VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

J- COMPONENT (REVIEW 1)

Information Security Analysis and Audit

TITLE-DATA SECURE SMART HOME AUTOMATION SYSTEM

PRESENTED BY:

Satrunjay Kumar – 18BIT0040

SLOT – **G1**

TO:

SUMAIYA THASEEN I

OBJECTIVES

1. Create Deep speech brain using deep learning
2. Create task brain using deep learning
3. Frame a secure encryption pathway towards main server using homomorphic encryption.
4. Handle bot attacks

MY OBJECTIVE

Create task brain using deep learning (CNN)

INTRODUCTION

The Objective of task brain is to successfully handle the encrypted message given to the server and to predict what to do based on the received message or we can also say that the use of task brain is to classify the messages for certain actions which we want to perform and since we are given messages as texts so we only need to classify these text into certain tasks.

LITERATURE SURVEY

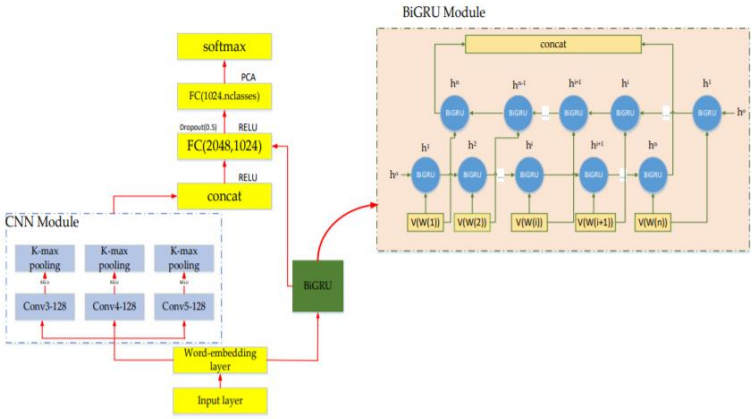
1. **Feature Fusion Text Classification Model Combining CNN and BiGRU with Multi-Attention Mechanism**
   →

   The feature fusion model is divided into a multiple attention (MATT) CNN model and a bi-directional gated recurrent unit (BiGRU) model. The CNN model inputs the word vector that has been labeled. Obtaining the influence intensity of the target keyword on the sentiment polarity of the sentence, and forming the first dimension of the sentiment classification, the BiGRU model replaces the original BiLSTM and extracts the global semantic features of the sentence level to form the second dimension of sentiment classification. Then, using PCA to reduce the dimension of the two-dimensional fusion vector, finally result is obtained by combining two dimensions of keywords and sentences.

Motivation

Although CNN has made great breakthroughs in the field of text categorization, it pays more attention to local features and ignores the contextual meaning of words, thus affecting the accuracy of classification. Therefore, improvements need to be done.

Architecture



Dataset used

To verify the validity of the model proposed in this paper, the experiment used movie review data (MRD) created by Cornell University's film evaluation data and adopted SemEval2016 datasets. Among them, the MRD consists of movie review data, with a positive attitude review of about 1000 articles, a negative attitude review of 1000 articles, a label of five-character sentences of 5331 sentences, and a sentence with 5000 subject sentences. During the experiment, 1000 sentences were randomly selected as the training set and 400 were used as the test set. SemEval2016 was the dataset of the semantic evaluation game task 4, which contains user reviews in the fields of laptop and restaurant, and the emotional polarity of the data samples was divided into positive, negative and neutral.

