**Review and Consolidation of Advanced Deep Learning Techniques For Text Classification**

*Submitted in partial fulfillment of the requirements for the degree of*

**Bachelor of Technology**

**in**

**Electronics and Communication Engineering**

*by*

**SARTHAK GOSWAMI: 17BEC0904**

**AYUSH BISHNOI: 17BIS0066**

**Under the guidance of**

**Prof. Dr. Naveen Kumar Vaegae**

**SENSE**

**VIT, Vellore.**



May, 2021

**DECLARATION**

I hereby declare that the thesis entitled “Review and Consolidation Of Advanced Deep Learning Techniques For Text Classification” submitted by me, for the award of the degree of Bachelor of Technology in Programme to VIT is a record of bonafide work carried out by me under the supervision of Prof. Vaegae Naveen Kumar.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 31/05/2021

**CERTIFICATE**

This is to certify that the thesis entitled “Review and Consolidation Of Advanced Deep Learning Techniques For Text Classification” submitted by Ayush Bishnoi 17BIS0066 & Sarthak Goswami 17BEC0904, SENSE, VIT, for the award of the degree of Bachelor of Technology in Programme, is a record of bonafide work carried out by him / her under my supervision during the period, 01. 12. 2020 to 30.04.2021, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : 31/05/2021

Signature of the Guide

Internal Examiner External Examiner

Head of the Department

Electronics and Communication Engineering

**Executive Summary**

Text classification is a widely popular field today used in almost every industry. There is an immense amount of text online today, and with advancements in deep learning and neural networks, analyzing it provides a lot of unique benefits in various situations, namely in places that require categorizing articles or attaching tags to them, inspecting sentiments in the large amount of data available from social media creating profiles of users based on it. Although classical machine learning techniques still work really well with text, there’s a huge number of deep learning techniques that can be used on this text today. It hence becomes extremely important to see the applications of these situations in the real world and in different industries. Through our work, we intend to shed light on different uses of text classification in major industries today. This paper gives the different requirements in different industries and different factors affecting. It then gives a semi-universal metric to evaluate training time and accuracy relative to other similar methods, this can be used to effectively select the best method for a certain deployment, it also summarizes popular methods used for text classification today and gives their pros and cons in regards to real world use. Using popular datasets and frameworks, it gives a comparison of different methods on sample articles.

Contents

[1. Introduction 9](#_Toc73462575)

[2. Proposed Method 10](#_Toc73462576)

[3. Literature Review 11](#_Toc73462577)

[4. Comparative Analysis 13](#_Toc73462578)

[4.1. Feed Forward Neural Networks 13](#_Toc73462579)

[4.2. RNN 14](#_Toc73462580)

[4.3. CNN 16](#_Toc73462581)

[4.4. Capsule Neural Networks 17](#_Toc73462582)

[4.5. Attention Mechanism 18](#_Toc73462583)

[4.6. Transformers 20](#_Toc73462584)

[4.7. Reinforcement Learning 21](#_Toc73462585)

[5. Database and performance metrics 21](#_Toc73462586)

[6. Experimental setup 22](#_Toc73462587)

[7. New Metric 22](#_Toc73462588)

[7.1. Reasoning behind this metric 22](#_Toc73462589)

[7.2. Normalization 22](#_Toc73462590)

[8. Results 23](#_Toc73462591)

[8.1. Graphs: 23](#_Toc73462592)

[8.1.1. Classification Accuracy vs Number of Epochs for AG-News: 23](#_Toc73462593)

[8.1.2. Classification Accuracy vs Number of Epochs for DBpedia: 26](#_Toc73462594)

[8.1.3. Classification Accuracy vs Number of Epochs for IMDb: 29](#_Toc73462595)

[8.2. Accuracy 32](#_Toc73462596)

[8.2.1. AG News 32](#_Toc73462597)

[8.2.2. DBPedia 33](#_Toc73462598)

[8.2.3. IMDb 33](#_Toc73462599)

[8.3. Proposed New Metric 34](#_Toc73462600)

[8.3.1. AG News 34](#_Toc73462601)

[8.3.2. DBPedia 34](#_Toc73462602)

[8.3.3. IMDb 35](#_Toc73462603)

[9. Observations 35](#_Toc73462604)

[10. Discussions 35](#_Toc73462605)

[11. Conclusion 36](#_Toc73462606)

[12. Areas of Improvement 36](#_Toc73462607)

[13. References 38](#_Toc73462608)

1. **List of Figures**

[Figure 1: two broad text classification methods 9](#_Toc73460050)

[Figure 2: Feed Forward Neural Network and it's components 12](#_Toc73460051)

[Figure 3: RNN's underlying LSTM structure 13](#_Toc73460052)

[Figure 4: an RNN interpreting a basic sentence 14](#_Toc73460053)

[Figure 5: CNN underlying structure in text classification 15](#_Toc73460054)

[Figure 6: Capsule Neural Networks and it's various building blocks 16](#_Toc73460055)

[Figure 7: Hierarchial representation of an Attention Mechanism 18](#_Toc73460056)

[Figure 8: Building blocks of a Treanformer 19](#_Toc73460057)

[Figure 9: Accuracy vs epochs on AG-News dataset of Feed Forward Neural Networks 22](#_Toc73460058)

[Figure 10: Accuracy vs epochs on AG-News dataset of RNNs 22](#_Toc73460059)

[Figure 11: Accuracy vs epochs on AG-News dataset of CNNs 23](#_Toc73460060)

[Figure 12: Accuracy vs epochs on AG-News dataset of Capsule Neural Networks 23](#_Toc73460061)

[Figure 13: Accuracy vs epochs on AG-News dataset of Attention Mechanisms 24](#_Toc73460062)

[Figure 14: Accuracy vs epochs on AG-News dataset of Transformers 24](#_Toc73460063)

[Figure 15: Accuracy vs epochs on DBpedia dataset of Feed Forward Neural Networks 25](#_Toc73460064)

[Figure 16: Accuracy vs epochs on DBpedia dataset of RNNs 25](#_Toc73460065)

[Figure 17: Accuracy vs epochs on DBpedia dataset of CNNs 26](#_Toc73460066)

[Figure 18: Accuracy vs epochs on DBpedia dataset of Capsule 26](#_Toc73460067)

[Figure 19: Accuracy vs epochs on DBpedia dataset of Attention Mechanisms 27](#_Toc73460068)

[Figure 20: Accuracy vs epochs on DBpedia dataset of Transformers 27](#_Toc73460069)

[Figure 21: Accuracy vs epochs on IMDb dataset of Feed Forward Neural Networks 28](#_Toc73460070)

[Figure 22: Accuracy vs epochs on IMDb dataset of RNN 28](#_Toc73460071)

[Figure 23: Accuracy vs epochs on IMDb dataset of CNNs 29](#_Toc73460072)

[Figure 24: Accuracy vs epochs on IMDb dataset of Attention Mechanisms 29](#_Toc73460073)

[Figure 25: Accuracy vs epochs on IMDb dataset of Transformers 30](#_Toc73460074)

1. **List of Tables**

[Table 1: Accuracy of each text classification model on the AG-News dataset after 15 epochs of training 32](#_Toc73460146)

[Table 2: Accuracy of each text classification model on the DBpedia dataset after 15 epochs of training 33](#_Toc73460147)

[Table 3: Accuracy of each text classification model on the IMDb dataset after 15 epochs of training 33](#_Toc73460148)

[Table 4:Relative performance score for AG-News dataset 34](#_Toc73460149)

[Table 5: Relative performance score for DBPedia dataset 35](#_Toc73460150)

[Table 6: Relative performance score for IMDb dataset 35](#_Toc73460151)

# Introduction

Text classification and analysis is a huge field for machine learning and deep learning. It basically aims to look at different sources of text and provide insights. This insight can be in the form of tags or themes prevalent in text, or it can be about semantics and the language of the text. All these insights can be used in a lot of ways today due to the large amount of text being available today, especially through sources such as messages, chats, emails, comments, reviews, tweets, posts and many other. It can also be used by people writing articles in virtually every field, since the increasing amount of online information creates a huge need to categorize all this data. However, a huge challenge is this massive amount of data also brings challenges and constraints that require different methods for different approaches. Deep learning is a great approach to solve this problem, as it is able to use abstraction at a high level [1] on this data through various mechanisms with different hierarchies.

