,

agreeableness, and neuroticism. All these features are combined with baseline sarcasm features at hidden layer.

(Tsur, et al., 2010; Joshi, et al., 2015) the core model with a smaller number of features dimension will produce better results but more features demand transformation. These former researchers utilized the core model SVM. But results in former research are not better but it was not more than 0.92 F1. (Joshi, et al., 2015; Poria, et al., 2016) the baseline model CNN reduced the dimensionality of tweet features when 20000 tweets word vectors squished out from a hidden network of the CNN. The default baseline features coupled with pretrained features at the hidden layer to classify the sarcasm with a score of 0.97 F1 which was not better than the baseline score of 0.95 F1 over 20000 tweets.

Former author worked on combining various features in deep learning model at different layers (Ren & Ren, 2018). Proposed two contextual based neural networks to sense the contextual clues. The research collected context information from the target tweet and removed the redundant information. The CNN model is supported with conversation-based context information; the model is called CNN-ALL. The CNN-ALL contains six layers of information like the input layer, convolution layer, pooling layer, non-linear combination layer, SoftMax layer, and output layer. The input tweets are about conversation tweets and target tweets. The author collected 1,500 tweets about context and 6774 are about history-based tweets. All these input tweets were used to train the model with 10-fold validation. The results exhibit the indicator that the performance would be

better from the 0.58 F1 score. Further, it is necessary to get better performance after integrating the features of the conversation context at hidden layers, thus improving the results slightly with 0.61 F1.

(Kolchinski & Potts, 2018) the former researcher proposed bidirectional RNN with GRU cells. Gated recurrent units (GRUs) are a gating mechanism in RNN. The GRU is like a long short-term memory (LSTM) with a forget gate but has fewer parameters than LSTM. These bidirectional RNN supported a dense embedding method that allows complex interaction between the author of the comments. The author claimed that embedding has more variation in the form of vector encoding. The data was collected about users' comments from 533 M Reddit News that were self-annotated. The user comments were tokenized into words with the support of dictionaries like Glove and Wikipedia. The two BiGRU layers concatenated each other by running through fully connected layers. It was evidence that the BiGRU produced better representations of texts than the CNN-based model (Hazarika, et al., 2018). 'No embed' model is no contextual features, which achieved 0.66 F1 on the full balanced tweet and 0.70 F1 on the politics balanced dataset. The BiGRU is much better in performance on political tweets classification with 0.70 F1. Here, the bidirectional model selection has decided whether concatenation required auxiliary context in the form of the feature. Therefore, to get better performance, there is a need a network that remembers the context.

(Kumar, et al., 2020) proposed a hybrid model, which was a combination of BiLSTM soft attention-based Bidirectional Long Short-Term Memory, a model that extracts the features in both directions regardless the order of the features (reverse or forward) at hidden layers. In the feature engineering phase manually extracted words, emotions, and exclamation marks from the tweets. There are eight layers in BiLSTM with an attention layer that gained 0.89 F1 and 0.91 accuracy. The attention feature vector was the output of the BiLSTM that supported context-sensitive vectors. These context-sensitive vectors are feature set like incongruity and pragmatic features. The last layer activated the sigmoid function to get the output into sarcastic and non-sarcastic. The dataset SemEval 2015 Task 11 was evaluated by the model BiLSTM-CNN. This attention-based BiLSTM-CNN model was discussed in the paper with multiple activation functions. The work addressed the importance of auxiliary features (incongruity and pragmatic features) with attention-based context vector. Further, it is required that these features integrated with a hybrid model for multiple social domains.

2.8 Transfer learning

Transfer learning's primary purpose is to transfer knowledge from one domain to other. Transfer learning classified the NLP sarcasm task into three categories.

- 1) Whether the source and target source handle the same task.
- 2) Nature of source and destination domain.
- 3) The order of the task that must be performed.

There are many transfer learning types, but here focus is on the transfer learning domain adaptation. The transfer learning types are given as follows:

Transudative Transfer Learning: If the task is the same as the source domain, it is called Transudative Transfer Learning. Further subdivision is *domain adaptation* if the domains are different like Amazon and Twitter. *Cross-lingual learning* learns the diverse languages among source and destination domains.

Inductive Transfer Learning: Label data in the target is different tasks like sentiment and sarcasm that are two distinct tasks from source to the destination domain. If the task is learned simultaneously, it is called *multi-tasking*. If a task is learned sequentially, it is called *sequential transfer learning*.

This research will use domain adaption so that the source and destination tasks are learned at the same time. Different languages are referred to as cross-lingual transfer learning that deals with cross languages however, it is beyond the scope of the study.

The main objective of the research is to find the common features in both domains. The research plans to propose a new framework that has the capabilities to search for the best model for multiple domains, integrate the algorithmic features, and optimize the hyperparameters.

In this regard, the purpose is to adapt domains that focus on two main questions:

1. Can the pretrain the model optimizes performance to other domains like Amazon reviews and Twitter tweets?
2. Can the pre-train model be deployment perspective?

According to the direction of the above questions, the primary focus was to select multiple domains as the benchmark datasets. These multiple domains are Amazon reviews and forum dialog discussion. Different domains have several challenges that humans can encounter due to contextual limitations like the size of the dataset and the commonality of features.

Transfer learning is the process of knowledge transfer from one domain to another domain. The focus of this research is to transfer knowledge from one domain to other like Amazon reviews and dialog discussion. Here domains are considered under NLP, and more specifically under the social media domain. The social media domain divides into two categories formal and informal domains. Twitter has an informal text in the tweets and Amazon reviews are the formal text. Different knowledge transfer learning strategies help to transfer knowledge from one domain to the other domain. These strategies are the last layer, full layers, and chain-thaw. (Felbo, et al., 2017) the chain-thaw strategy freeze one layer at a time (detail in Chapter 5). Considering transfer learning strategies like chain-thaw which applied to various domains like NLP social media domains, YouTube video, and visual image.

One of the initial strategies was full freezing that freezes all layers in the model. The work was on the domain of visual detection (Erhan, et al., 2010). Full layers freezing strategy was proposed that train model belief Network Restricted Boltzmann Machine (RBM) on the new dataset. Initially, pretrain the model on the dataset and saved it. Thereafter, the trained model freezes all layers to optimized to other dataset for better performance.

Another benefit is robust learning of features, the pretraining model is better in the generalized features. The pretrain based model learns more generalized features. The small number of layers train the model with better performance as compared to the larger layers, the optimization of hidden layers or dense layers depends on the problem. Another point is that the small number of layers is the non-convex optimization, it is the problem where convex regions have multiple points those need to optimize so the problem can be avoided. It works like a variance reduction technique, it is worked to reduce the training error that works to reduce the lower training to obtain the lower training error on a larger dataset. The famous optimization approach Gradient Descent optimizes the model while reducing errors in terms of actual value and predicted value. The approach used full training of all layers then freeze all layers and transfer knowledge while training the model to other domains. This approach observed more hidden layers learned a broader set of highly predictable features.

Another strategy was the “last” that leaves room to elaborate optimize to other domains (Donahue, et al., 2014). Freezing entire model layers except for the last layer when fine-tuning the model on

the different domain datasets. The former work focus was on the transfer learning strategies adapted to the video detection. The CNN model can extract generic features that adapt to other domains using the last layer strategy. The proposed method DeCAF visualize CNN layers' features while adjusting to the target domain. The t-SNE algorithms visualize the features in the 2-dimension embedding space or high dimensional feature space. The pretrain CNN shared knowledge to other domains such as Amazon images, webcam images, and DSLR camera image datasets. The transfer learning was performed with a "last" layer strategy over a small dataset but with low accuracy. Further, it is necessary to evaluate the last layer strategy to this research classification task like sarcasm.

Another aspect that was the prediction of words and sentences using pretrain models. Initially, the **BERT** model was proposed by (Devlin, et al., 2018), this model predicts the words and sentences. The model proposed bidirectional LSTM on word prediction, a few words tokens were masked to the LSTM input layer to predict any word in the sentence. The pretrain model masks words to predict a few words. The second task was to predict the sentence, where sentence B followed the sentence A. The pretrain model was train using 800M book words and 2,500M Wikipedia words. (Rajpurkar, et al., 2016) the pretrain model fine-tuned over Q/A dataset. The second dataset Multi-Genre Natural Language Inference (MNLI) was a large-scale dataset that fine-tuned for the crowdsourced classification task (Williams, et al., 2017). The goal is to predict whether the second sentence is contradicted in polarity. The other dataset QQP Quora Question Pairs is a binary classification task where the goal was to determine if two questions asked on Quora were

semantically equivalent (Chen, et al., 2018). For instance, it will predict the Quora domain whether two questions are analogous to each other or not. The question can be raised whether BERT can apply to sarcasm detection problems. To answer this question, there is a requirement to experiment with the model over the dataset.

(Felbo, et al., 2017) the research was conducted on multiple tasks classification like sentiment, emotion, and sarcasm. The model BiLSTM pretrain over 1.2 billion tweets and then saved and load to fine-tune over the other domains like YouTube video comments. The task was emotion sentences classification from Headline News and Tweets dataset. The sarcasm detection was performed on Debate forum comments (Walker, et al., 2012). The results were compared between the standard LSTM and DeepMoji transfer learning model. DeepMoji outperformed standard LSTM with the 0.92 F1 and 0.87 F1. The transfer learning strategies like ‘chain-thaw’ were proposed by the author. The result of chain-thaw strategy was 0.75 F1 on the debate forum for sarcasm detection and it outperformed other strategies. However, author proposed new model with new transfer learning strategy ‘chain-thaw’. It would be novel to observe that another new strategy that gains better performance over sarcasm detection.

(Van Hee, et al., 2018) this work is an extension that will detect irony over the competition dataset. The dataset SemEval 2018- Task 3 was given at the input layer in the form of a pre-trained word embedding vector learned from the GLOVE dictionary. (Pennington, et al., 2014) GLOVE is an embedding which provides vector representations for words. Training is performed on the word-

to-word co-occurrence statistics from a corpus, and the results represented linear substructures of the word vector. The pre-trained word vector representation is called as embedding which learned for 5 million tweets. Deep learning models like LSTM are better to detect the verbal irony because it performed significantly with 0.71 F1.

(Peters, et al., 2018) proposed the general-purpose multitask NLP model based on LSTM architecture, which trained over the Wikipedia text database. Embeddings of words feed as the vector to the input layer of ELMO. These embeddings are required in numerous NLP tasks such as sentiment, emotion, and other classification. Moreover, this model is widely used in industry and research. ELMO top layer takes input of word vector with contextual dependencies for word sense or dis-ambiguous tasks. Lower layers represent the aspect of syntax like POS tags. It records all the layer representations like POS tags, then the model learned a linear combination of these representations. Firstly, the model takes input as the context-independent word taken at a higher layer. Then, the model BiLSTM and CNN represent context-sensitive information. It works by freezing all ELMO layers and then adding ELMO vectors and concatenating to RNN.

Table 2.8: Transfer Learning Review Matrix

Dataset Sources	Datasets	Model	Methodology
(Erhan, 2010)	Visual Video	RBN	Full layer freeze strategy to transfer knowledge
(Donahue, et al., 2014)	Image /Videos from Amazon Birds UCCD	CNN	DeCaf method /TSNE for visualization, Last layer freeze strategy to transfer knowledge
(Felbo, et al., 2017)	NLP, Tweets, Dialog, YouTube	BiLSTM, MDL/MTL DeepMoji	Chain thaw freeze strategy for sarcasm
(Devlin, et al., 2018)	Wikipedia, Book Data	BiLSTM with attention, BERT, 2018	Word/Sentence prediction, Multiple purpose NLP
(Van Hee, et al., 2018)	SemEval Tweets	BiLSTM, LSTM, and CNN feature concatenation	Verbal Irony and other types
(Peters, et al., 2019)	Wikipedia Text	LSTM	NLP – multitasking

2.9 Automated Machine Learning (AutoML)

Automated Machine Learning initiate its task with a feature extraction step that converts the variable length input text into fixed-length numeric vectors (features). Further, step is required to apply the Machine Learning technique classifies the vectors. An example is a bag-of-words representation, where each numeric feature represents the count of a specific word selected based on frequency from the lexicon token. (Islam, et al., 2018) proposed a classifier may learn the hopelessness from depressed people in the text. The step of extracting features represents the text which is vital because a significant amount of information loss can occur. For example, in the bag-of-words representation, the order of the words is discarded.

Machine Learning classifiers are not good in the visualization of features rather good in classification. Lexicon-based approaches utilized dictionaries to represent features, which require a mapping between word and term. However, lexicon and rule-based approaches produced vector feature which is more understandable from human understanding. The rules are visible in the form of features extracted from these approaches. In contrast, it is hard for neural networks to represent the baseline features when processing these features into vectors in the form of embedding at hidden layers. However, it is difficult to interpret the features.

(Feurer, et al., 2015) proposed the Auto-Sklearn framework; it was the primary work that focused on the probability model to discover the relationship between hyperparameters and model. During

the iteration of model search, the framework will evaluate the hyperparameter like dropout. The model used tree-based optimization technique, named SMASH. SMASH is firmly recommended as the fast cross-validation technique. It evaluated one-fold and discarded poorly performing hyperparameter settings at an early stage. Firstly, it improved the AutoML framework work; it added a meta-learning step to start the Bayesian optimization, which boost the efficiency. Secondly, this research added an automated ensemble construction step to use all classifiers found by Bayesian optimization. Finally, this model observed over 140 datasets of multiclass classification applying Bayesian hyperparameter in Auto-Sklearn. (Hutter, et al., 2011) the SMAC optimization which run 24 hrs. with 10-fold cross-validation on two-thirds of the data and stored the resulting best performance. Therefore, it was concluded that it was a good configuration that run the ML framework. The computation cost of finding the optimized parameters is the main issue.

(Olson, et al., 2016) the tree optimization model begins with the data cleaning phase and entered the automated phase. Automation will start with the feature removal phase and create a new feature from the existing model. Further, the model selection phase selects the training model, followed by parameter optimization to make an accurate model. The validation phase of the model trains the dataset. These automation steps were performed by a tree-based pipeline optimization tool (TPOT), developed by genetic analysis over the prostate cancer dataset. The success of the automation technique TPOT find out better competitive classifiers by discovering novel pipeline operators such as synthetic feature constructions. (Feurer, et al., 2019) this author devised a new

ML Auto-Sklearn a general-purpose python Machine Learning library. The synthetic features (binary features) learn from the decision tree and combine genetic markers that generate binary feature pairs to classify. (De Rainville, et al., 2012) the DEAP library evolves a sequence of pipeline operators with a genetic algorithm that is applied over the dataset and parameters. These operators mutated and crossover with parameters to generate multiple trees thereafter select the best feature pairs using the random forest concept. e.g., the number of trees in the random forest to select features.

(Kotthoff, et al., 2017) the Auto-WEKA framework proposed the Weka tool and is available in the Sklearn library. The model was another initiative to discover the best model and its associated hyperparameters using CASH optimization technique. This ML is having many advantages as compared to other AutoML. Firstly, the regression algorithms are part of Auto-Weka for the task of classification. Secondly, it supports the optimization of all performance metrics in WEKA. Thirdly, it supports parallel runs (on a single machine) to find useful configurations faster and save N best design at each run instead utilized best available configuration. Fourthly, Auto-WEKA 2.0 is now fully integrated with WEKA. Because the focal point of Auto-WEKA is its simple usage. Therefore, providing a push-button interface that requires no knowledge about the available learning model and optimized hyperparameter. Additionally, it is fast due to limited memory (1 GB) consideration besides user dataset separate memory. The overall running time of the AutoML framework was set to 15 minutes by default to accommodate impatient users; longer runs allow the Bayesian optimizer to search more space; it would take several hours for production runs. It

was not a robust framework to include any type of deep learning model because running time of the deep learning models. The second aspect was the possibility to optimize hyperparameters using Weka's grid search and multi-search packages. However, these packages only permit tuning one learner and one filtering method at a time. Grid search optimizes only one hyperparameter that is dropout however, it is an important parameter in the deep learning model. It impacts the performance of the model whether dropout is necessary or not. The benefit was to apply techniques to modern deep learning models to get good performance.

(Jin, et al., 2019) it proposed another famous framework AutoML-fastText, unlike former AutoML, which primarily focused on deep learning tasks different from shallow models. The cloud-based AutoML made complicated configurations of Docker containers (A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code and runtime) and Kubernetes (open-source system for automating deployment, scaling, and management of containerized applications), which was not easy to configure and required more expertise. The API is easy to learn so that no need for a rich user experience. Secondly, local computer resources were limited; therefore, the graphical processing units (GPU) in the cloud required different configurations according to different environments. The Keras benefit is that it is available to every CPU instead of solely to the cloud Dockers and Kubernetes. The searcher algorithm that containing the Bayesian optimizers and Gaussian process which optimize the model parameters. The module trainer is responsible for GPU training and it trains the training data in separate modules in a parallel process.

Another, one is the graphical module, which is responsible to generate the neural network graphically. The model is trained among the pool of trained models, but the neural network size is large so that train the model in the pool therefore, it cannot be stored in a single memory. The search of the model was initiated to select the best model on the given dataset. Finally, build a graph module to generate neural architecture from simulation to the real neural network and stored in the memory. The new neural architecture was copied to the GPU memory RAM. Model Trainer trains the model on the new dataset. After that trained model is saved into the storage. The model's performance will send back as feedback to the model searcher to update the Gaussian process to select the best parameters and performance.

(Takahashi, 2019) the proposed framework that incorporates the different models of ML, raises two issues, firstly, how to include the different datasets and interfaces to the ML model, it is difficult to include all models in one unified framework. The second issue is scheduling the model search across different machines. The author proposed framework solved this issue, it consists of four modules, driver, hyperparameters, tuner, scheduler, and several executors. The user provides a dataset, an evaluation metric. The driver module will take these and pass them to the hyperparameters tuner. It used the grid search to pass hyperparameters to the model search. The driver will be activated to query the scheduler, the scheduler must be executed to select the subset of configuration. The scheduler will balance the load of the model among executors by assigning each task based on profiling information. The driver module then runs training tasks on the

executor using Apache Spark. It maps functions and obtained trained predictive models. These models are executed trained by executors' modules and the best model is selected. To answer the issues, the driver always executes and initiates predictive models using a common interface and executes all tasks at the local machine. The uniformed format data is embedding vector number that is going to be converted into a specified format before training the model, this solves the issue of data format. The framework picks the dataset physics event HIGGS¹ and signals dataset SECOM². The framework will execute model searches with 1 to 32 parallel tasks measuring their execution times. The cluster machines are used for parallel runs. The AutoML Sklearns model search on multi-node clusters with Apache Spark. The results will be compared with other AutoML like spark-MLib and spark-Sklearn. This framework is having a common interface between the framework and the ML implementations, but the performance of the dataset was less than these AutoML. Another thing is that framework is having a degree of parallelism due to its profile-based scheduling, that work will finish work on time. The degree of parallelism means that a given completion depends on concurrent tasks that run parallel to each other.

(Li, et al., 2019) this paper was about Semi-Supervised Learning (SSL) using the AutoML. However, SSL do not learn the meta-features for meta-learning which implement semi-supervised

¹ <https://archive.ics.uci.edu/ml/datasets/HIGGS>

² <https://archive.ics.uci.edu/ml/datasets/secom>

learning through AutoML. SSL is also large separation method that finetunes the hyperparameters and alleviates the performance. The proposed method bridges the gap between meta-features and SSL because meta-features have little to do during the SSL process. It explored the distribution related to meta-features that directly influence the SSL. These meta-features helped to quickly train the SSL; however, it does not support fine-tuning. Therefore, hyperparameters large marginal technique was proposed to fine-tune the hyperparameters that select the best model where the margin of hyperparameters is more. These large margins are indicators of the maximum performance of the model. The performance of AutoML-SSL has shown that performance is better as compared to SSL techniques such as SVM, TSVM, and other techniques. Another limitation was explored that Auto-Sklearn performance cannot be compared to AutoML-SSL because it worked better on unlabeled data whereas AutoML-SS works only on label data.

(Howard, et al., 2020) aimed to benchmark multiple text feature representation methods for social media posts and compared downstream with Automated Machine Learning (AutoML) models. The primary dataset was collected from the peer support forum, reachout.com (Milne, et al., 2019). (Chen, et al., 2018) the elf-reported Mental Health Diagnoses (SMHD) datasets contain posts labeled for perceived suicide risk or moderator attention in the context of self-harm. Specifically, they assessed the methods' ability to prioritize posts that a moderator would identify for immediate response. It used 1,588 labeled posts from the Computational Linguistics and Clinical Psychology CLPsych (2017) shared task collected from the Reachout.com forum. Posts were represented using lexicon-based tools, including Valence Aware Dictionary and Sentiment Reasoner, Empath,

Linguistic Inquiries, and Word Count. The AutoML used pre-trained artificial neural network models including DeepMoji, Universal Sentence Encoder, and Generative Pretrained Transformer-1 (GPT-1). It used the Tree-based Optimization Tool and Auto-Sklearn as AutoML tools to generate classifiers to fine-tune the posts. The top-performing system used features derived from the GPT-1 model, which was fine-tuned over 150,000 unlabeled posts from Reachout.com. The top system had an averaged F1 score of 0.572, providing a new state-of-the-art result on the dataset of the CLPsych 2017 task, the pretrain model result was not very good. It was achieved without additional information from metadata or preceding posts. Error analysis revealed that this top system often misses expressions of hopelessness. Besides, it had presented visualizations that aid in the understanding of the learned classifiers. This study found that transfer learning is an effective strategy for predicting risk with relatively little labeled data and noted that fine-tuning of pre-trained language models provided further gains when large amounts of unlabeled text were available.

(Anton, 2020) the work was on an automation learning framework that fast automation and finds autonomously best-learning methods. The datasets were used MINIST (LeCun, 1998) and the Titanic dataset by (FISHER, 1936). The learning methods are the data engineering and learning phase. The data engineering produces valid and optimized input for the learning phase. The learning phase will take this input and find the best algorithm which turns these input independent variables into predictions. The feature engineering initiated with data cleaning, one-hot encoding that is the binary representation of the word, and principal component analysis which select the

best possible features for learning algorithms using co-occurrences. Then the model is optimized to global optimum using Machine Learning. The results were compared with state-of-the-art AutoML frameworks like TPOT (Olson, et al., 2016) and Auto-Keras (Jin, et al., 2019) with accuracies approximate to 99% on benchmark MNIST image dataset. But the question arises here, whether deep learning models can classify the task over multiple domains.

(Giovanelli, et al., 2021) it performs optimization over the pre-processing pipeline and considered the pre-processing transformations in the prototype. The dataset SDSS³ contains galaxies photograph of more than a quarter of the sky. Preprocessing pipeline prototypes are transformation techniques that applied normalization which scales the values between 0 and 1. There is another preprocessing technique is called as imputation that will fill up the missing data with average or median value. The preprocessing pipeline works for transformation with validation rules that applied over baseline method. The order of the transformation technique is also important. The framework applied five transformation techniques with all possible orders of these prototypes. Furthermore, after transformation train the data over Machine Learning techniques Bayesian, KNN, and random forest. The limitation of this approach is that it is unable to handle the NLP

³ The data is also used in Microsoft's Worldwide Telescope (<http://www.worldwidetelescope.org>) and Google Sky (<http://www.google.com/sky>).

preprocessing transformation applied with planned training. Further, all datasets and AutoML approaches are described in below Table 2.9.

Table 2.9: Automated Machine Learning Models (AutoML) Review Matrix

Data Sources	Datasets	Model/Approach	Model/Approach Summary
(Feurer, et al., 2019)	140 Dataset	Auto-Sklearn	Bayesian optimization and SMASH tree-based classifier
(Olson, et al., 2016)	Prostate cancer	TPOT	Tree-based optimization and genetic algorithm
(Kotthoff, et al., 2017)	Amazon, Car, medical data, and many more	Auto-Weka, Multiple tree-based and basic core models.	Tree-based optimization using random forest, Genetic algorithm CASH
(Jin, et al., 2018)	Image /NLP	Auto-Keras, deep learning-based approach	Bayesian network, grid search, GPU graph generation on search model
(Takahashi, 2019)	Higgs- physical events dataset Secom- signal dataset	AutoML – Common	- Common interface to train on other nodes -Comparison among AutoML-Keras and AutoML- Sklearn
(Li, et al., 2019)	Meta-data features	AutoML - SSL	- Worked on meta-features

			- Comparison among semi-supervised AutoML – SSL vs AutoML-Keras
(Howard, et al., 2020)	-Reachout.com posts (SMHD dataset) -CLPysco 2017-task	Sklearn, DeepMoji	- Transfer learning application to NLP problem - Compared pretrain model DeepMoji - fine-task on NLP post
(Anton, 2020)	Minist Lecun et al. (1998), Iris Fisher, R.A (1936),	AutoML- fast learning	- Fast automation / best learner -Compared performance of TPOT vs. Auto-Keras
(Giovannelli, et al., 2021)	Sloan Digital Sky Survey (SDSS) – data is about galaxy	AutoML- Preprocessing	- Preprocessing applied transformation over data Compared performance of KNN, logistic, and SVM

2.10 Gap Analysis and Findings

(Kreuz & Caucci, 2007; Donahue, et al., 2014; Bamman, 2015) these former authors explored the categories of features like hyperbole, linguistic, and pragmatic. (Davidov, et al., 2010; González-Ibáñez, et al., 2011; Reyes, et al., 2012) the work was on pattern-based features to extract verbal clues for sarcasm in the sentence. (Liebrecht, et al., 2013) the statistical-based method SMO detect the pattern-based features, it is a part of the Weka toolkit. On the other hand, few authors extracted

incongruity features which are the main clues of sarcasm that are detected with a core model logistic (Riloff, et al., 2013; Joshi, et al., 2015).

(Davidov, et al., 2010) work was on domain sarcasm classification using patterns, dictionaries, and pragmatic features like punctuation. This work was on the validation technique, which is a 5-fold, however, other validation techniques will also be applied like cross-corpus validation. The better performance still required context-sensitive patterns like verbal patterns. (González-Ibáñez, et al., 2011) the former author proposed method significance is dictionary utilization and POS patterns, but it is unable to extract n-gram patterns i.e., number of words. Both approaches lack context-based features in the form of polarity contrasting between words and phrases. Like the word term, “love” positive polarity will contrast with the “being ignored” phrase positive negativity.

To overcome the context-based feature coverage in the research, one of the initiatives taken by (Riloff, et al., 2013). That work was on pragmatic features like punctuation, capital words, and implicit incongruity features that is term and phrase polarity contrast. The work was on terms and phrases contrasting polarity, which is called implicit incongruity however, it lacked the explicit incongruity that is terms polarity contrast in a tweet. The work was based on multiple sentiment dictionaries that discover that the positive sentiment which has more coverage than negative sentiment. Extracted all verbal words or terms' polarity would not get covered from these sentiment dictionaries. Thus, to solve this problem, the third-party API would cover polarity almost all word.

The sarcasm detection context was initially explored by (Riloff, et al., 2013). But it covers only implicit incongruity, on the other hand, another author (Joshi, et al., 2015) was focused on explicit incongruity. The former work was on contextual incongruity and baseline features combinedly outperformed the classification results. Another limitation was that the former author explored short text sentences domain Twitter tweets therefore, it is necessary to explore other long text domains like Amazon reviews as well.

(Buschmeier, et al., 2014) former research was on deep learning model CNN, it combined features into the model, these features fall into three sub-categories such as sentiments, emotions, and personality. These features author claimed to have worthy impact on sarcasm detection. (Walker, et al., 2012; Maynard & Greenwood, 2014) these features are concatenated at the hidden layer with baseline model features. These authors proposed a CNN model to learn baseline features from the network when passing numeric embedding of 20,000 tweets. Further, boosting the performance of the model by concatenating the baseline features with pragmatic features. *Therefore, this research will follow a similar method but with automation of deep learning models that automatically integrate baseline features with incongruity features.*

Initially, deep learning was proposed by (Poria, et al., 2016); this work utilized a pre-trained model for feature extraction with integration of other features like personalities, sentiments, and emotions. The deep learning model predicted sarcasm using the tweet context that found between verbal terms of opposite polarities. The benefit of the pretrain model is that it can be applied over

multiple domains. To overcome the limitation of detection of sarcasm in long sentences, the pretrain model BiLSTM was proposed by (Felbo, et al., 2017). The crux of the research was on multitasks like sentiments, and emotion classifications using a single pretrain model deep learning BiLSTM. The novelty of the deep learning model is the transfer learning strategy, it is called as the chain-thaw freezing strategy. The limitation of the work is pretraining the model over emoji-based tweets, but all social media domains do not have emojis in the form of clues.

(Bamman, 2015) the work of the author was on sarcasm detection using contextual features. These context features are about the tweets such as author interactions, relationship with the audience, and immediate communicative context. The author also considered these features to detect the other types of irony. (Reyes, et al., 2012; Maynard & Greenwood, 2014) like former authors have considered all types of irony.

An investigation reveals that the implicit context, explicit context, other contexts of tweets like author interactions, and immediate context of the long sentence are required to explore multiple social domains. The common features, which extracted from emotion presented in the text that is the vital clue for irony or sarcasm. Further, the training of the deep neural model can be domain adapted for multiple social media domains. The transfer learning will recognize sarcasm from multiple domains like Twitter tweets and Amazon reviews. Therefore, it is a contribution to society to invent a new pretrain model part of AutoML framework that must have capabilities to transfer knowledge effectively to any domain. To accomplish this task, there is a need to investigate a new

transfer learning strategy with the new deep learning model-based framework for sarcasm detection among multiple domains.

Verbal irony is often recognized as sarcasm; it has two types. The first one is recognized as the phrase polarity contrast named as implicit incongruity, and the second one deal with the term polarities contrast named as an explicit incongruity. Verbal irony means a polarity contrast, containing an expression whose polarity (positive, negative) is inverted between the literal and the intended evaluation. For instance, in this sentence, “*I love waking up with migraines, not*”, it has incongruity or polarity contrast between term “love” and term “migraines”, this is an example of explicit incongruity. Another example, “*I love this year's summer; weeks and weeks of awful weather*”, ‘love’ is a positive term in the literal phrase but inverted with the term ‘awful’. It was also observed that polarity inversion occurred between two successive words within a sentence. For example, ‘I love being ignored’. Here the positive word, 'love', and a negative word 'ignored' contrast in polarity that occurred in the tweet.

The context of sarcastic sentences was expressed as the pragmatic features and contextual incongruity features (Riloff, et al., 2013; Joshi, et al., 2015). *These former researchers proposed a set of features list, but none has described the algorithmic-based features. This research aims to invent the algorithm for incongruity.*

The pretraining of the model over video, visual, and image scene was initiated by (Yang, et al., 2007); the SVM-A technique was developed for domain adaption and ensemble classifier. (Erhan, 2010). One of the strategies is a full freeze approach, freezing all the layers and evaluating performance of Restricted Boltzmann Machine (RBN) deep learning model. Similarly, here's prominent strategy was the last layer proposed by (Donahue, et al., 2014), that is CNN based model freezing all layers except the last layer. *Here, synthesis is that the last and full layer freezing strategy needs to be observed. The purpose is to observe existing and to develop new strategy to confirm which strategy has more efficacy for sarcasm detection.* (Jiang & Zhai, 2007) the domain adaptation task investigated bio-natural language processing (Bio-NLP). The data is sourced from PubMed Medical Abstract, abstract text utilized the log-likelihood as the weight between target and source instances. Amazon party dataset was used for the domain adaptation with Machine Learning techniques like SVM and LR (Joshi, et al., 2012). Though, existing work do not support the common model that trains over other domains because not all domains supported common features. (Felbo, et al., 2017) this work was the contribution to multiple tasks classification and BiLSTM with the attention-based mechanism for various benchmark domains: YouTube, Tweets, and dialog discussion comments. The method in this research pretrain the model over a million tweets. The model BiLSTM is a multi-tasking and multi-domain model; however, it can adapt to observe over a single and multiple task for multiple domains.

The strategy proposed is named as chain-thaw that freezes each layer. It was claimed that the proposed strategy outperforms other former strategies like 'last' and 'full'. *One of the facts about*

these strategies is that these strategies applied to another dataset to fine-tune the model. It is part of the plan to compare these strategies and to observe any gap for contribution.

Two aspects like to explore to fill the existing research gap. Firstly, to observe the performance of pretrain model using transfer learning strategies applied over the models BiLSTM and CNN. The pretrain model will also be performed using AutoML pipeline model search and hyperparameters optimization. Secondly, the research methodology relied on an algorithmic approach to extract incongruity features and integrate into models during the model search pipeline. This research plan is to observe a semi-supervised algorithm to extract features.

(Feurer, et al., 2019) AutoML-Sklearn, the primary purpose of the AutoML framework is the automation of the preprocessing, feature engineering, model search, hyperparameter optimization, and model architecture. (Olson, et al., 2016) TPOT, another technique that was a tree-based ensemble technique that optimized the parameters and searches for the best model. TPOT and AutoML-Weka, the AutoML automation model search like AutoML-Sklearn but bundled with genetic algorithms. All the automation model focus was on the multiple model architecture but also on integration of features in the model during search pipeline.