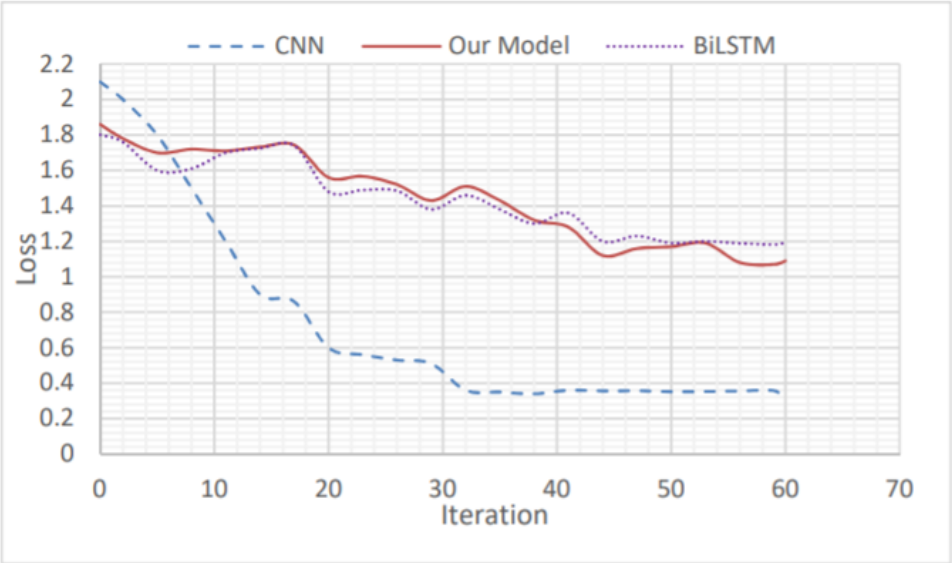


Figure: loss function comparison

Tabular Comparison on Acuuracy on MRD and SemEval2016 dataset

Model	Laptop	Restaurant	MRD
SVM	65.17	74.18	70.13
CNN	65.23	69.90	68.43
AM-CNN	65.42	77.67	74.32
CNN+BiLSTM	63.20	**79.54**	73.28
ATT-LSTM	68.22	75.30	67.23
BiLSTM-ATT-G	73.34	79.12	69.89
IAN	**73.24**	77.40	78.19
MATT-CNN+BiGRU	**74.21**	78.47	**79.22**

2. A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis

→

This hybrid model uses LSTM and very deep CNN model named as Hybrid CNN-LSTM Model to overcome the sentiment analysis problem. First, they used Word to Vector (Word2Vc) approach to train initial word embeddings. The Word2Vc translates the text strings into a vector of numeric values, computes distance between words, and makes groups of similar words based on their meanings. Afterword embedding is performed in which the

proposed model combines set of features that are extracted by convolution and global max-pooling layers with long term dependencies.

Motivation

CNN helps to learn how to extract features from the data. However, it also requires many convolution layers to captures the long-term dependencies, capturing dependencies becomes worse with the increase of input sequence of length in a neural network. Basically, it leads towards a very deep layer of convolution neural networks.

Architecture

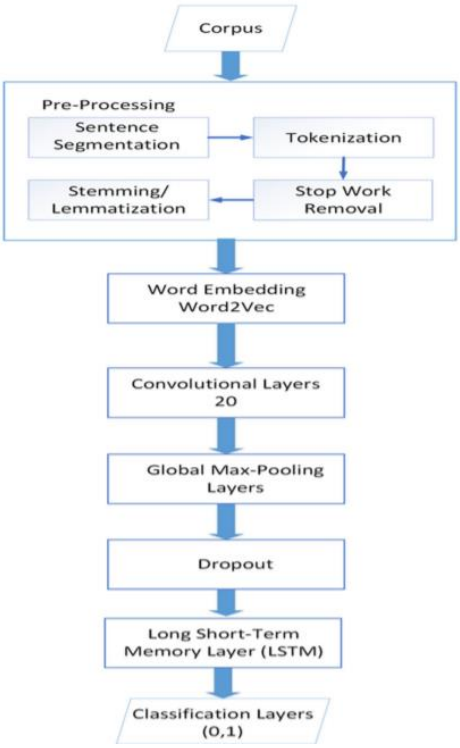


Figure: Hybrid CNN-LSTM Architecture

Dataset used

They tested proposed Hybrid CNN-LSTM Model on two datasets IMDB and Amazon.