Initially, all this classification was done manually, by reading all the text and assigning different values for it. However there is too much text to do this manually now, and this is where machine learning techniques have become popular. Classical machine learning works with manually created features that have to be gathered from a document. These features are then written for classifiers, which make predictions based on these. However, these methods have a lot of shortcomings, mostly with the formation of features that is simply impossible with the amount of data available today. Also, since this uses very specific domains for situations, it can’t be expanded universally. Furthermore, these models can’t make use of the valuable data today since most things have to be pre- defined. But the idea still remains that for a lot of situations such as article classification, these techniques are the only possible approaches due to the complexity of work. However, the advent of deep learning techniques has proven to be extremely successful. These techniques can make use of a large number of layers [1] to give results otherwise not possible with earlier techniques and find new patterns in data not observed before. They can also be applied to multiple situations since they work in an adaptive, end to end fashion. Feed forward neural networks provide the simplest architecture possible to make use of this, and still work very effectively. Using these layers and networks in different ways can yield many different results. Convolutional neural networks, traditionally used for image classification has proven to be successful in several situations regarding text classification as well [2], since it provides the necessary complexity to deal with this data. Recurrent neural networks are the preferred technique since they take the semantics of text in consideration while giving insights. They have a lot of use in newspaper article classification [3]. The latest methods try to combine favorable traits in hybrid methods to get the best of all words. There are many upcoming techniques as well such as capsule learning, attention mechanisms and transformers.

Through this paper, we intend to work on different techniques and characterize their algorithms. I also wish to give the different merits and demerits these methods have in industrial use, as well as motivation provided for further research. By the end, I compare popular datasets available online (on TensorFlow) and compare efficiencies of different methods. This paper aims to look at almost all popular techniques possible for all types of text classification techniques possible, and then look at them from a real world perspective. It aims to use different methods and research into them and actualize them in a real life framework, looking at how these techniques are actually used by individuals and organizations. It looks at these theoretical concepts in the real world and provides real world context into their research, the importance for further experimentation and research and the various important reasons to use different techniques in different situations.

Another flaw in the current approach to text classification is comparing only accuracy for methods that take massive time compared to other approaches, hence not actually telling us how much more accuracy and performance they provide with the extra time taken. We aim to compare these methods in a more equitable manner by taking this time taken into account for the same processing setup.

# Proposed Method

The next section looks at each method in isolation, and lists down the basic important features. It gives the basic definition, algorithm and popular uses for each popular method. It then gives pros and cons for each method, and which method improves these specific caveats, or loses out on benefits earlier methods provide. It separates methods using traditional methods that use text as a bag of words, and modern methods that instead use methods like word2vec, that also take semantics and surrounding words as features.

The next section compares these methods on popular databases and gives basic error metrics to compare them in different situations, namely sentiment analysis and article categorization, providing basic classification.

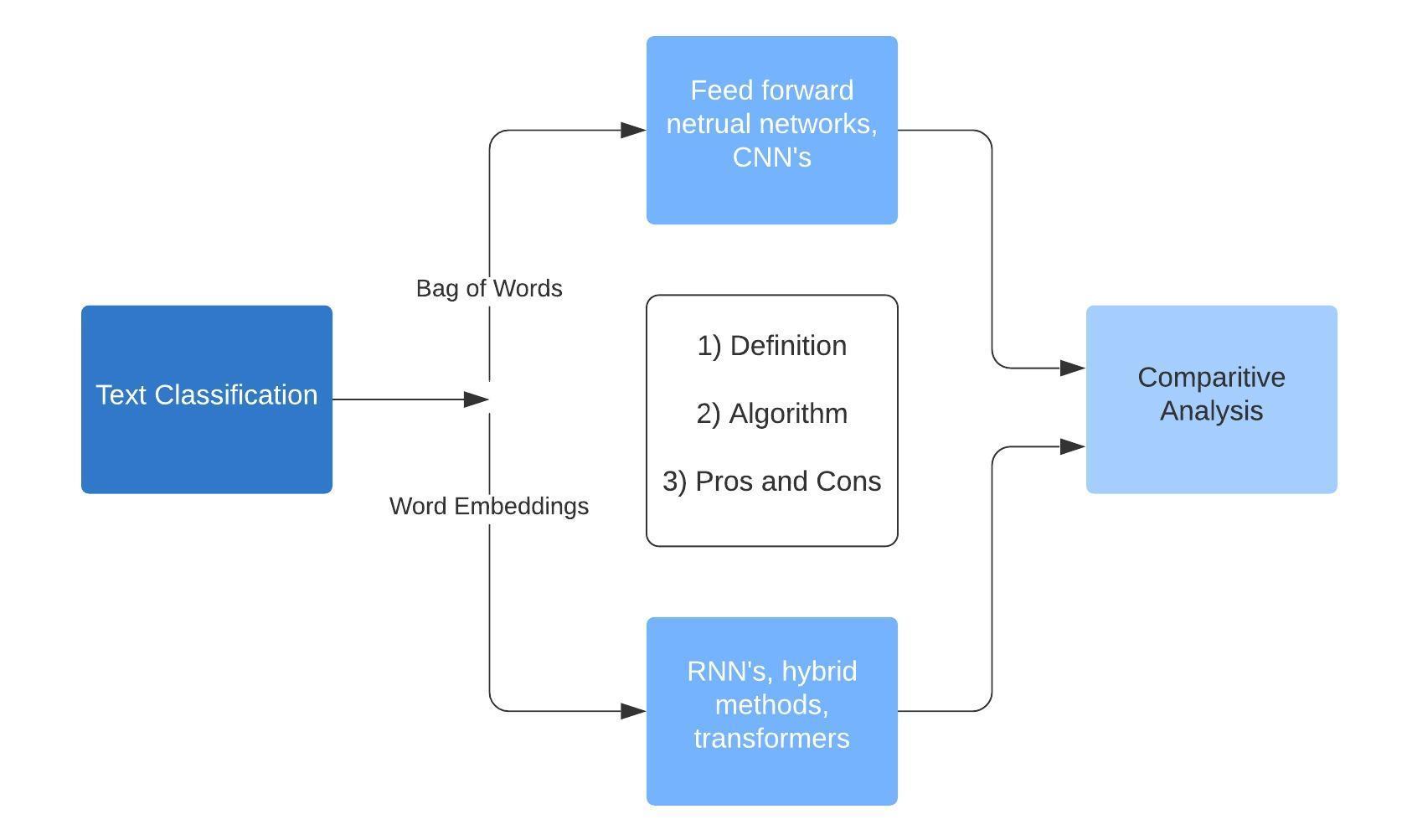


Figure 1: two broad text classification methods

# Literature Review

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Title** | **Author(s)** | **Dataset** | **Metrics** |
| 1 | A Comparative Study of Neural Network Models for Sentence Classification | Phuong Le-Hong, Anh-Cuong Le | UIUC Corpus | Accuracy |
| 2 | Text Classification Using KM-ELM Classifie | Neethu K S, Jyothis T S, Jithin dev | Iris, Diabetes, Lenses, 20 Newsgroups, ACL anthology network, Blood Pressure Estimation | Accuracy |
| 3 | Text Sentiment Analysis Based on Long Short-Term Memory | Dan Li, Jiang Qian | Comments from the website JD.COM, travel comments from ctrip and English movie reviews | Accuracy, recall rate |
| 4 | Convolutional Recurrent Deep Learning Model for Sentence Classification | Abdal Raouf Hassan, Ausif Mahmood | SSTb, IMDb | Accuracy |
| 5 | Term-Based Pooling in Convolutional Neural Networks for Text Classification | Shuifei Zeng, Yan Ma, Xiaoyan Zhang, Xiaofeng Du | HIT question category corpus | Train set / Test set accuracy |
| 6 | Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis | Alec Yenter, Abhishek Verma | ACL IMDb | Maximum accuracy |
| 7 | Text Classification using Capsules | Jaeyoung Kima , Sion Janga , Eunjeong Parkb,∗ , Sungchul Choia,∗ | 20news, Reuters10, MR, TREC-QA, MPQA, IMDb | Average accuracies for 5 turns |
| 8 | Siamese capsule networks with global and local features for text classification | Yujia Wua , Jing Li a,∗ , Jia Wub , Jun Changa | MR, MPQA, SUBJ, SST1, SST2, TREC | Accuracies, Hyper Parameters |
| 9 | Knowledge attention sandwich neural network for text classification | Zhiqiang Zhana,b , Zifeng Houa,b,∗ , Qichuan Yangc , Jianyu Zhao d, Yang Zhang d, Changjian Hu | TREC, SUBJ, MR, SST-5 | Accuracies |
| 10 | Relation classification via knowledge graph enhanced transformer encoder | Wenti Huang Yiyu Mao Zhan Yang Lei Zhu Jun Long | SemEval, TACRED | Accuracy |

# Comparative Analysis

This section lists out notable aspects about all methods and compares them.