AutoML DeepConcat starts with preprocessing; it required minimum human intervention like previously developed AutoML models (Feurer, et al., 2019; Olson, et al., 2016). However, feature engineering tools like TransmogrifyAI support general attributes or fixed data types are part of

AutoML, further, it is complex to adapt to various datasets. Auto-Sklearn (Feurer, et al., 2019), Auto-Weka (Kotthoff, et al., 2017), and TPOT (Olson, et al., 2016) frameworks lack generality but can perform feature engineering to rely on user specifications. Auto-Sklearn required user input to convert categorical data into integers (e.g., using label encoder). On the other hand, TPOT (Olson, 2016), Auto-Keras (Jin, et al., 2019) frameworks do not support preprocessing and feature engineering however, support users to perform manual data preprocessing and feature engineering.

One of the authors worked on AutoML-Keras by (Jin, et al., 2019), that work was on model optimization using a Bayesian optimization but with the model random search criteria. *The former AutoML supports feature engineering that cannot apply to all domain datasets due to the metadata of various datasets. The research plan is to adapt the AutoML-Keras like optimization strategy of random and Bayesian optimization, however; the proposed model integrates the general features based on the semi-supervised algorithm.*

This research likes to cover the gap as italicized in former research. *The main gap is the AutoML model, which can integrate the proposed algorithm features and pretrain the model to adapt to multiple domains.* The adapted methodology will be consisting of new strategy and fills the gap as highlighted italicize.

2.11 Problem Statement

The research scope is to derive performances, complexity in existing research, and trends. The trends in the literature confirms the importance of sarcasm in natural language more specifically in social media domains like tourism, shopping sites, and social platforms. The focus of this research is to overcome the problem identified in existing research. *None of the existing research focused on Automation Techniques (AutoML) to identify sarcasm that search model, optimize model by hyperparameters and recognize sarcasm among multiple social domains using a novel domain adaption strategy. Thus, pertaining the model from the beginning using the medium volume of data from formal and informal data like Tweets and News. The model with the integration of incongruity and pragmatic features over the pretrain model using the AutoML framework that would be the novel criterion.* Identifying the gap, this research would evaluate the existing models to find the best-automated model for sarcasm on multiple domains. The synthesis of the research gap in existing literature found that the AutoML framework is needed to achieve efficient model performance based on the integration of generalized features. Furthermore, this is planning to overcome two aspects of the research using the AutoML framework. The first aspect deal with model uniformity across the domains so that generalized features integrate during the AutoML pipeline model search. Secondly, our AutoML framework will outperform existing benchmark datasets for sarcasm detection using pretrain models. It will overcome the problem of sarcasm into multiple domains therefore, this model fine-tune over other domains. These domains are Amazon reviews, Twitter tweets, and dialog discussion comments.

2.12 Summary

The project's scope is not limited to contextual features but also covered other features like pragmatic and incongruity. The multitasking approach was proposed by (Felbo, et al., 2017), which was a deep learning based BiLSTM model that classifies sarcasm from dialog discussion dataset, however, other domain datasets were not considered for sarcasm. Additionally, the existing work collected a large dataset of tweets about 1.2 million that was input to pretrain BiLSTM model named as DeepMoji. The DeepMoji train over emojis-based tweets and adapted to other domains like Twitter, Amazon reviews, Dialog discussion dataset, and YouTube Comments. The former BiLSTM model classifies multiple tasks like sentiment, sarcasm, and emotions. There is need of AutoML pipeline model selection and feature integration with domain adaption strategies like last, full, and chain-thaw. The purpose of the domain adaption strategies like chain-thaw, full, and last is to transfer knowledge to other domains like Amazon reviews, Twitter tweets, and Dialog Discussion comments.

CHAPTER 3

Research Methodology

3.1 Existing Research Methodology

(Davidov, et al., 2010; Joshi, et al., 2015) few authors worked on the hashtag and context-based extraction methodology. But hashtag-based sarcasm methodology is not effective than context-based feature extraction. The hashtag-based methodology collects the Tweets which might be wrongly tagged. The contextual-based tweets identify the patterns for the sarcasm.

The context-based identification of sarcasm was more effective, therefore, this research like to explore methodology which investigate that how these features are integrated into the deep learning model. (Riloff, et al., 2013) the former author extracted verbal irony in the tweet therefore, a negative phrase followed by a positive phrase in the tweet was the context aware. It is concluded that the methodology of extraction the features lacks the concept true incongruity according to which extracted verbal phrase contrasts in polarity with other verbal terms.

This paragraph discussed the main methodologies of existing researchers. (Poria, et al., 2016), extracted the features based on the deep learning models like CNN, LSTM, and BiLSTM. The model integrated other features like sentiment and emotion at the hidden layer before SoftMax

(classification layer). The methodology was presented by (Rangwani, et al., 2018) the model LSTM that was concatenated the features at the hidden layer. Below Table 3.1 presented a few important steps for methodologies developed from existing research. It identified that features related to the context of the tweet that was followed by (Riloff, et al., 2013; Joshi, et al., 2015). Similarly, the concept of integration was presented by former authors (Poria, et al., 2016; Felbo, et al., 2017; Rangwani, et al., 2018). Therefore, to adapt the existing methodology there are a list of methodologies presented to observe like in Table 3.1.

Table 3.1: Few existing methodology overviews

Author	Methodology steps	Adapted Summary
(Davidov, et al., 2010) (Tsur, et al., 2010)	<ol style="list-style-type: none"> 1. Data gathering 2. Preprocessing 3. Feature patterns selection 4. Pattern's extraction 5. Patterns matching 6. Punctuation features 7. Data enrichment 8. Classification 	It was adapted partially such as preprocessing steps.
(Riloff, et al., 2013) (Joshi, et al., 2015)	<ol style="list-style-type: none"> 1. Data gathering 2. Preprocessing 3. Iterative feature extractions (incongruity) 4. Sub feature building 5. Classification 6. Validation and evaluation 7. Error analysis 	The features extraction methodology of both authors was adapted and further taken for improvement.
(Poria, et al., 2016)	<ol style="list-style-type: none"> 1. Data gathering 2. Preprocessing 3. Word embedding and model layers preparation 4. Model Training <ol style="list-style-type: none"> a. Sentiment features b. Emotion features c. Personality features 5. Model training without features (Baseline model) 6. Merge model with features from step 4 (at hidden layer) 7. Classification layer (SoftMax) 8. Evaluation and Results 	It was adapted based on the model concatenated the features at the hidden layer. The pretrain based model features are taken from the deep learning models and concatenated with the baseline and discrete features.

(Felbo, et al., 2017)	<ol style="list-style-type: none"> 1. Data gathering 2. Preprocessing 3. Word embedding and model layers preparation 4. Pre-training model (1 single domain dataset) 5. Transfer learning on pretrain model (different domain dataset) <ol style="list-style-type: none"> a. Fine-tuning with freeze layers b. Optimize the model 6. Attention layer (word and phrase features coverage) 7. Classification layer (SoftMax) 8. Pre-train model comparisons 9. Benchmarking 	The transfer learning strategies adapted will be compared. The methodology pertains the model so that adapted to other domains. This research would like to follow a similar methodology.
(Rangwani, et al., 2018)	<ol style="list-style-type: none"> 1. Data gathering 2. Preprocessing 3. Iterative Feature extraction (context and other) 4. Word embedding and model layers preparation 5. Pre-train model with CNN and features (1 single domain train set) 6. Transfer learning on pre-train model WITH CNN (same domain dataset another split) <ol style="list-style-type: none"> a. Fine-tuning with freeze layers b. Optimize the model 7. Attention layer (word and phrase features coverage) 8. Classification layer (SoftMax) 	The methodology integrated features in the deep learning model.
(Feurer, et al., 2015)	<ol style="list-style-type: none"> 1. AutoML pipeline 2. AutoML Bayesian parameter optimization 3. A validation technique of SMASH for parameter optimization 	The methodology of pipeline steps like model search and hyperparameters optimization with validation technique SMASH was explored.

Here, discussed some of the possible adaptations of existing methodologies in this research. This research like to adapt the methodology of preprocessing step from the existing authors (Joshi, 2015; Rangwani, H. 2018). The deep learning-based baseline model features concatenated with discrete features at the hidden layer (Poria, et al., 2016). These baseline features are in the form of

vector embeddings which are manipulated using activation functions and concatenated with discrete features at the hidden layer of the deep learning model. The discrete features were extracted from incongruity algorithms (Riloff, et al., 2013; Joshi, et al., 2015). The adapted methodology aims at concatenation of model-based baseline features and discrete features at the hidden layer of the proposed deep learning model. Thus, to train the model extra attention is required to optimize the parameters with features in the model.

The second aspect of the adapted methodology is the domain adaptation among domains. Therefore, domain adaptation is the concept of transfer learning which is using strategies to pretrain the model on the large dataset. Finally, this research methodology pretrain the models like the former authors (Felbo, et al., 2017; Rangwani, et al., 2018). However, it is compulsory to discuss the experiment using existing strategies like last layer, full layer, and chain-thaw. The transfer learning strategies were applied over pretrain model during the AutoML framework. There are few pipelining steps model selection, and hyperparameter optimization. This research will adapt a similar methodology like the former AutoML based model Auto-Keras (Jin, 2019). The Proposed model search pipeline is the framework that will decide which model is better on the baseline feature after training. These models will adapt while training using AutoML framework that consists of pipelining steps like model selection and hyperparameter optimization.

3.2 Adapted Methodology

The adapted methodology will follow few steps to solve the research problem. The first step is to gather or collect the data from sources, which are state-of-the-art sources. Next step is to

preprocess the data which derived from the existing literature. The feature extraction or feature engineering is in the direction of pattern-based features reviewed in the literature. This research will follow the existing core state-of-the-art approaches SVM, Logistic, and KNN for the experiment.

The most prominent and novel model was BiLSTM proposed by the renowned author (Felbo, et al., 2017). Few authors implement the new strategies to optimize or transfer knowledge to other domains with the pretrained model. The former author proposed chain-thaw strategy (Erhan, 2010), this strategy will unfreezing each layer step by step. This research will observe all of the existing strategies of optimizing the pretrain models to other domains Chapter 7. Finally, like to adapt novel methodology which will lead this research to propose new AutoML framework and incongruity algorithm that would contribute to the existing literature.

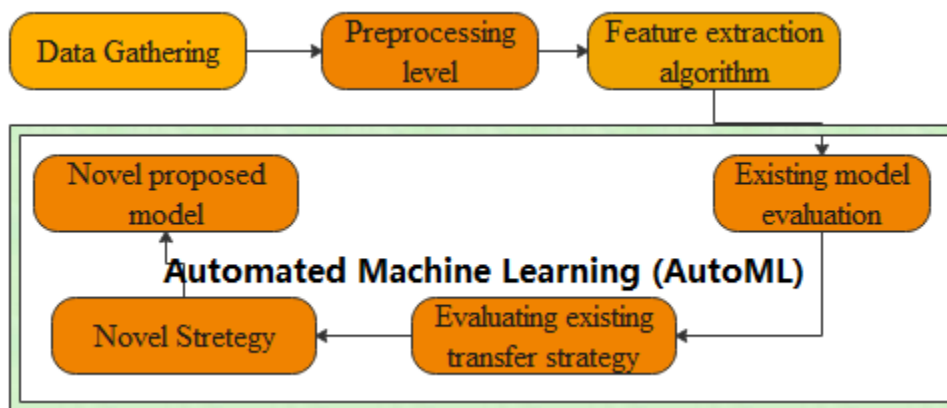


Figure 3.2: Adapted Research Methodology

Adapted methodology will initiate from data acquisition followed by preprocessing levels where data will be given to the model using the preprocessing levels like hashtag removal, stops word removal, punctuation removal, and stemming removal. Thereafter, a novel feature extraction algorithm extracts the incongruity features into implicit and explicit incongruity. Algorithms will learn features from datasets of different domains. Furthermore, existing deep learning and core models will be evaluated using these extracted features, while keeping the focus on the outcome which is to select the best method for sarcasm detection. The methodology will propose a novel model which is the goal to devise a new strategy that applicable to other domains. The domain adaption is involved multiple domains like Amazon, Twitter, and Dialog discussion comments. The proposed methodology main skeleton is new AutoML framework which will automate model search mechanism and evaluation pretrain with hyperparameter optimization.

3.2.1 Data Collection

In below Table 3.2.1a, there are datasets which are part of scope of study from various domains.

Table: 3.2.1a: Data Collection

Dataset	Purpose	Data Sources
SemEval-2018 Task3	It is utilized for feature extraction for incongruity algorithms.	(Van Hee, et al., 2018)
Twitter dataset	It is benchmark purpose, evaluation, and training of existing models.	(Riloff, et al., 2013)
Amazon Reviews		(Filatova, 2012)
Dialog Discussion Comments		(Walker, et al., 2012)
Reddit News	Reddit News data will be used to pretrain the model.	(Khodak, et al., 2017)
Twitter's Tweet dataset	Twitter tweets will be used to pretrain the model.	(Ghosh, et al., 2015)

This research planned few datasets for benchmark, evaluation, and training. (Van Hee, et al., 2018) initially this research will train the model from extracted features using algorithm applied over existing dataset SemEval-2018 Task3: A dataset famous for irony and irony type detection which is about Tweets. Thus, the first step is to develop the incongruity algorithm that will extract incongruity features.

(Van Hee, et al., 2018) the author collected tweets from the SemEval-2018 Task 3, which is a competitive dataset which was given for two of important tasks that is Task A and Task B. Task A aim is to predict the verbal irony or sarcastic tweets. Task B target is to predict the irony types like regular irony, situational irony, other ironies, a non-ironic tweet. Following Table 3.2.1b describes the distribution of tweets of task A. Task A is about verbal irony, which will extract polarity contrast features founded in the tweet. Task B is about irony and its types however it is not part of our scope. The polarity contrast features fall into two categories: implicit incongruity and explicit incongruity.

Table 3.2.1b: Irony Dataset

	Task A	
	Sarcastic	Non-sarcastic
SemEval	125	488

The dataset SemEval-2018 Task3 contains the tweets in the form of text. The aim is to extract the features of the given dataset SemEval-2018 Task3 tweet dataset. The invented implicit incongruity

algorithm aim is to extract term and phrase polarity contrast. The explicit incongruity algorithm will aim is to extract the total number of positive sentiments, the total number of negative sentiments, the distance between positive sentiment words and negative sentiment, and the sentiment of overall tweets. Besides incongruity features, the pragmatic features are also part of extraction like punctuation, exclamation marks, capital letters, and emoticons. Further, deep learning models will be part of the proposed AutoML framework, the deep learning models will be pretrained over the formal text source like Reddit News (Khodak, et al., 2017), and informal text in the form of tweets (Ghosh, et al., 2015). The purpose is to observe the performance impact over pretrain models and to evaluate whether formal text like Twitter tweets get higher accuracy than informal text Amazon reviews. Finally, the better pre-trained model will be founded which has better fine-tuned performance over other domain datasets like Amazon reviews and dialog discussion comments.

Benchmarking datasets are Amazon reviews (Filatova, 2012), Twitter tweets (Riloff, et al., 2013), and SCv2-GEN (Walker, et al., 2012) discussion dataset, these benchmark datasets selected for fine-tuning or transfer learning. Further, it is planning to apply transfer learning and evaluation in comparisons among core and deep learning models. Amazon reviews were given a rating from 1 star to 5 stars; however, it is classified into sarcastic and regular reviews. Even 1-star reviews are identified as non-sarcastic they are negative reviews but lack clues of sarcasm. The 5-stars are more regular than sarcastic but suspected due to clues of the negative and positive polarity found in a single tweet. Twitter tweets are divided into two categories: 35000 positive tweets were extracted based on hashtag sarcasm. On the other side, the author selected 140,000 tweets which

were without sarcasm hashtags, thus considered negative sarcastic tweets. SCv2-GEN dialog discussion comments are considered as benchmark dataset, where already experimented by (Felbo, et al., 2017; Walker, et al., 2012). Before discussing the methodology, this research has taken sample data from dataset like SemEval2018 task3 (Van Hee, et al., 2018). This dataset extracted features are in the form of explicit and implicit incongruity. The algorithm extracted features that have verbal irony or sarcastic tweets. Moreover, the dataset was experimented by numerous researchers for verbal irony and its types. Further, the plan is to test the pretrain model on other domains like Amazon dataset (Filatova, 2012).

3.2.2 Preprocessing

The sarcasm is expressed by hashtags, pragmatic features, URLs, and incongruity features in various domains like Amazon reviews, YouTube comments, and Twitter. The scope of the study is not limited to incongruity features, but the wide range of pragmatic features to detect sarcasm.

Tokenizing: The tweet tokenized into words using the UDPipe tokenizer (Straka & Straková, 2017), which is a public dictionary of 40 million words. The tweets or reviews split into words using POS tag. There are other tokenizer dictionaries, but this research filtered POS verbal tags associated with each token. The UDPipe in the tool is in the form of a library and coverage of verbal features are verbal forms that have a position and participle. But, before the word tokenization, the preprocessing of the tweets will perform to clean the tweet for better feature extraction.

- Removing punctuation is not compulsory because it was observed that it is an essential pragmatic clue represent emotional human linguistic expression found in the tweet (Zhao, et al., 2011). On the contrary, punctuations are vital clues to identify the radical text in the former work (Riloff, et al., 2013; Joshi, et al., 2015).
- Remove numbers: Numbers are not essential in the context. That is the reason it is likely to remove from the desired feature list. Numbers are not essential in the context that is the reason would like to remove from the desired feature list. (Filatova, 2012; Prasad, 2010) numbers are not part of the feature list on the other hand former authors considered numbers as essential, but due to large scope, this research will consider it in future.
- Remove URL: It is compulsory to prune the text from the URL (Felbo, et al., 2017). Thus, to remove the URL, the qdapregex library is utilized (Liu, et al., 2018). The plan is to test preprocessing levels with/without its impact on sarcasm.
- Remove Stop words: It is necessary to remove the stop words because it reduced the index space and improved the response time (Altrabsheh, et al., 2014; Prasad, 2010; Zhao, et al., 2011). For instance, “I”, “me”, and “myself”, there is a list of words that are not essential in the tweets, therefore, required to clean from the text.
- Stemming: Removal of stem part in the research is required to perform on the word, the verbal participle, as it is essential to extract the correct meaning of sarcasm (Riloff, et al., 2013). For example, “studies” and “studying” were stemmed from the tweets. However, removing may change the polarity of the word from negative to positive.

Other removals are #tag, and @ symbols, because it's not necessary to take as the clue of sarcasm feature sets. Some of the tags like “#sarcasm” and “@john” that appeared in the tweets depict the sarcasm. Few other symbols like punctuations and upper-case characters were not removed during pre-processing because of the importance of sarcasm detection (Riloff, et al., 2013; Joshi, et al., 2015; Poria, et al., 2016).

3.3 Exploring Features Methodology

It is planning to explore preprocessing steps and all features in the existing literature. To finalize the useful features based on existing research literature.

3.3.1 Features Desired

It is usually expected that the URL extracted from tweets is not the right clue for emotional content. Instead, URL is a noisy factor in tweets. It was expected that these emojis and emoticons are not nosier labels, but rich with emotional clues which indicate the tweet as sarcastic. Therefore, like to remove the URL. However, it is unlikely to remove the emojis and emoticons due to strong clues of sarcasm. Here, the question arises, do all or any domain contains emojis like tweets? it was considered capital letters as the pragmatic feature; a strong candidate for emotional content that is part of sarcasm analysis (Davidov, et al., 2010). Thus, it is part of the scope of features due to important clues for sarcasm.

3.3.2 Preprocessing Strategy

In the existing research, sarcasm harvesting is done by various features like the Hashtag, pragmatic features, and incongruity in several domains like Amazon reviews, YouTube comments, and Twitter tweets. The scope of this research has expanded the feature space with a wide range of pragmatic and incongruity features, which would boost the performance of the model. However, before feature extraction, the proper strategy is required to preprocess the tweets.

The first step of the methodology is initiated by the tokenization process (Bouazizi & Ohtsuki, 2016), using the UDPipe tokenizer (Straka, et al., 2016). The pragmatic feature are capital letters which were found as an essential characteristic of the emotional content during sarcasm analysis in existing research (Carvalho, et al., 2009; Reyes, et al., 2012). Emoticons are pragmatic features for criticism and contempt sentences in social media.

(Ghosh & Veale, 2016; Poria, et al., 2016; Van Hee, et al., 2018) the former authors preprocess the tweets and other domains' text to prune. Punctuation is an important pragmatic feature, it has emotional meanings (Pang & Lee, 2008; Maynard & Greenwood, 2014). It is not necessary to prune the text entirely however, few clues will prune the text to get better performance of sarcasm classification. The tweet is required to be clean from numbers besides its importance as the sarcasm clue in this research. Even URL count is vital in existing research (Joshi, et al., 2015), on the contrary, it has noisy effects so pruning the text from URL is mandatory (Donahue, et al., 2014). Therefore, it is required a proper plan to test and evaluate the model performance.

(Barbieri, et al., 2014; Straka, et al., 2016) former researchers selected the feature, the consecutive occurrences of exclamation marks. It is essential to remove stop words because it reduces the index space and improves response time. Stemming will remove the verbal participle from the term and words (Reyes, et al., 2012). It is essential to preprocess the text from stemming because it influenced the classification.

(Reyes, et al., 2012; Riloff, et al., 2013) the capital letters have emotional meaning; therefore, it is mandatory to decide that it is not compulsory to process tweets into small letters. (González-Ibáñez, et al., 2011; Joshi, et al., 2015) these exclamation marks' consecutive occurrences are essential markers for sarcasm detection, thus like to include as a feature.

On the other hand, the former researchers claimed some markers exclusion and inclusion like punctuations must be expunge during preprocessing because it would lessen the performance (Reyes, et al., 2012). It was not necessary to preprocess the text stemming because it influences the classification performance (Riloff, et al., 2013). Former research preprocesses the tweets, it is essential to apply stemming and punctuation with the proper preprocessing plan. Therefore, to fulfill the gap, it was planning to preprocess the tweets into different levels to evaluate the performance over state-of-the-art models and deep learning models. Further, proposed the preprocessing plan level as given in Table 3.3.2.

Table 3.3.2. Preprocess levels for the processing cleaning of tweets.

	Preprocessing Level	P1	P2	P3	P4	P5
P1	Remove Hashtag	Y	Y	Y	Y	Y
P2	Remove URL		Y	Y	Y	Y
P3	Remove Special Tags			Y	Y	Y
P4	Remove number				Y	Y
P5	Remove space				Y	Y
P6	Remove capital letters					Y

At the preprocessing level P1 likes to consider the removal of the hashtag, for instance, “#sarcasm”. At preprocessing level P2 likes to observe the removal effect of the punctuation on the model performance, however, it cannot be ignored like in former papers (Riloff, et al., 2013; Ptáček, et al., 2014; Joshi, et al., 2015; Poria, et al., 2016). Therefore, it is more impactful to keep punctuation because it is a pragmatic feature thus keeping it as a pragmatic feature at the level P2. It was necessary to remove all special characters like @ and hashtag because none of the existing papers defined special characters as the clue. Similarly, level P3 considered the removal of special tags, which are special tags part of the tweets, it is required to prune the tweet.

Furthermore, level P4 removes numbers and P5 removes space, and blanks. However, the numbers would be the clue of sarcasm, however, besides its importance, it is out of the scope. Finally, at level P6 plan to capital letters. Therefore, will like to perform the preprocessing according to levels given in Table 3.3.2 and evaluate the model performance for each preprocessing level separately.

3.3.3 Cognitive Algorithm for Irony Detection

The purpose of the cognitive algorithm for irony detection is to implement an algorithm that extracts the implicit and explicit incongruity features with an automated ML. These features will help to detect the sarcasm from the various domains Twitter tweets, Amazon reviews, and discussion dialog comments. (Joshi, et al., 2015) the author proposed verbal irony detection technique which was better than previous technique to detect sarcasm (Riloff, et al., 2013; Joshi, et al., 2015). It was better than the human intervention-based annotation of extracted polarity (Ramteke, et al., 2015). (Joshi, et al., 2015) that work proposed different types of incongruity features which are implicit and explicit. There is required the semi-supervised algorithms that fill-up the gap of extracting incongruity features extraction.

3.3.4 Features Extraction

It is important to find a better feature extraction method. Pruning the data from special characters and URL is necessary due to noise. Feature extraction techniques will be able to extract various category of features like hashtag, lexical, and contextual. Feature extraction techniques are either in the form of algorithms or procedures. Feature anatomy divides the feature extraction using algorithm for incongruity and procedure for pragmatic features. Another type of features are lexical-based features as illustrated below Figure 3.3.4. This research will follow both types of feature extraction techniques, algorithmic and procedural.

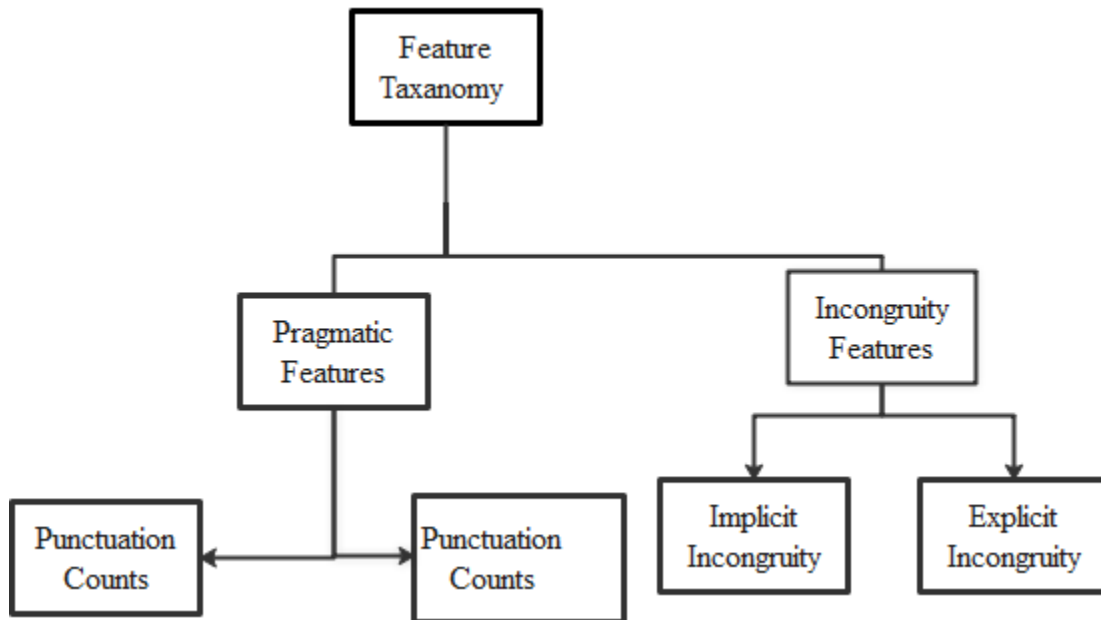


Figure 3.3.4: Feature Taxonomy

3.4 Feature's taxonomy

Features are categorized into lexical, contextual, and pragmatic features, which are extracted based on patterns like bag-of-words, n-gram patterns, and POS tags. (Camp, 2012) the bag-of-words are about the total occurrences of the words in the document. The frequency of words is not an indicator of sarcasm but rather it indicates the strength of the words. The author's focus was on bag-of-words features; this selection was based on human judgment on sarcastic sentences (Kreuz & Caucci, 2007). However, the contextual clues are true indicators, thus it will consider two categories of features pragmatic, and incongruity. Initially, the work of incongruity was observed in the form of contradiction of polarity among words and phrases. But in existing research, proposed the algorithm that analyzes the incongruity or polarity contrast features. The contextual

clues were investigated by (Riloff, et al., 2013) the polarity contrast among terms and phrases defined as the positive polarity word followed by a negative polarity phrase.

(Joshi, et al., 2015) work was on the incongruity types where incongruity is classified into implicit and explicit incongruity. However, the feature is related to polarity contrast among terms and phrases, but other features are also part of the experiment. The features are total positive/total negative, the distance between positive and negative terms in the tweet, and the overall sentiment of the tweet. The distance between positive and negative terms denotes the co-occurrence proximity which exhibit that given sentence is complex and/or simple. However, complex sentences are also sustained with sarcasm, it is required to mark as a feature. Furthermore, similar features would be extracted from a semi-supervised feature extraction algorithm.

Further, extending the concept of domain incongruity features among terms and phrases. This section explained the incongruity features which is divided into two categories: implicit and explicit incongruity features. Implicit incongruity is defined as contrasting features according to which a positive candidate term and negative phrase contrast each other. The explicit incongruity is defined as contrasting terms found in the sentence. For instance, in this sentence, 'My tooth hurts! Yay!', the negative word "hurts" is incongruous with the positive term "Yay." It denotes the explicit incongruity feature between the two terms. Furthermore, elaborating the concept of harvesting the tweets for polarity contrast among terms and phrases. (Tsur, et al., 2010) the methodology like to extract incongruity feature that is based on seed word like "love" if it found in any tweet, it will select that tweet to analyze the negative sentiment found in that tweet. But it

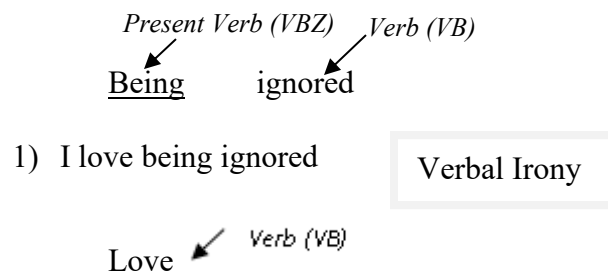
is not a uniform methodology because it depends on seed words (fixed terms will be search into entire corpus) only rather focusing on all the other terms.

Another milestone of the research is the generality of the features of different categories (Figure 3.3.4). The generality would achieve with the inclusion of the verbal patterns and intensifiers which would annotate after extracting words and phrases from patterns such as 1-gram (one word), 2-gram (two words), and 3-gram (three words). It is considered similar incongruity features that fall under the class of lexical n-gram patterns. Features belong to the hyperbolic category such as interjections, punctuation marks, intensifiers (like a verb and adverb). The combination of lexical and intensifier features is also important to extract that is part of the incongruity feature category. The reason is that when 1-gram and 2-gram verbal patterns are extracted, it will contrast in polarity among words and phrases. However, more than 3-gram are not considered due to the limitation of verbal patterns in the context of the tweet.

The features are divided into the lexical, contextual, and pragmatic categories which are based on verbal patterns, intensifiers, and symbols found in the tweet. However, other features like syntactic features and POS tags are the main categories. Therefore, these sub-categories lexical, POS tags, n-gram patterns, and syntactical features are strong clues for sarcasm.

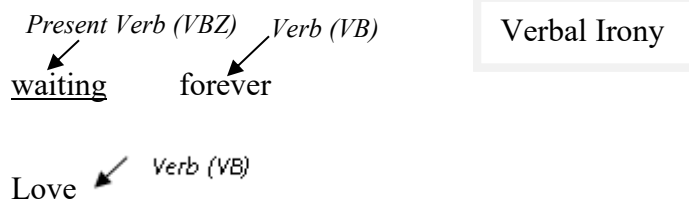
3.4.1 Verbal irony Detection

According to former authors, the verbal irony is also referred as sarcasm (Wilson, 2006; Wallace, 2015; Joshi, et al., 2015). The implicit and explicit incongruity clues are reasons to detect the sarcasm or verbal irony in the former research. Verbal irony would find in below given sentence.



Sentence 1 is composed of the term “love”, which is a positive polarity verb proceeding negative polarity phrase “being ignored”. The part of the participle of the terms is the present participle verb and the past participle verb. In the example, the negative phrase is “being ignored” contrasting in polarity with “love”, which is a candidate term. The candidate term occurred in the first position as an action verb in the tweet. The contrast of polarity between the candidate term and the phrase is one of the main clues of verbal irony or sarcasm (Riloff, et al., 2013).

2) “I love waiting forever for the doctor #sarcasm”



Example 2 might indicate a tweet that is not sarcastic or regular irony. These tweets were searched by API in R by the “Tweetr” library.

3.4.2 Incongruity Features

(Joshi, et al., 2015), the existing approach extracted seven patterns of incongruity features that occurred among terms. There are seven verbal patterns and its combinations which are important: for instance, V+V, V+ADV, ADV+V, "to" +V, V+NOUN, V+PRO, and V+ADJ. The negative phrases were extracted using verbal pattern V+V and expressed as the present participle. The second verbal term is described as the past participle for instance, "being ignored" and "getting hit."

The verbal irony or sarcasm is defined as the concept of incongruity (contrasting polarity) among terms however there is no algorithmic based features extracted approach proposed by former researchers (Wilson, 2006; Wallace, 2015). To fulfill this major gap, this research will propose the algorithms that categorized as implicit and explicit incongruity.

3.4.3 Syntactic Features

There were many features investigated by previous authors: POS and n-gram patterns. Like, the author (Ptáček, et al., 2014), explored the essential category of features to detect sarcasm like n-gram and POS. Similarly, (Rajadesingan, et al., 2015) explored POS and n-gram features that were experimented with a semi-supervised approach. The former research used POS tag to annotate the

tweet to extract verbal features and used n-gram to get the verbal features with the help of unigram, bigram, and trigram patterns. The verbal patterns of bigram are based on patterns like verb and verb (V+V) occurrences and verb and adverb (V+ADV) in the sentences. Due to structural consideration, both patterns verb and adverb (V+ADV) and adverb and verb (ADV+V) are needed.