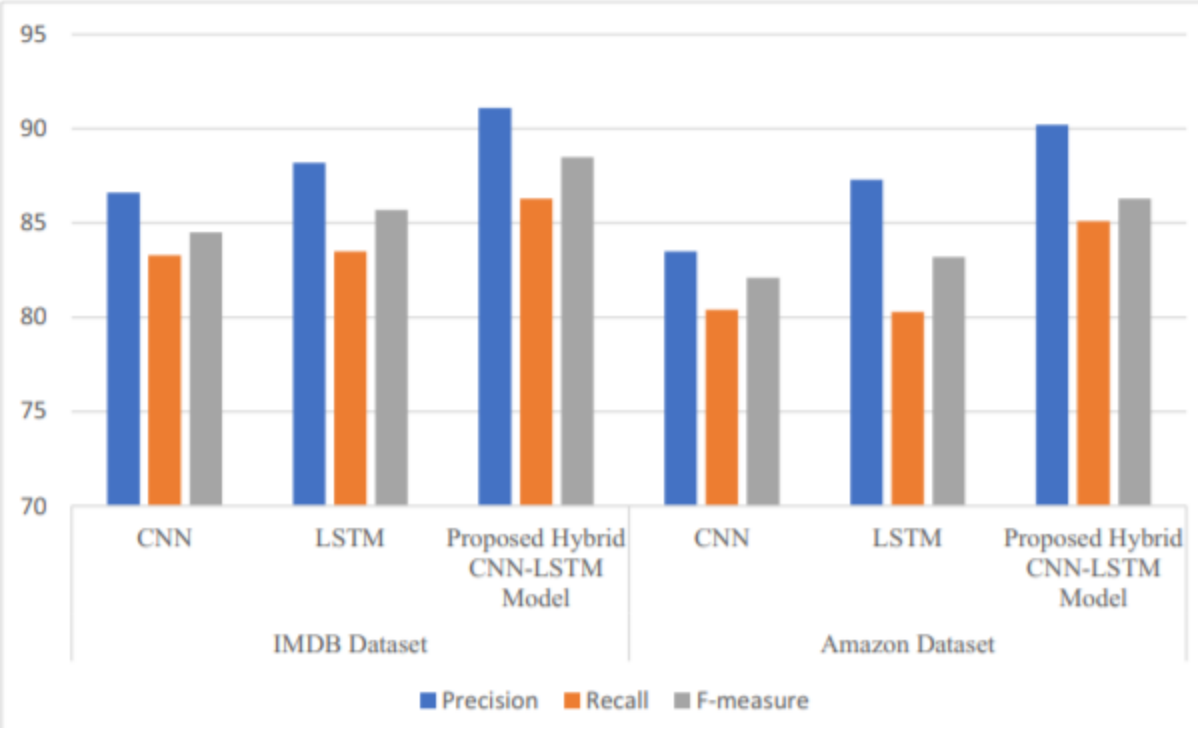


Fig: Comparison of Proposed Hybrid CNN-LSTM model with CNN and LSTM w.r.t Precision, Recall and F-measure on IMDB and Amazon Movie Reviews Dataset

Tabular comparison

Comparison on IMDB dataset

Models	Precision	Recall	F-measure
CNN	86.2	82.8	84,3
LSTM	87.7	82.7	86.0
Hybrid CNN-LSTM	91.3	86.2	87.8

Comparison on Amazon dataset

Models	Precision	Recall	F-measure
CNN	82.8	80.9	82.3
LSTM	87.2	80.9	82.7
Hybrid CNN-LSTM	90.1	85.0	86.8

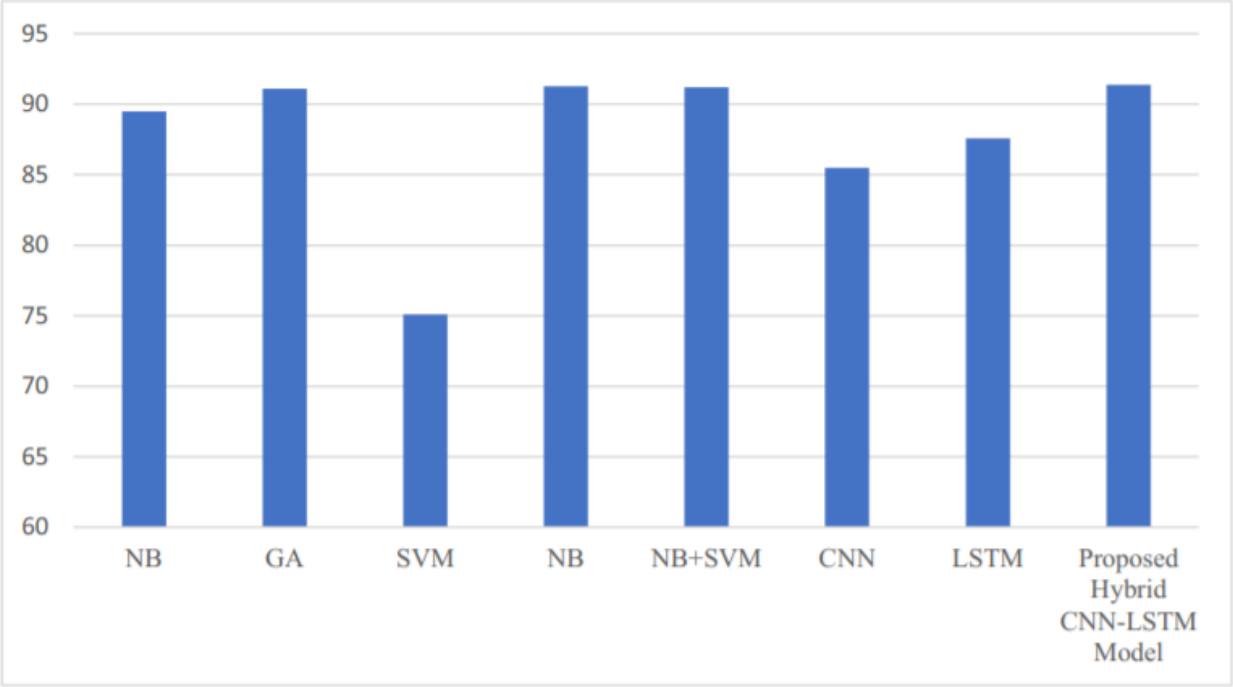


Figure: Accuracy comparison with traditional models on IMDB dataset.