## Feed Forward Neural Networks

These models use the bag of words approach, this is a basic and fairly easily implementable neural network where flow of information is unidirectional – Forward – from the input

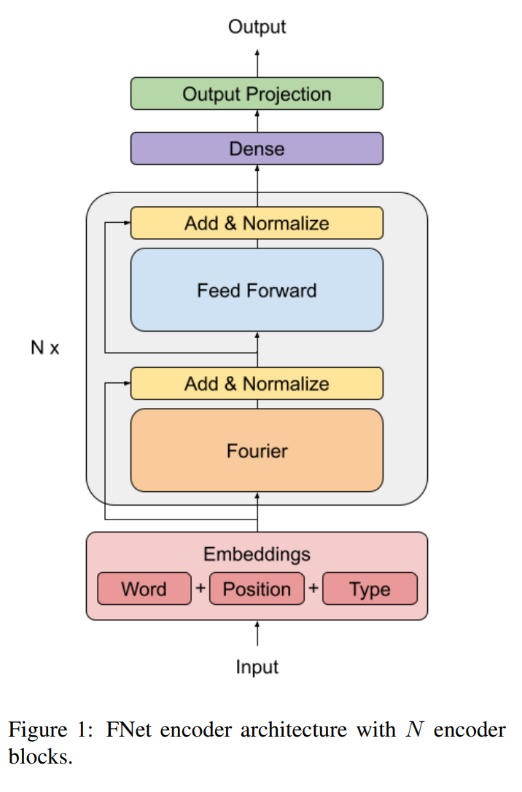
****

Figure 2: Feed Forward Neural Network and it's components

***Algorithm***

1. Count instances of every word, removing commonly used words.
2. Assign different required values on words if necessary (n-grams) for more accurate results [4].
3. We can also add paragraph vectors to get some contextual information.
4. Pass it through multiple layers using Multi Layer Perceptrons.
5. Use different classifiers such as logistic regression, Support Vector Machines etc.

***Pros***

1. Less pre processing required.
2. Simple method.
3. Works fast even for large amounts of data.
4. Gives sufficient results in most categorization functions.

***Cons***

1. Does not capture sequence of words, hence not as accurate.
2. Does not take semantics and context into consideration, hence not the best possible accuracy.

## RNN

Models using RNN’s view text as sequential data, and hence focus on capturing context and semantics. They focus on the structure of available text.

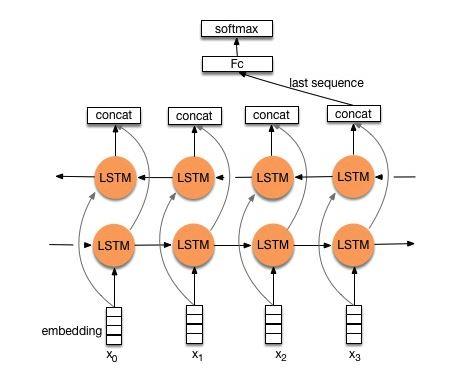
****

Figure 3: RNN's underlying LSTM structure

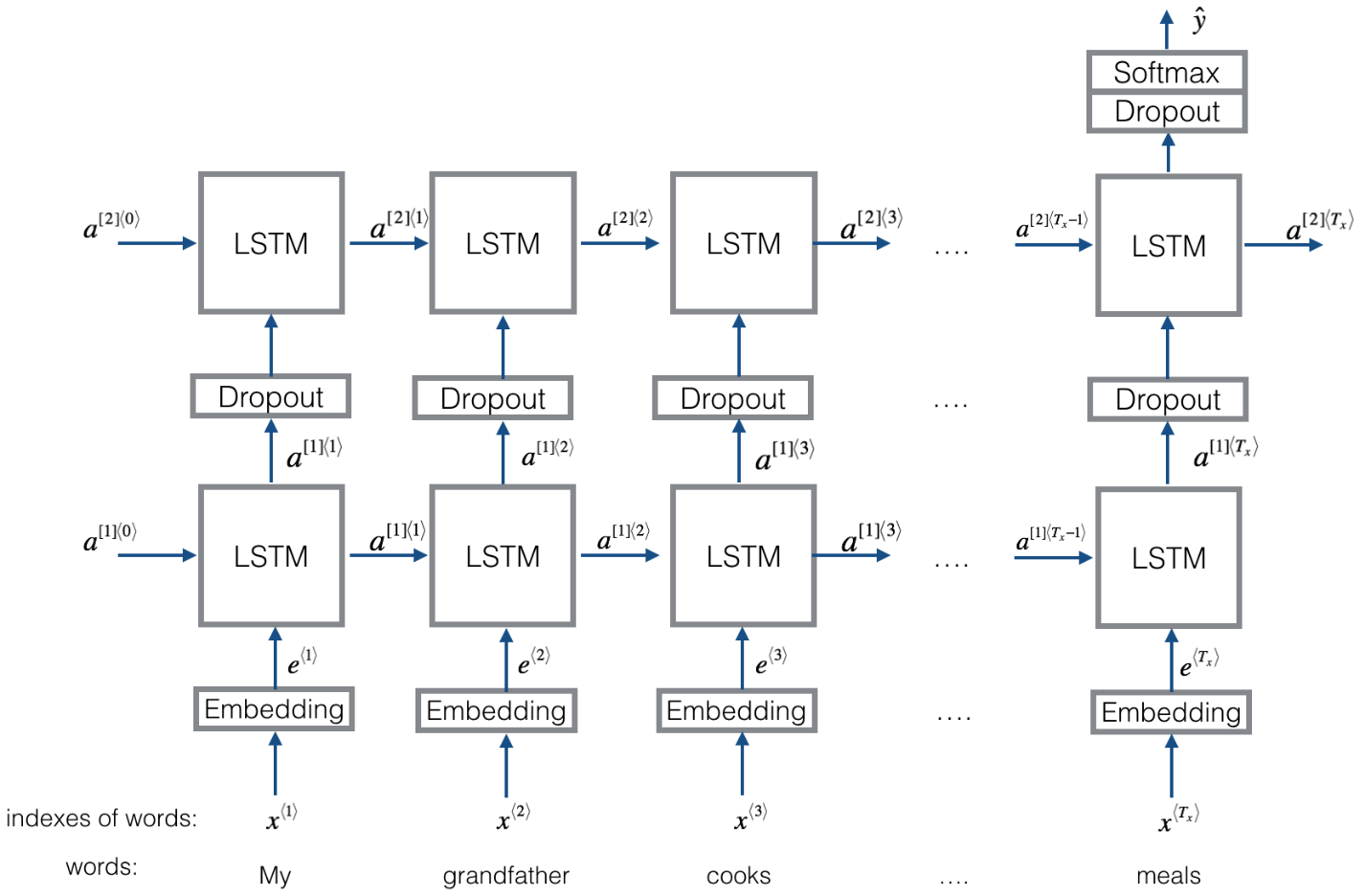


Figure 4: an RNN interpreting a basic sentence

***Algorithm***

1. It uses a mechanism called long term short memory, which processes a series of data instead of singular points.
2. They provide connections for feedback.
3. They use word2vec to vectorize the text.
4. It stores the length of given text, and then takes this text as input as an ongoing sequence of text.
5. The vectors of the text are added cumulatively by weights to get representative vector.
6. A Softmax layer is hence used to give classification.

***Pros***

1. Remembers the various dependencies of all words, and takes the text as a sequence.
2. Takes grammar, structure and other details into account.