3.4.4 POS Tag

POS tag is used to represent the patterns present in the English language, there are mainly five POS tags 1) Noun 2) Pronoun 3) Adjectives 4) Verb 5) Adverb 5) Preposition. Different combinations of POS tags are utilized to detect sarcasm.

The former authors have opted tags like the noun, verb, and adjective, the ratio of noun vs. adjectives, the ratio of verb vs. adverb, and the number of negative verbs. For example, positive verbs are extracted with the negative situation according to the former author (Riloff, et al., 2013). The verbal pattern “being ignored” which is having present participle “being” followed by past participle “ignored,”. All the POS tags are shown in Table 3.4.4.

Table 3.4.4: POS Tag

POS tag used	Abbreviation	Research Sources
Noun, verb adjective Verb (present participle) + Verb (past participle) Adverb and verb To and verb Verb and noun Verb and prop Verb and adj	N+ADV V+V ADV+V TO+V V+N V+PRO V+ADJ	(Riloff, et al., 2013) (Ptáček, et al., 2014) (Joshi, et al., 2015)
Verb, noun, adverb adjectives The ratio of verb vs. adverb The ratio of nouns vs. adjectives	V+N+ADV	(Akhtar, 2010)
Verb	V	(Kouloumpis, et al., 2011)
Adverb and Adjective, Adjective and Noun, Adjective and adjective	ADV+ADJ ADJ+N ADJ+ADJ	(Oraby, et al., 2017)

(Riloff, et al., 2013; Ptáček, et al., 2014; Joshi, et al., 2015) the former researchers used seven verbal patterns for phrase extraction using lexical 2-gram pattern as mentioned in the above Table 3.4.4. The algorithm would consider 3-gram patterns in feature space by authors (Kouloumpis, et al., 2011; Oraby, et al., 2017). However, this research will consider an experiment with all possible parts of participles to get maximum combination of patterns, which would be vital for sarcasm detection.

3.4.5 Lexical Feature

The lexicons are the tokens, which are generated with the tokenizer UDPipe (Straka, et al., 2016). Further, these tokens were filtered using verbal patterns, verb-and-verb pair that is consecutive occurrences in the tweet. In recent research, n-grams words or sequence of words were identified using unigram (one word) and bigram (two sequence words), and trigram (three sequence words)

patterns. Few researchers found that unigram patterns were useful than bigram and trigram for sarcasm detection (Riloff, et al., 2013; Joshi, et al., 2015; Kreuz & Caucci, 2007). The comparison among all these patterns unigram, bigram, and trigram was not explored by recent researcher (Zhang, et al., 2016). These n-gram patterns' main benefit is dimensionality reduction because the reduced features space would scale the performance of deep learning model. This procedure is called tokenization using 1-gram and 2-gram patterns. But the main purpose of tokenization is to filter important patterns thus useful patterns will be included.

The sarcasm occurred in the situation or event where negative phrases present with positive humor terms in the sentence. These features are called incongruity features because negative situation phrases contrasting with positive terms or words. For example, in this sentence “it was a laughing hectic situation during drama play”. Here, humor “laughing” is a positive term, but “hectic situation” is a negative situation or phrase. Thus, contrasting polarity is expressed in the form of incongruity to reflect sarcasm. Initially, the sarcasm theory is called as verbal irony in the verbal communication, which is expressed with opposite polarity in the tweet. It was observed that these opposite polarities founded in the former research (Colston & Gibbs, 2007). (Stock & Strapparava, 2005) the former research findings were opposite concepts of producing funny senses. (Reyes, et al., 2012), the work was on humor vs. irony that was evaluated with the concept of incongruity or contrasting polarities in a situation. Therefore, the verbal irony is identified from tweets with the concept of opposite polarities, similar humor evaluated with the concept of sentiment polarities.

3.4.6 Incongruity Lexical Features

The context of sarcastic sentences was expressed by incongruity and pragmatic features (Riloff, 2013; Joshi, 2015). Initially, there are two kinds of incongruity features explicit and implicit.

3.4.7 Incongruity features Methodology

The implicit incongruity feature is the concept which is expressed in sentence as polarity contrast occurred between positive candidate term and negative phrase (Riloff, et al., 2013). Similarly, another researcher observed a polarity contrast between candidate terms and phrases (Joshi, et al., 2015).

The terms and phrases are annotated with opposite polarities, where positive phrases follow the negative candidate term. The order of phrase and term is not important. Therefore, a similar methodology would like to adapt but the order is irrespective during extraction, that is positive term may occur first or the negative phrase.

There are seven 2-gram lexical patterns are given in Table 3.4.5. The negative phrases will extract from POS tag V+V patterns where verb “V” represents present participle, and “V” as the past participle, for instance, “being ignored” and “getting hit”. The verbal patterns that occurred as polarity contrast in the tweets as shown in Table 3.4.7.

Table 3.4.7: Incongruity Features verbal types with bigram patterns

Candidate Term (Verb Positive/Negative)	Positive/Negative Patterns
VB	V+V
Verb (VBP) Present Participle	V+ADV
Verb (VBG) Gerund	ADV+V
Verb (VBN) Past Participle	TO+V
Verb (VBD) Past Form	V+PROP
Verb (VBZ) Present Participle 3 rd Person Singular	V+ADJ
	V+NOUN

(Camp, 2012) initially, the work of incongruity was on embeddings of the words and phrases. But the incongruity or contrast theory defined the phrase polarity and term polarity contrast. (Riloff, et al., 2013) the specification of phrases was defined as the positive term followed by a negative phrase. (Joshi, et al., 2015), further classification of incongruity was expressed like implicit and explicit features nevertheless these verbal features were extracted from bigram lexical patterns as given in Table 3.4.7. The explicit incongruity is handled by prominent researchers (Ramteke, et al., 2015; Riloff, et al., 2013), where two co-patterns contrast to each other by polarities. For instance, an incongruity is observed in the sarcastic sentence like ‘My tooth hurts! Yay!’. Here, the negative word “hurts” is incongruous with the positive word “Yay.”. (Joshi, et al., 2015) which built many sub-features based on existing explicit incongruity features: total positive/total negative, the distance of the largest sequence of positive and negative term pairs, and overall sentiment of the tweet. This methodology will opt similar features, however, calculate the distance between two terms whether it is largest or smallest among terms in the sentences of the tweet. However, it can be assumed that proximity between the terms might get better performance whereas larger proximity would change the contextual meaning.

(Riloff, et al., 2013) the former research the implicit incongruity features were extracted positive words and negative phrases. (Joshi, et al., 2015) this research extracts the negative noun phrases and positive verb phrases though extracted with a verbal participle. This research will adapt a similar methodology with the larger set of verbal participles. (Riloff, et al., 2013) the phrases ‘being ignored’ extracted with some of the limitations, such as ‘being’ extracted with the present verbal participle (VBP) and ‘ignored’ as a past verbal participle (VBP). It is a plan to consider future participles but filtered out auxiliary patterns (VBZ) from the list.

3.4.8 Pragmatic Feature Methodology

(Riloff, et al., 2013) the existing literature has presented the clues for sarcasm. These clues are lexical features like unigram, bigram, n-gram with hyperbole intensifier features. For instance, the role of interjection ‘gee’, ‘gosh’, punctuations like question marks ‘?’ and ‘!’. (Kreuz & Caucci, 2007; Carvalho, et al., 2009) numeric features like punctuation are one of the important pragmatic features. (Joshi, et al., 2015) another promising research initiative in deep learning was the integration of pragmatic features like punctuation count.

The adapted methodology has shown significant usage of features such as laughter expression, hyperbole, and useful markers heavy punctuation for sarcasm detection, for example, “Protein shake for dinner!! Great!!!”. Similarly, the usage of emoticons was observed in this example “I LOVE it when people tweet yet ignore my text X- (” and capital letters in “SUPER EXCITED TO WEAR MY UNIFORM TO SCHOOL TOMORROW!! :D lol.”. The use of interjections in these

examples are as follows: “3:00 am worked YAY. YAY.” and “Your intelligence astounds me. LOL”. However, like to include these extracted features from the former researchers (Kreuz & Caucci, 2007; Carvalho, et al., 2009; Joshi, et al., 2015; Rangwani, et al., 2018).

3.4.9 Emoticons / Emojis

Emoticons are expressions often expressed by people in natural language using various social platforms. For example, when a person likes to express a smile, the smileys are expressed as emoticons. Emojis are a version of emoticons that represented the more intense feeling of people in natural language. For example, a fine-grain emotion icon expressed happiness and a smile.

(Barbieri, et al., 2014; Carvalho, et al., 2009) initially, former authors discovered that emoticons expressed literal meanings for sarcasm. (González-Ibáñez, et al., 2011) selected smileys and frown emoticons for sarcasm detection. It helped how to identify negative and positive tweets during sarcasm detection. Further, proposed positive and negative emotions with a lot of laughs “lols” feature by (Joshi, et al., 2015). Similar feature extraction methodology will consider a lot of laugh “lol” and smileys emoticons as given in below Table 3.4.9. (Felbo, et al., 2017) this research collected 64 different types of emojis based on the tweets in million which are an important clue of the sarcasm. It is not considering these emojis features, because, of exclusive features towards a particular domain. Further, it would consider generality for multiple domains but not all domains are having similar emojis. Although, there are many domains where emojis have limited usage (Ben Eisner., et al., 2016).

Table: 3.4.9: Emoticon features by former researcher

Emoticons Features	Research Sources
Positive emoticons are smileys	(González-Ibáñez, et al., 2011)
Expressions for laughter	(Filatova, 2012)
Emoticons with lols	(Joshi, et al., 2015)
Emojis of 64 sets of features	(Felbo, et al., 2017)

3.5 Existing Machine Learning Techniques

At the earlier era, the popular core techniques for sarcasm detection are regression technique (Davidov, et al., 2010; González-Ibáñez, et al., 2011; Bamman, 2015), SVM (González-Ibáñez, et al., 2011; Riloff, et al., 2013; Maynard & Greenwood, 2014; Ptáček, et al., 2014; Joshi, et al., 2015) and decision tree (Reyes, et al., 2012). (Poria, et al., 2016; Felbo, et al., 2017; Van Hee, et al., 2018), Few authors proposed deep learning models for the detection of sarcasm. The former author worked on the deep learning technique, the BiLSTM pretrain model was a transfer learning model that proposed chain-thaw strategy for multiple domains.

3.5.1 Regression Technique (LR)

The regression model predicts the sarcasm in the tweet. It identifies the relationship between the dependent and non-dependent variables. In the case of sarcasm, the classifier detects the sarcasm into the sarcastic and non-sarcastic categories. The independent variables are author interaction, interactive audience, and the response features of the user in the tweet. The response of other users for the tweet are expressed in the form of the interaction using emojis. The prediction of the model LR classified the tweet into sarcasm and non-sarcasm classes. The result was

significant if it is more than 0.5 then output is classified as sarcastic. Initially, it performed worse with a score of 0.49 F1 on the LogR model with positive and negative emotion in the tweet (González-Ibáñez, et al., 2011). (Bamman, 2015) another author proposed the baseline model evaluated on the tweet without pragmatic features like punctuation count with a 0.47 F1 score, which was similar performance. Therefore, the regression models are not effective models for sarcasm detection. In this research, the contextual features are treated as incongruity and pragmatic features, therefore, adapted methodology will not support interaction and communication features for the experiment.

3.5.2 SVM

SMO is an algorithm that is a variation of an SVM classification model. It demarcates the two classes by drawing a hyperplane between two types. For the sarcasm, it performed moderately with 0.57 accuracy over two classes sarcasm and non-sarcasm compared to the regression technique (González-Ibáñez, et al., 2011). The dataset was analyzed for the ridicule sentences and phrases of the tweets, e.g., “I love #sarcasm,” whereas ridicule tweets containing phrase “lol thanks” in sentence “I can count on you for comfort # sarcasm”. Although the inclusion of discrete features will be part of proposed research experiment, the model will extract features like the former author (Joshi, et al., 2015). The author performance on task was effective than the previous authors (Riloff, et al., 2013; Maynard & Greenwood, 2014), it was observed over tweets effective performance of 0.88 F1 as compared to the 0.47 F1 and 0.41 F1 respectively. Therefore, this research like to automate the feature extraction with polarity contrast concept as mention by the

former work. Therefore, incongruity features are essential feature for the detection of sarcasm. However, there is a need to get better performance for SVM over sarcasm detection using features.

3.5.3 Deep Learning Model for Big data

(Poria, et al., 2016) the former researcher proposed a deep learning model (CNN) with the baseline features combined with pretrain model-based features at the hidden layer. These features fall into three sub-categories such as sentiments, emotions, and personality features. These features are having the phenomenal influence for sarcasm detection. Like proposed by the former authors (Ptáček, et al., 2014; Bamman, 2015) the CNN convolution neural network to learn baseline features however, without discrete features.

Further, improving the performance of the model by giving input in the form discrete features to the core model despite the correlation analysis of pragmatic features (Karoui, et al., 2017). Following the similar methodology given by former authors (Karoui, et al., 2017; Poria, et al., 2016), all features' methods had profound impact on sarcasm detection when baseline features are combined with discrete features. The real problem is the methodology that how to integrate the discrete features at hidden layer of Deep Learning Model. The author raised two solutions one is that integrate the features at the hidden layer. Input layer takes embedding vector so further concatenated features at hidden layer (Poria, et al., 2016; Aggarwal, 2018). It was observed that

model CNN and CNN-SVM combined approach produces F1 scores of 0.97 and 0.95 on dataset of tweets however, it outperformed the existing method for sarcasm detection (González-Ibáñez, et al., 2011; Joshi, et al., 2015). It proved that the baseline models CNN-SVM and CNN combined the features of personality, emotion, and sentiment achieved the highest performance. The adapted model will integrate the baseline features with discrete features in the deep learning models. Further, discussed the details of adapted methodology with architectural aspects. The convolutional neural network will apply convolution operation which is the main advantage to reduce dimensionality. Further, these discrete features concatenated at the hidden layers with strong features resulting from pool max layer. These embedding or representations are based on embedding dictionaries like Wiki-news-300d. The size of the embedding layer is important to set at larger value because of fine-tuning of pretrain model over multiple domains.

The model will take the input as a vector at the input layer after mapping each word to vector space. The matrix size is not uniform because Amazon reviews have long text input on the other hand the Twitter tweet has short text. The matrix size is uniform, for instance, 64-word size to 256 for the long text. Therefore, the vocabulary size is 40,000, that is the maximum number of word vector will participate in training such as in case of the domain of Amazon reviews. However, in the case of short text, the vocabulary size is 10,000 likes in the case of tweets. The other operations aim to reduce the dimensionality of the features by applying an input to the filter with convolution operation and getting a strong vector feature. The filter operation is a dot product between image and filter of size (5,5). These filters are features for the given task; it also reduced the

dimensionality from (128,128) space to (5,5). Each filter is a dot scalar product and produce a single value by sliding at each step (Appendix A). The strong features are the result of the convolution operation get from the max-pool operation. After max-pool, the features will be concatenated together with baseline features and passed to dense layer. The discrete features are incongruity and pragmatic features extracted using the algorithm as proposed in the next Chapter. Further, the dense layers will perform ReLU operation (the rectified linear activation function or ReLU is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero) and finally, the SoftMax layer will output the binary classification results into two classes sarcastic and non-sarcastic.

3.5.4 Deep Learning Model (LSTM)

LSTM model's main characteristics are information retention over the sequence of words context in the sentence and removing unimportant information. The main objective of this model is to retain information from the previous state if it is useful, then it will combine with the next state and pass it to the sigmoid function. Then input pass gate will get the value from the previous and current state to the tangent and sigmoid activation functions. However, this proposed research would like to adapt the LSTM rather than a variant of LSTM like GPU-based LSTM proposed by Haifeng (2018). According to former authors, the performance of LSTM was better than core models that is another reason to select the model. Like few authors (Felbo, et al., 2017; Poria, et al., 2016) proposed BiLSTM for multiple tasks such as sentiment, emoticons, and sarcasm. Therefore, like to do the work in similar direction that is why deep learning models selected for

the experimentation purpose, the models are LSTM, CNN, and BiLSTM. These models are having baseline features integrating with incongruity and pragmatic features.

3.6 Transfer Learning Strategies

The deep learning model is a renowned and popular model to solve multiple tasks of NLP sentiment, emotion, and sarcasm analysis. These models are specific to the domain but can transfer knowledge to any other domain when input context is generalized features of sarcasm. Transfer learning will transfer knowledge among multi-domains; however, the model proposed LSTM transfer knowledge for multiple domains and work on classification of multiple tasks (Felbo, et al., 2017). Therefore, this research would like to work on domain adaption problems which is called multiple domain learning (MDL) but with single task. The proposed model can train the dialog discussion comments and would have the ability to transfer knowledge among other domains. The adapted models proposed by the former author for sarcasm tasks for multiple domains.

Transfer learning can be understood by the teacher-student relationship. For example, a teacher has years of experience in the topic which she/he can teach. With all this accumulated information, students' lectures are a concise and brief overview of the topic. It is some sort of transfer of knowledge to students effectively. Therefore, similar concepts would apply in which a neural network is trained on the data and knowledge. These weights can easily transfer from one dataset to another datasets. Therefore, it will save time for neural networks because no need to train the entire model from scratch.

There are many strategies defined in existing research which are part of the existing methodologies. One of the early works was on unfreezing the last layer which fine-tunes the new dataset with freezing all other layers (Jeff, 2014).

Another strategy proposed by (Erhan, et al., 2010) a common model where all layers were frozen to fine-tune. (Felbo, et al., 2017) the chain-thaw strategy sequentially freezing the layer one at a time however, results were effective on another domain dataset. The transfer learning strategy chain-thaw fine-tune the model over multiple datasets of multiple domains and evaluated better accuracy as compared to the “last” and “first” layer freeze strategy. The pretrain model will fine-tune with other strategies to experiment however purpose is to devise a new strategy.

3.7 Evaluation Matrices

The task is to classify sarcasm using the core models and deep learning models: BiLSTM, LSTM-CNN, and LSTM. Here, the scope is limited to word-based BiLSTM instead of a character based BiLSTM model. Secondly, limiting the model to a single task on verbal irony or sarcasm thus, incongruity and pragmatic features were extracted. Initially, the experiment will train the model with the integration of the features over the sample of SemEval-2016 dataset. The evaluation criteria are based on the three types of evaluation metrics as given below.

$$\text{Accuracy} = (\text{True Positive} + \text{True Negative}) / (\text{Total Number of Instances}) \quad (1)$$

$$F1 = 2.(\text{Precision}.\text{Recall}) / (\text{Precision} + \text{Recall}) \quad (2)$$

F1 tries to find the distance between precision and recall.

Precision:

It is correct positive results divides by number of positive results predicted by the classifier.

$$Precision = TP / (TP + FP) \quad (3)$$

Recall

It is the number of correct positive results divided by the number of all relevant samples

$$Recall = TP / (TP + FN) \quad (4)$$

The third metric is the AUC, which observes the output points of the area under the curve in the evaluation of the prediction results based on the deep learning model and core models. The expression and definition are defined as ratio between TPR and FPR.

True Positive (TP): Actual Positive and Predicted as Positive

True Negative (TN): Actual Negative and Predicted as Negative

False Positive (Type I Error): Actual Negative but predicted as Positive

False Negative (Type II Error): Actual Positive but predicted as Negative

Now let us look at what TPR and FPR.

$$TPR = TP / (TP + TN) \quad (5)$$

$$FPR = FP / (FP + TN) \quad (6)$$

AUC-ROC is nothing but area under the TPR and FPR as illustrated below.

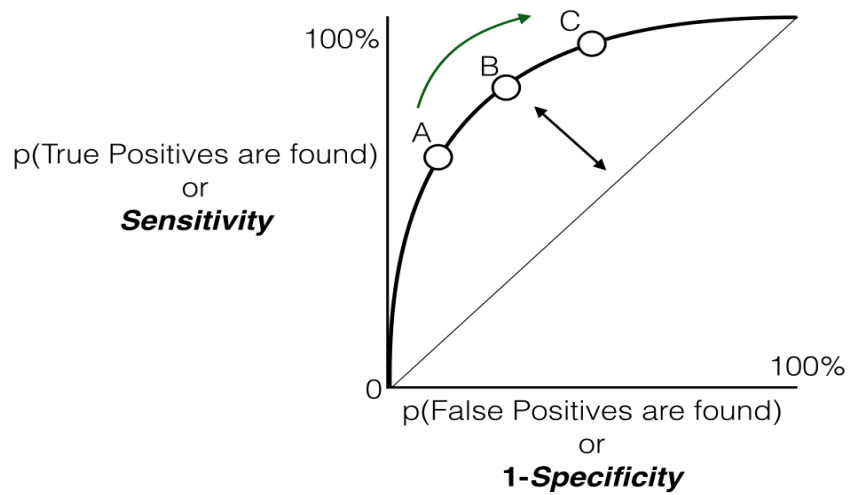


Figure 3.7: Source Creative Common

CHAPTER 4

Experimentation, Algorithm, Models, Results, and Evaluation

4.1 A novel comparison of core models vs. deep learning for sarcasm detection

Many state-of-the-art Machine Learning approaches detected the sarcasm using various methods like patterns, pragmatic markers, seed/bootstrapping, and linguistics. Initially, the detection was not significant due to the improper features and scaling techniques. This research will observe the impact of scaling techniques over the models, which fall under the category of core model and deep learning. Initially, it was concluded that various models behave differently due to impact of scaling technique. For example, KNN performed better when min-max scaling applied as compared to SVM when lambda applied. The research aim is to select the new baseline model, therefore, will like to compare the performance of state-of-the-art baseline core models SVM, logistic, KNN with deep learning-based models: CNN, LSTM, BiLSTM, CNN-LSTM, and CNN-BiLSTM. It is concluded that the deep learning-based models, the CNN and BiLSTM, are better for the detection of sarcasm as compared to core models SVM and KNN. On the other side, adapted SVM is better among core models in the presence of pragmatic and incongruity features. The experiment methodology will develop to propose the incongruity algorithm, which will extract the features into the categories of implicit and explicit incongruity. It is planning to train the models

based on preprocessing plan to evaluate which preprocessing level is more appropriate for performance (Chapter 3, Table 3.3.2). The algorithms will extract features with the tokenization concept like the 2-gram. These features are term/phrase polarity contrast, the highest polarity contrast, total positive/negative terms count, and the sentiment of the tweet.

4.2 Irony Detection Algorithm

This chapter is about the extraction of lexical, syntactic, and semantic features extraction. The purpose is to implement an algorithm that will extract the implicit and explicit incongruity features with an automated approach rather than based on human judgment. These features will extract from various sources datasets like tweets, Amazon reviews, and discussion dialog comments.

4.2.1 Implicit Incongruity Algorithm (IIA)

The incongruity features algorithm extracts implicit features that will recognize the polarity contrast among terms and phrases in the tweet. Furthermore, the implicit incongruity algorithm extract features which occurred in the tweet as opposite polarities among terms t_{ip} and phrases. Mathematically, 2-gram patterns phrases are denoted as t_{p1}, t_{p2} , the annotation of verbal words that belong to the tweet set $T_i = \{T_1, \dots, T_n\}$ and extracted based on seven verbal patterns.

Algorithm 1: Implicit Incongruity Algorithm (IIA)

Input: W_{ai} Annotated verbal words, T_i tweets, $W_{ai} \in T_i$

Output: F_{cong} Subset of term t_{pf} and t_{1p}, t_{2p} pair of 2-gram phrase

- 1: $F_i: t_{vi} \in W_{ai}$ Filter the first term as a verbal candidate term
 - 2: $F_j: (t_{v1}, t_{v2}) \subseteq W_{ai} \wedge (t_{v1}, t_{v2}, t_{v3}) \subseteq W_{ai}$ Find the verbal patterns 2-gram and 3-gram
 - 3: $F_{ij} = \forall \left\{ \exists (t_{pi} \in P_i, (t_{p1}, t_{p2}) \subseteq P_j) \right\}$ Take polarities of pair $F_{ij} \in (t_{pi}, t_{pj})$ where t_{pi} and t_{pj} pair of terms represent 2-gram phrase and term polarity
 - 4: $F_{cong} = \forall \left\{ \left(\exists t_{pi} \in P_i \mid P(t_{pi}) \text{ is } + \right) \right\} \neq \left\{ \exists (t_{p1}, t_{p2}) \in P_j \mid P(t_{pj}) \text{ is } - \right\}$ the subset of contrasting polarities
 - 5: **return** F_{cong}
-

Finally, the subset F_{cong} is the output of the algorithm that matches all negative/positive contrasting pairs called as the implicit incongruity features at step 4, where F_{cong} , pairs belong to the classes (\pm) . Below Table 4.2.1a, the sample subset is taken from Dataset DS1 and DS2. Total verbal pairs $(t_{v1}, t_{v2}) \subseteq W_{ai}$ belong to the classes: $class(VV) = 32$, $class(VA) = 16$ and $class(VM) = 9$. The last step found 58 implicit incongruity features (F_{cong}) returned from the algorithm out of 800 training tweets where found 27 + and 28 - contrasting 2-gram phrases polarities $(t_{p1}, t_{p2}) \in P$ and the t_{pi} verbal terms polarity.

Table 4.2.1a: Polarity of 1-gram and 2-gram.

Doc-id	1-gram	2-gram	Polarity
D13	Feeling	Feeling blessed	Negative/Positive
D51	Spent	realizing looks	Positive/Negative
D301	Find	looking realize	Positive/ Negative
D394	Using	took long	Negative/positive
D563	Made	Go wasted	Negative/Positive

The first step is to discover the candidate term for each tweet. The first term filtered out all tokens in the form of POS tags using the udpip dictionary. Its position is stored in a frame; token position is the result of the udpip annotation with POS tag and its participles for given term. In total, 11,294 tokens were extracted from the 613 documents with all tags SemEval-2018 Task 3 (Van Hee, et al., 2018).

Table 4.2.1b: Dictionary tokenization and part of speech (POS)

Token_id	Token	UPOS	XPOS
1	Can	AUX	MD
2	U	PRON	PRP
3	Help	VERB	VB
4	More	ADJ	JJR
5	Conservatives	Noun	NNS
6	Needed	VERB	VBN
7	On	ADP	IN

Part of the speech tag represented by XPOS which denote the short form of verb, for instance, past participle verb (VBN). The next step is to find the F_i candidate subset term that is the first verb extracted from the tweets. This subset F_i is found by tokenization as presented in Table 4.2.1b. The

subset results are the verbal patterns these are the combination of nouns, pronouns, adverbs, and adjectives belong to any participle. The verb can be past participle, present participle, infinite, finite, or simple verb.

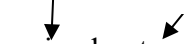
Table 4.2.1c: Dictionary tokenization and part of speech (POS)

Doc-id	Token_id	Token	POS
D1	5	Need	VERB
D1	12	Go	VERB
D2	2	Walk	VERB
D2	5	Starbuck	VERB
D2	7	Ask	NOUN
D3	3	Win	VERB

Here doc_id refers to the single tweet that uniquely identify each token from the dictionary. These tokens are sequence from the dictionary, the first element is the verb term extracted as shown in above Table 4.2.1c. The result is the first occurrence of the verb term in the matrix F_i . Total of 360 verbal patterns were selected from 1,522 verbal patterns, those belonged to 613 tweets. The next step is to harvest the phrases using 2-gram (two consecutive words) and 3-gram (three successive terms) patterns.

2-gram Patterns

3) Loooovvveeeeeee when my phone gets wiped not

Verb (VB) *Verb (VB)*


In the sentence, "gets wiped," a 2-gram pattern is extracted based on the seven patterns inspired by (Riloff, et al., 2013). These patterns are verbal pair (VV), verb/pronoun (VM), adverb/verb

(AV), verb/ adverb (VA), to/verb, verb/noun (VN), and verb/adjective (VADJ). Despite these auxiliary patterns “are” and “be”, the other auxiliary patterns combined with an adverb, main verb, and verb. The reason behind the exclusion of these unwanted auxiliaries is ambiguity (the performance of classification results will be affected by the inclusion of these gerund features). These patterns are employed for the task of classification; therefore, opted seven verbal patterns. Here, verbal combinations are 2-gram patterns extracted with the POS intensifiers such as VBZ (comes after 3rd person singular, e.g., takes), VBD (past form of the verb, e.g., took), VBN (past verb participle, e.g., accepted), VBG (verb gerund present participle, e.g., taking), VBP (verb, sing. present, non-3d take) and VB (verb base form, e.g., bear). For instance, the verbal past participle combined with a gerund formed the candidate phrases these are contrasting polarities among the terms and phrases.

In above sentence 3), it would depict as character's exaggeration problem: repeated characters in the above sentence. Further, analyzed 2-gram phrases based on the seven verbal patterns, therefore, extracted the total of 8,769 unigram patterns out of 11,294 tokens. There are 604 2-gram pairs of patterns extracted from 8,769 tokens (Table 4.2.1d).

Table 4.2.1d: 2-Gram Pattern-based intended phrases

Doc-id	Pattern	Token	POS
D1	VN	Paid posting	VERB NOUN
D5	VV	Sleeping Mate	VERB VERB
D10	VV	Stop worrying	VERB VERB
D36	VM	Feels more	VERB ADVERB
D19	VN	Post my	VERB NOUN

“I love context and large ensemble Fridays!!!! Der my most favourite #Sarcasm”.

The above sentence observed the 2-gram pattern out of seven patterns, such as the word “love” is a positive verb. The polarity of words taken using datatumbox API; covers a broader vocabulary range than any other source. In this example, 2-gram patterns include the intensifier symbol “!!!! Der,” which is matched to the verbal noun pattern. However, the symbol “!!!!” represents the punctuation feature extracted that belonged to the pragmatic category. These three terms have the highest frequency “love,” “know,” and “think” which were found in all tweets document. Further, the bootstrapping process will search the term “love” in the entire tweet corpus. The sentences are tokenized into the word tokens that include low-frequency words. In total 250 verb and pronoun combinations were found by verbal pairs.

It is important to find the incongruity subset by matching the negative/positive contrasting words. (Joshi, et al., 2015) according to occurrences of the polarity action or verbal words, the first term is the candidate term, and there were 45 terms in the different documents that were extracted as candidate terms. Interestingly, the pattern “heading” in document number 153 appeared 23 times.

Table 4.2.1e: Polarities contrast Incongruity Subset

Doc-id	1-gram	2-gram	Polarity
D5	Boring	Sleeping mate	Negative/Positive
D10	Tell	Stop worrying	Positive/Negative
D42	Love	Fixed by	Positive/ Negative
D52	Find	Ask hiring	Negative/positive
D152	Want	Feeling to	Negative/Positive

After analyzing thoroughly all the verbal combinations and other patterns, the best seven general patterns were found. In comparison to the former author (Riloff, et al., 2013), the extraction method is more robust to check the polarities among terms/phrases in any order, that is positive/negative terms may occur before or after the positive/negative phrases. (Riloff, et al., 2013; Maynard & Greenwood, 2014) the former researcher worked on a seed-based approach, whereas the proposed algorithm was robust to extract any candidate term that contrasts polarity with the phrase. It found 22 implicit incongruity patterns at the last step of the algorithm F_{cong} as given in Table 4.2.1e. This algorithm is not static but robust enough to extract the verbal polarity terms from the tweet. This algorithm is observed the 3-gram patterns as well, but unfortunately, the frequency of extracted patterns of 3-gram is shallow. Overall, it extracted 8 3-gram patterns with the combination of verb, pronoun, adjective, and adverb. Thus, 3-gram patterns are not considered to extract from other domains.

3-gram Patterns

The 3-gram patterns are extracted based on seven general types of verbal patterns: verb and adverb mixtures, an infinitive VP that includes an adverb, a verb and noun phrase, a verb and prepositional phrase, a verb, and adjective phrase, or an infinitive VP and adjective, noun, and pronoun. Finally, the algorithm found that 8 patterns are matched according to the general verbal combination given in (Table 4.2.1f). It was planning to discard the 3-gram patterns due to low frequency, therefore, adapt the similar methodology proposed by the former authors (Riloff, et al., 2013; Joshi, et al., 2015).

Table 4.2.1f: Pattern incongruity Subset

Doc-id	Pattern	3-gram
D7	VPN	Complain about my
D10	VPN	Worrying about it
D42	VMN	Love when my
D52	AVN	unintentional buying Bogs
D152	AVN	nice work Wednesday

4) “People who tell people with anxiety to "just stop worrying about it" are my favorite kind of people”.

Sentence 4 extracted the phrase “worrying about it”, which is based on pattern VPN. Similarly, phrase “Loovee when my” is extracted based on verb-main-infinitive (VMN) and extracted the phrase “unintentional buying bogs” is based on adverb-verb-Noun (AVN) pattern.

Table 4.2.1g: Polarity contrast incongruity 3-gram and 1-gram subset

Doc-id	1-gram	3-gram	Polarity
D42	Love	fixed by pm	Positive/Negative
D149	Want	DONE WITH FINALS	Negative/Positive
D152	Start	Feeling to myself	Positive/ Negative

There are 8 3-gram patterns were observed which are contrasting in polarity with candidate term, as shown in Table 4.2.1g. Due to low frequency of 3-gram patterns, these patterns will not consider part of test and training set.