Tabular comparison of Accuracy

Models	Accuracy (IMDB dataset)
NB	**89.2**
GA	**91.0**
SVM	**75.0**
NB	**91.1**
NB+SVM	**91.1**
CNN	**85.2**
LSTM	**87.0**
Hybrid CNN-LSTM	**91.3**

3. Design and Investigation of Capsule Networks for Sentence Classification

. A capsule is a group of neurons whose activity vector represents different attributes of a specific type of entity such as an object or an object part. The vector's length represents the probability that the entity exists, and the orientation of the vector represents the attributes of the entity.

Here, capsules are adapted to represent a sentence or document as a vector. The model consists of four layers: (1) an n-gram convolutional layer, (2) a capsule layer, (3) a convolutional capsule layer, and (4) a fully connected capsule layer.

Motivation

CNNs classify images or texts by using successive layers of convolutions and pooling. Although pooling operations identify salient features and reduce the computational complexity of convolution operations, they lose information regarding spatial relationships and are likely to mis-classify entities based on their orientation or proportion.

Architecture

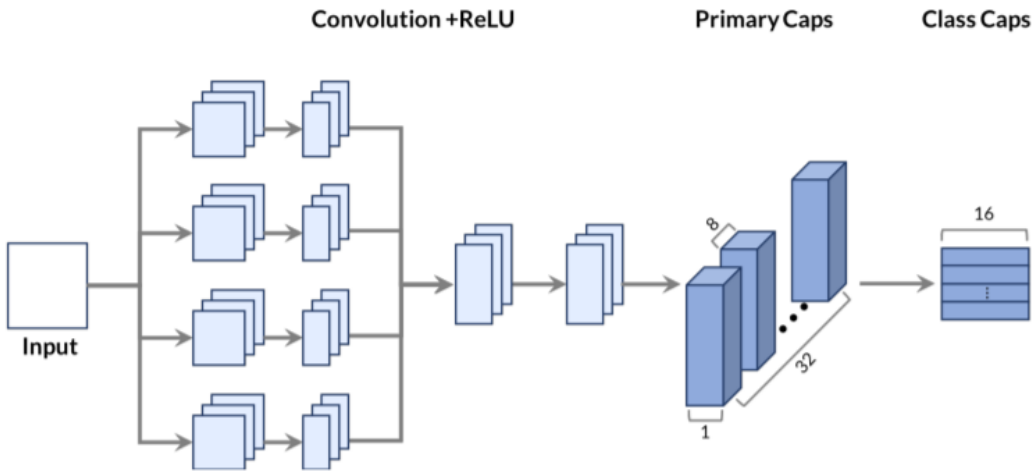


Figure -. Proposed CapsNet architecture for sentence classification.

Dataset used

The datasets used in the experiment are the following:

• MR Dataset: movie review dataset (positive/negative review)

• TREC Dataset: 6 class Text REtrieval Conference dataset (Entity, Human, Location, · · ·).

• SST-2 Dataset: modified version of the MR dataset provided on a phrase and sentence level.

• Subj Dataset: Subjectivity dataset (subjective/objective sentence)

• MPQA Dataset: contains mostly short news articles from a variety of sources manually annotated for opinion polarity

Table 1. Details of datasets used in the experiments.

Dataset	No. of Class	Dataset Size	Max. Sent. Length	Vocab. Size
MR	2	10662	56	18758
TREC	6	5952	37	8760
SST-2	2	9613	53	16185
Subj	2	10000	120	21323
MPQA	2	10606	36	6236

Tabular comparison

Table 6. Sample sentences correctly predicted by CapsNet and Incorrectly predicted by CNN.

Prediction Confidence			Sample Sentence
CapsNet	LSTM	CNN	
0.91555923 (Negative)	0.996865 (Negative)	0.9800223 (Positive)	depicts the sorriest and most sordid of human behavior on the screen, then laughs at how clever it's being
0.84724176 (Positive)	0.5029957 (Positive)	0.99953365 (Negative)	rarely, a movie is more than a movie go
0.9158743 (Positive)	0.9931028 (Positive)	0.943629 (Negative)	a sharp satire of desperation and cinematic deception
0.9007900 (Positive)	0.9957867 (Negative)	0.8162323 (Negative)	a painfully funny ode to bad behavior
0.8918062 (Negative)	0.99717844 (Negative)	0.9997436 (Positive)	pretty much sucks, but has a funny moment or two

Table 7. Sample sentences incorrectly predicted by CapsNet but correctly predicted by CNN.