***Cons***

1. They are good at using local small scale structures, but struggle to remember long range dependencies over a huge amount of text.
2. Classic RNN models do not perform well in most situations, and have to be paired up in different configurations with different methods.
3. Long term short memory provides a useful way to record context, but it often only selects part of information, and often overlooks a lot of valuable features over all the iteration it works for.

## CNN

Models that use CNN, traditionally used for image classification, instead look for patterns in text such as important phrases and such through its complex structure and huge number of layers available.

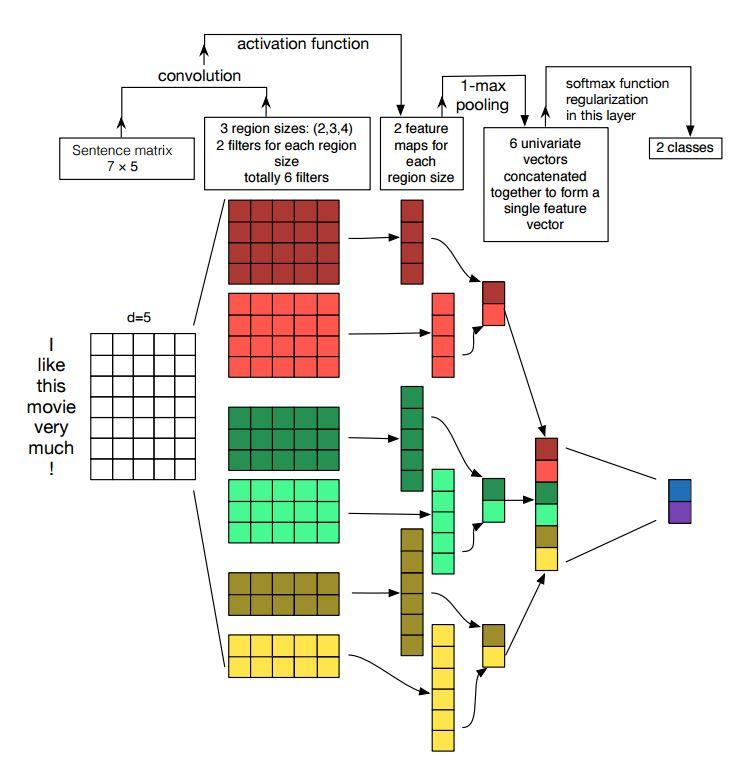


Figure 5: CNN underlying structure in text classification

***Algorithm***

1. While RNN time sequential data and over time develop results, CNN’s focus on taking the entirety of data and then finding patterns across it.
2. They use multiple successive layers that help it use convolution on the text and then pooling.
3. It first makes a matrix consisting of all sentences using vectors for each word appearing in a sentence.
4. A complex and convolutional structure then uses these layers to generate a feature map over each sentence that uses both short and long range connections for features.
5. The parameters for each layer are chosen through successive iterations and training, and depend on the level and size of the layer being used.
6. Multiple word embedding methods can be used such as randomly assigning values, using values by pre training model or changing these values dynamically based on every iteration.

***Pros***

1. Work best for detecting features where position is not important.
2. Articles with certain important phrases are easily classified with good speed.
3. It dynamically identifies important features and values, and is able to reduce its complexity by itself.

***Cons***

1. It does not capture semantics and context.
2. It has a very complicated architecture.
3. It does not track information regarding space occupied by words, and can easily go wrong with data based on different proportion or orientation.
4. In most situations, shallow neural networks or networks that do not use convolution provide similar results and use much less time, hence being better for extremely large datasets which require classification.

## Capsule Neural Networks

It fixes the problem of losing important information that CNN’s, and has recently found use in fields of text classification.

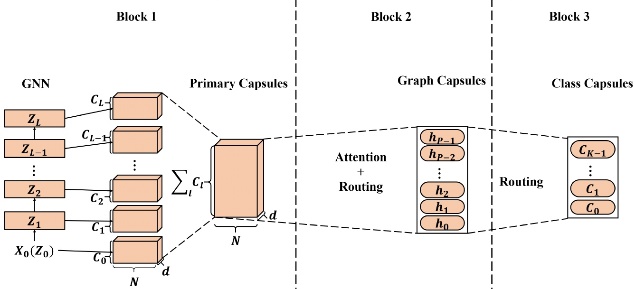


Figure 6: Capsule Neural Networks and it's various building blocks

***Algorithm***

1. It uses capsules, i.e groups of multiple neurons, having activity vectors that work for multiple properties of different entities such as words, sentences or paragraphs.
2. The length of each vector is used to store the probability of an object existing, and the orientation stores important characteristics.
3. They use a hierarchy where a parent capsule collects information from capsules present in lower layers, and collectively pools this information up to the final layer.
4. This process is called routing and can be achieved dynamically or statically.
5. The model works with:
   1. N-gram convolution layer
   2. Capsules
   3. Convolutional Capsules
   4. Connected capsule layers
6. The lower layers generate maps of features and then a linear unit to keep useful information.
7. The convolutional layer finds overall features by aggregating this information.

***Pros***

1. Solves loss of information faced by CNN’s in certain situations.

***Cons***

1. Words that are not related to any document categories may be given preference, and have to be solved by presence of noise capsules.
2. These noise capsules interrupt the main working.

## Attention Mechanism

It is used to identify relations between different words in text, and is a useful addition in several existing models. This method mostly acts as a layer of extra improvement on preexisting methods to increase accuracy and reliability, these have two unique features:

1. It can be used to pay ‘attention’ to the hierarchal structure of documents
2. It has two implementable levels of attention mechanisms that can be used at word level and sentence level to enable the model (to which Attention Mechanism is being used to improve) to understand and interpret information on different levels within a sentence or throughout a folder

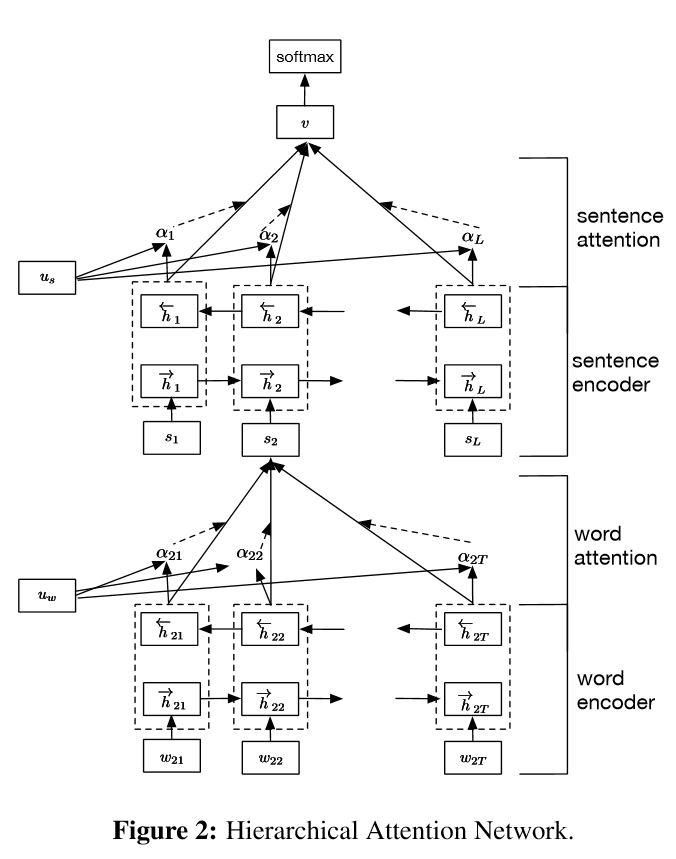


Figure 7: Hierarchial representation of an Attention Mechanism

***Algorithm***

1. They work on the principle of how we pay attention to certain parts of an image or an object, and groups different sections of a text.
2. It is basically a technique using weights indicating importance.
3. For each word, it checks its relation or “attention” paid to other parts of the text. It adds these up with different weights decided by the attention mechanism to predict another word.
4. The way this works is that after each prediction, it only uses words that are actually relevant to the given word and finds the concentration of this information using information weights.
5. The encoder encodes information as it used to, but the decoder compares everything with the context vectors it deems important.
6. Each relation is checked against an alignment model that tries to find the level of match between input and output.
7. It thus decides information that is necessary for each part of text and assigns scores to them.