4.2.2 Explicit Incongruity Algorithm (EIA)

The explicit incongruity algorithm (EIA) is the opposite polarity among terms/words in the tweets. There are four features which are extracted using proposed algorithm such as positive and total negative, incongruity count, the token sequence between terms (positive and negative), and sentiment of the tweet. These features are about the word contrasting polarities which indicate the mocking behavior of the users in the terms.

The expression for subset of terms which have polarities is defined as follows.

$$F_{ij} : \exists \{t_{aij} \in w_{ai}, t_{pij} \in P_{ij}, X(t_{vij})\} \quad (1)$$

Expression (1) is the subset F_{ij} that contains terms t_{aij} which annotate the verbal term that have t_{pij} polarities (positive and negative) and $X(t_{vij})$ is the term position.

$$t_{ij} = \exists(\{t_{pi} | t_{pi} \in P_i \wedge t_{pi} = -\}, \{t_{pj} | (t_{pj} \in P_j \wedge t_{pj} = +)\}) \quad (2)$$

Expression (2) expresses contrasting polarity terms which occurred in a single tweet. Here features are t_{p1} belonged to the polarity positive and t_{p2} belonged to the negative polarity, here expression denoted contrasting terms occurred in a single tweet represent explicit incongruity t_{ij} .

Algorithm 2: Explicit Incongruity Algorithm (EIA)

Input: Annotated verbal words, tweets

Output: $F_{explicit}$ Subset of four Features: t_{ij} contrasting pair of term,
 $maxseq_{ij}$ is the distance between the position of terms, $T_{pi} \in \{pos, neg\}$ T_{pi} is
the total the positive and negative polarity of the term $S_i = Sent(T_i)$ is the
sentiment of tweet doc T_i document

- 1: $F_{ij} : \exists \{t_{aij} \in w_{ai}, t_{pij} \in P_{ij}, X(t_{vij})\}$ create a subset F_{ij} that contains terms t_{vij}
belong to annotated verbal Terms, t_{pij} is the polarities of the Terms, and
 $X(t_{vij})$ function get the position of terms
- 2: **for** $W_{ai} \in T_i$ **do**
- 3: Transpose rows to the column for each document $t_i \in T_i$ Add ($W_{ai}, t_{pij} \in P_{ij}$) as
column, W_{ai} : a term that is annotated verb
- 4: **end for**
- 5: **for** $t_{ij} \in T_i$ **do**
- 6: $t_{ij} = \{t_{pi} | t_{pi} \in P_i : t_{pi} = \{-\} \wedge t_{pj} \in P_j : t_{pj} = \{+\}\}$ Find the contrasting terms polarities from
each pair of terms t_{ij}
- 7: Find $maxseq_{ij} = \{\exists Pos(t_{ij}) | \max(Pos(t_{ij}) - Pos(t_{ij})) : t_{ij} \in P_{ij}\}$, the maximum sequence between
contrasting polarities terms
- 8: $\{\exists t_{pi} | t_{pi} \in P_{ij} : t_{pi} Total Positive \wedge t_{pj} \in P_j : t_{pj} Total Negative\}$ Find the total positive/negative
terms
- 9: **end for**
- 10: $S_i = Sent(T_i)$ the sentiment of the overall tweet document

11: Add $(t_{ij}, \max seq_{ij}, t_{pij}, S_i)$ to the $F_{explicit}$

At step 11 $F_{explicit}$ expressed the output into four features. The first feature is the polarity contrasting terms represented by t_{ij} and the second feature is the largest distance between positive/negative polarity terms expressed as $\max seq_{ij}$. The third feature is total positive and total negative terms expressed as the t_{pij} , and the final expression referred to as the overall sentiment of the tweet that denoted S_i .

These features are listed in below Table 4.4.2a, these features were adapted from the work of the former author (Joshi, et al., 2015). (Poria, et al., 2016) the author considered the syntactic patterns to extract the verbal patterns. According to the former paper, "love" is a positive, but other words are negative in the tweets that is polarity contrast among terms. (Tsur, et al., 2010) the verbal patterns are generalized to extract the terms based on fixed patterns like the seed words "love".

$F_{explicit}$ is a subset that contains four features,

$F_{explicit} : (t_{ij}, \max seq_{ij}, t_{pij}, S_i)$. Below Table 4.4.2a represents these features which are part of the

explicit incongruity features subset $F_{explicit}$.

Table 4.4.2a Explicit incongruity subset: $F_{explicit}$

Doc_id	PosCount	NegCount	TokenSeq	Sentiment
D5	1	2	3	-
D6	1	2	4	+

At Step 2-4 Explicit Incongruity Algorithm (EIA) algorithm will transpose all terms into rows from columns. Each row denotes the tweet terms with its characteristics, and its columns are polarity, position, and token sequence (that is term distance from positive to negative). The terms will rearrange into the columns of the tweet; however, the column may expand to many columns when large text tweet occurred. Each tweet or document has a different length of sentences like document 1 is having 13 columns. The first column represents the position, the second one token, and the third one is polarity. Each term has three subsequent columns in a row and will repeat these columns readily dynamically for all terms of the tweet. The longest document is having 153 which comprised of more than 100 tokens or words (Table 4.4.2b).

Table 4.4.2b: Polarities contrast incongruity subset

Doc-id	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5	Pos 6
D1	Help	Neutral	6	Needed	Neutral	10
D2	Walked	Negative	7	Asked	Neutral	14

In the step 5-9, the algorithm will harvest the features total positive t_{pi} , total negatives t_{pj} , $maxseq_{ij}$, the longest sequence of positive and negative words and overall sentiment of the tweet s_i . As given in Table 4.4.2c, document D5, the positive term follows the two negative terms represented as t_{ij} where explicit incongruity count is 1 (it means two terms is having polarity contrast) and the longest sequence of the terms which is the fourth feature which represents the distance among terms belonged to the tweet denoted as Tokenseq ($maxseq_{ij}$).

Table 4.4.2c: Explicit incongruity features

Doc-id	PosCount	NegCount	Neutcount	Incongcount	Tokenseq
D5	1	2	1	1	3
D6	1	2	1	1	4

4.3 Pragmatic Features

(Ghosh, et al., 2015; Maynard & Greenwood, 2014) the other clues are pragmatic features that detect the verbal irony. The verbal irony in comparison to context-based features proposed by former authors (Davidov, et al., 2010; Giovanelli, et al., 2021; Liebrecht, et al., 2013; Buschmeier, et al., 2014).

4) “Well, no! clubbing / putting up eyes (p1), is not violent it does respect human rights (p2)!!!”

In the above sentence, phrase 2 (p2) is having the pragmatic feature in the form of punctuation utterances. However, phrase 1 (p1) is not having any clue. Similarly, the proposed methodology will select the features like emoticons, laughter, and interjections to detect sarcasm. A similar feature is punctuation count that builds number of times punctuation occurred in the tweet with the sign ‘!’, it is a strong clue of sarcasm and its low occurrence means a weak clue.

4.4 Methodology

This research likes to adapt similar methodology proposed by former authors (Poria, et al., 2016; Felbo, et al., 2017). These approaches are the LSTM with fewer layers, LSTM with Convolution Neural Network (CNN) proposed a hybrid approach, and LSTM concatenated with various features. The transfer learning strategies will be applied among different models such as LSTM,

LSTM-CNN, and bi-directional LSTM for multiple social media domains. The experiment plan will implement the AutoML framework which will select the model with its hybrid combinations. The second plan is to propose the freezing strategy that will allow pretrain models to work on multiple domains.

4.4.1 SVM

The baseline SVM integrated with 2-gram and 3-gram features to classify the sarcasm. (Davidov, et al., 2010; Riloff, et al., 2013) the former authors opted the LIBSVM library to classify sarcasm with unigram, bigram features and RBF kernel. The lexical-based features extracted with the support of 2-gram and n-gram patterns. SVM will classify sarcasm based on the input of the pragmatic, implicit incongruity, and explicit incongruity features. (Ptáček, et al., 2014; Joshi, et al., 2015) the incongruity features were extracted which are main clue for the sarcasm in the sentence where terms have opposite polarities. Similarly, another author worked on integrating the features into the core and deep learning models. (Prasad, 2010) the adapted methodology integrated the features such as lexical and pragmatic utterances.

4.4.2 Logistic Regression

In the recent past, more experiments were conducted in the domain of NLP for sarcasm detection. In this direction, (Davidov, et al., 2010) the research methodology detected the sarcasm using logistic and naïve Bayesian core models over the small benchmark dataset of 60000 Amazon reviews. The performance of logistic regression was not profound, even applied preprocessing and

integrated the pragmatic features like emojis and slang words. In former research, pragmatic markers have low performance of 0.40 F1 and 0.75 F1, because the pragmatic clues cannot impact the performance of the model standalone. Therefore, the features list needs to be extended to include the pragmatic features alike capital letters, punctuation, emoticons, interjections, and incongruity contrast features that occurred among terms.

4.4.3 KNN

In the past, the Semi-supervised Sarcasm Identification Algorithm (SASI) was proposed by former author (Tsur, et al., 2010). The algorithm has two modules: the first one works on the semi-supervised pattern that learns the sarcastic patterns using the classifier and the second module is the classifier that classifies each Twitter tweets and Amazon reviews into sarcastic or non-sarcastic classes. The algorithm classifies multiple domains like 60000 Amazon reviews and selected 1500 small sample tweets hashtag filtered from 5.9 million Twitter tweets. The feature-set learns the pattern with the pattern-based algorithm which is founded on bootstrapping. The bootstrapping algorithm searches the fixed term “love” in all tweets document. The performance of the pattern-based algorithm is 0.54 F1 on tweets, however, it was not as good as it has been due to the absence of context features.

4.4.4 Neural Network

(Walker, et al., 2012) the author proposed DNN and hybrid combinations with CNN. Proposed multimodal for the text and visual images are the fusion of both domains. In the scope of study text patterns of 2-gram, and sentiment will be considered but visual features are out of the scope.

During the experiment of the model CNN, the performance was compared with DNN using the input 1-gram pattern of the tweet and visual features of the image, but the performance of multimodal was not effective because of low score 0.69 F1. (Felbo, et al., 2017) another research experiment was conducted with the bidirectional LSTM model that was applied to multiple social domains. The experiment was on multiple domains to get good performance over sarcastic data source like dialog discussion comments. It was another research experiment conducted over multiple social domains using the Bidirectional Long-term Short Memory (BiLSTM).

4.4.5 Adapted hybrid model (LSTM-CNN)

The modes BiLSTM/LSTM and CNN models formed the adapted hybrid model LSTM-CNN as given in the below Figure 4.6.5c. The model with BiLSTM or LSTM is illustrated in Figure 4.6.5a and CNN in Figure 4.6.5b. These are the basic layers which defined the hybrid model formulation.

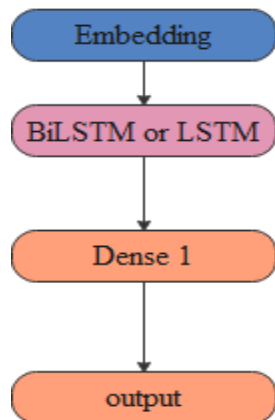


Figure 4.4.5a: BiLSTM or LSTM Baseline

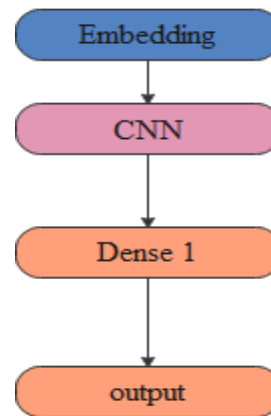


Figure 4.4.5b: CNN Baseline Model

Model

The elaboration of hybrid model is as follows: the first layer is the embedding layer where vectors are input. Initially, the word vector is generated from the embedding dictionary of 1.6 billion Glove. In proposed methodology, it is preferred to specify dictionary due to the coverage of billions of words in the form of vector space, which is a large vector space as compared to Wikipedia embedding (like to have more word vectors and similarity). These vectors lookup in the word embedding matrix $W \in R^{I*V}$. The sequence of the word vectors is concatenated together in a matrix to represent each tweet in a row. Further, the concatenation leads to the padding procedure for each sentence because these sentences have a different length, therefore padding will make uniform length for each sentence. It will prepare the matrix formulated with the sequence of word vectors of the tweet. The words vector size is 64 the maximum length sequence for each tweet. But if the length of one of the sentences is shorter than other than the padding procedure will expand the length while replace it with zeros.

Initially, the word vocabulary size is 40,000 because of Amazon reviews, which is a longer length of text, on the contrary shorter text required less dimensional space of vector terms. The maximum length of sentence is set to 128 of tweet. The maximum length of Amazon reviews is required larger dimensionality space for vector terms. The maximum epochs are 10 during training of the model, the training reflect the inflection point that marks the training limitation, which is 10 episodes. (Van Hee, et al., 2018) the dataset SemEval-2018 Task 3 considered the 613 tweets sample size to train over multiple models. The model split the dataset into 65% training set and

35% test set. (Chollet, 2015) the Keras `texts_to_sequence` function of Keras library in R language converts text into integer sequences, but text of tweets is unequal in length so padded further. The matrix is further divided into validation and training matrix. The x of the model is divided into vector-matrix padding space and y is pre-train labels. Here x represents the padding space that pads each word into embedding vector at horizontal space.

According to characteristics of CNN (Figure 4.5.5b), the image space reduction is called as dimensionality reduction, that is the results from the convolution operation between kernel and image that is sum of product with the filter. These features after convolution operation concatenated together at the hidden layer. The concatenation operation happened with the discrete features, pragmatic, and incongruity features at the dense layer. The bidirectional LSTM layers (Figure 4.5.5a) have 128 sequence lengths. After that, the dense layers, or fully connected layers where concatenation occurred with the incongruity and pragmatic features. The final layer is the SoftMax layer that will classify sarcasm, where 1 stand for sarcastic and 0 for non-sarcastic.

The hybrid model BiLSTM-CNN defined and illustrated below in Figure 4.6.5c, the model architecture with all layers. The model has the bidirectional layer of 100 input units and 128 is the length of the sequence of words. The parameter size of the model is as follows: filter size is 64 and kernel size is 5*5 CNN, dense layer size is 128-unit size, and output layer size is 1. A sigmoid activation function will process the output either sarcastic or non-sarcastic tweets.

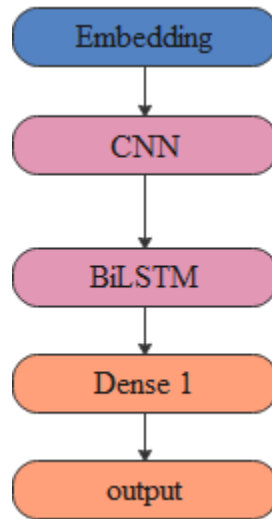


Figure 4.4.5c: Baseline CNN-BiLSTM Models.

Finally, this research will further explore the features like pragmatic and incongruity features concatenation at the dense layer of the model.

4.4.6 Data Collection

(Riloff, et al., 2013; Ptáček, et al., 2014) this research opted two datasets Twitter tweets and Amazon reviews. (Riloff, et al., 2013) the state-of-the-art benchmark dataset DS1 constitute of the tweets sample with 1,367-training and 588 test sets. (Ptáček, et al., 2014) DS2 dataset split tweets into 30,010 regular and 29,055 sarcastic tweets. The proposed research model will incorporate these benchmark datasets to observe the impact using the core and deep learning models for sarcasm.

4.5 Results and Discussion

4.5.1 Experiment

This research has compared the core model with the deep learning model using evaluation metrics; F1, accuracy, and AUC. The feature extraction method is based on the incongruity algorithm to extract 401 pragmatic and incongruity features from the 1,367 tweets sourced from DS1. The words in the tweets are tokenized and mapped words to vectors using a publicly available dictionary named as Glove. These numeric representations of the word feed as the input at the embedding layer of the baseline deep learning models CNN, LSTM, CNN-BiLSTM, and CNN-LSTM. The second dataset DS2 split data into 800 training and 200 test features that feed in the form of input to the core and deep learning models.

Table 4.5.1: Description of benchmarking dataset split.

Identifier	Study	Baseline Model	Train	Test
			(Features + Lexical)	
DS1	(Riloff, et al., 2013)	SVM	321	80
DS2	(Ptáček, et al., 2014)	CNN	800	200

In Table 4.5.1, (Ptáček, et al., 2014) the author builds the baseline model CNN that will evaluate the performance over the dataset DS2 that comprised of 1000 features. The SVM was the baseline model that was evaluated over the dataset DS1 that comprised of 401 features (Riloff, et al., 2013).

4.6 Evaluation

It is planning to evaluate the datasets DS1 and DS2 after splitting into 80% training and 20% test datasets. The model trained for 15 epochs; the reason is to set the limitation of epochs to avoid

overfitting, but the model will be overfitted after 19 epochs. The main reason of the limitation of epochs is the leverage of the computation resources on a single machine. But, to ensure that the model trained well and converged, there is a need to observe the inflection point in the training plot. These metrics AUC and F1 score are representation of model performance over imbalance datasets given in Table 4.5.1. All the experiments were conducted in RStudio using Keras and TensorFlow libraries. The hardware platform is Intel core-duel i7-7500 with 12 GB RAM.

4.7 Baseline Results

The datasets DS1 and DS2 were evaluated with 15 epochs during training the model but DS1 is overfitted after 15 epochs because it was observed with the inflection point. There are evaluation metrics selected for the experiment such as F1, accuracy, and AUC.

4.7.1 Evaluation of Data Set (DS1)

There is necessity to observe the performance of core and deep learning baseline models. The core model input is incongruity and pragmatic features. The performance measures F1 was 0.93 as evaluated over the DS1 imbalanced dataset. Consequently, the KNN approach is a better than core model for classification with a score of 0.93 F1. Thus, adapted core models SVM 0.83, logistic 0.91, and KNN 0.93 F1 outperformed as compared to current state-of-the-art approach SVM 0.82 and 0.51 F1 as given in below Table 4.7.1a.

Furthermore, presented the comparison of core models and deep learning models over DS1. On the other hand, the core model integrates the incongruity and pragmatic features to get performance

measures with better efficacy. However, deep learning model CNN F1 was 0.95 which is better than core model KNN F1 0.93. But the model has an average performance of 0.78 AUC over KNN. But adapted core models SVM, logistic, and KNN outperformed other states-of-the-art core model approaches when integrating the extracted features of incongruity and pragmatic (Ptáček, T. 2014; Riloff, 2013; Poria, 2017). Nevertheless, according to former research literature review for the sarcasm the core models are not better in efficacy than deep learning models. The evaluation of the model KNN over DS1 produced 0.93 F1 and CNN got 0.95 F1 that shows effective performance which is better than all other deep learning models, however, it outperformed existing state-of-the-art deep learning model (Poria, et al., 2016).

Table 4.7.1a: Comparisons of benchmarking dataset DS1 with the baseline

Identifier	Study	F1	Accuracy	AUC
SVM	All features (pragmatic + incongruity)	0.83	0.90	0.63
Logistic	All features (pragmatic + incongruity)	0.91	0.83	0.72
KNN	All features (pragmatic + incongruity)	0.93	0.88	0.78
Neural Network	All features (pragmatic + incongruity)	0.93	0.88	0.78
CNN	Lexical	0.95	0.78	0.53
LSTM	Lexical	0.78	0.65	0.45
BI-LSTM	Lexical	0.79	0.66	0.46
CNN-LSTM	Lexical	0.82	70.0	44.4
CNN-BI-LSTM	Lexical	0.86	0.77	0.58

The CNN model achieved 0.95 F1 due to the proper preprocessing and scaling. It is a far better performance as compared to the existing CNN deep learning benchmark model, the result is presented in Table 4.7.1b (Ptáček, et al., 2014; Poria, et al., 2016).

The second-best technique is CNN-BiLSTM that achieved 0.86 F1. Model parameters set the default vocabulary size 10,000 for all tweets. After evaluating the model over DS1, it is concluded that the deep learning model CNN is better among all baseline deep learning models and core models.

Table 4.7.1b: Benchmark Dataset

Author	F1
(Poria, et al., 2016)	0.92
(Ptáček, et al., 2014)	0.94

4.7.2 Evaluation of Data Set (DS2)

Further, aim is to evaluate the benchmark dataset DS2 reference from former research (Ptáček, et al., 2014).. Additionally, it was planned to observe the DS2 dataset with the scaling techniques. The scaling techniques comparison is also presented in Table 4.7.2a, which shows that the lambda scaling technique is the most significant technique when applied over SVM, that depicts better than without scaling features. In this direction, KNN is better when the scale type is min-max, but it performed moderately when lambda and range scaling is applied. Thus, different models have the different scaling impact over the final performances. But, overall, SVM is better among all

existing state-of-the-art core techniques because features performed better when lambda scaling is applied and is evaluated with the AUC metric as given in Table 4.7.2b. The deep learning models CNN-BiLSTM and CNN are better among other deep learning techniques with 0.97 F1. However, the performance metrics AUC over SVM is better among other core models due to the scaling.

Table 4.7.2a: Comparisons of benchmarking dataset DS2 with the baseline

Models	Methods	F1	Accuracy	AUC
SVM	All features pragmatic + incongruity	0.94	0.94	0.94
Logistic	All features pragmatic + incongruity	0.90	0.90	0.93
KNN	All features pragmatic + incongruity	0.94	0.94	0.94
LSTM-DNN	Lexical	0.96	0.96	0.97
BI-LSTM-DNN	Lexical	0.93	0.97	0.97
CNN-DNN	Lexical	0.97	0.97	0.96
CNN-LSTM-DNN	Lexical	0.96	0.96	0.97
CNN-BI-LSTM-DNN	Lexical	0.97	0.98	0.97

Table 4.7.2b: Scaling effect on SVM

Models	Range	Min-Max	Lambda
SVM	0.94	0.88	0.94
Logistic	0.90	0.83	0.90
KNN	0.92	0.94	0.90

4.8 Conclusion

This Chapter was about selecting the adapted technique for the rest of the experiments. It was observed that the comparisons of the baseline core model with deep learning. Factually,

experiment validate the proof of concept that baseline core model performances cannot be better than deep learning baseline method on DS1 and DS2, however, the core model's performance is equal to deep learning models. Chapter 5 will like to evaluate performance over another domain dataset. Particularly, to validate that, it was observed that the deep learning CNN model outperformed core models SVM, KNN, and logistic. However, CNN-BiLSTM-DNN and CNN both models have equal performance, therefore, will adapt for next experiment. The core model SVM on DS2 performed better than existing state-of-the-art core techniques due to inclusion of all feature's method in the form of contextual features. On the contrary, the core model's performance on the dataset DS2 achieved similar conclusions while examining the features with different scaling techniques like range, max-min, and lambda. Overall, it proved that different models performed using different scaling techniques. Like KNN performed well when applied min max whereas SVM performed better among all core models when features scale with lambda scaling. Therefore, scaling techniques would enhance the performances of the core model with different efficacy.

In the next Chapter, these pragmatic and incongruity features will be considered to integrate with the deep learning model for better evaluation than state-of-the-art deep learning models. Another point is that it is required to prepare the plan for the preprocessing of tweets at different levels like removing hashtags and removing stemming at a different level and each level will be evaluated as given in Table 5.7.3-Chapter 5. Finally, it is the plan to propose the deep learning-based automated Machine Learning framework that will examine the models with better results over multiple datasets using different parameters.

CHAPTER 5

The newly developed AutoML framework for Sarcasm Detection

The last Chapter outcome is the best available baseline model which will be elaborate with useful outcomes. In this Chapter, aim is to discover the best available model using the AutoML framework to detect sarcasm. The main objective of this research is the newly developed AutoML DeepConcat framework that searches and evaluates the best model. The automated framework models set hyperparameters randomly set at drop_out layer. The parameter value of drop_out layer will help to reduce the network weights. Further, it is required to integrate incongruity features extracted from algorithms IIA and EIA. These features are integrated during the model search pipeline and further framework will search for the best model after evaluating performance using Bayesian optimization. But there is need to evaluate the existing deep learning and core models using the Automated Machine Learning framework with hyperparameters optimization. Few techniques worked on a single domain due to no domain adaptivity to other domains. Before discussing existing techniques in detail, here would like to draw the reader's attention to these selected baseline techniques. These are the best available baseline model that detects sarcasm evaluated in Chapter 4. The baseline models are CNN-BiLSTM and CNN therefore, these baselines were selected after performance evaluation in the last Chapter.

Recently, AutoML has shown considerable growth and industry application of Machine Learning. AutoML has emerged to improve the learning task by saving time and effort in repeated tasks like preprocessing, feature engineering, model selection, hyperparameters, and model architecture. AutoML proposed with newly developed feature engineering incongruity algorithms named as explicit incongruity algorithm (EIA) and implicit incongruity algorithm (IIA). These algorithms extract features from the existing datasets belonging to domains Twitter tweets, Amazon reviews, and dialog discussion comments. The core idea is to automate the AutoML pipelines like model search, hyperparameters optimization, and model architecture. Further, developed preprocessing plan with various levels, where level represents a single cleaning task of the text where model performance may vary with less and more preprocessing levels. After, preprocessing the feature extraction initiated with the tokenization process by selecting the 2-gram patterns in the form of phrases. The 1-gram and 2-gram patterns were extracted as sub-features to get incongruity contrast using the incongruity algorithms. The explicit incongruity algorithm extract features that are term/phrase polarity contrast. These four features are extracted as a contrast: the highest polarity contrast, total positive/negative terms count, and the sentiment of the tweets. The AutoML DeepConcat framework automates the model selection by concatenating these features into the five deep learning models during the model search pipeline and evaluating the hyperparameters like dropout and learning rate. After that, the learning models' outcomes, it will get multiple performance metrics and hyperparameters to find out best model.

5.1 Adapted DeepConcat Model

Initially, proposed a DeepConcat model that is part of the AutoML framework. The experiment was conducted with the baseline methods SVM, Logistic, and KNN. Following that, deep learning model architecture concatenated with features extracted into pragmatic and incongruity. Finally, proposed the DeepConcat model which constructs the two-layer CNN-CNN with BiLSTM layers with a dropout layer. The skip layer mechanism will select multiple combination of models such as LSTM, BiLSTM, and CNN during the model search pipeline at each iteration. The proposed model will also regularize the hyperparameter like dropout with random and fixed values.

This research proposed the framework that will evaluate all models' performances iteratively with metric F1 on the various dataset using preprocessing levels, extracted features, and scaling techniques. The performance evaluation was outstanding after the inclusion of these pragmatic features. It is necessary to clean the URL due to noise which will not harm performance of model for sarcasm detection. The aim is to evaluate the performance of all datasets comparable to existing approach (Ptáček, et al., 2014; Ghosh, et al., 2015). The tangent function creates better performance at the last two layers of DNN (further see Chapter 6 for the results of the transfer learning faded-out strategy). Further, the model will train the embedding layer with a Glove dictionary of 6 billion embedding vectors.

DNN layers have 64 units, BiLSTM contains 128 input sizes, and the vocabulary size is 10,000. The DeepConcat model is composed of multilayer CNN, BiLSTM, and DNN see Figure 5.2. For better optimization, it is planning to evaluate three datasets with the preprocessing levels and

dropout parameter. The skip-layer mechanism during the model architecture pipeline switch among layers to formulate all possible combinations like an ensemble method. Therefore, like to propose skipping layers-based combinations that will select all possible model architecture sequentially trigger by Bayesian optimization. These are model layers shared at model search pipeline, which will select anyone of the model from LSTM-DNN, BiLSTM-DNN, CNN-DNN, CNN-BiLSTM-DNN, CNN-LSTM-DNN.

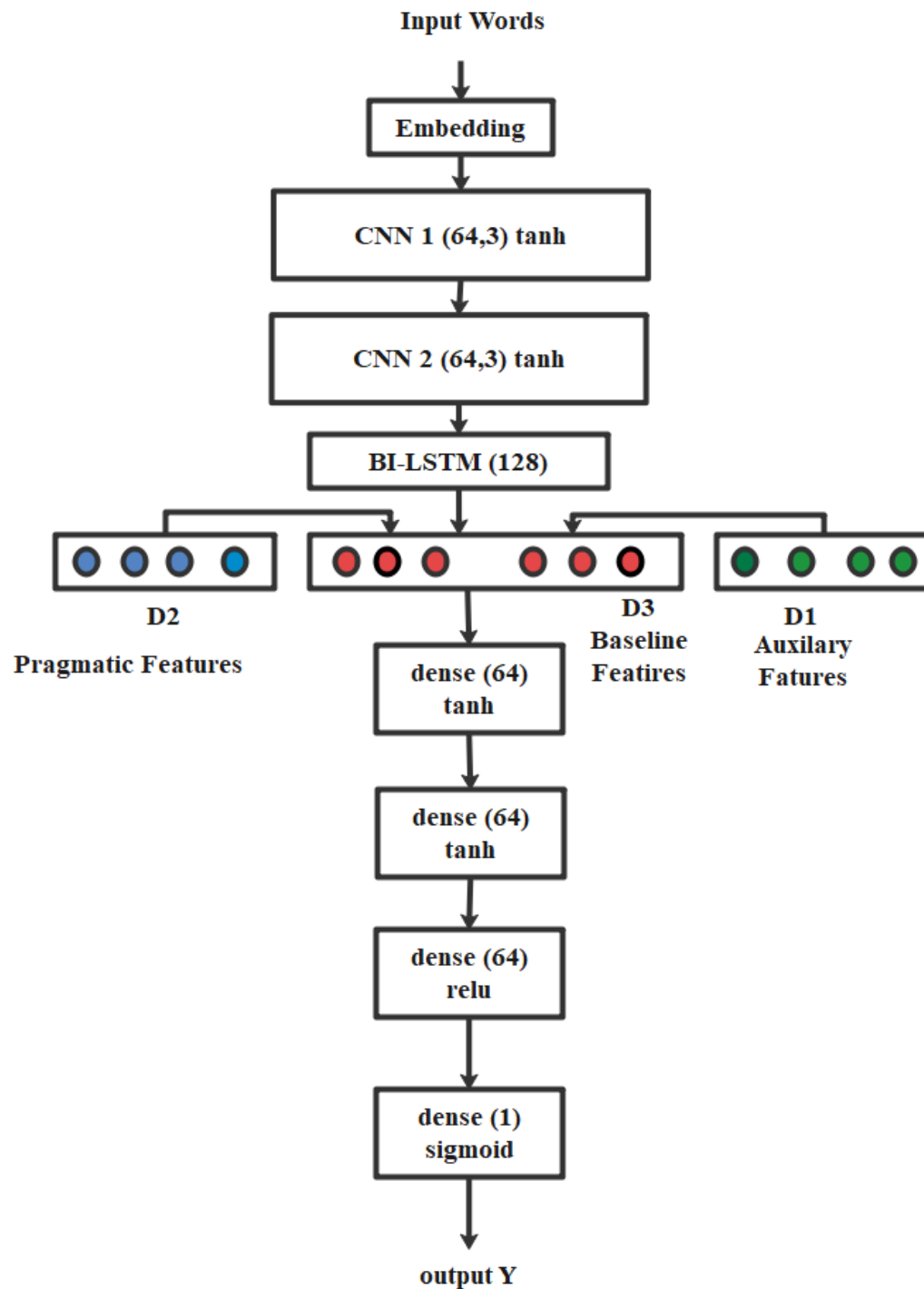


Figure 5.2. DeepConcat Model

The proposed model has eight layers.

- Input Layer: The input is tweet/news/amazon reviews/dialog discussion comments at this layer of the model x_1, x_2, \dots, x_n where x_i refers to individual words.
- Embedding Layer: Each word maps to the input vector using Glove 1.6 billion vectors that is called encoding.
- Convolution Layer: The convolution operation's purpose is to generate the convolve operation which output the feature vector.
- Down-sampling Layer: The max-pooling operation is used as the downsampling strategy for features.
- BiLSTM layer: The previous layer's output gets the strong feature vector feed output to BiLSM. This layer learned features D_3 .
- Concatenation layer: Feature vector auxiliary features categorized into the incongruity features D_1 and pragmatic features D_2 . These layers concatenate all features categories to produce a single set of combined features $D_1, D_2, D_3 \subseteq D$.
- Representation Layer: A fully connected three layers activate the features with tangent function at the first dense layer and second dense layer. The third dense layer performs the Relu and final layer will perform sigmoid activation function to generate the output Y as predicted variable.

The input gate (flow the information in), forget gate (tells the cell state which information to forget by multiplying 0 to a position in the matrix), output gate (flow the information out), and input cell state of BiLSMT activation of each cell unit can be calculated using Equations 1-6.

$$i_t = \sigma (w_{ix}x_t + w_{ih}h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma (w_{fx}x_t + w_{fh}h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma (w_{ox}x_t + w_{oh}h_{t-1} + b_o) \quad (3)$$

$$s_t = \tanh (w_{sx}x_t + w_{sh}h_{t-1} + b_s) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot s_t \quad (5)$$

$$h_t = \tanh(c_t) \odot o_t \quad (6)$$

Here some of the operations are denoted as follows.

- \odot is the element-wise product.
- w_i, w_f, w_o, w_s are the weight used for mapping hidden layer input to the gates those are input gate, forget gate, and output gate.
- b_i, b_f, b_o, b_s are the bias factors.
- σ is the sigmoid activation function that processes the information between 0 and 1.
- Tanh is a hyperbolic function that outputs values between 1 and -1 and finally, the output is $Y_t = \{y_1, \dots, y_t\}$.