Prediction Confidence			Sample Sentence
CapsNet	LSTM	CNN	
0.695352 (Positive)	0.692398 (Positive)	0.999564 (Negative)	a stuporously solemn film
0.842644156 (Negative)	0.99852017 (Negative)	0.78954309 (Positive)	though everything might be literate and smart, it never took off and always seemed static
0.833856 (Negative)	0.998444 (Negative)	0.9367719 (Positive)	Elvira fans could hardly ask for more
0.719898 (Positive)	0.993055 (Positive)	0.999979 (Negative)	there might be some sort of credible gender provoking philosophy submerged here, but who the hell cares?

Comparison: Accuracy of our CapsNet model and other models on different datasets.

Network	MR	TREC	SST-2	Subj	MPQA
CNN	79.8	91.6	84.7	93.3	90.2
LSTM	81.6	94.5	86.5	93.5	90.0
CapsNet	83.8	96.0	86.2	94.2	91.1
2-phase CapsNet	**84.53**	**96.46**	86.4	**95.1**	**91.32**
CNN-static [1]	81.0	92.8	86.8	93.0	89.6
CNN-non-static [1]	81.5	93.6	87.2	93.4	89.5
CNN-multichannel [1]	81.1	92.2	88.1	93.2	89.4
Capsule-B [4]	82.3	92.8	86.8	93.8	-
Capsule-static [5]	81.0	94.8	-	-	90.6
MCapsNet [19]	83.5	94.2	88.6	94.5	-
SVMS [8]	-	95.0	-	-	-
DCNN [11]	-	93.0	86.8	-	-
LR-Bi-LSTM [28]	82.1	-	-	-	-
CCAE [9]	77.8	-	-	-	87.2
Bi-LSTM+SWN-Lex [29]	-	-	89.2	-	-
BLSTM-2DCNN [18]	82.3	96.1	**89.5**	94.0	-

4. Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN

→ Peng et al. proposed a graph-CNN based deep learning model to first convert text to graph-of-words, and then use graph convolution operations to convolve the word graph. They showed through experiments that the graph-of-words representation of texts has the advantage of capturing non-consecutive and long-distance semantics, and CNN models have the advantage of learning different level of semantics.

Motivation

Traditional approaches simply use bag-of-words and have achieved good results. However, when there are a lot of labels with different topical granularities, bagof-words representation may not be enough.

Architecture

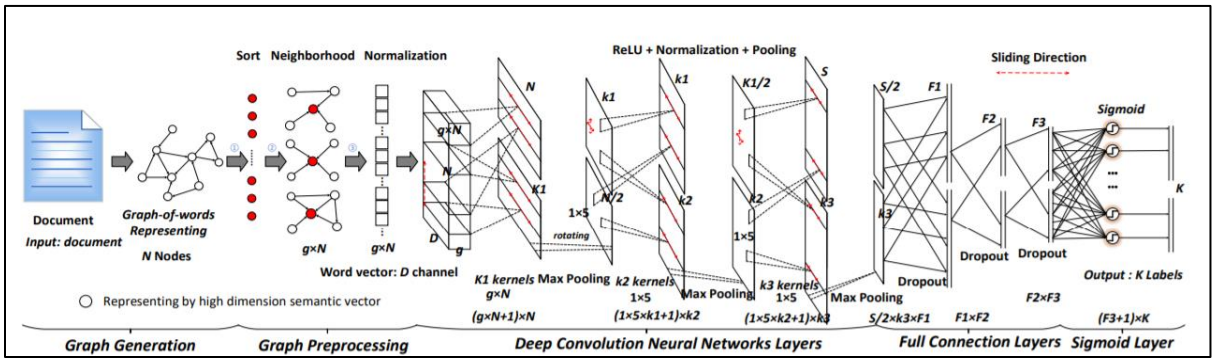


Figure : A typical configuration of Deep Graph-CNN for text classification.

Dataset used

The two datasets we used are RCV1 and NYTimes datasets.

• **RCV1** : RCV1 dataset is a manually labeled newswire collection of Reuters News from 1996-1997. The news documents are categorized with respect to three controlled vocabularies: industries, topics, and regions. They used the topic-based hierarchical classification as it has been most popular in evaluation. There are 103 categories including all classes except for root in the hierarchy.