***Pros***

1. For longer sentences, traditional neural networks compress information into fixed vectors that do not change, which leads to a worse performance.
2. After training, for basic encoder decoder groups, there are multiple problems when dealing with sentences that are longer than the largest available sentence during training.

***Cons***

1. Very time consuming.
2. Does not parallelize well.

## Transformers

Solves shortcomings of RNN’s that do not allow it to be used simultaneously in situations (parallelization). It is used to train large models of different languages.

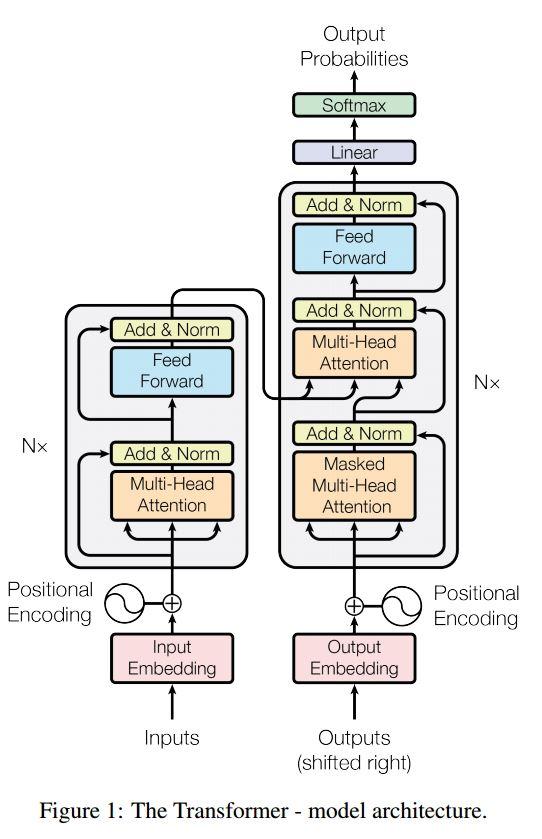


Figure 8: Building blocks of a Treanformer

***Algorithm***

1. It first uses self attention to calculate scores for each word in a sentence (attention score) parallel.
2. Through this way, it is able to view the entire text in a single go as a whole, instead of only sequentially by RNNs.
3. It stacks multiple encoder decoder blocks together.
4. Each word embedding is passed on to an encoder, transformed and then sent to next encoder.
5. The stack computes the required information in a single trial.

***Pros***

1. Give us the best method for parallelization.
2. Even though CNN’s do not use sequential data, they still suffer as sentences become large.

***Cons***

1. Requires a huge amount of training data and training time.
2. Basic transformers do not work good on datasets that are not that big.

## Reinforcement Learning

It is a method that uses continuous interaction with the environment during learning.

***Algorithm***

1. It performs specific tasks according to a set of rules, and uses a system to optimize reward.
2. It chooses the most important words in text that are relevant to the task.
3. It is used to discover key phrases in a sentence and then compile it.

***Pros***

1. Performs the same function as a CNN but with less complexity and much quicker.

***Cons***

1. Each system of rewards has to be designed differently for different situations.
2. Does not have many situations it can be readily used in.
3. Only used for a handful of classification problems.
4. Probabilistic models are better since they’re easier to interpret.

# Database and performance metrics

This section talks about the datasets chosen in detail, and the reasons for choosing them. They’re the most popular datasets used for analyzing text classification algorithms taking into account different functions of sentiment analysis, news categorization and text categorization. They’re also varying in the number of possible labels, entries and splits, to get a complete picture.

**IMDb**: The IMDb dataset is one of the most popular datasets for sentiment analysis. It has a total of 50,000 entries evenly split for testing and training, and has either a positive or negative category, with the same number for both categories as well.

**AG News**: The AG News dataset consists of 2,000 news articles by an academic news website ComeToMyHead. There are 1,20,000 training samples and 7,600 test samples, making it a good judge of training capability.

**DBpedia**: The DBpedia dataset is generally chosen for being a large scale dataset that is multilingual. It uses common information boxes from Wikipedia, and it’s updated monthly with different classes and properties being changed. There are 5,60,000 training samples and 70,000 testing samples, and 14 overall categories.

# Experimental setup

All tests were run on a local machine with an Nvida RTX 2070 SUPER and 16 gigabytes of RAM, and an i7-4770k, but since all our models were CUDA accelerated, the GPU was used. A local machine with a static configuration was used to ensure elimination of variables from server farms. This also gave enough time for the smaller datasets to produce tangible results since the output was not instant

# New Metric

## Reasoning behind this metric

It was observed in our literature review that an ‘accuracy’ metric was preferred by most researchers. Although a lot of paper preferred using error too, this leads to a skewed representation of the actual data as it does not take into account training time. Training time is an extremely important measure since the huge differences in these training times show us that comparing accuracy for methods taking less time with those that take a much larger time is not fair, and training time is extremely important for long term functioning of these algorithms in real life scenarios. Not all deployments have the liberty or resources to train their model for days or on state of the art hardware, our solution to that was to standardize the epochs, time takes, and dataset for the given methods to gain a true understanding of the power and ease of use of the given methods. This metric allows us to understand the relative performance of each method at a glance. based on the deployment a suitable method can be chosen.   
We tried to eliminate as many variables as possible, hence all the tests are done at 15 epochs with the regular training/testing splits given by the most popular datasets.

## Normalization

A function value between 0.5 to 1 is preferred as, if normalization was done on a 0 to 1 scale, it would have left us with a 0 value which would not have been reliable going forward. it would have led to miss representation is certain cases and the accuracy could be interpreted as 0

Using minmax was a more favorable metric than a z score ranging between -1 to 1 as it led to certain negative values misrepresenting data.