The proposed framework will integrate algorithmic features and produced pretrain model which is a novelty. The next section proposed the AutoML framework which will automate the trained model along with other combinations of deep learning models: CNN, LSTM, and CNN-BiLSTM, CNN-LSTM. The newly developed AutoML framework automates the pretrain models which is novelty and contribution to the task of detection of sarcasm.

5.2 Adapted AutoML DeepConcat

This research will adapt the newly proposed AutoML DeepConcat framework for the NLP problem of sarcasm however, multiple task classification is not the aim. The model proposed here with feature engineering method that will integrate at dense layer.

5.3 Model Selection, hyperparameters optimization, and Architecture Search

A newly developed AutoML DeepConcat framework will allow the features engineering method that will integrate the general features at dense layer of the model dissimilar to the former AutoML TPOT (Olson, et al., 2016) and Auto-Keras (Jin, et al., 2019). The newly proposed algorithm is classified into two broad categories: explicit and implicit. The proposed AutoML DeepConcat framework will overcome the research gap of integrating the general nature of features that can be adapt to any domain, however, this Chapter's focus was on the integration of generalized features. Finally, proposed framework named as DeepConcat as illustrated in Figure 5.4a.

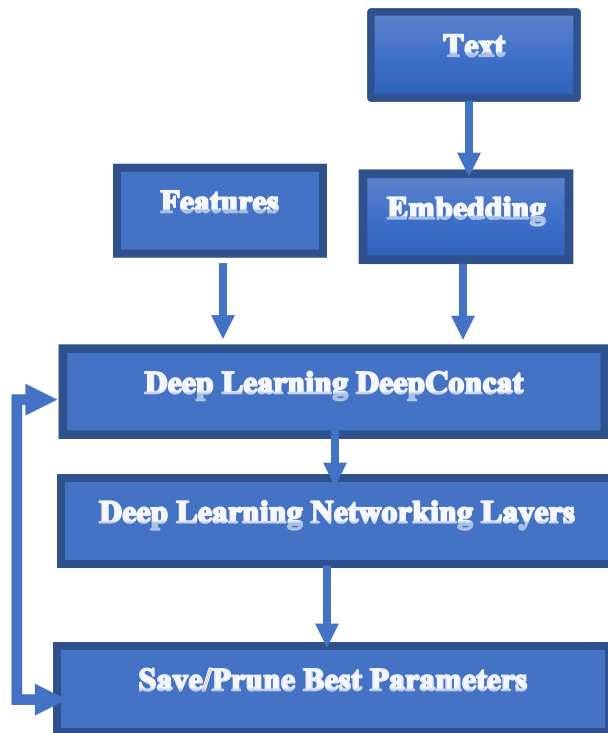


Figure 5.4a: AutoML DeepConcat Framework

The most popular hyperparameter selection methods are grid search, random search, and Bayesian search SMAC (Hutter, et al., 2011). The new framework DeepConcat is adapting the similar hyperparameter based automation machines like Auto-Keras and Auto-Sklearn that utilized the optimization method SMAC3, a better version of hyperparameter optimization than previous SMAC. The proposed framework AutoML DeepConcat set the grid-based search model pipeline that iterate to optimize five models' hyperparameter like dropout and learning rate. The model evaluated over datasets using the model with hyperparameters like dropout and learning rate. The

model will be saved with the best parameters and better performance metrics F1, accuracy, and AUC.

Initially, preprocess the tweets and Amazon reviews that select the best preprocessing level i.e., removal of punctuations and capital letters that is part of plan of preprocessing levels. These levels clean the tweets and reviews then evaluated over the model. AutoML framework DeepConcat selects the best model for each preprocessing level. Further, AutoML framework models will concatenate the features extracted from the implicit IIA and explicit EIA algorithms. All the features were concatenated at the hidden layer of the model during the model search pipeline. The ultimate purpose is to select the best model using hyperparameters optimization. In former research, the AutoML optimization technique is based on Bayesian optimization. The outcome of the AutoML is optimized parameters, that will select the best model which will be saved along parameters; however, the best model will be selected after multiple time iterations to get the best performance. If the existing model is less optimized, then a newly trained model at a particular training cycle then it will replace it. It is the AutoML DeepConcat framework that is based on the feature's integration. It is pretrain model that fine-tune for any domain, and it optimized the hyperparameters at each iteration. The pretrain model will optimize and evaluate the model performance. The automation would be time-consuming to train the best model at each training cycle.

AutoML framework train the models of CNN, LSTM, and BiLSTM which have many layers, the first layer is the embedding layer that takes input as numeric vectors (representation of the terms).

In case of CNN, at the convolution layer applied filter with convolution operation that convolve the input with filters to extract the features (this process is called a dot operation defined in Chapter 1, Section 1.6.1). The BiLSTM layer shape size is 32 timesteps and 100 sequence size for the input feature vectors at the embedding layer. As illustrated in Figure 5.6a, the concatenation layer concatenates baseline and user-defined features at the hidden layer. Here, user-defined features are divided into two categories: incongruity features and pragmatic features. The incongruity features are classified into implicit and explicit categories, that concatenated auxiliary features and baseline features at the hidden layer. Furthermore, Dense layers (dense 1, dense 2, and dense 3) consist of fully connected layers with activation function tangent and Rectified Linear Activation (ReLU) as shown in the below Figure 5.4b.

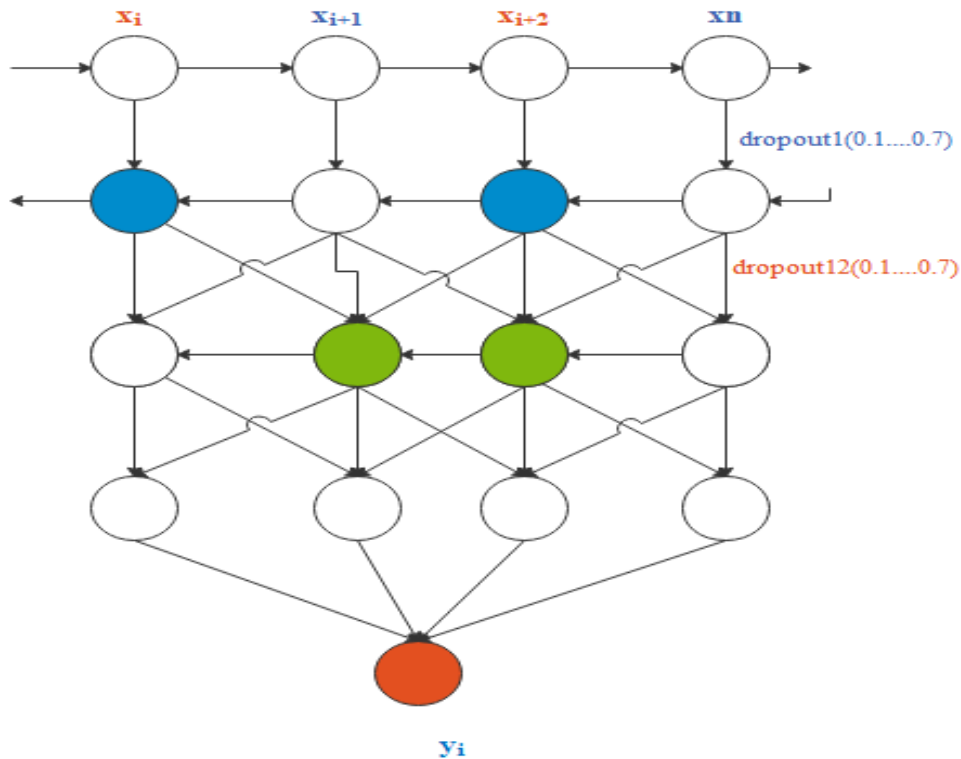


Figure 5.4b: Fully Connected Network

The last unit applied the Sigmoid function, which will output whether tweet is sarcastic or non-sarcastic.

$$y = \sigma(w_0 * C_{IJ} + b_0) \quad (1)$$

The proposed AutoML DeepConcat framework comprises of five models to train and evaluate during the model search and architecture search pipelines. Further, LSTM is elaborated comprehensively, Figure 5.4c illustrated that the model framework architecture with all layers, the sigmoid function is an activation function. The model has one bidirectional layer with 100 input unit and a 40,000 sequence of words length, 2 hidden layers are with 128 units, and has 1 output layer. Rectified linear activation function and tangent function process the features at the hidden layer, and a sigmoid activation function classifies the output into sarcastic and non-sarcastic output.

The last layer is the activation function Sigmoid that will classify Twitter tweets and Amazon reviews that classified the output into two classes sarcastic and non-sarcastic. The main aim of the AutoML based deep learning models is to classify tweets for sarcastic output.

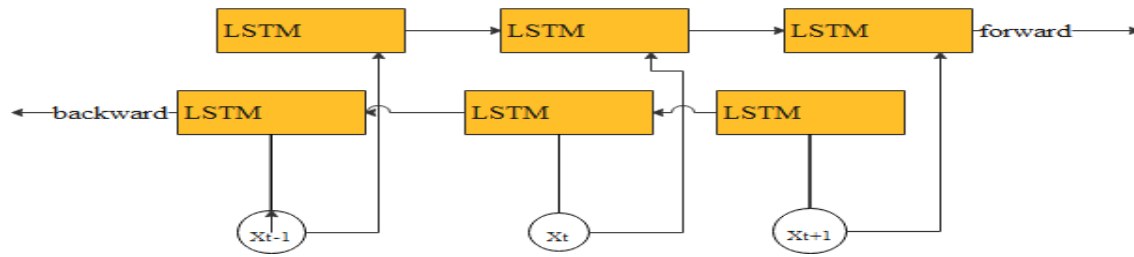


Figure 5.4c: LSTM feed-forward and backward

5.4 Result and Discussion

The plan is to preprocess levels for all models with automation of the AutoML DeepConcat framework. The preprocessing will automate that the model search pipeline, hyperparameter optimization, and selection of model architecture.

Further, the plan is to evaluate baseline ML models like SVM, Logistic, and KNN with integrated features. In addition to incongruity algorithm-based features, the pragmatic features are also part of the experiment.

The planned experiment setup involves optimal parameters.

[1] Optimize the hyperparameters dropout and model performance evaluated using Bayesian optimization.

[2] Firstly, the pre-processing levels are evaluated with multiple trainings. Thereafter, the automation initiated with the best model search, hyperparameter optimization, and model architecture selection.

5.4.1 Experiment Setup

The experiment setup will follow the steps of execution for the AutoML framework as given below.

- [1] The optimal drop-out strategy and model performance are evaluated using the Bayesian optimization technique, which will select the best parameters during model search and training.
- [2] The AutoML framework will execute a few operations during the experiment. The pre-processing levels are selected with planned levels (levels define the removal of the hashtag, stemming, hyperlink and special tags that is separate level for each cleaning) and automation of the best model, the optimization strategy is based on hyperparameters dropout, learning rate, and performance metrics.

Datasets

The experiment plan has three benchmark datasets DS1, DS2, and DS3 for the task of sarcasm detection (Riloff, et al., 2013; Ptáček, et al., 2014; Ghosh & Veale, 2017). The implicit incongruity and explicit incongruity algorithm extracted features from these three DS1, DS2, and DS3 datasets and provided input to the models of AutoML DeepConcat framework. As mentioned earlier DS1 benchmark datasets consist of 1,367 tweets training set and 588 test sets. Dataset DS2 contains 30,010 regular and 29,055 sarcastic tweets. The DS3 dataset have 24,453 sarcasm tweets and 26,736 non-sarcasm tweets. The description of these datasets is given in below Table 5.4.1a.

Table 5.4.1a: Description of benchmarking dataset

Identifier	Data Sources	Train (Features)	Test (Features)
DS1	(Riloff, et al., 2013)	321	80
DS2	(Ptáček, et al., 2014)	800	200
DS3	(Ghosh & Veale, 2017)	800	200

Setup

In the beginning, this research has considered various performance metrics for the evaluation: F1, accuracy, and AUC. The validation of the model performed by 10-fold validation. Further, compared the model performances with these metrics to get the best model performance and saved the best model with optimized parameters after training. The experiment will consider the parameters with/without dropout hyperparameter. The comparison of the framework models was conducted with the existing core models using evaluation metrics: F1, accuracy, and AUC. The model classified sarcasm with the integration of incongruity features which were extracted from newly proposed algorithms EIA and IIA. These features will feed further in the form of input to the core models SVM, logistic, and KNN, and integrate at hidden layers of deep learning models.

5.4.2 Preprocessing over DS3

The preprocessing level plan is investigated in Table 5.4.2 which shows that the AutoML DeepConcat framework automates the model's evaluation using a skip layer mechanism with Bayesian optimization and performance metrics. The skip layer selects layer at each cycle and skip

all other layers to formulate the deep learning model. The best metrics will be selected after hyperparameters optimization. The preprocessing level P1 remove hashtag as given in Table 5.4.2, it proved that it has significant result of F1 when train with BiLSTM among all other levels such as P1-P2, P1-P3, and P1-P5 and models. The second highest significant of P1 is 0.98 F1 over CNN-DNN model, thus both models will opt P1 level of cleaning that is Hashtag. The level P1-P2, P1-P2-P3 is not having more significance F1 than P1 level over all models, however, P1-P2-P3-P4 levels performed equally well 0.98 F1 over BiLSTM but over CNN-DNN model it is 0.96 F1 thus cannot be selected further. Bayesian optimizations trains the models with random dropout at grid search and evaluate the models using Bayesian optimization.

Table 5.4.2: Preprocessing Level F1 results on DS3 dataset

	LSTM -DNN	CNN -DNN	BiLSTM	CNN-LSTM-DNN	CNN-BiLSTM-DNN
P1	0.97	0.98	0.99	0.97	0.96
P1-P2	0.97	0.96	0.97	0.96	0.95
P1-P2-P3	0.97	0.95	0.97	0.97	0.98
P1-P2-P3-P4	0.97	0.96	0.99	0.96	0.96
P1-P2-P3-P4-P5	0.97	0.97	0.97	0.97	0.96

Figure 5.4.2 and Table 5.4.2 shown the result of preprocessing level of all models and all preprocessing levels have more than 95% significance. Preprocessing is required to clean the tweets with hyperlinks, tags, and other context information that would enhance the performance of the model, but cleaning everything from the tweet might not be significance. The P1 and P1-P4

levels achieved 0.99 F1 with BiLSTM that is better than P1-P2 0.98 F1 on BiLSTM and 0.99 F1 on CNN-BiLSTM-DNN. Therefore, it is important to select either P1 or P1-P4, further these preprocessing levels will be part of method for experiments, it means there is no need to prepare the data with complete cleaning levels thus opted the P1 or P1-P2-P30-P4 level.

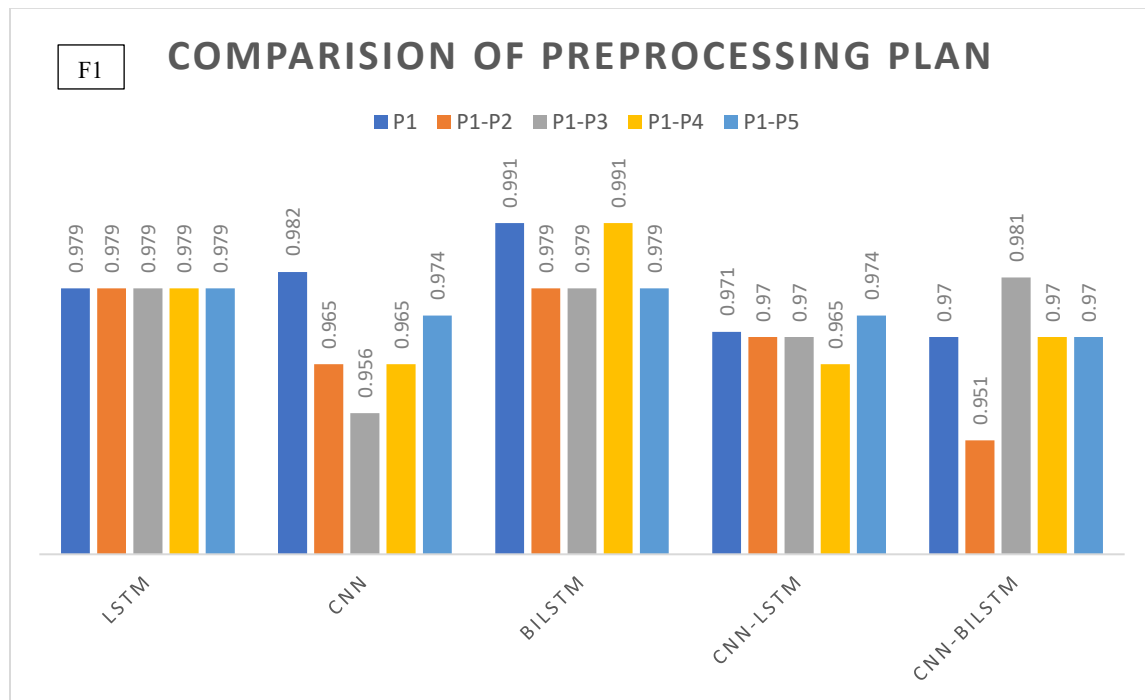


Fig. 5.4.2. AutoML DeepConcat models result with impact over preprocessing Levels

The diagram above validates the proof that AutoML DeepConcat models performed different over different preparations, however, BiLSTM performed better when applied with Hashtag only P1 level. The AutoML automates all model's preparation separately using model search pipeline so need to train each model separately.

5.4.3 AutoML DeepConcat Evaluation

It was observed that the proposed AutoML DeepConcat framework search the best model. The AutoML DeepConcat Framework models produced the best results at preprocessing level P1. It was also observed that inclusion of all preprocessing levels was not effective in existing research (Riloff, et al., 2013; Joshi, et al., 2015). Interestingly, hyperparameters of dropout layer set dropout value between the dense layers, one after the convolution layer, and one after pool max. AutoML DeepConcat framework skip layers mechanism or architecture pipeline iteratively select the model as pre-selection criteria while keeping input and output layers fixed. The optimal parameters were the outcome of results with the best preprocessing level P1, as given in Table 5.4.3a.

Table 5.4.3a: All features, incongruity, and pragmatic features impact

Methods	Metrics/ Parameters	LSTM -DNN	CNN -DNN	BiLSTM	CNN- LSTM- DNN	CNN-BiLSTM- DNN
All Features + Baseline	Dropout_1	0.086	0..086	0.086	0.086	0.086
	Dropout_2	0.013	0.03	0.013	0.013	0.013
	Accuracy/F1	97.5%/0.98	96.6%/0.97	98%/0.98	96%/0.97	96%/0.97
	AUC	0.973	0.96	0.98	0.96	0.96
Incongruity Feature + Baseline	Dropout_1	.086	.086	0.086	0.5	0.086
	Dropout_2	.013	.013	0.013	0.2	0.013
	Accuracy/F1	97.6%/0.98	97.6%/0.98	98%/0.98	95%/0.96	95%/0.96.
	AUC	0.978	0.97	0.97	0.946	0.967
Pragmatic Feature + Baseline	Dropout_1	0.28	0.089	0.086	0.12	0.086
	Dropout_2	0.13	0.013	0.013	0.26	0.013
	Accuracy/F1	97.6%/0.98	96.5%/0.97	97.5%/0.98	96%/0.97	96%/0.97
	AUC	0.97	0.96	0.97	0.95	0.96

5. The newly developed AutoML framework for Sarcasm Detection

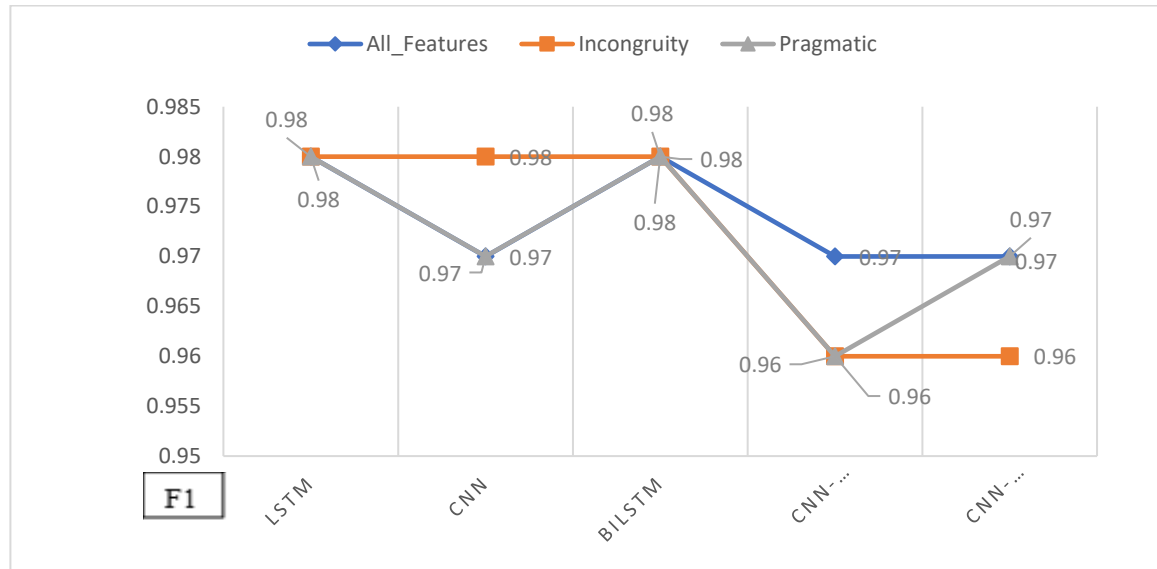


Figure 5.4.3a AutoML DeepConcat optimization and robust framework F1 impact over features and its categories

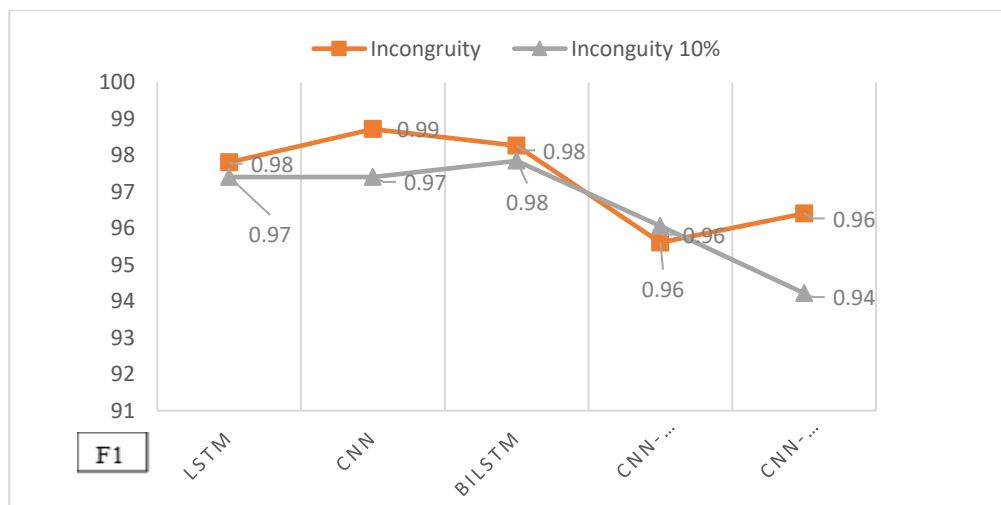


Figure 5.4.3b AutoML DeepConcat optimization with 10% fixed dropout and AUC impact over features and their categories

All the results were evaluated and compared with methods baseline + all-features, baseline + pragmatic, and baseline + incongruity as given in Table 5.4.3a. Here, the features have better significance for the sarcasm detection when setting the dropout values in the model. Therefore, the model will set fixed dropout values between 0.5 and 1 during the model search pipeline. Following that, set the dropout rate at 0.10 and F1 is less in the case of heuristic-based dropout values. The performance was also compared with randomly set dropout values using grid-based Bayesian optimization. The merits of optimization are more than the demerits of computational cost. The computation cost is minimized with the newly proposed DeepConcat AutoML framework during the training and prediction of the models. The best method among all categories is the baseline + all-features as shown in Figure 5.4.3a and Table 5.4.3a because it produced better results for the model BiLSTM, the score is F1 0.98 as compared to the other methods baseline + incongruity and baseline + pragmatic features over other models. The performance of the baseline + incongruity feature method was significant on CNN-DNN and BiLSTM among other models.

Similarly, baseline + pragmatic feature method is significant on BiLSTM, CNN-LSTM-DNN, and CNN-BiLSTM-DNN rather than on CNN-DNN and LSTM-DNN. But the baseline + all-features method outperformed the other methods baseline + pragmatic and baseline + incongruity methods. A comparison of these methods was illustrated in Figure 5.4.3a and Figure 5.4.3b. It depicts that AutoML DeepConcat framework models with level P1 and all-features-based methods performed better than the baseline method. Specifically, the all-features method in comparison to incongruity and pragmatic features is better as given in Table 5.4.3a. The baseline + all-features

method was more effective over BiLSTM model while less effective when evaluated over datasets DS1 and DS2. The results of comparisons of F1 on core models and BiLSTM is presented in Table 5.4.3b and as illustrated in Figure 5.4.3c.

Table 5.4.3b: All-features method comparison on F1 with core models and the DeepConcat best model BiLSTM

	BiLSTM	SVM	KNN	Logistic
DS1	0.96	0.83	0.937	0.91
DS2	0.95	0.95	0.95	0.90
DS3	0.98	0.73	0.74	0.69

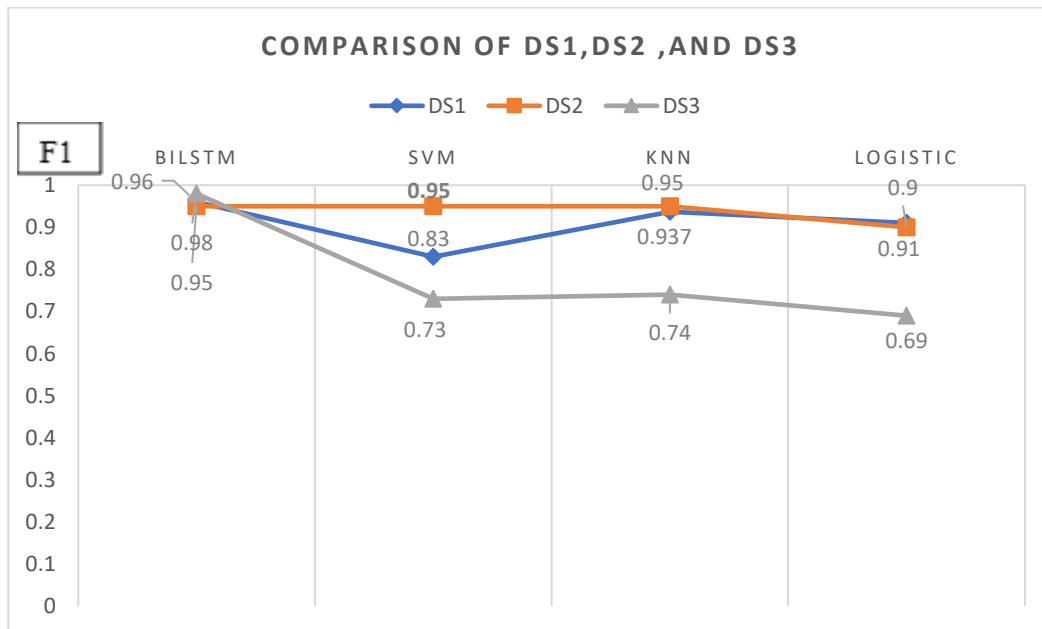


Figure 5.4.3c. BiLSTM, SVM, KNN, and Logistic models' F1 impact over DS1, DS2 and DS3

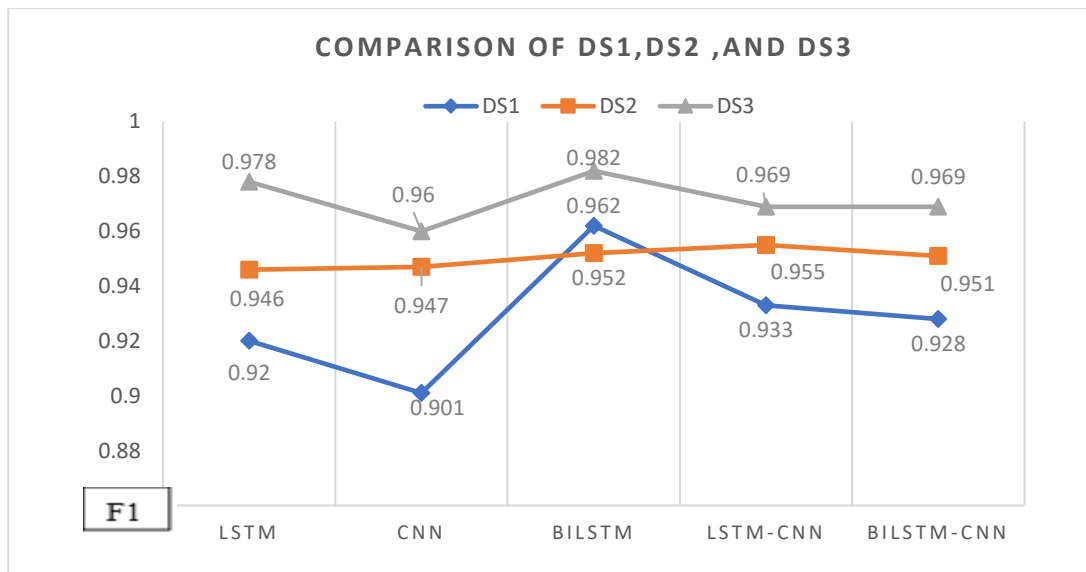


Figure 5.4.3d. AutoML DeepConcat framework all models F1 comparison on Datasets

It was observed that core model SVM is better on DS2 as compared to other core models KNN and logistic, however, DeepConcat framework model BiLSTM outperformed all core models in comparisons. The comparison of all datasets over the DeepConcat AutoML framework based BiLSTM model is more effective, but it performed better when evaluated over DS3 as compared to any other datasets in Figure 5.4.3d. It was observed in experiment that deep learning models are better for sarcasm detection than core models. However, drop-out value with the random generation during Bayesian optimization for model search pipeline is an effective performance of the model. Moreover, compared the performance of deep learning models, which depicts that BiLSTM is better among all other models. The new framework is outstanding in performance for the sarcasm but also time-efficient because it finished training in few hours rather automate for longer hours. The incongruity features extraction is novel algorithm that is the contribution in this research. The method is new because integrating the algorithmic features in the novel automated framework AutoML DeepConcat.

5.5 Conclusion

The newly developed AutoML DeepConcat Framework was explored with hyperparameter optimization. AutoML DeepConcat framework has proven that the performance of new framework is better than the non-automated ML technique. The results indicated that the best model selection is based on a random dropout hyperparameter that will optimize the model as compared to the heuristic-based dropout values. Two factors influence the performance of the proposed AutoML DeepConcat framework. Firstly, the preprocessing plan levels to get the best preprocessing

because, if select all preprocessing levels that would degrade the performance, therefore, it was needed to decide which level of cleaning is more effective for sarcasm detection after training. Secondly, the feature extracted with the novel algorithm will integrate into models of DeepConcat framework. Automation framework gives performance in term of metrics like F1, AUC, and accuracy. It is required to consider all metrics but none of metrics can be compared. Because AUC would be equal to F1 in performance up to sample accuracy rather than evaluating entire dataset on all episodes, so it is not useful to consider real comparison between F1 and AUC. According to the experiment in this Chapter, it was explored that optimization with preprocessing plan levels produced better results. But not all the levels have an equal contribution to model performance and optimization, thus concluded that the minimum cleaning is better for sarcasm detection because text might contain clues in the form of text context.

All the datasets have diversity in performance at preprocessing level that was observed from the AutoML DeepConcat framework models. Integrated features are essential with the parameter optimization, which proved that all features (incongruity and pragmatic) method is more significant with F1 than other methods as given in Table 5.4.3. Individually, it means that pragmatic features are not a better contributor to model performance than incongruity features. Consequently, all features method is the best method that outperformed other methods like pragmatic and incongruity. Lastly, the novelty of the AutoML DeepConcat framework is the concatenation of incongruity features extracted from the novel algorithms EIA and IIA, which is the main contribution to existing literature. Furthermore, these features are generalized and can be adapted to any social media domain.

Another contribution is the automation of the AutoML DeepConcat framework, which is comparative and performance persuasive. The feature engineering is performed with conventional AutoML-Keras, but it is limited to features extraction due to specificity to a particular dataset thus not all datasets support common features. Therefore, proposed the AutoML DeepConcat framework that would transfer knowledge using common features among different domains, which is another concept of transfer learning, that is common knowledge shared among social media domains and other NLP domains. The deep learning-based framework incorporated various models and evaluated performances of these model's LSTM, CNN, BiLSTM, CNN-LSTM, and CNN-BiLSTM. The results also concluded that the BiLSTM model is better than all other models in search pipeline. It is also proved that deep learning models are better than core models. All datasets were evaluated, which were shown remarkable results with the deep learning-based framework in comparison to the core approaches. Searching the model required extra computation cost to run for multiple iterations to select the best model.

In the future, it is planning to observe the AutoML DeepConcat framework models whether it can be adapted to other domains like Amazon reviews, Forbes News, and dialog discussion comments. Further. the hypothesis can observe pretrain model for the optimization, hyperparameter, and model search for multiple domains. Therefore, the model would adapt to various domains using generalized features.