• **NYTimes** : This corpus contains nearly every article published in the New York Times between January 01, 1987 and June 19th, 2007. As a large scale corpus, NYTimes was widely used in document routing, document categorization, entity extraction, cross document coreference resolution, and information retrieval, etc. They used the standard of taxonomic classifier labeled in this corpus to test large-scale hierarchical text classification.

Models	#Classes	Macro-F_1	Micro-F_1	Classes	Macro-F_1	Micro-F_1
LR	103	0.3281	0.6921	137	0.5339*	0.8008*
SVM	103	0.3295	0.6905	137	0.5472*	0.8082*
HSVM	103	0.3329	0.6932	137	–	–
TD-SVM	103	0.3368	0.6964	137	0.3415*	0.7134*
HR-LR	103	0.3217	0.7159	137	0.5581*	0.8123*
HR-SVM	103	0.3858	0.7275	137	0.5656*	0.8166*
HLSTM	103	0.3102	0.6726	137	0.3201	0.7019
HAN	103	0.3268	0.6964	137	0.3411	0.7211
RCNN	103	0.2931	0.6859	137	0.3219	0.6952
XML-CNN	103	0.3007	0.6955	137	0.3106	0.7149
DCNN-3	103	0.3987	0.7323	137	0.5843	0.8169
DCNN-6	103	0.3479	0.7158	137	0.5013	0.8072
DGCNN-1	103	0.3631	0.7418	137	0.5495	0.8168
DGCNN-3	103	0.4322	0.7611	137	0.6182	0.8175
DGCNN-6	103	0.3905	0.7404	137	0.5637	0.8149
HR-DGCNN-1	103	0.3682	0.7481	137	0.5649	0.8166
HR-DGCNN-3	103	**0.4334**	**0.7618**	137	**0.6586**	**0.8255**
HR-DGCNN-6	103	0.3992	0.7489	137	0.5623	0.8142

Figure: Comparison of results on RCV1 dataset.

5. Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function

→

In this paper, Devendra do a careful study of a bidirectional LSTM network for the task of text classification using both supervised and semi-supervised approaches. He implemented LSTM with cross entropy loss and is able to achieve competitive results compared with the more complex models. s. Furthermore, in addition to cross-entropy loss, by using a combination of entropy minimization, adversarial, and virtual adversarial losses for both labelled and unlabelled data, he reported new state-of-the-art results for text classification task on four benchmark datasets.

Motivation

It has been reported in previous works that it is difficult to optimize the LSTM network for text classification tasks and therefore in order to perform well, its parameters required pretraining with either a language model or a sequence auto-encoder [2]. A disadvantage of this approach is that it

can take a long time to train a language model or a sequence auto-encoder and thus this additional step may not be always feasible.

Architecture

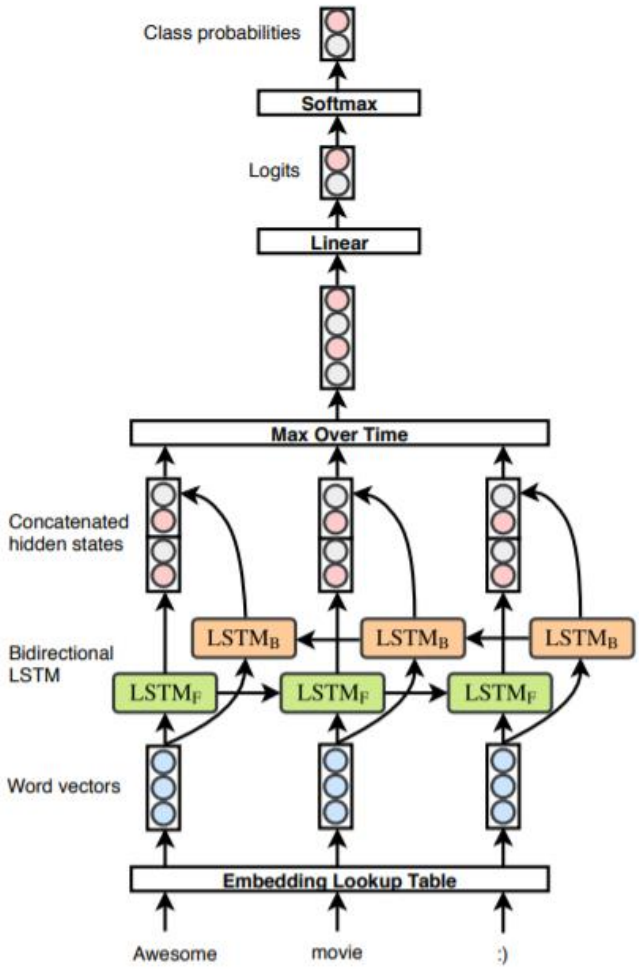


Figure: BiLSTM : Text classification model architecture.