# Results

## Graphs:

### Classification Accuracy vs Number of Epochs for AG-News:

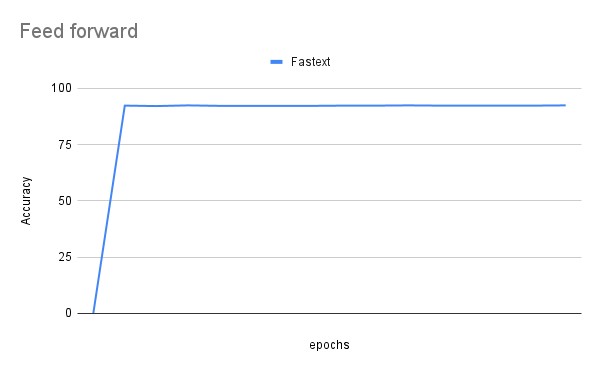
****

Figure 9: Accuracy vs epochs on AG-News dataset of Feed Forward Neural Networks

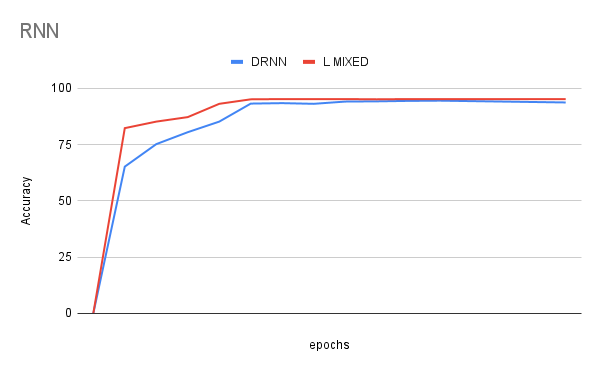
****

Figure 10: Accuracy vs epochs on AG-News dataset of RNNs

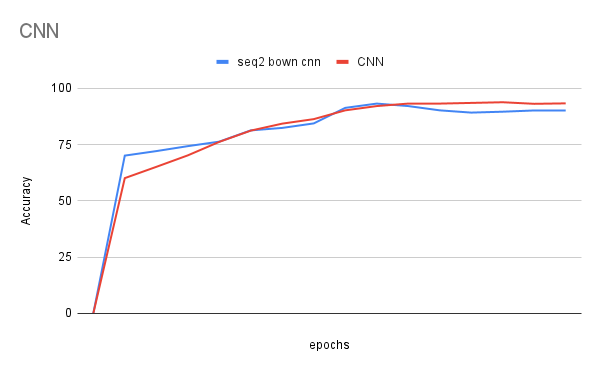
****

Figure 11: Accuracy vs epochs on AG-News dataset of CNNs

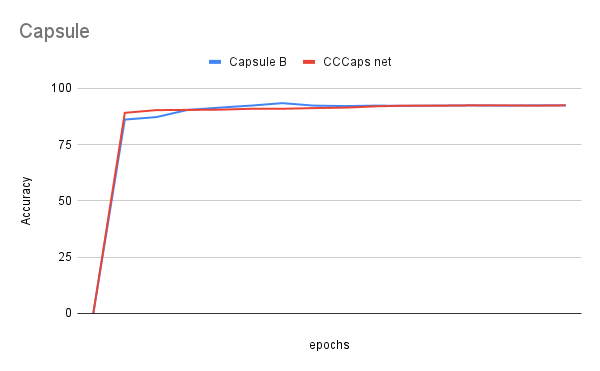
****

Figure 12: Accuracy vs epochs on AG-News dataset of Capsule Neural Networks

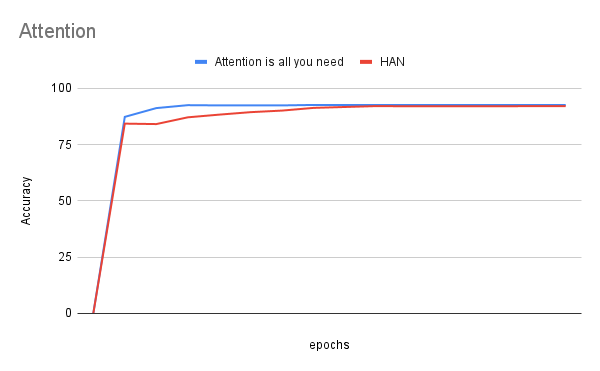
****

Figure 13: Accuracy vs epochs on AG-News dataset of Attention Mechanisms

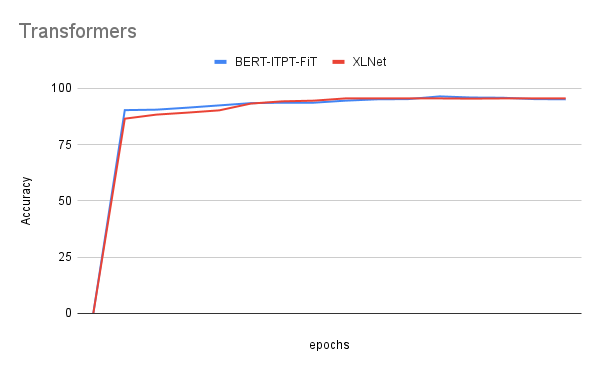
****

Figure 14: Accuracy vs epochs on AG-News dataset of Transformers

### Classification Accuracy vs Number of Epochs for DBpedia:

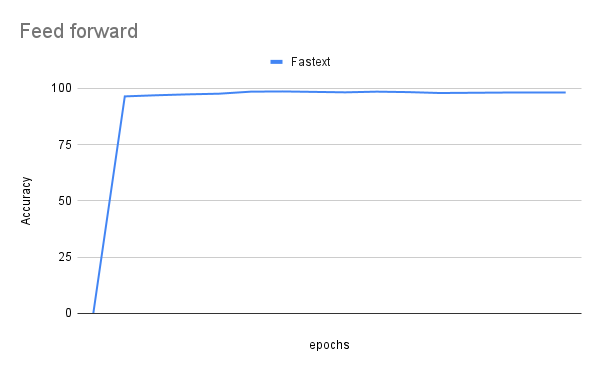
****

Figure 15: Accuracy vs epochs on DBpedia dataset of Feed Forward Neural Networks

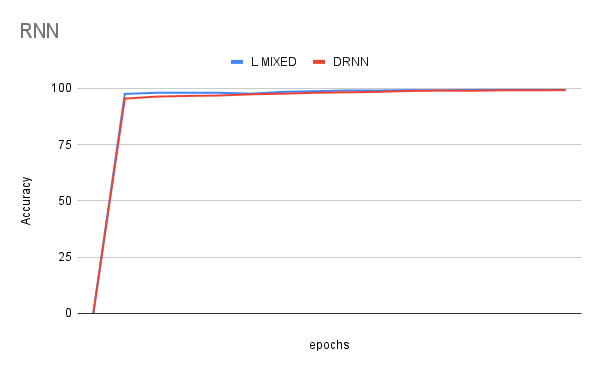
****

Figure 16: Accuracy vs epochs on DBpedia dataset of RNNs

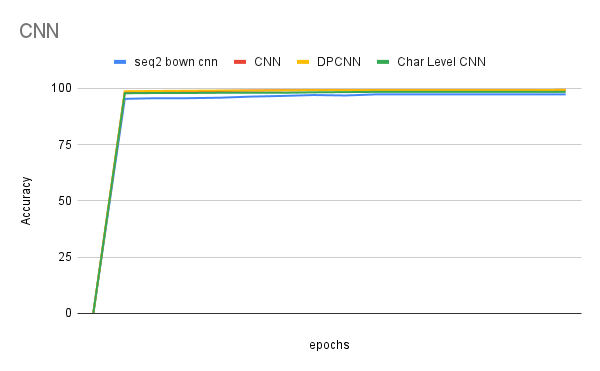
****

Figure 17: Accuracy vs epochs on DBpedia dataset of CNNs

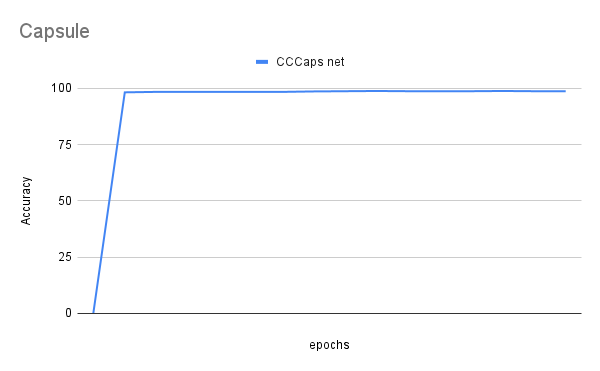
****

Figure 18: Accuracy vs epochs on DBpedia dataset of Capsule

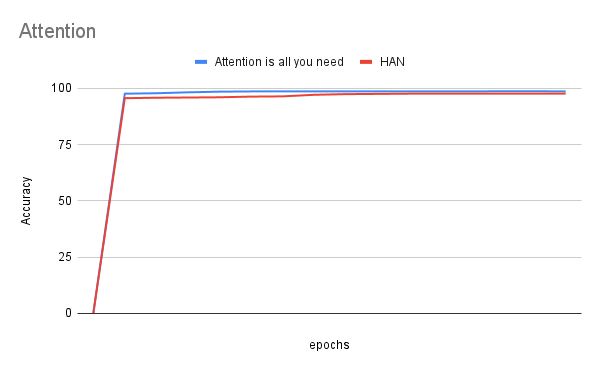
****

Figure 19: Accuracy vs epochs on DBpedia dataset of Attention Mechanisms

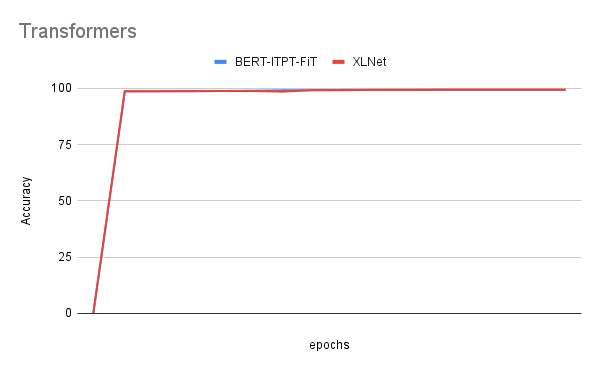
****

Figure 20: Accuracy vs epochs on DBpedia dataset of Transformers

### Classification Accuracy vs Number of Epochs for IMDb:

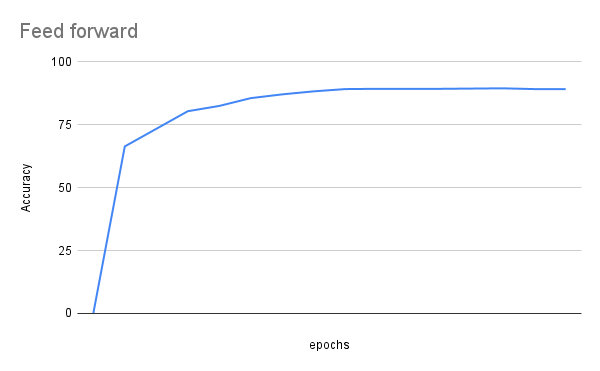
****

Figure 21: Accuracy vs epochs on IMDb dataset of Feed Forward Neural Networks

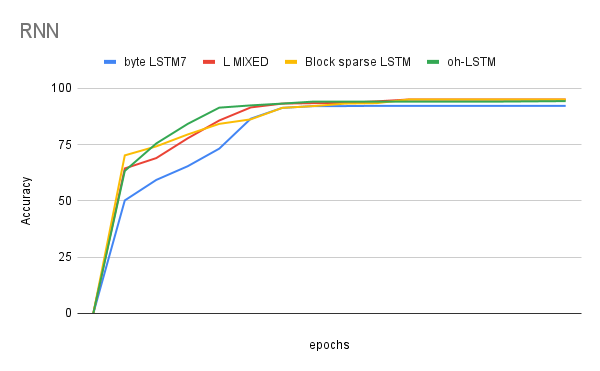
****

Figure 22: Accuracy vs epochs on IMDb dataset of RNN

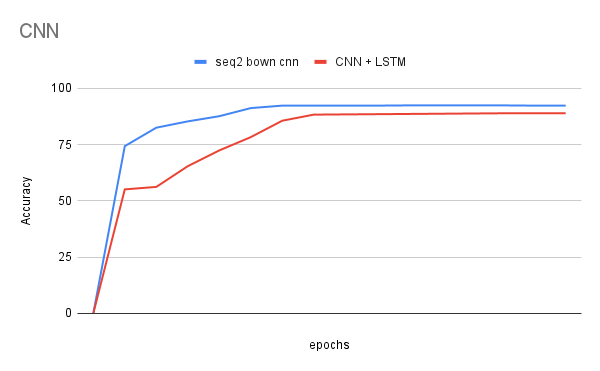
****

Figure 23: Accuracy vs epochs on IMDb dataset of CNNs

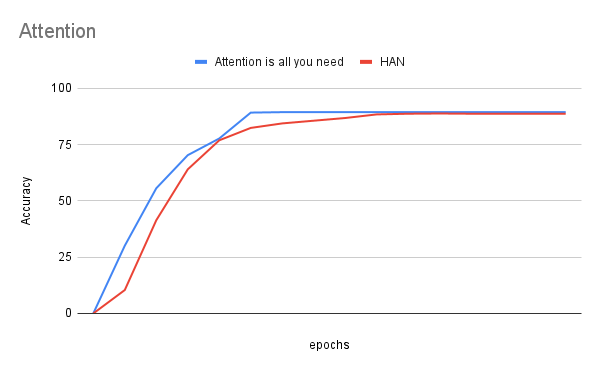
****

Figure 24: Accuracy vs epochs on IMDb dataset of Attention Mechanisms

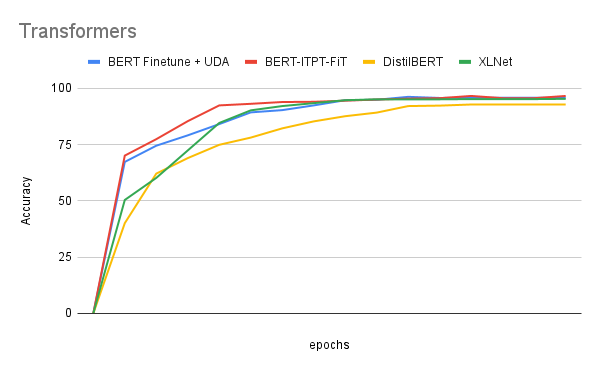
****

Figure 25: Accuracy vs epochs on IMDb dataset of Transformers

## Accuracy

Accuracy was tested after each epoch with the remaining testing data, throughout each run it was noticed that each method responded differently overtime with respect to accuracy based on how larger the training data was, this was especially certain in DBpedia that has the most samples, hence the accuracies in that dataset are some of the highest we have seen.

### AG News

|  |  |  |
| --- | --- | --- |
| **Category** | **Method** | **Accuracy original** |
| Feed forward | fasttext | 92.21 |
| RNN | L MIXED | 95.03 |
|  | DRNN | 94.46 |
| CNN | seq2 | 90.33 |
|  | CNN | 93.45 |
| Capsule | Capsule B | 92.62 |
|  | CCCaps net | 92.38 |
| Attention | Attention is all you need | 92.64 |
|  | HAN | 92.02 |