CHAPTER 6

Transfer learning - Domains Adaptation

The text data is either long or short that fall into various domains so presented in various ways. Twitter tweets have a hashtag in the form of the short text. Likewise, source dialog discussion where a question has a response. The dialog discussion comments have varied sizes and informal sentence structures. Additionally, the Amazon domain reviews have noisy text that found in the sarcastic text that would impact the product sale because of low ratings. Therefore, it is important to identify the sarcasm with the AutoML DeepConcat framework models, which will recognize the sarcasm among multiple domains. This research will propose multiple domain adaption using the pretrained ep learning models.

6.1 Data Source

The pre-train model train over multiple datasets that belong to various data sources. (Poria, et al., 2016; Felbo, et al., 2017) the former authors proposed a method of pretraining the model over informal text of the tweets. The model pretrain over a dataset using formal and informal text to observe the impact of structured text with the accuracy. The formal news dataset taken 1.3 million news taken from (Khodak, et al., 2017; Ghosh, et al., 2015), the informal text is the tweet dataset as described in below Table 6.1a.

Table 6.1a: Imbalance dataset for pretrain

Data Sources	Total Sample	Train	Test
Reddit News (Khodak, et al., 2017)	60000 News	48000	12000
Tweets (Ghosh, et al., 2015)	50000 Tweets	40000	10000

The benchmarking dataset split the dataset into training and test dataset with 80:20 ratio, here huge text required to pretrain the model so later fine-tune over other domain text. The formal text represented by Reddit News and informal text are represented by Tweets. The model will pretrain using news and tweets text to evaluate its performance to transfer knowledge to other domains. The domain adaption was performed by former research using diverse datasets like the dialog discussion dataset (Oraby, et al., 2017) and the Amazon dataset (Filatova, 2012). The dialog discussion domain provides a rich source of data in the form of discussion comments. The dataset chosen as the benchmark dataset; it has 3 columns and more than 10,000 discussion records. It acquired the data from the discussion dialog comments which are generally comprised of questions.

The Amazon reviews have star rating associated with reviews that indicate the liking and disliking of the Amazon products. Sarcastic reviews constitute of positive words but with negative opinion words as well. The negative opinion reviews are usually rated with low stars in reviews.

Mostly the sarcastic reviews are written by people who give low ratings to the products refer to the Table 6.1b, 59% of reviews have 1-star and 74% regular reviews rated 5-stars. The reviewers have the general understanding of sarcasm when submitting reviews of products. The rating of reviews changes the positive literary meaning of a text utterance to negative. Thus, machine-intensive work is required to find the sarcastic reviews among low and high star ratings.

Table 6.1b: Distribution of stars assigned to reviews

		1*	2*	3*	4*	5*
Sarcastic	437	262	27	20	14	114
Regular	817	64	17	35	96	605

The low star rating reviews have a more sarcastic ratio due to a negative utterance in the sentence. The reason is to include low-star rating reviews because those have more sarcastic responses, which will be analyzing by ML technique as given in Table 6.1b. The 1-star reviews belong to the regular category reviews that are limited in numbers therefore selected all-star ratings such as 2-star, 3-star, 4-star, and 5-star review.

6.2 Methodology

(Felbo, et al., 2017) the former researcher has applied the transfer learning technique using BiLSTM. The word embedding feeds to the first layer of BiLSTM. Further, the BiLSTM model takes these vectors from the first layer and passed to the hidden layers which will transform the

linear combination into the value between 0 and 1 using the activation function. But the abstraction of features at hidden layers is not visible due to hidden complexity of the network. LSTM transfer learning strategy and its integration with CNN classify the verbal irony (Rangwani, et al., 2018). However, the tokenization with discrete features will classify the tweet into verbal irony that is better than any other approaches like n-gram proposed by a former author (Ptáček, et al., 2014).

Conceptually, these embeddings are the vector representation of the words that feed to the deep learning network input layer. (Felbo, et al., 2017) the former author proposed a model DeepMoji applied transfer learning to pretrain the model over 1.2 billion tweets and finally pretrain model fine-tune to other domains. The outcome of the words vector from final layer classifies the tweet with SoftMax function. Similarly, the output of important words with the attention mechanism has different levels of confidence score, which indicate the encoding will provide important word vectors information to decoder. The attention mechanism is the word weightage criteria using the concept of word strength so that unimportant words have low confidence score. To move in a similar direction, there is need to explore discrete features that concatenated the baseline features at the hidden layer of CNN, the model proposed by (Poria, et al., 2016). The proposed approach aim is to classify sarcasm over multiple domains with the contribution to literary society.

6.2.1 Transfer Learning Strategies

Domain adaptation is the main approach to transfer knowledge from one domain to other. This approach's main objective is to change the underlying data distribution by finding common features among domains. Thus, this research plan is to devise new framework with the domain adaptation characteristics, which classify the sarcasm using the AutoML framework-based deep learning models.

There are strategies for freezing layers like the last layer, full layer, and chain-thaw. The last layer strategy was the famous strategy proposed by (Donahue, et al., 2014). The concept of all layers of the model will freeze all layers except the last layer then fine-tune to the target dataset. Alternatively, another common strategy was proposed full layer by (Erhan, et al., 2010), that freeze all layers of the model then fine-tune the other domain dataset.

(Felbo, et al., 2017) proposed the new transfer learning strategy which is called “chain-thaw”. The new ‘chain-thaw’ strategy achieved better performance over fine-tune tasks of other domains compared to ‘last’ and ‘full’ strategies. That model is multi-domain and multi-task which transfer knowledge among multiple domains and classify multiple tasks such as sentiment, emotions, and sarcasm. Further, the aim is to explore these strategies ‘last’ layer, ‘full’ and ‘chain-thaw’ with comparison of the performances in experiments.

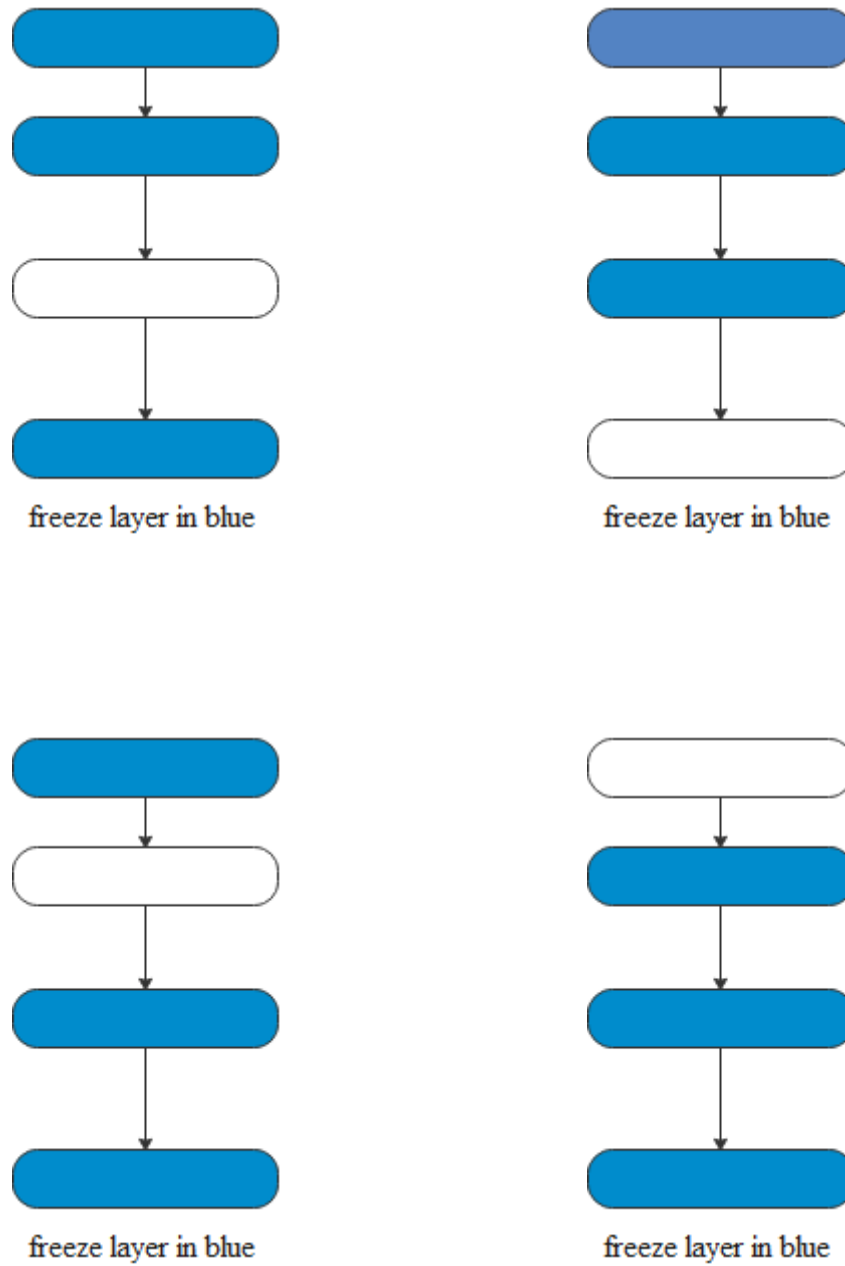


Figure 6.2.1: Chain-thaw strategy to freeze all layers sequentially.

Chain-thaw freezes fine-tune the model by sequentially freezing the layers of model. Here, one layer unfreezes at a time as illustrated above in Figure 6.2.1.

6.3 Pretraining Supervision

Distance supervision is the training that uses data obtained from heuristic and domain expertise where data is noisy, and patterns are pre-defined by (Felbo, et al., 2017). The former research trains the model on tweets which involve emoji and collected 1.2 billion tweets. The multitasking model DeepMoji proposed for sentiment analysis, emotion analysis, and sarcasm detection.

6.4 Experiment

Domain adaptation in the model will be applied in three steps.

- 1) Load an existing model and add some dense layers
- 2) Train the extended model on your data
- 3) Add more dense layers train and fine-tune the model over data.

6.5 Existing Pre-train model

BERT is developed by Google AI; it is a good initiative for multiple domains and multiple task detection. Usually, BERT performance is better on language translation tasks, but the alternative pretrain LSTM model proposed by (Felbo, et al., 2017). It was created with attention layer outperformed slightly BERT on multiple tasks classification like sentiment, emotion, and sarcasm.

Therefore, to prove this hypothesis, the AutoML pretrain model LSTM and its combination which are a better alternative than existing pretrain models like BERT and fastText. These pretrain models are evaluated on benchmark dataset SemEval-2018 Task3 as given in below Table 6.5.

TABLE 6.5: Performance of Pre-Train Existing emerging models on Semeval-2018 Task3 test dataset A.

	Accuracy	Train
Bert	72% approx.	100
fastText	56% approx.	100

One thing was observed during experiment that BERT is slow in training due to more dense layers. That is why it is not possible to implement model at a single core dual or quad-core local machine (Disha S, 2017). Instead of building a model from starting point to solve a similar situation, we use the model to train on other issues. For example, if one wants to build a self-learning car. One can spend years building a decent image recognition algorithm from scratch instead take a pre-trained model from Google, built on ImageNet data to identify images in pictures. A pre-trained model may not be 100% accurate on multiple tasks, but it saves enormous effort to build and train the model from scratch for multiple tasks. However, it may be good for one task rather than on another task.

How can I use Pre-trained Models?

(Poria, et al., 2016) the pretrain model proposed the framework that integrate baseline CNN with features of emotion, personality, and sentiment categories. The pretrain model extracted sentiment features from the sarcastic dataset Semeval-2014 (Rosenthal, et al., 2019). Sentiment features feed in the form of input to the CNN model as baseline features then pre-trained model integrates with baseline features at hidden layer. The pretrain model was initially applied over dataset SemEval-2014, extracted features that classify the output into positive, negative, and neutral categories. These categories of features combined with the baseline features at hidden layers to classify the sentiment tweets.

The former framework concatenates the multiple features at pretrain CNN model hidden layer like emotion features of six categories extracted from the dataset (Aman & Szpakowicz, 2007). The baseline model will set input node according to number of features like 100 input features so 100 nodes. The pretrain of model methodology is not complex however training time will be more, whereas fine-tune time of the model over other dataset is minimal. For this reason, the LSTM model has proposed the embedding layer, which is the input and output layer.

Model will be trained using the last layer strategy (Donahue, et al., 2014). (Howard, et al., 2020) experimented with a gradual unfreezing plan that is one by one layer unfreezing and get results over pretrain model, the model proposed was Universal Language Model Fine-tuning (ULMFIT)

that model fine-tune the other dataset using the transfer learning technique that can help in various NLP tasks. (Peters, et al., 2019) later find the relative performance to fine-tune the model on the target task of other datasets.

The objective is to pretrain the deep learning model to identify the correct weights for the network with multiple feedforward and backward iterations. The feedforward and backward strategy will update weight during each iteration. The pretrain models trained on large datasets and saved for fine-tuning later for other datasets. The model will be load with architecture and weights then transfer all knowledge with help of freezing layer strategy. (Felbo, et al., 2017) concluded that the pretraining the model over single task is better than multiple tasks, however, not all datasets fine-tuning performances were better like over dialog discussion comments. It leaves the room to elaborate further to enhance performance proposing single task pretrain model for multiple domains.

6.6 AutoML Pretrain Proposed Model

The model pretrain over informal text like tweets, however, it is also having the impact of training the model over formal text like News. AutoML based pretrain model has two data sources to pretrain over formal and informal text sources such as Reddit News and Twitter tweets. AutoML framework searches the best pretrain model during the model search pipelines. The BiLSTM model will pretrain using the Bayesian optimization and hyperparameter drop-out parameters which will search the best model. The grid search optimization pipeline sets the random drop-out

parameters to minimize the cost of computation for each epoch. Finally, the model will be saved with extension dot “h” format with all the weights and layers at the local drive (Figure 6.6).

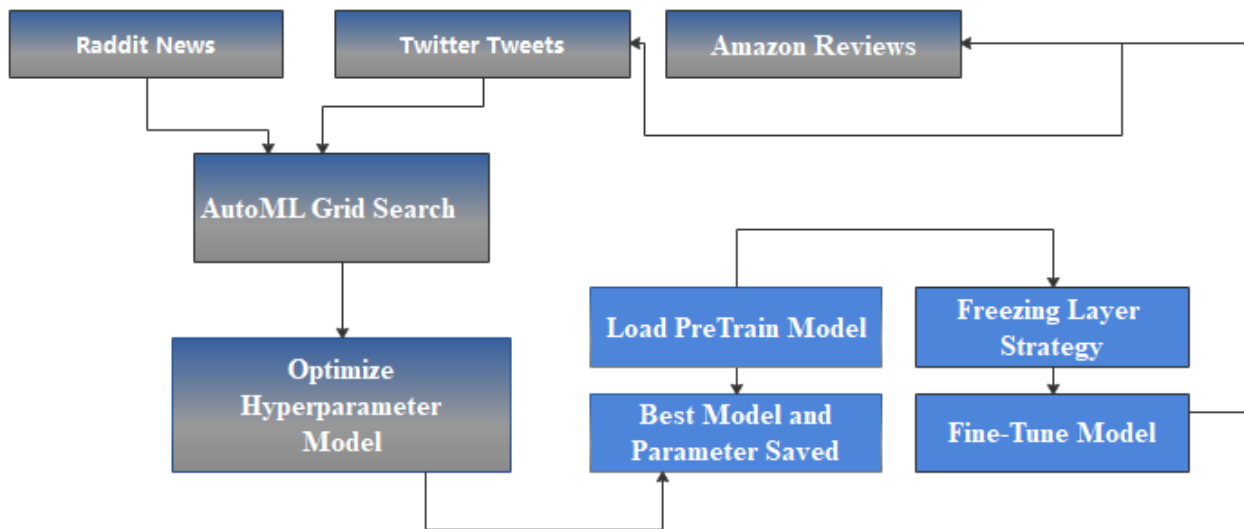


Figure 6.6: Pretraining and Fine-tuning of models of Proposed DeepConcat Framework

Several pretrain models are available to transfer knowledge using effective strategies to fine-tune model for other domain datasets. These strategies are effective on domains adaptivity over datasets. The proposed framework will implement the freezing strategies like last, full, and chain-thaw to fine-tune the pretrain model, however, will also plan to devise new strategy. The adapted models from the previous Chapter will be part of the framework AutoML DeepConcat for pretraining. That comparison concluded that BiLSTM is the best for sarcasm detection however, CNN is also effective among other models.

Firstly, the model pretrain on formal datasets like the Reddit news, and training is ubiquitous due to the nature of the data. Reddit dataset has the volume of 1 million news datasets but due to computation limitations, this research cannot choose all of datasets for pretraining. The dataset size was 50k, but it is still a large enough dataset to train using BiLSTM and CNN. But due to the more layers, the training will take more time in hours. Therefore, it restricts model training up to two epochs in the case of BiLSTM because training took 3-4 hours. However, not all models have similar training time like CNN, which is having 15 epochs because it converges at episode 13 after observing the inflection point. The hyperparameters optimization is performed using Bayesian optimization with grid search. The framework will minimize the grid search time by setting the drop-out based on the randomly generated and heuristic-based values. The heuristic-based dropout value is fixed when training the model. On the other side, randomly generated dropout values are varied between thresholds. These dropout parameters are vital because it will optimize the result. The dropout and learning rate parameters optimize the model with the planned episodes. It takes more than 3-4 hours to pretrain the model BiLSTM and CNN with the preprocessing level P1. The lower layer of the model represents the general features whereas the higher layers represent domain-specific features that is problem-dependent. During the experiment, it was observed the weights in the network adjusted on the contrary the freezing layer where weights were unchanged. The transfer learning strategies freeze more layers to avoid overfitting. One question arises here whether the chain-thaw has a better freezing layers strategy so fine-tune the model over new dataset. But the concept of chain-thaw will take more computation. On the other side, a large

dataset is selected to pretrain the model by training more layers for multiple domains. Therefore, evaluated the model performances using ‘last’ layer and ‘full’ layer strategies.

The “last” layer strategy experimented with a few optimal parameters for computation reasons. The last layer strategy feed model with 50k news to pretrain the model, which takes 11 min for each epoch. The model trains up to 10 epochs and learns 40k vocabulary words using the Glove embedding dictionary of 1.6 billion-word vectors. The limitation of epochs is due to training the model for hours in a single machine that would not converge. The pretrain model will be saved and loaded. The rule is that all layers freeze except the SoftMax layer of the model. Consequently, the model will be pretrain over the informal text of the tweets and optimized the parameters as well. The framework will pretrain the model over the imbalance dataset of Reddit news with grid search during iterations 2 to 5. On the other hand, choosing 10 epochs convergence takes more time as it is computationally expensive due to the large dataset.

Table 6.6: Pretrain Models

Pretrain Models	AutoML Parameters	Data Sources	Train	Test	Accuracy	Recall	AUC	F1
BiLSTM	Epochs 5 Grid 1-2	Redditt News (Khodak, et al., 2017)	48000	12000	62.34	93.74	75.42	0.73
	Episode 15 Grid 2-5	Tweets (Ghosh, et al., 2015)	40000	10000	65.71	72.68	59.88	0.71
CNN	Epochs 5 Grid 1-2	Redditt News (Khodak, et al., 2017)	48000	12000	63.42	80	62.27	0.72
	Grid 1-2 Grid 2-5	Tweets (Ghosh, et al., 2015)	40000	10000	64.55	88.22	60.34	0.74

The experiment is performed to pretrain the above models in Table 6.6 which will transfer knowledge from one domain to other domain by applying strategies of transfer learning. The proposed Framework DeepConcat outcome is best pretrain model on Reddit News and tweets. The results were evaluated and presented in Table 6.6 the train model will save as the pretrain model. The work of the AutoML framework was based on pre-train models after comparison and optimization. In comparison, the performance metric F1 is better to pretrain model BiLSTM as compared to CNN on the news formal dataset. However, the performance of CNN F1 is 0.74 but, BiLSTM is at 0.71 that indicated that performance F1 is better on formal text. Although most of the domains are informal even CNN performance of F1 is 0.74 after more epochs it is proven that BiLSTM is better when train and fine-tune to other domains. However, CNN train up to 15 epochs, and the Grid model iteration is set to 2-5, which produced a better F1 score over informal text like tweets. The pretrain models will pretrain over an unsupervised dataset that is why the accuracy rate is also issue. The accuracy rate will decide the model's pretraining efficiency over big data. In

the case of BiLSTM epochs are 15 and grid search set to 2, the total time taken for pretraining is more than 6 hours and produced an accuracy of 65%. The best dropout rate is 0.086 and 0.12, whereas the learning rate is 0.94. Further, the framework pretrain model will transfer knowledge to other domain datasets. As most of the benchmark datasets are informal thus like to opt the BiLSTM for fine-tuning task using various strategies.

6.6.1 Last strategy

(Donahue, et al., 2014) the strategy was proposed named as ‘Last’, it will freeze all layers except the last layer as given in Figure 6.6.1. The domain adaptation is performed for the benchmark dataset of Twitter’s tweets, (Ghosh, et al., 2015), and Dialog SCV2-Gen discussion, (Oraby, et al., 2017), and Amazon reviews, (Filatova, 2012). The last layer strategy only unfreezes the last activation layer. The “Last” layer strategy will optimize performance using pretrain models on the Twitter tweets dataset (Ghosh, et al., 2015) and Amazon reviews dataset (Filatova, 2012). Nevertheless, this strategy ‘last’ performed better than the existing state-of-the-art ‘last’ strategy with the score of 0.88 F1 Table 6.6.1.

Here in this diagram on the left indicate the freezing of all layers, however, on the right ‘last’ layer unfreeze only however, all layers freeze as shown in Figure 6.6.1.

Figure 6.6.1: Last layer strategy from pretrain BiLSTM with 512 units

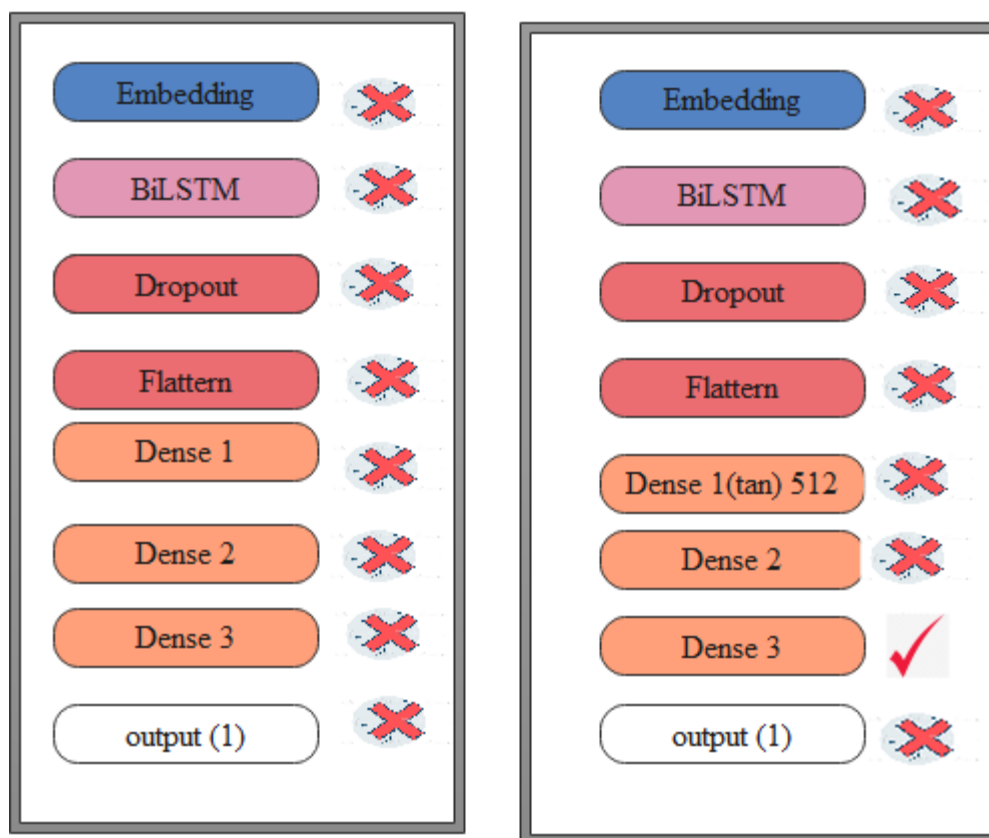


Table 6.6.1: Benchmark imbalance dataset performance on AutoML-Deep pretrain framework

(Last Approach)

Data sources	Data Sample	F1	AUC	Accuracy
Tweets (Ghosh, 2015)	1000	0.73	0.49	58%
		0.70	0.59	61%
Amazon (Filatova, 2012)	100	0.69	0.44	55%
		0.74	0.44	59%
SCV2-Gen (Oraby, et al., 2017)	2000	0.86	0.57	76%
		0.88	0.72	78%

The results in Table 6.6.1 explained that pretrain model over dialog discussion dataset was excellent in terms of F1 and accuracy.

6.6.2 Chain-Thaw Strategy

Initially, this strategy was discovered by (Felbo, et al., 2017), that train one layer at a time. In the proposed AutoML-DeepConcat framework, it was evident that three dense layers following the BiLSTM, framework plan to fine-tune the dense layers with one layer at a time concept, the model will converge when all layers trained one by one as shown in diagram Figure 6.6.2. The BiLSMT layer will train as shown on the left and then on the right freeze BiLSTM layer and train another layer drop-out. This freezing will continue until all layers will be train one by one.

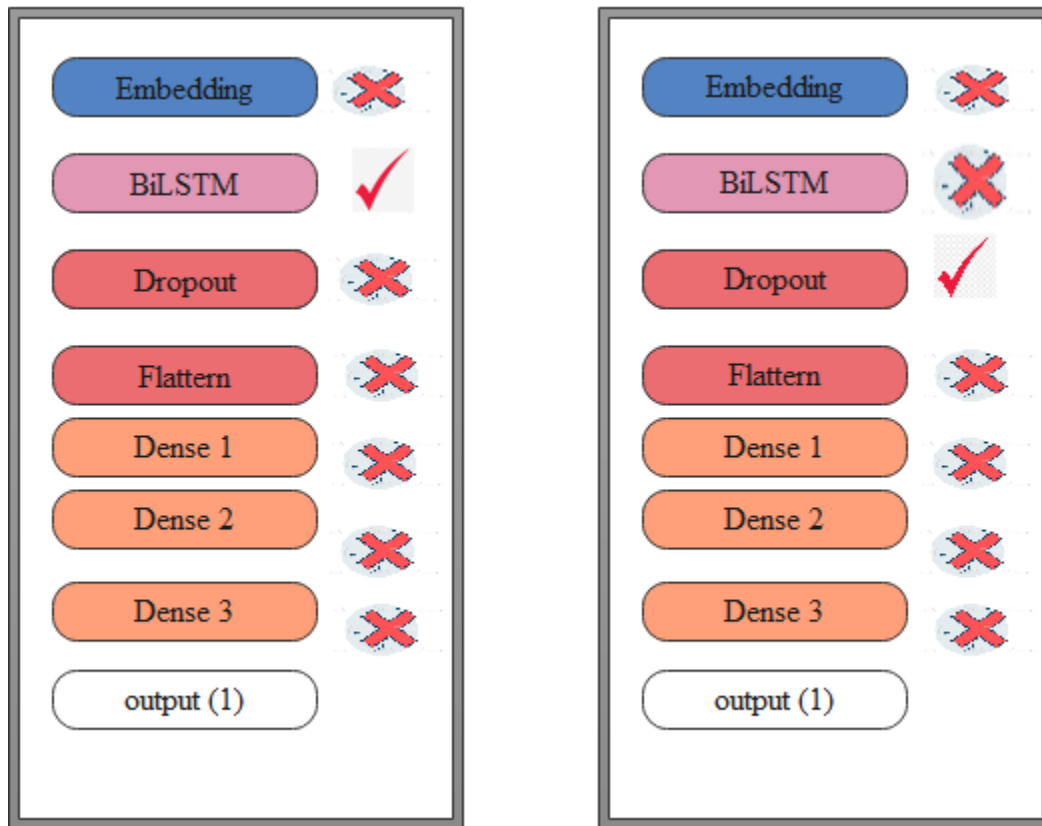


Fig 6.6.2: Freezing one layer at a time pre-train CNN/BiLSTM Reddit dataset

However, it is planning to compare the performance of these transfer learning strategies “last”, “full”, and with newly devise strategy. The computation of chain-thaw was not possible using a single machine therefore evaluation would not present here in experiment. The cloud required extra computational configuration and cost therefore, it was not possible for this research.

6.6.3 Proposed Strategy Faded out

The strategy of freezing the layers was explored in different ways like ‘last’, ‘chain-thaw’, and ‘full’. The real purpose is to enhance the performance gain; therefore, like to invent different strategy than the model freezing layer by layer concept. Thus, this research would like to adapt the strategy by freezing all layers except the dense layers which will remove one by one from the saved pre-train model as shown in Figure 6.6.3a. Thus, proposed new strategy faded-out will keep freeze all the layers of BiLSTM despite all dense layers. Therefore, it will freeze all layers of the model embedding, BiLSTM layers but instead of unfreezing the one layer at a time, the new proposed faded-out strategy will drop each layer of DNN one at a time to evaluate the performance. This approach is computationally inexpensive and will like to produce best results compared to existing strategy ‘last’ and ‘full’. Here, the proposed faded-out strategy produced better results than chain-thaw strategy when compared with the state-of-the-art result as given in (Felbo, et al., 2017). The framework will evaluate the pretrain model over the dialog discussion dataset with the proposed faded-out strategy.

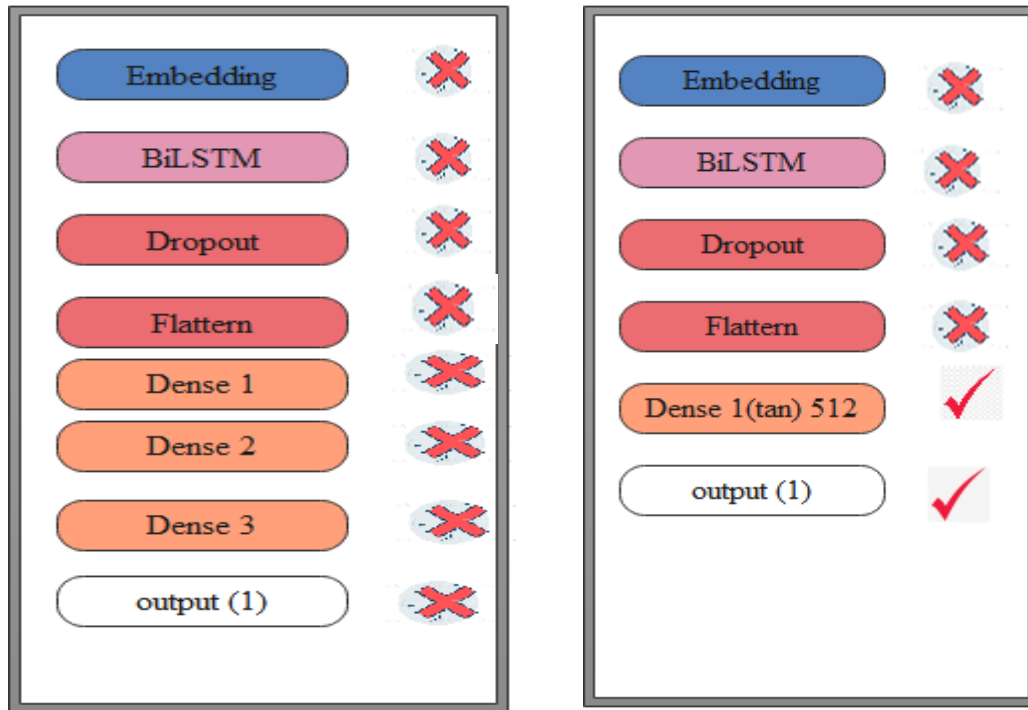


Figure 6.6.3a: Removing dense layers sequentially and evaluating the performance

Similarly, followed a similar strategy of chain-thaw concept here when fine-tune the CNN model, the division of layers classified into the four categories: input or embedding, filter layer, max pool, and dense layers. Here freeze all three categories except dense later, thus, removed the dense layers sequentially and finally concluded that a single dense layer outperformed the performance in comparison to existing strategy. The results indicated that comprehensive improvement over dialog discussion comments data. Even the performance F1 over tweets dataset is far less than the performance F1 over dialog discussion dataset. The CNN has shown significant improvement over dialog discussion data. The pretrain model BiLSTM has shown training performance far better

than CNN in Table 6.6. It is also required to understand that proposed DeepConcat pretrain models of AutoML framework is suitable to various other social media domains like YouTube, Instagram, and Facebook. Therefore, CNN pretrain model illustrated as below in Figure 6.6.3b.