Dataset used

In this work, they experimented with four datasets that are summarized in Table 1. ACL-IMDB [15] and Elec [8] datasets are for sentiment classification of movie reviews and Amazon product reviews [16] respectively while AG-News and DBpedia are for topic classification of news articles and Wikipedia articles respectively.

Dataset	Training	Test	K	ℓ
ACL-IMDB	25K	25k	2	268
Elec	25K	25K	2	125
AG-News	120K	7.6K	4	46
DBpedia	560K	70K	14	56

Table 1: Dataset summary statistics. K: Number of classes. ℓ: Average length of a document in the training set.

Method	ACL-IMDB
ML Supervised Baseline	93.43
+ VAT (L)	94.25
+ VAT (L + U)	94.68
+ VAT (L + U) + EM	**95.53**
+ VAT (L + U) + Word Dropout	95.26
+ VAT (L + 4×U) + EM	94.63

Table 6: Test accuracy on ACL-IMDB dataset when trained using mixed objective function (L_{MIXED}) on variants of our best BiLSTM model. L: labeled data. U: unlabeled data, 4×U implies that the size of unlabeled data was four times to that of labeled data.

Method	ACL-IMDB
Baseline	93.43
+Word Dropout	**93.57**
Random Embeddings	92.36
Static Embedding	91.83
5K words per batch	**93.50**
1K words per batch	93.02
256-D hidden size	93.33
64 batch size	93.14
32 batch size	**93.60**
Default Adam parameters	93.37
30K Vocab	92.22
300 max length	93.31
gradient norm (5.0)	92.99

Table 5: Test accuracies on ACL-IMDB dataset for variants of our best BiLSTM model trained using maximum likelihood objective (L_{ML}).

Tabular comparison

Models	ACL-IMDB	Elec	AG-News	DBpedia
Semi-supervised CNN	7.67	7.14	6.88	0.88
CL-CNN	-	-	9.51	1.55
Proposed Model	**6.40**	**7.40**	**5.62**	**0.91**

Table:Test set error rates when the model is trained using maximum likelihood objective

Models	ACL-IMDB	Elec	AG-News	DBpedia
Semi-supervised CNN	7.67	7.14	6.88	0.88
CL-CNN	5.94	5.55	6.57	0.84
Adversial semi supervised	5.91	5.40	-	0.76
Semi supervised sequence learning	5.91	-	-	1.19
Proposed method	**4.32**	**5.24**	**4.95**	**0.70**

Table: Test set error rates when the model is trained using mixed objective function

Method	ACL-IMDB
Baseline	93.43
+Word Dropout	**93.57**
Random Embeddings	92.36
Static Embedding	91.83
5K words per batch	**93.50**
1K words per batch	93.02
256-D hidden size	93.33
64 batch size	93.14
32 batch size	**93.60**
Default Adam parameters	93.37
30K Vocab	92.22
300 max length	93.31
gradient norm (5.0)	92.99

Table : Test accuracies on ACL-IMDB dataset for variants of the best BiLSTM model trained using maximum likelihood objective

6. CryptoNet Applying Neural Networks to Encrypted Data with High Throughput and Accuracy

Dowlin et al. focus on using only fully homomorphic encryption - which is a new cryptographic technique that allows one to evaluate functions on encrypted data without decryption or access to the secret key - to evaluate Convolutional Neural Networks (CNNs) on encrypted data. Dowlin et al. were the first to propose using FHE to achieve privacy-preserving DL, offering a framework to design NNs that can be run on encrypted data. They proposed using polynomial approximations of the most widespread ReLU activation function and using pooling layers only during the training phase to reduce the circuit depth of their neural network.

Motivation

Allows evaluation of functions on encrypted data without decrypting or access to secret key.

Architecture

1. *Convolution Layer*: The input image is 28×28. The convolution has windows, or kernels, of size 5×5, a stride of $(2, 2)$, and a mapcount of 5. The output of this layer is therefore $5 \times 13 \times 13$.
2. *Square Activation Layer*: This layer squares the value at each input node.
3. *Scaled Mean Pool Layer*: This layer has $1 \times 3 \times 3$ windows, and again outputs a multi-array of dimension $5 \times 13 \times 13$.
4. *Convolution Layer*: This convolution has a kernel size of $1 \times 5 \times 5$, a stride of $(1, 2, 2)$, and a mapcount of 10. The output layer is therefore $50 \times 5 \times 5$.
5. *Scaled Mean Pool Layer*: As with the first mean pool, the kernel size is $1 \times 3 \times 3$, and the output is $50 \times 5 \times 5$.
6. *Fully Connected Layer*: This layer fully connects the incoming $50 \cdot 5 \cdot 5 = 1250$ nodes to the outgoing 100 nodes, or equivalently, is multiplication by a 100×1250 matrix.
7. *Square Activation Layer*: This layer squares the value at each input node.
8. *Fully Connected Layer*: This layer fully connects the incoming 100 nodes to the outgoing 10 nodes, or equivalently, is multiplication by a 10×100 matrix.
9. *Sigmoid Activation Function*: This layer applies the sigmoid function to each of the 10 incoming values.

Dataset used

This model is tested in the MNIST dataset. This dataset consists of 60,000 images of hand written digits. Each image is a 28x28 pixel array, where each pixel is represented by its gray level in the range of 0-255. They used the training part of this dataset, consisting of 50,000 images, to train a network and the remaining 10,000 images for testing. The accuracy of the training network is 99% (it mislabels only 105 out of the 10, 000 test examples).

Table 1. Breakdown of the time it takes to apply CryptoNets to the MNIST network

Layer	Description	Time to compute
Convolution layer	Weighted sums layer with windows of size 5×5, stride size of 2. From each window, 5 different maps are computed and a padding is added to the upper side and left side of each image.	30 seconds
1^{st} square layer	Squares each of the 835 outputs of the convolution layer	81 seconds
Pool layer	Weighted sum layer that generates 100 outputs from the 835 outputs of the 1^{st} square layer	127 seconds
2^{nd} square layer	Squares each of the 100 outputs of the pool layer	10 seconds
Output layer	Weighted sum that generates 10 outputs (corresponding to the 10 digits) from the 100 outputs of the 2^{nd} square layer	1.6 seconds

Tabular comparison

Table 4: MNIST HCNN vs CryptoNets [14] complexity for homomorphic inference

Layer	MNIST HCNN				CryptoNets			
	Input Neurons	Output Neurons	# of Multiplications		Input Neurons	Output Neurons	# of Multiplications	
			HMultPlain	HMult			HMultPlain	HMult
1	28×28 = 784	5×12×12 = 720	25×720 = 18,000	-	29×29 = 841	5×13×13 = 845	25×845 = 21,125	-
2	5×12×12 = 720	5×12×12 = 720	-	720	5×13×13 = 845	5×13×13 = 845	-	845
3	5×12×12 = 720	4×4×50 = 800	25×800 = 20,000	-	5×13×13 = 845	1×1×100 = 100	100×845 = 84,500	-
4	4×4×50 = 800	4×4×50 = 800	-	800	1×1×100 = 100	1×1×100 = 100	-	100
5	4×4×50 = 800	1×1×10 = 10	10×800 = 8,000	-	1×1×100 = 100	1×1×10 = 10	10×100 = 1,000	-
	Total		**46,000**	1,520	Total		106,625	**945**

Table: Accuracy of different models on MNIST dataset

Model	Runtime (sec)		λ	Accuracy (%)	Dataset
	Total	Amortized time			
CryptoNets	570	69.580×10^{-3}	80	99.00	MNIST
E2DM	28.590	450.0×10^{-3}	80	98.01	MNIST
FCryptoNets	39.100	39.100	128	98.71	MNIST
A*FV	5.160	0.630×10^{-3}	82	99.00	MNIST
A*FV	5.710	0.340×10^{-3}	175	99.00	MNIST

REFRENCES

[1] Zhang, J.; Liu, F.; Xu, W.; Yu, H. Feature Fusion Text Classification Model Combining CNN and BiGRU with Multi-Attention Mechanism. *Future Internet* 2019, *11*, 237.

[2] Rehman, Anwar Ur & Malik, Ahmad & Raza, Basit & Ali, Waqar. (2019). A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis. Multimedia Tools and Applications. 10.1007/s11042-019-07788-7.

[3] Fentaw, H.W.; Kim, T.-H. Design and Investigation of Capsule Networks for Sentence Classification. *Appl. Sci.* 2019, *9*, 2200.

[4] Peng, Hao & Li, Jianxin & He, Yu & Liu, Yaopeng & Mengjiao, Bao & Wang, Lihong & Song, Yangqiu & Yang, Qiang. (2018). Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN. WWW '18: Proceedings of the 2018 World Wide Web Conference. 1063-1072. 10.1145/3178876.3186005.

[5] Sachan, Devendra & Zaheer, Manzil & Salakhutdinov, Ruslan. (2019). Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function. Proceedings of the AAAI Conference on Artificial Intelligence. 33. 6940-6948. 10.1609/aaai.v33i01.33016940.

[6] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2018, June). Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In International Conference on Machine Learning (pp. 201-210).