| Transformers | BERT-ITPT-FiT | 95.01 |
|  | XLNet | 95.65 |

Table 1: Accuracy of each text classification model on the AG-News dataset after 15 epochs of training

### DBPedia

|  |  |  |
| --- | --- | --- |
| **Category** | **Method** | **Accuracy original** |
| Feed forward | fasttext | 98.13 |
| RNN | L MIXED | 99.27 |
|  | DRNN | 99.18 |
| CNN | seq2 | 97.26 |
|  | CNN | 99.19 |
| Capsule | Capsule B | 98.22 |
|  | CCCaps net | 98.73 |
| Attention | Attention is all you need | 98.68 |
|  | HAN | 97.57 |
| Transformers | BERT-ITPT-FiT | 99.33 |
|  | XLNet | 99.41 |

Table 2: Accuracy of each text classification model on the DBpedia dataset after 15 epochs of training

### IMDb

|  |  |  |
| --- | --- | --- |
| **Category** | **Method** | **Accuracy original** |
| Feed forward | fasttext | 89.29 |
| RNN | L MIXED | 95.53 |
|  | DRNN | 93.78 |
| CNN | seq2 | 92.35 |
|  | CNN | 88.89 |
| Capsule | Capsule B | 89.96 |
|  | CCCaps net | 89.15 |
| Attention | Attention is all you need | 89.42 |
|  | HAN | 88.77 |
| Transformers | BERT-ITPT-FiT | 95.65 |
|  | XLNet | 96.23 |

Table 3: Accuracy of each text classification model on the IMDb dataset after 15 epochs of training

## Proposed New Metric

### AG News

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Method** | **Time** | **Normalized** | **1 - Normalized** | **Accuracy original** | **New Score** |
| Feed forward | fasttext | 12 | 0 | 1 | 92.21 | 92.21 |
| RNN | L MIXED | 746 | 0.5 | 0.5 | 95.03 | 47.52 |
|  | DRNN | 664 | 0.444 | 0.556 | 94.46 | 52.52 |
| CNN | seq2 | 232 | 0.15 | 0.85 | 90.33 | 76.78 |
|  | CNN | 152 | 0.095 | 0.905 | 93.45 | 84.57 |
| Capsule | Capsule B | 358 | 0.236 | 0.764 | 92.62 | 70.76 |
|  | CCCaps net | 345 | 0.227 | 0.773 | 92.38 | 71.41 |
| Attention | Attention - IAYN | 93 | 0.055 | 0.945 | 92.64 | 87.54 |
|  | HAN | 150 | 0.094 | 0.906 | 92.02 | 83.37 |
| Transformers | BERT-ITPT-FiT | 516 | 0.343 | 0.657 | 95.01 | 62.42 |
|  | XLNet | 539 | 0.359 | 0.641 | 95.65 | 61.31 |

Table 4:Relative performance score for AG-News dataset

### DBPedia

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Method** | **Time** | **Normalized** | **1 - Normalized** | **Accuracy original** | **New Score** |
| Feed forward | fasttext | 44 | 0 | 1 | 98.13 | 98.13 |
| RNN | L MIXED | 2660 | 0.5 | 0.5 | 99.27 | 49.63 |
|  | DRNN | 2441 | 0.458 | 0.542 | 99.18 | 53.76 |
| CNN | seq2 | 779 | 0.14 | 0.86 | 97.26 | 83.64 |
|  | CNN | 526 | 0.092 | 0.908 | 99.19 | 90.06 |
| Capsule | Capsule B | 1359 | 0.251 | 0.749 | 98.22 | 73.57 |
|  | CCCaps net | 1178 | 0.217 | 0.783 | 98.73 | 77.31 |
| Attention | Attention - IAYN | 315 | 0.052 | 0.948 | 98.68 | 93.55 |
|  | HAN | 541 | 0.095 | 0.905 | 97.57 | 88.3 |
| Transformers | BERT-ITPT-FiT | 1872 | 0.349 | 0.651 | 99.33 | 64.66 |
|  | XLNet | 1968 | 0.368 | 0.632 | 99.41 | 62.83 |