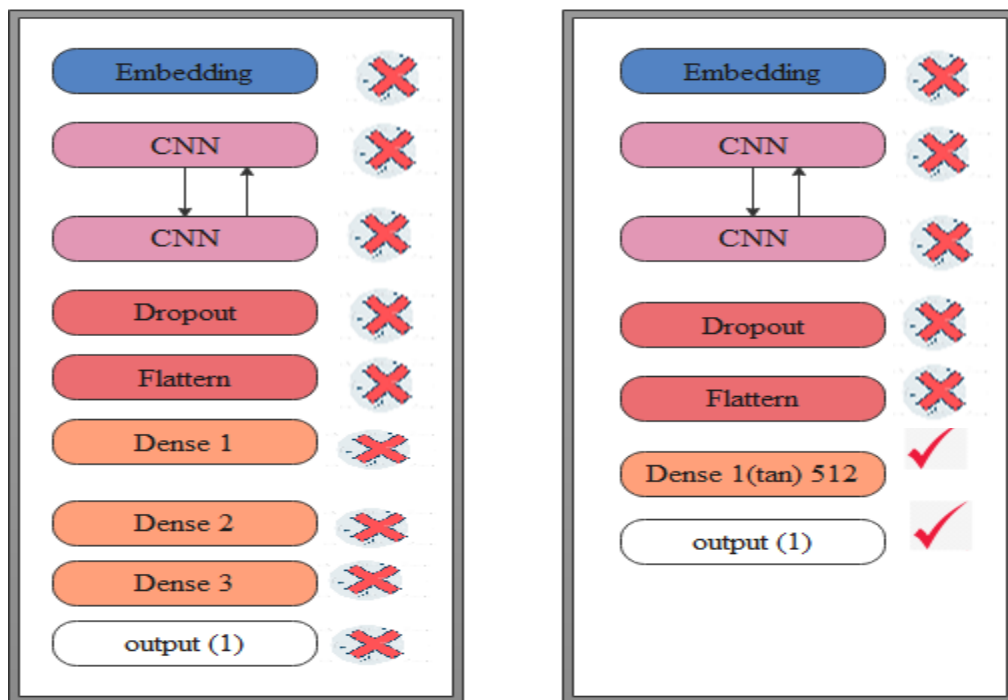


Figure 6.6.3b: Removing dense layer at CNN with 512 units

6.7 Results and Discussion (Faded-out)

Here, result of new faded-out strategy evaluated with pretrain models BiLSTM and CNN over Twitter data, Amazon reviews, and dialog discussion comments datasets. Interestingly, results are promising for the Twitter tweet dataset as it produces significant results over DS3 as compared to DS2 and DS1 datasets. The tweets sample experimented over the same AutoML framework-based

pretrain models like CNN and BiLSTM to produce the result. Below mentioned Table 6.7, is showing the results that proposed strategy ‘faded-out’ has remarkable performance as compared to existing strategies, ‘full’ and ‘last’. The faded-out based fine-tuning of the model outperformed the existing strategies over the Twitter tweets dataset with 0.98 F1, however, state-of-the-art performance was 0.92 F1.

(Filatova, 2012) another dataset is sourced from Amazon reviews, which shows that the domain adaptivity over the dataset has shown low performance over BiLSTM as compared to the CNN. In comparison, the AutoML-Deep framework has shown significant results over dialog discussion comments and Twitter tweets domain dataset as compared to former techniques and models. The dataset of dialog discussion dataset taken from the SCV2-Gen dataset (Oraby, et al., 2017) and evaluated by model BiLSTM proposed by (Felbo, et al., 2017), it was observed that pretrain model BiLSTM have significance score of 0.75 F1. The performance of existing state-of-the-art was 0.75 F1, which is comparatively less than proposed pretrain AutoML-Deep framework model CNN and BiLSTM which is 0.88 F1 and 0.87 F1 as given in the Table 6.7. As one can observe in Figure 6.7 that illustrated the training inflection point which is the point that indicate the training limitation during the training iteration of the dialog discussion dataset SCV2-Gen. It shows accuracy over the 10 epochs thereafter, it reflects no change in accuracy thus restricting the training up to that point because it will also help to avoid exploding or vanishing Gradient problem (it is the problem vanishing or exploding gradient problem, it occurred when gradient too small or too large because of this problem algorithm do not converge).

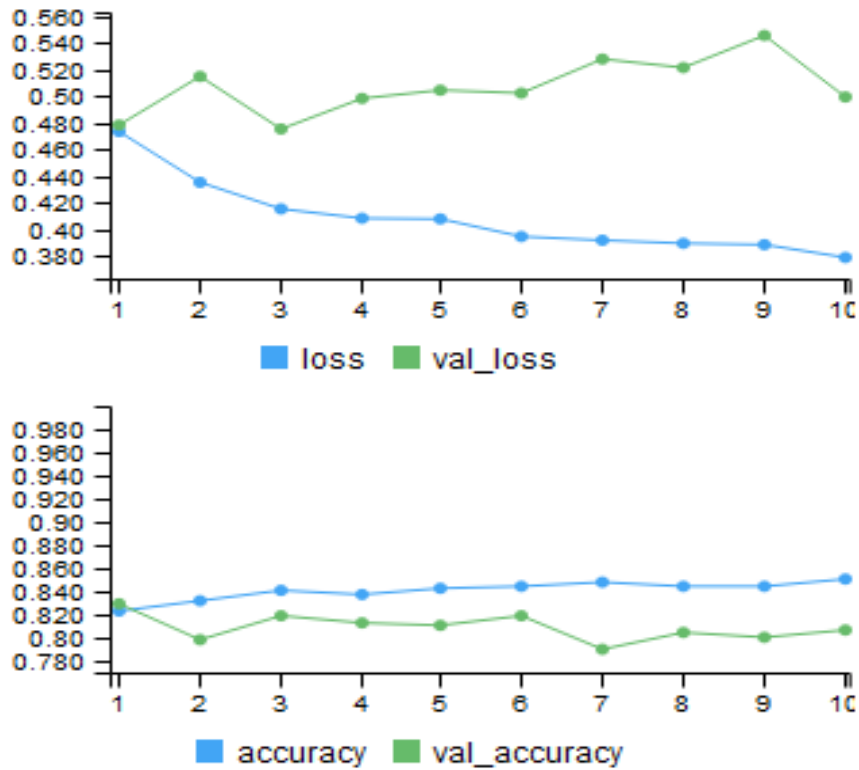


Figure 6.7: Inflection Point

Above shown the training plot for the dialog dataset SCV2-Gen over pretrain model AutoML framework. The percentage indicate the accuracy with the corresponding validation, which is also proportionally increasing, however the accuracy loss is decreasing as compared to validation loss which reflect effective fine-tuning of the pretrain model.

6.0 Transfer Learning – Domains Adaptation

Table 6.7: Imbalance dataset performance on proposed AutoML-DeepConcat pretrain Framework

Data Source	Existing State-of-the-Art Model	Existing State-of-the-Art (F1)	Proposed AutoML Deep Pretrain Models	Sample Data	Evaluation Metrics		
					F1	AUC	Accuracy
Tweets (Ghosh, et al., 2015)	CNN-LSTM-DNN (Ghosh, et al., 2015)	0.92	BiLSTM	1000	0.98	0.978	98%
	CNN-CNN (Ghosh, et al., 2015)	0.87	CNN		0.70	0.568	59%
Amazon (Filatova, 2012)	CNN (Zhang, et al., 2016)	0.91	BiLSTM	100	0.85	0.87	74%
			CNN		0.86	0.88	77%
SCV2-Gen (Oraby, et al., 2017)	BiLSTM- (Felbo, et al., 2017)	0.75	BiLSTM	2000	0.87	0.56	77%
			CNN		0.88	0.66	79%

6.8 Conclusion

In this Chapter, AutoML-DeepConcat framework based pretrain model has shown significant performance over cross-domain datasets such as SCV2-Gen (Oraby, et al., 2017), Twitter tweets (Ghosh, et al., 2015), and Amazon reviews (Filatova, 2012). To validate this concept, this research particularly evaluated the pretrain model over other datasets when fine-tune with strategies like ‘last’, ‘full’, and newly proposed strategy ‘faded out’. The performance of the prtrain model much better when transfer knowledge among cross domains using proposed strategy ‘faded-out’. Notably, it is a contribution to the pretrain models’ strategies, AutoML DeepConcat framework-based pretrain models will randomly select the model by the model search pipeline. The second aspect of the framework is the optimization of pretrain model hyperparameters such as dropout and learning rate.

It was also experimented that evaluated existing pretrain models like BERT and fastText over a small sample of SemEval-2008 Task3 irony. The results concluded that NLP task when evaluated over openly available pretrain models like BERT and fastText, it is not necessary that these modes are suitable for all types of NLP Tasks. It is observed that pre-train model fastText is not efficient in performance as compared to BERT. BERT is mostly useful for figurative tasks like multiple language translation problems.

Therefore, the main problem is the leverage of local resources to train the model over a huge volume of 50,000 news formal datasets. Thus, the objective is to overcome the leverage of extra computation with minimum epochs when pretrain the BiLSTM model, on the contrary, it takes 3-5 hours to train. The time complexity is due to the bidirectional nature and more dense architecture. On the other hand, CNN takes 1-2 hours to pretrain the model due to the model dimensionality which reduced due to convolution and pool max operations. Further, it is a desire to evaluate the formal and informal text in both pretrain AutoML frameworks models. Further, it is highly recommended to overcome the leverage of computational resources when pretrain the model because it required cloud resources.

CHAPTER 7

Discussion, Conclusion, and Recommendation

In Chapter 1 discussed following topics: Introduction of the research, the research questions, and aims are part of this chapter. Chapter 2 -Literature Review: reviewed the research literature on different topics to find a research gap and synthesis of the existing research gap that can contribute to literary society. In Chapter 3 – Research Methods: the study investigates the existing methods to observe the existing models, techniques, feature extraction techniques and algorithms, and all available benchmark datasets. The detail experiment is presented in these Chapters 4, 5, and 6. Chapter 4 is about the experiment of core techniques vs. baseline deep learning models' techniques to find the best baseline model. Chapters 4 and 5, proposed a novel feature extraction algorithm for linguistic features that will integrate with AutoML DeepConcat framework. The novel AutoML DeepConcat framework proposed the integration of features with baseline models. The results were evaluated on three benchmark datasets and outperformed them with proposed new method. Chapter 6, the outcome of this Chapter is pre-train models transfer learning strategy faded-out which will be applied over pretrained saved model DeepConcat with the support of AutoML DeepConcat framework. Entire thesis was about domain adaptation framework that discover best pretrain model with novel strategy faded-out that fine-tune the model to other domains like Amazon reviews, Twitter tweets, and Dialog discussion

Chapter 7– Conclusion and Recommendation: This chapter is about the conclusion of the research, research limitations, and suggestions with future research recommendations.

7.1 Discussion

In the previous Chapters 4,5,6, presented the experiments comprehensively with new findings using three datasets. These chapters presented the results obtained from the evaluation of experimented dataset and the presented best model part of this research outcome. Following Chapter 7, the studies presented the conclusion and recommendation.

The first section of the Chapter solved the research questions and provide the answers obtained throughout all chapters of the thesis. This is followed by the thesis summary, which briefly summarizes the thesis, after which the research limitations are discussed. The study will suggest promising future research and closing remarks.

7.2 Recall of Research Questions

The study recalls the research questions as below:

7.2.1 What are the existing methodologies and techniques to detect sarcasm? What is technique and algorithm which used to extract features from various social media domains?

Existing techniques for sarcasm detection were explored and observed in Chapters 4,5 and 6. These techniques are fall under the broad categories of core techniques or state-of-the-art deep learning techniques, which are part of new AutoML DeepConcat framework with the integration of novel algorithmically extracted features. The core techniques like SVM have been explored by various authors with context-based features (Davidov, et al., 2010; González-Ibáñez, et al., 2011; Riloff, et al., 2013). Interestingly, multiple domains adaptation techniques need to be discovered for sarcasm like these domains are Amazon reviews and Twitter tweets. (Ghosh & Veale, 2016; Joshi, et al., 2015; Felbo, et al., 2017) the work was on the generalizability of features that help to classify the tweet using deep learning. The work focus was on a pretrain model that trains with 1.2 million tweets, the pretrain model transfer knowledge to multiple domains like dialog discussion, YouTube comments, and Twitter tweets for multiple tasks classification like sentiment, emotion, and sarcasm analysis. (Joshi, et al., 2012) Fed-LR and Fed-SVM learn the best classifier like SVM and LR among all datasets and domain results. The methodology of applying the pretrain model to the different domains was initiated by former research. (Yang, 2014) ELMO model proposed for the multiple tasks learning (MTL) and multiple domain learning (MDL) that is based on a matrix where one categorical variable is shared among multiple domains. (Felbo, et al., 2017) DeepMoji proposed pretrain BiLSTM is to classify sarcasm, emotions, and sentiments that work was on multiple domains adaption. (Devlin, et al., 2018) Bert BiLSTM masks the

words to the predictor network. These pertains models are based on methodology which proposed deep learning models have a better adaptation for the multiple social domains.

7.2.2 What is the best methodology to extract the features related to various categories like pragmatic and incongruity features? Which category of features are more suitable for the sarcasm detection?

The core models like SVM, regression, and KNN has experimented on the tweet's dataset (Riloff, et al., 2013; Joshi, et al., 2015). The core model is adapted with features of explicit and implicit incongruities. The novelty is features extraction that extracted by two of the prominent algorithms like Explicit Incongruity Algorithm (EIA) and Implicit Incongruity Algorithm (IIA), which takes polarity contrast among terms and phrases. The features extracted by the IIA algorithm are in the form of contrasting polarities of terms and phrases. The features of EIA are total-positive, total-negative, total-neutral, the overall sentiment of the tweet, and the distance between negative and positive terms. The experiment was conducted category-wise and build three methods for experiments 1) incongruity + baseline 2) pragmatic + baseline 3) all-features + baseline. The baseline features are feed to the model as the input and concatenated at the hidden layer.

The proposed algorithm is further integrated with the existing core model to validate the hypothesis that the core model has similar or better performance than the deep learning model.

The invented algorithm is written mathematically in Chapter 4 Section 4.4.1 implicit incongruity algorithm and explicit incongruity algorithm 4.4.2 that proved the concept of incongruity. Further results were tested with these concatenated features at hidden layer with baseline features. The all-features + baseline including incongruity and pragmatic were best over sarcasm detection. The second-best performance gain was incongruity features + baseline over all models. In comparison incongruity feature methods are better than pragmatic features concatenated with baseline features.

7.2.3 Which Machine Learning technique is the best to evaluate the best results over benchmarking compared to the core approach? How to select the best model as the baseline model?

The experiment initiated with investigation among baseline core models compared with the deep learning models. Mainly, it was observed that the models' performance to detect sarcasm and concluded that the contextual clues in the form features and scaling/normalization techniques influence the performance. To prove the betterment of the model, initially experiment conducted over the dataset DS1, it was evaluated that deep learning CNN model outperformed core baseline models SVM, KNN, and logistic. These baseline core models outperformed the existing techniques due to contextual clues in the form of features. On the contrary, the model's performance on the dataset DS2 achieved a similar conclusion when examining features using different scaling/normalization techniques; these techniques are range, max-min, and lambda scaling techniques.

The baseline SVM with the lambda scaling technique outperformed existing core models as presented in Chapter 4 Table 4.9.2b. it depicts that scaling techniques influence the performance of core techniques. The experiment examined the best baseline model among deep learning and core models, which are BiLSTM and CNN.

7.2.4 How the transfer learning will be implemented using the AutoML automation?

It significantly impacts the AutoML pipeline's performance, such as preprocessing plan, feature engineering, and model search optimization criteria. Therefore, it is important to observe the adapted model performance and the proposed model with different features, preprocessing levels and hyperparameter optimization. The existing models are part of automated machines like AutoML-Keras (Jin, et al., 2018), TPOT (Olson, et al., 2016), AutoML-Sklearn (Feurer, et al., 2019), and AutoML-Weka (Kotthoff, 2017). It cannot incorporate the domain-specific features except AutoML-Keras but it has limitation that it cannot incorporate generalized features. The proposed AutoML framework has pipelines model search, feature engineering and hyperparameter optimization. The AutoML framework automates the model search and parameters optimization, on the other hand, this research focus is on semi-automated framework that applied preprocessing plan and feature extraction with algorithms prior to automation of framework pipelines. Thus, the during the model search pipeline and architecture selection the extracted features will integrate.

Initially, the model will train and then automation pipelines will initiate the step of automation to explore the best pretrain models using Bayesian optimization. During the feature engineering, it extracted features through algorithms thereafter integrated into the models. Formally, AutoML was initiated with a random search mechanism and applied Bayesian optimization with hyperparameters. Finally, the evaluation of each category of features falls into broad categories, these features are integrated at each iteration of the AutoML DeepConcat framework.

7.2.5 What are the strategies to transfer the knowledge from the existing domain to a new domain? What is the new strategy of transfer learning that improve the performance? How the transfer learning will be implemented using the AutoML automation?

There are various strategies that will help to transfer the knowledge from one domain to another domain using deep learning. But the question arises here, these strategies can be adapted regardless of the different areas and domains like image, audio/video visual detection, and NLP. (Yang, et al., 2007) domain knowledge transfer between NDTV channel news videos and CNN channel news video using ensemble-based classifier Adaptive SVM (SVM-A). (Erhan, et al., 2010) proposed the RBN deep learning neural network that freeze last layer to transfer

knowledge of the video domains whereas pre-train the model in one domain and transfer the weights to another domain. The strategy of transfer knowledge is that it freezes only the last layer of the CNN model that transfers knowledge among images of birds. Based on the evidence presented, the strategy of freezing full layers needs to be examined with proposed pretrain model of new AutoML DeepConcat framework.

. (Felbo, et al., 2017) proposed the multiple domains and multiple tasks based BiLSTM models which transfer domain knowledge. The BiLSTM attention-based network pretrain the model over Twitter tweets and adapted to multiple domains like YouTube comments and dialog discussion comments. (Felbo, et al., 2017) the former author handled multiple tasks like emotion analysis, sentiment, and sarcasm using the chain-thaw strategy. The chain-thaw strategy transfers knowledge with one-layer training at a time (as illustrated in Chapter 6, Figure 6.5.2). After comparison of the strategies last, full, and faded-out (newly proposed), it was observed that best strategy can be adapted. After experiment of all these former strategies, it is concluded that proposed strategy, named as faded out, is more effective for fine-tuning the other domains datasets for sarcasm. The difference between the strategy “full”, “last”, and “faded-out” is the that it sequentially removes the layers, on the opposite side, the former strategies freeze all the layers or few layers of the model. This concept of altering the pretrain model layers validate the concept of hyperparameter optimization, which is true direction of transfer knowledge. The data was collected from Forbes News and Twitter tweets in the form of big data of 50k to pretrain the AutoML framework-based models CNN and BiLSTM. The training time is computationally expensive

though it was taking more time, therefore, planning of pretraining the BiLSTM complex layers within few iterations. Still, CNN took less time of training even with more epochs, due to dimensionality reduction concept. The pretraining of models with the AutoML framework over the datasets of Amazon reviews, Twitter tweets, and dialog discussion comments. The proposed strategy “faded out” achieved the best results as compared to the former strategies like “full” and “last”; therefore, it novel to contribute as the transfer learning new strategy.

7.3 Discussion

This research proposed the incongruity features extraction algorithm is based on contrasting phrases and terms. These features extracted then further will be integrate into models before pretrain but during model search pipeline, thus it is the concept of feature engineering with semi-automation framework. After that, the proposed automated AutoML DeepConcat Framework that will select the best baseline models among five deep models CNN, LSTM, BiLSTM, CNN-LSTM, CNN-BiLSTM with hyperparameter optimization. This research also discovered a new strategy “faded-out” to transfer domain knowledge to other domains. The pretrain model train over formal and informal text using tweets and news datasets and observed the best performance. Further, the AutoML DeepConcat framework based pretrain model transfer knowledge using novel strategy “faded-out” for multiple domains.

7.4 Conclusion and Recommendation

The discussion is presented in the previous section with the answers that how the AutoML DeepConcat framework incorporates the features extracted from the novel semi-supervised incongruity extraction algorithms. Then, proposed a new strategy “faded out” that is contribution to the AutoML DeepConcat framework to achieve the goal of knowledge transfer among domains. It was observed that pretrain model selection through the AutoML framework was not proposed in former research. In the below section, presented the summary of the thesis and conclusions.

7.4.1 Thesis Summary

The research aim is to prove that feature engineering or feature extraction AutoML architecture outperforms single and multiple domains without user interventions. This study achieved the objectives in Chapter 1. It was found interesting when systematically exploring the study that reveals the sarcasm detection in Chapter 2. Systematically the gap in existing research defined the need for incongruity feature extraction algorithms and the AutoML framework automation pipelines to find the best pretrain model with the integration of incongruity feature extracted algorithms.

The research gap found in Chapter 2, then further like to explore the existing research methodology in Chapter 3. Furthermore, like to define the adapted methodology from

existing literature, the process steps are given in the form of AutoML framework such as data gathering, preprocessing, feature extraction, and model search among existing core models, and deep learning models. In that stage, this research explored the adapted methodology with evaluation criteria and metrics.

Following Chapter 3, three chapters are about experiments Chapters 4,5, and 6. Initially, in Chapter 4, this Chapter is about evaluated the existing adapted methodology after integrating features extracted from the algorithm into the core models and deep learning models. Therefore, identify the useful baseline method based on existing core models and deep learning models. Further methods were pragmatic and incongruity features these were based on novel algorithmic features extraction.

Further required each model to optimize with minimum human intervention in Chapter 5. In this Chapter proposed the AutoML DeepConcat framework that automates the model search, hyperparameters optimization, and model architecture pipelines. Firstly, the preprocessing plan and extracted features integrate into the AutoML DeepConcat framework models at model search pipeline. Secondly, these features are integrated into these selected models using different architectural layers, these models are LSTM-DNN, CNN-DNN, BiLSTM-DNN, CNN-LSTM-DNN, and CNN-BiLSTM-DNN. Thirdly, the Bayesian hyperparameter is the drop-out that optimized during each step of training iteration. The result is the best model output with optimized hypermeters such as dropout and learning rate. In my opinion, the model search is computationally

expensive, but it controls the model search with five iterations using the random dropout value set at each iteration.

Finally, investigated the adaptation of AutoML into multiple domains like Amazon reviews, dialog, and Twitter domain in Chapter 6. The models CNN and BiLSTM were pretrain which are part of the AutoML DeepConcat framework using formal text of Forbes news and informal text Twitter tweets. The pretrain AutoML DeepConcat framework applied over benchmarking dataset sourced from Amazon reviews, Dialog discussion comments, and Twitter's tweets.

7.5 Conclusion

Initially, proposed the framework that applied Bayesian optimization technique that works with AutoML framework hyperparameter optimization. That AutoML framework has proven better performance than the non-parametric approach, as given in the above section. The results indicated that the best pretrain model selection is based on random drop-out optimization during the AutoML framework search model pipeline, which is better than conventional fixed value-based drop-out values. Further two factors are there which enhanced the performance of the proposed AutoML DeepConcat framework models. The experiment has been conducted in steps.

- Firstly, the pre-processing level plan was prepared which is the main purpose to observe the impact over model. Model trained for each level separately then P1 was selected as the best preparation level.
- The features are extracted with incongruity algorithms which categorized as explicit and implicit. The algorithmic based features extracted using patterns 2-gram, 3-gram and n-gram patterns. The useful patterns were 2-gram which are extracted with the concept of incongruity that is positive and negative polarity contrast. The algorithm of each category is written to extract the contrasting polarities of terms and phrases.
- To decide which model category is suitable for extracted features, thus experimented in Chapter 4 to ensure among core techniques and deep learning models. It proved that deep learning models are better in performance and chosen adapted models. The model performance evaluated over Twitter tweets and dialog discussion datasets; the results indicated the F1 score of pretrain models BiLSTM and CNN of AutoML DeepConcat outperformed existing state-of-the-art models.
- Further, adapted pretrain models BiLSTM and CNN compared features integration methods like extracted features + baseline features, baseline + incongruity features, and baseline + pragmatic features. All features + baseline features were best among all methods with deep learning model BiLSTM.
- The transfer learning strategies were planned to be part of experiment to devise new proposed AutoML DeepConcat model, the experiment outcome is new strategy named as faded-out.

Then transfer learning performed on pretrain model, which outperformed all existing techniques on same datasets.

Further, elaborating the experiment in detail with performance metrics, methods, and models. However, pretrain model performance over long text domains like Amazon reviews was less than state-of-the-art. The performance of AUC and F1 metrics are equal as defined in Table 6.7 for BiLSTM model however, F1 is slightly better than AUC but these metrics cannot be equal in performance, it is due to less epochs for training might set AUC equal or better than F1. It was also explored the fact that optimization together with preprocessing plan levels produced effective results. But not all the levels have an equal contribution to model performance thus minimum preprocessing is required to get better results.

Nevertheless, the different dataset has a diverse preprocessing level effect that would optimize the performance of the models of AutoML framework. Another fact is that all categories of features with baseline and all-features method is better than pragmatic and incongruity features methods. It was observed that the pragmatic features with the baseline method is not more significant than all-features method, however, the incongruity features method has more significance than pragmatic features. Lastly, the incongruity features extracted from newly developed incongruity algorithms IIE and EIA as defined in Chapter 4, that is a contribution to existing literature. Alternatively, these features are generalized and can be adapted to any NLP domain.

Another contribution is the automation of the adapted DeepConcat framework, which is quite comparative and performance persuasive. It was observed that feature engineering with conventional AutoML-Keras cannot be fully adapted to other domains, thus it is hard to transfer knowledge due to domains specific data. The new framework AutoML DeepConcat contributes significantly due to the adaptation of generalized feature like incongruity and pragmatic, which supports any domain. The framework incorporated various models and trigger the searching of the model using model search pipeline. The results indicated that the BiLSTM model is better than all other models which are part of the AutoML DeepConcat framework. All datasets were evaluated, which shown remarkable results with proposed deep learning-based framework compared to core techniques.

The scaling techniques have shown significant efficacy in the case of core models over the dataset DS1. The result indicated that the lambda technique is better than the others. The search model pipeline required more computation to run for multiple iterations to select the best model and parameters. The proposed ‘faded-out’ strategy is the contribution to the domain adaptation. Therefore, proposed the new strategy ‘faded-out’ is part of the AutoML DeepConcat pretrain models. Another novelty is the pretrain model selection from the AutoML framework model search pipeline.

Finally, presented following findings based on the results of each chapter.

- It was evident from existing literature that incongruity was harvested through pattern or contextual-based markers for detection. Firstly, these features are extracted based on pattern-based and context-based techniques. Secondly, these features are extracted from the incongruity algorithm. The core model and deep learning comparison get the outcome in the form of best model for the detection of sarcasm. It was found scaling techniques have a good influence over features when input to the core models like SVM to enhance the performance. It was also found that even the core model cannot perform better than the deep learning model when features are integrated. However, deep learning models are better in performance.
- The generalized features are extracted in former research in the form of contextual features like incongruity features and pragmatic features which are the main clues for sarcasm. Proposed AutoML DeepConcat framework models integrate the features during the automation process to find the best model with optimized parameter drop-out for multiple domains. There are existing AutoML frameworks like AutoML-Keras (Jin, et al., 2018), TPOT (Olson, et al., 2016), and AutoML-Sklearn (Feurer, et al., 2019) which automate the pipelines feature engineering, model search, Bayesian optimization, and model architecture. However, existing frameworks are hard to adapt to other domains due to a lack of features generality. But proposed feature extraction algorithm can be adapted to any domain with

the deep learning models. It was also reviewed and evaluated that proposed pretrain AutoML DeepConcat framework models which were selected after training the multiple deep learning models. The AutoML DeepConcat framework will iteratively integrate the features at the dense layer of the deep learning models.

- One of the substantial finding is the exploration of the pretrain models: BiLSTM and CNN. The AutoML pretrain models are more effective when pretrain over the formal text like news as compared to informal text like tweets. The performance evaluation of the BiLSTM model over the dialog discussion domain dataset outperformed the existing pretrain BiLSTM model (Felbo, et al., 2017). Similarly, the evaluation of the pretrain model BiLSTM over the Tweet's dataset also performed well among other models. AutoML DeepConcat Framework pretrain models performed reasonably well over Amazon dataset but not outperformed existing state-of-the-art model performance.
- In this research, planned to prepare research paper, the paper is main contribution: "A new AutoML Framework: A DeepConcat framework for Context-aware Sarcasm Detection". It was recently submitted in the applied intelligence journal of springer. In this paper, like to explore a new AutoML DeepConcat framework in detail with the integration of features during automation of machine to select the best model.

- The pretrain models of the proposed AutoML DeepConcat framework BiLSTM model outperformed the existing state-of-the-art models CNN-LSTM and CNN-CNN with 0.98 F1. It was observed that proposed BiLSTM model is better than the existing pretrain model BiLSTM proposed by (Felbo, et al., 2017) when it was evaluated over dialog discussion dataset. However, multiple domains learning (MDL) and multiple tasks learning (MTL) based model is not effective in performance for many tasks of classification of multiple domains, however, single task-based MDL is better comparably. Similarly, the performance can be observed that it is better with 0.88 CNN and 0.87 BiLSTM over Amazon reviews and 0.85 F1 on dialog discussion dataset SCV2-Gen. Mainly, it was concluded that proposed pretrain models based on AutoML with hyperparameters optimization, preprocessing plan level and generalized algorithmic features like incongruity features produced better model for multiple domain sarcasm classification.

Below presented few characteristics that are associated with proposed framework DeepConcat.

- a) **Adaptivity:** The proposed framework model fine-tune over other domains, it is possible with new strategy faded-out.
- b) **Diversity:** The features of multiple domains can be extracted with generalized criteria so preserve the commonality for multiple domains.

- c) Efficiency: The AutoML DeepConcat framework is an initiative towards a best model that searches with good parameters which would be efficient model in terms of performance.
- d) Enhancement: The proposed AutoML DeepConcat framework can train many more models to select the best pretrain model.
- e) Autonomy: The main characteristic of the proposed framework is that it automates the model search pipeline so required less intervention with features integration, and hyperparameters optimization to get the best model.

7.5.1 Research Limitation

Further, defined the following challenges and issues in this section.

- The existing pretrain model BERT is a product of Google API however, it is quite expensive to implement in a single machine due to computation complexity. Thus, compared the performance of proposed framework models, which is comparatively better than BERT and fastText, however, computationally it is harder to train these commercial models over a single machine. The results of conventional pretrain models were not convincing as expected. Therefore, like to adapt proposed framework models with the feature's integration concept.
- The data dictionary and sentiment tool-based dictionaries FINN, BING, and NRC would not cover the terms polarity due to limited-term vocabulary. Therefore, it was explored Third-party Google API NLP services and libraries, but it lacks the coverage of polarities. Ultimately,

found the third-party API data dump which covered all terms and overcome this challenge but with a limitation of 1000 polarity words extracted per day.

- Another biggest challenge is to train the baseline models LSTM, LSTM-CNN, and BiLSTM over large datasets. Fine-tuning of the model was performed using AI-based transfer learning strategies to transfer knowledge among domains. One of biggest challenge is to train models with over 100,000 instances of dataset that are hard to train on single CPU, it would take many days to finish the training. Therefore, it is the limitation to pretraining the model on a single machine for a huge volume of data so, it demands the cloud-based service to overcome this challenge.

7.5.2 Future Direction

In the future, these pragmatic and incongruity features with other features like emojis would be integrated into the models for the evaluation so, further, enhance the performance. Finally, the deep learning model must be optimized using optimization parameters such as learning rate and batch normalization.

AutoML framework models adapted to the few domains like Twitter and Amazon, but it was not observed whether it has practicability to other domains like YouTube comments. Further, the hypothesis can be extended to observe the hyperparameter optimization and model search for many more social media domains; therefore, the model would adapt to various domains using generalized features and transfer learning strategies. The limitation of pretraining the model over

the huge volume can be overcome in the future with the improvisation of many dual cores in the cloud. To overcome the problem of the AutoML framework model over the single core, it is best to pretrain the model over the cloud platform with multiple cores. However, the data is still big, but more volume and velocity would change the scope of the research from local environment to cloud. There are possibilities to pretrain the models over cloud environments such as Azure and Google, but it required time in future to implement.

There are various issues still not covered related to this research, like aspect-based rating methodology from Tourism domain can be enhanced further to devise new aspect-based rating for many domains. This topic would be observed in next upcoming future research papers, however due to scope of the research it would not discuss here. But not all domains have aspect like Trip advisor reviews, but these aspects can be extracted from other domains and it would be novel to get common features or aspects from many domains. However, sarcasm detection with common aspect using AutoML DeepConcat would be tested.

In closing remarks, likes to discuss that this research contributed the novel incongruity features extraction algorithms, those integrated in models with automation of model search pipeline and skip layer mechanism, AutoML framework performed the hyperparameter optimization, search model pipelines, and devised a new strategy ‘faded-out’.

Bibliography

- Aggarwal, C., 2018. Neural networks and deep learning. Volume 10, pp. pp.978-3.
- Altrabsheh, N., Cocca, M. & Fallahkhair, S., 2014. *Learning sentiment from students' feedback for real-time interventions in classrooms*. cham, In International Conference on Adaptive and Intelligent Systems, pp. 40-49.
- Aman, S. & Szpakowicz, S., 2007. *Identifying expressions of emotion in text*. s.l., In International Conference on Text, Speech and Dialogue Springer, Berlin, Heidelberg., pp. 196-205.
- Amir, S., Wallace, B., Lyu, H. & Silva, P., 2016. Modeling context with user embeddings for sarcasm detection in social media.. *arXiv*.
- Anton, M., 2020. *Automated Machine Learning using Evolutionary Algorithms*.. s.l., s.n., pp. 101-107.
- Bamman, D. a. S. N., 2015. *Contextualized sarcasm detection on twitter*.. s.l., In Ninth international AAAI conference on web and social media..
- Bamman, D. a. S. N., 2015. *Contextualized sarcasm detection on Twitter*.. s.l., In Ninth international AAAI conference on web and social media..

-
- Barbieri, F., Saggion, H. & Ronzano, F., 2014. *Modeling sarcasm in Twitter, a novel approach..* s.l., In Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 50-58.
- Baziotis, C. et al., 2018. NTUA-SLP at SemEval-2018 Task 3: Tracking ironic tweets using ensembles of the word and character level Attentive RNNs.. *arXiv*.
- Ben Eisner., et al., 2016. *emoji2vec: Learning emoji representations from their description..* s.l., In 4th International Workshop on Natural Language Processing for Social Media.
- Blitzer, J., Dredze, M. & Pereira, F., 2007. *Biographies, Bollywood, boom-boxes :Domain adaptation for sentiment classification.* s.l., s.n., pp. 440-447.
- Blitzer, J., McDonald, R. & Pereira, F., 2006. *Domain adaptation with structural correspondence learning.* s.l., In Proceedings of the 2006 conference on empirical, pp. 120-128.
- Bouazizi, M. & Ohtsuki, T., 2016. A pattern-based approach for sarcasm detection on twitter.. *IEEE Access*.
- Brown, K., 2017. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA.. *The Journal of Machine Learning Research*, 18(1),, pp. 826-830.
- Buschmeier, K., Cimiano, P. & Klinger, R., 2014. *An impact analysis of features in a classification approach to irony detection in product reviews..* s.l., In Proceedings of the 5th Workshop

- on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis , pp. 42-49.
- Camp, E., 2012. Sarcasm, pretense, and the semantics/pragmatics distinction.. *Noûs*, 46(4), pp. 587-634..
- Carvalho, P., Sarmento, L., Silva, M. & de Oliveira, E., 2009. *Clues for detecting irony in user-generated contents: oh...!! it's" so easy"*. s.l., In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, pp. 53-56.
- Cheng, Z., Caverlee, J. & Lee, K., 2010. *You are where you tweet: a content-based approach to geo-locating Twitter users*. s.l., In Proceedings of the 19th ACM international conference on Information and knowledge management, pp. 759-768.
- Chen, Z., Zhang, H., Zhang, X. & Zhao, L., 2018. *Quora question pairs*., s.l.: URL <https://www.kaggle.com/c/quora-question-pairs>..
- Chollet, F., 2015. *keras*. s.l., s.n.
- Chung, C. & Pennebaker, J., 2007. The psychological functions of function words.. *Social communication*.
- Ciregan, D., Meier, U. & Schmidhuber, J., 2012. *Multi-column deep neural networks for image classification*. s.l., In 2012 IEEE conference on computer vision and pattern recognition, pp. 3642-3649.

- Cohan A, et al., 2018. A Large-Scale Resource for Exploring Online Language Usage for Multiple Mental Health Conditions. *arXiv*.
- Colston, H. & Gibbs, R., 2007. A brief history of irony.. *Irony in Language and Thought*. Taylor and Francis Group, pp. 3-24..
- Davidov, D., Tsur, O. & Rappoport, A., 2010. *Semi-supervised recognition of sarcastic sentences in Twitter and Amazon*. s.l., In Proceedings of the fourteenth conference on computational natural language learning . Association for Computational Linguistics, pp. 107-116.
- De Rainville, F. et al., 2012. *A python framework for evolutionary algorithms*.. s.l., In Proceedings of the 14th annual conference companion on Genetic and evolutionary computation, pp. 85-92.
- Devlin, J., Chang, M., Lee, K. & Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*.
- Disha S, 2017. Transfer Learning | Pretrained Models in Deep Learning.. *analyticsvidhya*.
- Donahue, J. et al., 2014. *A deep convolutional activation feature for generic visual recognition*.. s.l., In International conference on machine learning, pp. pp. 647-655.
- Donahue, J. et al., 2014. *A deep convolutional activation feature for generic visual recognition*.. s.l., n International conference on machine learning (pp. 647-655). PMLR., pp. 647-655.

- Dumitru Erhan, et al., 2010. Why does unsupervised pre-training help deep learning?. *Journal of Machine Learning Research (JMLR)*, Volume 11, p. 625–660..
- Erhan, D. C. A. B. Y. a. V. P., 2010. *Why does unsupervised pre-training help deep learning?*. s.l., In Proceedings of the thirteenth international conference on artificial intelligence and statistics , pp. 201-208.
- Erhan, D., Courville, A., Bengio, Y. & Vincent, P., 2010. *Why does unsupervised pre-training help deep learning?*. s.l., n Proceedings of tThe Thirteenth International Conference on Artificial Intelligence and Statistics., pp. pp. 201-208.
- Felbo, B. et al., 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm.. *arXiv*.
- Feurer, M. et al., 2015. *Efficient and robust automated machine learning*.. s.l., In Advances in neural information processing systems (pp. 2962-2970)., pp. 2962-2970.
- Feurer, M. et al., 2019. Auto-Sklearn: efficient and robust automated machine learning.. *In Automated Machine Learning Springer, Cham*.
- Filatova, E., 2012. Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing.. *In Lrec*, pp. pp. 392-398.
- FISHER, R. A., 1936. The use of multiple measurements in taxonomic problems. *Contributions to Mathematical Statistics, John Wiley*, pp. 179-188.

- Ghosh, A. & Veale, T., 2016. *Fracking sarcasm using neural network..* s.l., In Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis, pp. 161-169.
- Ghosh, A. & Veale, T., 2017. *Magnets for sarcasm: Making sarcasm detection timely, contextual, and very personal..* s.l., In Proceedings of 2017 Conference on Empirical Methods in Natural Language Processing, pp. 482-491.
- Ghosh, D., Guo, W. & Muresan, S., 2015. *Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words.* s.l., Conference on Empirical Methods in Natural Language Processing, pp. 1003-1012.
- Gibbs, R. & Colston, H., 2007. The future of irony studies. In R. *Irony in language and thought: A cognitive science reader*, p. 581–595.
- Gimpel, K. et al., 2010. art-of-speech tagging for twitter: Annotation, features, and experiments. *Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science..*
- Giovanelli, J., Bilalli, B. & Abelló Gamazo, A., 2021. *Effective data pre-processing for AutoML.* Nicosia, Cyprus, In Proceedings of the 23rd International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data, pp. 1-10.

- González-Ibáñez, R., Muresan, S. & Wacholder, N., 2011. *Identifying sarcasm in Twitter: a closer look*. s.l., In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers, pp. 581-586.
- Hazarika, D. et al., 2018. Contextual sarcasm detection in online discussion forums. *arXiv*.
- Hochreiter, S. & Schmidhuber, J., 1997. Long short-term memory. Neural computation. *Neural computation*.
- Howard, D. et al., 2020. Model Evaluation Study. *Journal of Medical Internet Research*, 22(5).
- Hussain., 2015b. Sentiment data flow analysis by means of dynamic linguistic patterns. *IEEE Computational Intelligence Magazine*., 10(4), p. 26–36.
- Hutter, F., Hoos, H. & Leyton-Brown, K., 2011. *Sequential model-based optimization for general algorithm configuration*. Berlin, Heidelberg., n International conference on learning and intelligent optimization.
- Islam, M. et al., 2018. Depression detection from social network data using machine learning techniques.. *Health information science and systems*, 6(1), pp. 1-12..
- Ivanko, S. & Pexman, P., 2003. *ontext incongruity and irony processing*. s.l., s.n., pp. 241-279.
- Jan N., Bernd B & Torgo, L., 2013. OpenML: Networked Science in Machine Learning. *IGKDD Explorations*, 15(2), pp. 49-60.

- Jiang, J. & Zhai, C., 2007. *Instance weighting for domain adaptation in NLP*. s.l., In Proceedings of the 45th annual meeting of the association of computational linguistics, pp. 264-271..
- Jin, H., Song, Q. & Hu, X., 2018. Auto-Keras: Efficient neural architecture search with network morphism.. *arXiv*.
- Jin, H., Song, Q. & Hu, X., 2019. *Auto-Keras: An efficient neural architecture search system*. s.l., In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. International Conference on Knowledge Discovery & Data Mining (pp. 1946-1956)..
- Jin, H., Song, Q. & Hu, X., 2019. *Auto-keras: An efficient neural architecture search system*. s.l., In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1946-1956).., pp. pp. 1946-1956.
- Joshi, A., Sharma, V. & Bhattacharyya, P., 2015. *Harnessing context incongruity for sarcasm detection*. s.l., In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. (pp. 757-762)..
- Joshi, M., Dredze, M., Cohen, W. & Rose, C., 2012. *Multi-domain learning: when do domains matter?*. s.l., In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning , pp. 1302-1312.

- Jousha, 2017. Natural language processing. *joshuakim*.
- Karoui, J., Zitoune, F. & Moriceau, V., 2017. Soukhria: Towards an irony detection system for arabic in social media.. *Procedia Computer Science*, pp. 161-168..
- Khodak, M., Saunshi, N. & Vodrahalli, K., 2017. A large self-annotated corpus for sarcasm.. *arXiv*.
- Kolchinski, Y. & Potts, C., 2018. Representing social media users for sarcasm detection.. *arXiv*.
- Kotthoff, L. et al., 2017. *The Journal of Machine Learning Research*, 18(1), pp.826-830..
- Kouloumpis, E., Wilson, T. & Moore, J., 2011. *Twitter sentiment analysis: The good the bad and the omg!..* s.l., In Fifth International AAAI conference on weblogs and social media..
- Kreuz, R. & Caucci, G., 2007. *Lexical influences on the perception of sarcasm.* s.l., In Proceedings of the Workshop on computational approaches to Figurative Language. Association for Computational Linguistics., pp. 1-4.
- Kreuz, R. J. & Glucksberg, S., 1989. How to be Sarcastic: The Echoic Reminder theory of Verbal Irony. *ournal of Experimental Psychology: General*, p. 118:374–386.
- Krizhevsky, A. & Hinton, G., 2009. *Learning multiple layers of features from tiny images.* *Master's thesis*, Toronto: s.n.
- Kumar, A. et al., 2020. Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM. *IEEE Access*, Volume 8, pp. pp.6388-6397..

- Kumar, A. et al., 2019. Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network.. *IEEE Access*, Volume 7, pp. 23319-23328..
- Lampert, C. H., Nickisch, H. & Harmeling, S., 2009. S. Learning to detect unseen object classes by between-class attribute transfer.. *CVPR*.
- LeCun, Y., 1998. *The MNIST database of handwritten digits*. [Online]
Available at: <http://yann.lecun.com/exdb/mnist/>.
- LeCun, Y. et al., 1995. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, p. 261(276).
- Liebrecht, C., Kunneman, F. & van Den Bosch, A., 2013. *The perfect solution for detecting sarcasm in tweets# not..* s.l., Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 29-37.
- Liu, P. et al., 2014. Sarcasm detection in social media based on imbalanced classification.. *In International Conference on Web-Age Information Management Springer, Cham..*
- Liu, P. et al., 2018. Generating wikipedia by summarizing long sequences.. *arXiv*.
- Li, Y., Wang, H., Wei, T. & Tu, W., 2019. *Towards automated semi-supervised learning..* s.l., In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 4237-4244.

- lson, R. et al., 2016. *Automating biomedical data science through the tree-based pipeline optimization..* s.l., In European Conference on the Applications of Evolutionary Computation, pp. 123-137.
- Lunando, E. & Purwarianti, A., 2013. *Indonesian social media sentiment analysis with sarcasm detection..* s.l., In 2013 International Conference on Advanced Computer Science and Information Systems (ICACISIS) (pp. 195-198). IEEE..
- Lunando, E. & Purwarianti, A., 2013. *Indonesian social media sentiment analysis with sarcasm detection..* s.l., In 2013 International Conference on Advanced Computer Science and Information Systems.
- Maynard, D. & Greenwood, M., 2014. *ho cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis..* s.l., In LREC 2014 Proceedings. ELRA..
- Milne, D., McCabe, K. & Calvo, R., 2019. Improving moderator responsiveness in online peer support through automated triage.. *Journal of Medical Internet*, 21(4).
- Nielsen, M. A., 2015. Neural networks and deep learning,. *Determination press the USA*, Volume 25.
- Olson, R. et al., 2016. *Automating biomedical data science through the tree-based pipeline.* s.l., In European Conference on the Applications of Evolutionary, pp. 123-137.

-
- Oprea, S. & Magdy, W., 2019. xploring author context for detecting intended vs perceived sarcasm.. *arXiv*.
- Oraby, S. et al., 2017. Creating and characterizing a diverse corpus of sarcasm in dialogue. *arXiv*.
- Owoputi, O. et al., 2013. *Improved part-of-speech tagging for online conversational text with word clusters..* s.l., In Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 380-390.
- Pang, B. & Lee, L., 2008. Opinion mining and sentiment analysis.. *Foundations and Trends® in Information Retrieval*, 2(1-2), pp. 1-135..
- Pang, B., Lee, L. & Vaithyanathan, S., 2002. Thumbs up? Sentiment classification using machine learning techniques.. *arXiv*.
- Pennington, J., Socher, R. & Manning, C., 2014. *Glove: Global vectors for word representation..* s.l., n Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).
- Peters, M. et al., 2018. Deep contextualized word representations. *arXiv*.
- Peters, M. et al., 2019. Knowledge enhanced contextual word representations.. *arXiv*.
- Poria, S., Cambria, E., Hazarika, D. & Vij, P., 2016. A deeper look into sarcastic tweets using deep convolutional neural networks.. *arXiv*.
- Prasad, S., 2010. *Micro-blogging sentiment analysis using bayesian classification*, California: s.n.

- Ptáček, T., Habernal, I. & Hong, J., 2014. *Sarcasm detection on czech and english twitter..* s.l., Proceedings of COLING 2014, the 25th international conference on computational linguistics:, pp. 213-223.
- Rajadesingan, A., Zafarani, R. & Liu, H., 2015. *Sarcasm detection on Twitter.* s.l., In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 97-106.
- Rajadesingan, A. Z. R. a. L. H., 2015. *Sarcasm detection on Twitter: A behavioral modeling approach..* s.l., In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (pp. 97-106)., pp. 97-106.
- Rajpurkar, P., Zhang, J., Lopyrev, K. & Liang, P., 2016. Squad: 100,000+ questions for machine comprehension of text.. *arXiv*.
- Ramteke, A., Malu, A., Bhattacharyya, P. & Nath, J., 2015. *Detecting turnarounds in sentiment analysis: Thwarting.* s.l., In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 860-865.
- Rangwani, H., Kulshreshtha, D. & Singh, A., 2018. *NLPRL-IITBHU at SemEval-2018, Task 3: Combining linguistic features and emoji pre-trained CNN for irony detection in tweets..* s.l., In Proceedings of the 12th International Workshop on Semantic Evaluation, pp. 638-642.

- Real, E., Liang, C., So, D. & Le, Q., 2020. Automl-zero: Evolving machine Learning algorithms from scratch. *arXiv*.
- Real, E., Liang, C., So, D. & Le, Q., 2020. *AutoML-zero: evolving machine Learning algorithms from scratch*. s.l., n International Conference on Machine learning, pp. 8007-8019.
- Ren, Y. J. D. & Ren, H., 2018. Context-augmented convolutional neural networks for twitter sarcasm detection.. *Neurocomputing*,, Volume 308,, pp. 1-7.
- Reyes, A., Rosso, P. & Buscaldi, D., 2012. From humor recognition to irony detection:. *he figurative language of social media. Data & Knowledge Engineering*, 74, pp. 1-12..
- Riloff, E. et al., 2013. *Sarcasm is the contrast between a positive sentiment and a negative situation*.. s.l., n Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 704-714.
- Rodriguez-Serrano, J., Perronnin, F. & Meylan, F., 2013. Label embedding for text recognition.. *In BMVC*, pp. 5-1.
- Rosenthal, S., Farra, N. & Nakov, P., 2019. Sentiment analysis in Twitter. *arXiv*.
- Shelley, C., 2001. The bicoherence theory of situational irony. *Cognitive Science*, 25(5), pp. 775-818.
- Soujanya Poria, et al., 1997. Long short-term memory.. *Neural computation*, 9(8), pp. 1735-1780.
- Sperber, D. a. W. D., 1981. Irony and the use-mention distinction. *Philosophy. Philosophy*.

- Sperber, D. & Wilson, D., 1981. Irony and the Use mention Distinction.. *In P. Cole (ed), Radical Pragmatics. New York: Academic Press*, p. 295–318.
- Sperber, D. & Wilson, D., 1986. Communication and Cognition.. *Cambridge, MA: Harvard University Press*..
- Stock, O. & Strapparava, C., 2005. *A computational humor system*.. s.l., n: Proceedings of the 43rd Annual Meeting on Association for Computational, p. 113–116..
- Straka, M., Hajic, J. & Straková, J., 2016. *UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing*.. s.l., In Proceedings of the tenth international conference on language resources and evaluation (LREC 2016)„ pp. 4290-4297.
- Straka, M. & Straková, J., 2017. *Tokenizing, pos tagging, lemmatizing and parsing 2.0 with udpipe*.. s.l., In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 88-99.
- Toyao, T. et al., 2019. Machine learning for catalysis informatics: Recent applications and prospects. *Acs Catalysis*.
- Tsur, O., Davidov, D. & Rappoport, A., 2010. *ICWSM—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews*.. s.l., In fourth international AAAI conference on weblogs and social media..

- Van Hee, C., Lefever, E. & Hoste, V., 2018. *Semeval-2018 task 3: Irony detection in English tweets*. s.l., International Workshop on Semantic Evaluation, pp. 39-50.
- Vargas-Govea, B., González-Serna, G. & Ponce-Medellín, R., 2011. Effects of relevant contextual features in the performance of a restaurant recommender system.. *ACM RecSys*, 11(592), p. 56.
- Walker, M. et al., 2012. *A corpus for research on deliberation and debate*. s.l., n Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12) (pp. 812-817)., pp. 812-817.
- Wallace, B. C., 2015. Computational irony: A survey and new perspectives,. *Artificial Intelligence Review*, 43(4), p. 467–483.
- Wang, P., Hu, J., Zeng, H. & Chen, Z., 2009. Using Wikipedia knowledge to improve text classification.. *Knowledge and Information Systems*, 19(3), pp. 265-281.
- Warriner, A., Kuperman, V. & Brysbaert, M., 2013. Norms of valence, arousal, and dominance for 13,915 English lemmas. Behavior research methods. *Behavior research methods*, 45(4), pp. 1191-1207.
- Williams, A., Nangia, N. & Bowman, S., 2017. A broad-coverage challenge corpus for sentence understanding through inference.. *arXiv*.
- Wilson, D., 2006. The pragmatics of verbal irony: Echo or pretense?. *Lingua*, 116(10), p. 1722.

- Wilson, D. & Sperber, D., 1992. On verbal irony. *Lingua*, 87(1), p. 53–76.
- Wilson, T., Wiebe, J. & Hoffmann, P., 2005. *Recognizing contextual polarity in phrase-level sentiment analysis*. s.l., In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing..
- Wu, C. et al., 2018. *Semeval-2018 task 3: Tweet irony detection with densely connected lstm and multi-task*. s.l., In Proceedings of The 12th International Workshop on Semantic Evaluation, pp. 51-56.
- Yang, J., Yan, R. & Hauptmann, A., 2007. *Cross-domain video concept Detection using adaptive SVM*. s.l., Proceedings of the 15th ACM international conference on Multimedia, pp. 188-197.
- Yang, Y. a. H. T., 2014. *arXiv*.
- Yang, Y. & Hospedales, T., 2014. *A unified perspective on multi-domain and multi-task learning*. s.l., arXiv preprint arXiv, p. 1412.7489.
- Yi, J., Nasukawa, T., Bunescu, R. & Niblack, W., 2003. *Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques*. s.l., In Third IEEE international conference on data mining, IEEE, pp. 427-434.
- Zhang Y & B., W., 2015. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *arXiv*.

- Zhang, M., Zhang, Y. & Fu, G., 2016. *Tweet sarcasm detection using deep neural network..* s.l., In Proceedings of COLING 2016, the 26th International Conference Computational Linguistics:, pp. 2449-2460.
- Zhao, S., Zhong, L., Wickramasuriya, J. & Vasudevan, V., 2011. *Analyzing twitter for social tv: Sentiment extraction for sports..* s.l., In Proceedings of the 2nd International Workshop on Future of Television, pp. 11-18.

Appendix

The convolution neural network (CNN) is filter-based. It works layer wisely. The first layer augments the input, but in some recognized representation, that is vector-based representation. These continuous vectors are based on embedding dictionaries like 1) google news embedding vectors 2) glove vectors 3) fastText wiki-news-300d embeddings. The vector representation yield of similarity vectors like some of the terms like cat and dog vectors is in closer proximity to each other. The second layer comprises vectors that aim to reduce the data dimensionality means to produce a feature map after applying a filter operation to the text's data sequence. The resultant feature maps are further divisible to multiple regions.

Illustrated, below the feature map, after applying an input to filter the features after applying convolution, which is dot product and sum over the entire sequence and produce feature map. The comprehensive process is explained with more details following illustration.

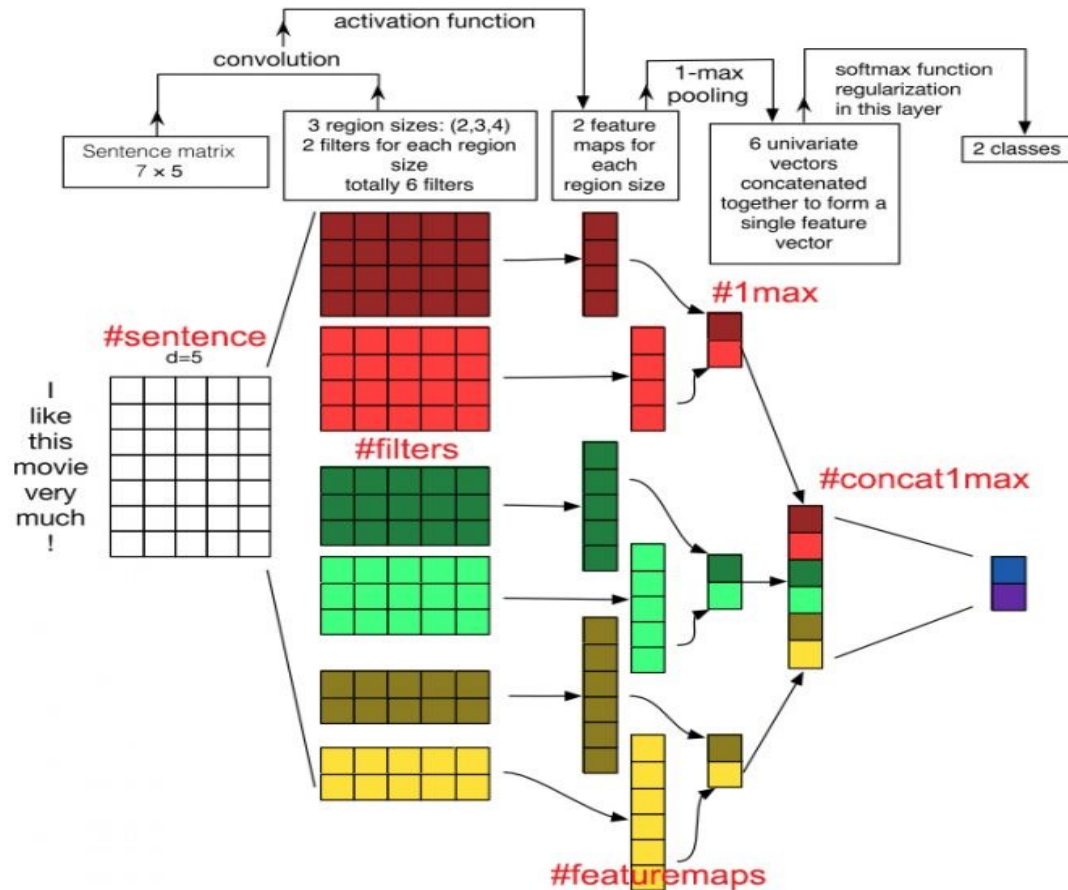


Figure 1: Convolution operation performed on a sentence sequence of 7 rows and 5 columns. There are 2 filters given in each region applied over sentence sequence with a dot product sum over the entire sequence.

Citing the first inscription here, to be talked about later. "Figure 1: Illustration of a CNN architecture for sentence classification. Models delineate three filter region sizes: 2,3,4, every one of which has 2 filters. Filters perform convolutions on the sentence matrix and create (variable-length) features maps; 1-max pooling is performed over each map, i.e., The biggest number from each feature map is recorded. In this manner, a univariate feature vector is produced for every one

of the six maps, and these 6 features are connected to form a feature vector for the penultimate layer. The last SoftMax later at that point gets this feature vector as input and utilizes it to classify the sentence; here accepted parallel grouping and henceforth portray two conceivable yield states."

Sentence

The precedent is "I like this motion picture without a doubt!", There are 6 words here, and the exclamation mark is dealt with like a word – a few scientists do this any other way and dismissed the exclamation mark – altogether, there are 7 words in the sentence. The creators picked 5 to be the dimension of the word vectors. Proposed the mean the length of the sentence and signify the dimension of the word vector. Consequently, presented have a sentence matrix of the shape $s \times d$, or 7×5 .

Filters

One of the alluring properties of CNN is that it preserves 2D spatial introduction in the PC version. Writings, like the pictures, have an introduction. Rather than 2-dimensional, texts have a one-dimensional structure where word sequence matters. This research likewise reviews that all words in the preceding are each supplanted by a 5-dimensional word vector. Consequently, it fixed one dimension of the filter to coordinate the word vectors (5) and change the region size, h . Region estimate alludes to the number of rows – speaking to word – of the sentence matrix filtered.

In the figure, #filters are the filters' illustrations, not what has been filtered out from the sentence matrix by the filter. The next paragraph would make this distinction clearer. Here, the authors chose to use 6 filters – 2 complementary filters to consider (2,3,4) words.

Feature Map

In this section, step-through through how CNN performs convolutions/filtering. I have filled some numbers in the sentence matrix and the filter matrix for clarity.

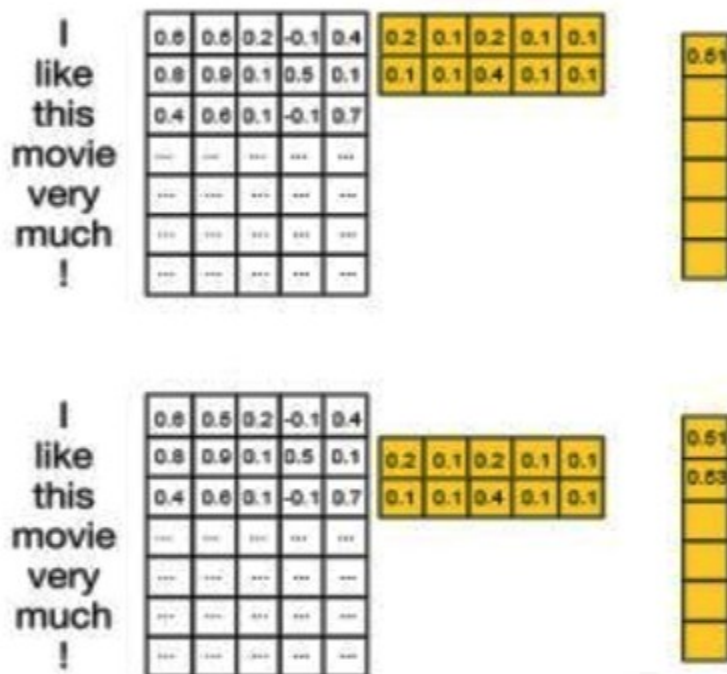


Figure 2: Feature Map

The above illustrates the action of the 2-word filter on the sentence matrix. First, the two-word filter, represented by the 2×5 yellow matrix w , overlays across the word vectors of “I” and “like”. Next, it performs an element-wise product for all its 2×5 elements, and then sum them up and obtain one number ($0.6 \times 0.2 + 0.5 \times 0.1 + \dots + 0.1 \times 0.1 = 0.51$). 0.51 is recorded as the first element of the output sequence, so, for this filter. Then, the filter moves down 1 word and overlays across the word vectors of ‘like’ and ‘this’ and performs the same operation to get 0.53. Therefore, so will have the shape of $(s-h+1 \times 1)$, in this case $(7-2+1 \times 1)$.

To obtain the feature map, c , it added a bias term (a scalar, i.e., Shape 1×1) and apply an activation function (e.g., ReLU). This gives us c , with the same shape as o ($s-h+1 \times 1$).

To obtain the feature map, c , it added a bias term (a scalar, i.e., Shape 1×1) and apply an activation function (e.g., ReLU). This gives us c , with the same shape as o ($s-h+1 \times 1$).

1-Max

Notice that the dimensionality of c is dependent on both s and h , in other words, it will vary across sentences of different lengths and filters of different region sizes. The authors employ the 1-max pooling function to tackle this problem and extract the largest number from each c vector.

Concat1-Mmax

After 1-max pooling, there is certain to have a fixed-length vector of 6 elements (= number of filters = numbers of filters per region size (2) x number of region size considered (3)). This fixed-length vector can then be fed into a SoftMax (fully connected) layer to perform the classification. The error from the classification is then back-propagated back into the following parameters as part of learning:

The W matrices that produced O .

The bias term that is added to produce C .

Transfer Learning

The pre-trained model is very different from the one on which the pre-trained model was trained, the prediction would be very inaccurate. For example, a model previously trained in speech recognition will work horribly if model try to use it to identify objects using it.

There are many pre-trained architectures are directly available for use in Kera's library. ImageNet data set has been widely used to build various architectures since it is large enough (1.2M images) to create a generalized model. The problem statement is to train a model that can correctly classify the images into 1,000 separate object categories. These 1,000 image categories represent object classes that come across in our day-to-day lives, such as species of dogs, cats, various household objects, vehicle types, etc.

These pre-trained networks demonstrate a strong ability to generalize to images outside the ImageNet dataset via transfer learning. The modifications to the pre-existing model by fine-tuning the model. Since assuming that the pre-trained network has been trained quite well, it would be modifying the weights too soon and too much. While modifying, one generally uses a learning rate smaller than those used for initial training in the model.

Ways to Fine-tune, the model

Feature extraction – if a pre-trained model used as a feature extraction mechanism. It can remove the output layer (the one which gives the probabilities of being in each of the 1000 classes) and then use the entire network as a fixed feature extractor for the new data set.

Use the Architecture of the pre-trained model – This research used the architecture of the model while it initializes all the weights randomly and trains the model according to the dataset.

Train some layers while freezing others – Another way to use a pre-trained model is to train it as partially. It can keep the weights of the model's initial layers freeze while retraining only the higher layers. It can test how many layers are to be freeze and how many are to be trained.

The below diagram should help one to decide on how to proceed with using the pretrained model in this case.

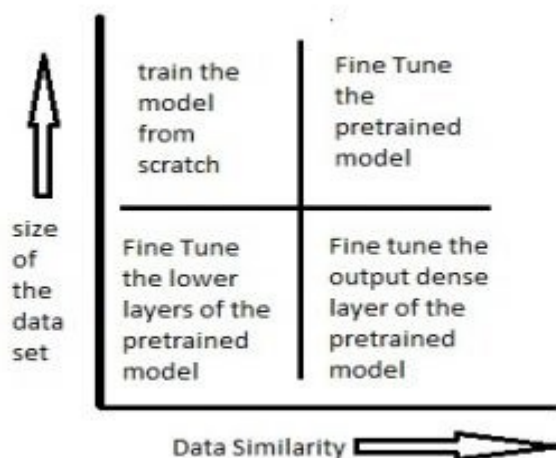


Figure 3: Training and Fine-Tuning Strategy

Scenario 1 – The size of the Data set is small while the Data similarity is very high – In this case, since the data similarity is very high, it does not need to retrain the model. All it needs to do is to customize and modify the output layers according to the problem statement. It uses the pretrained model as a feature extractor. Suppose it decides to use models trained on ImageNet to identify if the new set of images has cats or dogs. Here the images it needs to identify would-be like imagine. However, it just needs two categories as my output – cats or dogs. In this case, all it does is modify the dense layers and the final SoftMax layer to output 2 categories instead of 1000.

Scenario 2 – The size of the data is small and data similarity is very low – In this case, it can freeze the initial (let's say k) layers of the pretrained model and train just the remaining $(n-k)$ layers again. The top layers would then be customized to the new data set. Since the new data set has low similarity, it is significant to retrain and customize the higher layers according to the new dataset. The small size of the data set is compensated because the initial layers are kept pretrained (which have been trained on a large dataset previously), and the weights for those layers are frozen.

Scenario 3 – The size of the data set is large. However, the Data similarity is very low – In this case, since it has a large dataset, the neural network training would be effective. However, since the data is very different from the data used for training the pretrained models. The predictions made using pretrained models would not be effective. Hence, it's best to train the neural network from scratch, according to your data.

Scenario 4 – The size of the data is large as well, as there is a high data similarity – This is the ideal situation. In this case, the pretrained model should be most effective. The best way to use

Plot 2: Total 2-gram pattern frequency