Table 5: Relative performance score for DBPedia dataset

### IMDb

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Method** | **Time Taken (minutes)** | **Normalized** | **1 - Normalized** | **Accuracy original** | **New Score** |
| Feed forward | fasttext | 7 | 0 | 1 | 89.29 | 89.29 |
| RNN | L MIXED | 427 | 0.5 | 0.5 | 95.53 | 47.77 |
|  | DRNN | 386 | 0.451 | 0.549 | 93.78 | 51.49 |
| CNN | seq2 | 123 | 0.138 | 0.862 | 92.35 | 79.61 |
|  | CNN | 86 | 0.094 | 0.906 | 88.89 | 80.53 |
| Capsule | Capsule B | 222 | 0.256 | 0.744 | 89.96 | 66.93 |
|  | CCCaps net | 190 | 0.218 | 0.782 | 89.15 | 69.72 |
| Attention | Attention is all you need | 52 | 0.054 | 0.946 | 89.42 | 84.59 |
|  | HAN | 87 | 0.095 | 0.905 | 88.77 | 80.34 |
| Transformers | BERT-ITPT-FiT | 296 | 0.344 | 0.656 | 95.65 | 62.75 |
|  | XLNet | 304 | 0.354 | 0.646 | 96.23 | 62.16 |

Table 6: Relative performance score for IMDb dataset

# Observations

Using our new metric, we observe varying scores for methods with complex networks that take a long time, in huge proportions. This normalized time tells us how much the extra time taken influences the accuracy, and we observe that this change in accuracy is minimal compared to the massive difference in time taken. Hence, using our new metric, we can numerically point to how much the time taken actually affects result and whether this result is worth it. Comparing accuracy without comparing time taken was futile since accuracy does not give the complete picture, and our metric is able to help us directly compare these methods on a level playing field.

# Discussions

Using works that focus on specific methods, this paper is able to look all of them together and compare them for multiple situations. Listing them out gives ideal situations where they may be applied, and also gives us situations where applying these methods can give significant harms and perform bad. The comparative analysis shows the extent to how much the improvements in each subsequent method actually give a boost and whether this improvement is significant enough. Using multiple databases for popular websites, varying results are achieved and no particular method is ideal for every possible situation.

There are new advancements being made in the field of deep learning every day, which makes it tough to update every feasible technique but this paper tracks all popular current procedures. This research can also be extended into various other fields of text classification, and provide information relevant to each industry separately.

# Conclusion

Deep learning provides several advancements in text classification, and provides overlapping methods too. Processes that are reserved for image classification and the like can actually be used for text classification too, and give interesting results. Looking at each method’s algorithms, it is clear that different methods suit different situations, and methods that are supposed to give better performance don’t necessarily have to be used, and the cost of using them sometimes is more than the benefit they provide. It also gives us how multiple techniques are used in tandem, and how each model of working leads into the next by fixing shortcomings and giving us ways to solve them. However, the comparative table shows us how simpler methods sometimes give much better results than complex and deep architectures. It throws their superiority into question. It also shows us that many methods that perform better with some datasets actually work worse on others. This shows us that it is tough to predict which method will work best, even if the desired outcome and function that we need is the same. Therefore, it is useful to use multiple techniques in rotation and extensive testing should be done according to the need.

Using our new metric, we’re able to directly compare these varying methods evenly, as the time taken greatly influences the accuracy, and their results cannot directly be compared since the time taken is a huge factor to consider when comparing feasibility of a method in real life situations.

# Areas of Improvement

Simple improvements can be made to the testing methodology to achieve a more reliable metric score, the improvements are listed below:

1. **Testing more models, and testing against more datasets:** More models would allow there to be a more cohesive and unified database of all implementable models, and more datasets would grant us the ability to see how versatile a model is based on what the dataset is.
2. **Increasing epochs:** higher epochs would take more time and *ideally* increase accuracy, the two variables our metric relies upon, getting higher quality data would make the metric even more reliable.
3. **Confidence Testing:** running multiple samples of training to get multiple data points for time and accuracy, then doing a simple 5% or 15% confidence test would eliminate any outliers and give us a statistically accurate metric, this although has the drawback of increasing time taken by tenfold.
4. **Expanding the metric:** The metric can be expanded to take into account more variables like the TFlops of the processor or epochs, it is true that these do not scale predictably, but further research into a standardized metric would do wonders for the ease of choosing and implementing a function text classification algorithm, be it for a large scale deployment or a small project.

# References

[1] Le-Hong, Phuong, and Anh-Cuong Le. "A comparative study of neural network models for sentence classification." 2018 5th NAFOSTED Conference on Information and Computer Science (NICS). IEEE, 2018.

[2] Neethu, K. S., T. S. Jyothis, and Jithin Dev. "Text classification using km-elm classifier." 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT). IEEE, 2016.

[3] Li, Dan, and Jiang Qian. "Text sentiment analysis based on long short-term memory." 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI). IEEE, 2016.

[4] Hassan, Abdalraouf, and Ausif Mahmood. "Convolutional recurrent deep learning model for sentence classification." Ieee Access 6 (2018): 13949-13957.

[5] Zeng, Shuifei, et al. "Term-based pooling in convolutional neural networks for text classification." China Communications 17.4 (2020): 109-124.

[6] Yenter, Alec, and Abhishek Verma. "Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis." 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON). IEEE, 2017.

[7] Kim, Jaeyoung, et al. "Text classification using capsules." Neurocomputing 376 (2020): 214-221.

[8] Wu, Yujia, et al. "Siamese capsule networks with global and local features for text classification." Neurocomputing 390 (2020): 88-98.

[9] Zhan, Zhiqiang, et al. "Knowledge attention sandwich neural network for text classification." Neurocomputing 406 (2020): 1-11.

[10] Huang, Wenti, et al. "Relation classification via knowledge graph enhanced transformer encoder." Knowledge-Based Systems 206 (2020): 106321.

[11] Zulqarnain, Muhammad & Ghazali, Rozaida & Mazwin, Yana & Rehan, Muhammad. (2020). A comparative review on deep learning models for text classification. 10.11591/ijeecs.v19.i1.pp325-335.

[12] J SreeDevi, M Rama Bai, M Chandrashekar Reddy. (2020). Newspaper Article Classification using Machine Learning Techniques. 10.35940/ijitee.E2753.039520

[13] Wang, Dan & Gong, Jibing & Song, Yaxi. (2019). W-RNN: News text classification based on a Weighted RNN.

[14] S. Wang and C. D. Manning, “Baselines and bigrams: Simple, good sentiment and topic classification,” in Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2. Association for Computational Linguistics, 2012, pp. 90–94.

[15] Sebastiani, Fabrizio. (2001). Machine Learning in Automated Text Categorization. ACM Computing Surveys. 34. 1-47. 10.1145/505282.505283.

[16] Galán, Carlos & Torres, Jesús & Aguilar, Rosa & Diaz, Sergio. (2018). Using Deep Learning to Predict Sentiments: Case Study in Tourism. Complexity. 2018. 9. 10.1155/2018/7408431.

[17] Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification (cite arxiv:1607.01759)

[18] Miyato, T., Dai, A. M. & Goodfellow, I. (2016). Adversarial Training Methods for Semi-Supervised Text Classification (cite arxiv:1605.07725Comment: Published as a conference paper at ICLR 2017)

[19] Johnson, Rie & Zhang, Tong. (2016). Supervised and Semi-Supervised Text Categorization using One-Hot LSTM for Region Embeddings. Supervised and Semi-Supervised Text Categorization Using One-Hot LSTM for Region Embeddings..

[20] EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference. Association for Computational Linguistics, 2020. p. 445-455 (EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference).