Domain Classification of Text Conversationusing Deep Learning

[1]D. Tarun, [2]D. Varshik Reddy, [3]D. Venkata Sasikiran, [4]D. Vishnu Vardhan,[5]Sujit Das

[1,2,3,4] Department of AIML

School of Engineering,

Malla Reddy University, Maisammaguda,

Dulapally, Hyderabad, Telangana-500100

[1]dandetarun2002@gmail.com, [2]reddyvarshik9@gmail.com, [3]sasikiran.dv@gmail.com, [4]vishnunani71435@gmail.com

[5]Assistant Professor

Department of AIML

School of Engineering, Malla Reddy University, Maisammaguda,

Dulapally, Hyderabad, Telangana-500100

sujit.das@mallareddyuniversity.ac.in

Abstract:

The process of classifying text according to its domain or subject is known as domain categorization. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), long-short term memory networks (LSTMs), and natural language processing are just a few examples of the deep learning methods that may be utilised to categorise a domain (NLP). Recently, LSTM networks have proven to produce positive results. When textual input is given, the final product is classified according to the text's domain. This categorization can be used for various purposes, such as sentiment analysis, topic modelling, and even spam detection. Additionally, deep learning methods have shown promising results in improving the accuracy of domain classification compared to traditional machine learning techniques.

Keyword:

Domain classification, deep learning, and long-short term memory networks (LSTM).

Introduction:

The ability to analyse huge amounts of text data makes domain categorization a key job in natural language processing. Due to their capability to collect long-term dependencies and contextual information, deep learning approaches like Long-Short Term Memory Networks (LSTM) have been particularly successful in this field. These models have been effectively used in many different contexts, such as social media, news stories, and academic papers.

A text document is to be assigned a specified set of labels or categories based on its content, language, and style as part of domain categorization. The categories can range from general subcategories within those categories, such as news, sports, finance, technology, entertainment, and health, to more specialised categories themselves.

A model is trained on a labelled dataset, which is made up of texts tagged with their associated domains, in order to conduct domain classification, which is commonly accomplished using machine learning techniques. With the training data, the model extracts patterns and characteristics, which it then uses to categorise previously unread or fresh texts into the relevant domains.

The classification process is broken down into multiple parts, including pre-processing the text input (removing stop words, tokenizing, etc.), extracting pertinent features (bag-of-words, word embedding's, etc.), and using an appropriate machine learning algorithm (such as Naive Bayes, Support Vector Machines, or deep learning models like Convolutional Neural Networks or Transformers).

There are several practical uses for domain categorization. For instance, it may be used to group consumer evaluations into several product categories in an e-commerce environment, enabling firms to examine comments and make data-driven decisions. Domain categorization may be used in social media analysis to find recurring themes or subjects in user-generated material. It also helps with information retrieval, where categorising content into particular domains can increase search relevancy and precision.

Texts with unknown class labels are first classified using a text classification approach that uses pre-classified texts to train a classifier. This strategy may be used to classify text collections in a variety of areas, and, in comparison to knowledge engineering-based text classification approaches, the classification accuracy is somewhat increased.

Nevertheless, these models have drawbacks such as sparse feature vectors, dimensional explosion, and challenging feature extraction. They also need feature engineering and a lot of material and human resources.

The models made use of well-known machine learning methods such as Support Vector Machine (SVM), Naive Bayesian (NB), decision trees, k-means, etc. The simplest unit of language is a word, and machine learning models are constructed based on how words are represented as vectors. The word, the smallest semantic unit of the text, is the first object of NLP's work. The thesaurus generally represents each word as a high-dimensional vector (the dimension is the size of the thesaurus). The only dimension of the vector, with a value of 1, represents the position of the current word in the vocabulary, while all other dimensions, with a value of 0, represent the rest of the vocabulary.

This approach of one-hot word encoding is straightforward and efficient, but it is susceptible to the problem of dimension disaster and only represents words, which is independent of one another. It cannot take into account text word order information or represent the semantic similarity between words.

Literature Review:

Liu et al. [1] paper discusses domain classification techniques for text mining and presents an approach based on feature engineering and machine learning.

Akhtar et al. [2] provides a comprehensive review of deep learning techniques for domain-specific sentiment classification, which is closely related todomain classification of text.

Kowsari et al. [3] provides an overview of various text classification algorithms, including those used for domain classification, and discusses their strengths and limitations.

Wang et al. [4] proposed a multi-task learning framework for domain-specific sentiment classification, which involves identifying the sentiment in text from different domains.

Pan et al. [5] provides an overview of transfer learning techniques, which can be applied to domain classification by leveraging knowledge from related domains.

Jindal et al. [6] discussed about the problem of opinion spam detection, which can be relevant in identifying text from certain domains.

Wang et al. [7] presents a domain-specific word embedding technique that leverages a softmax-based attentive model to capture domain-specific semantic information.

Yang et al. [8] introduced an adversarial training approach for unsupervised domain adaptation of deep neural networks, which can be useful for domain classification tasks.

Ren et al. [9] presented a label-efficient learning method for transferable representations across domains and tasks, which can be applied to domain classification problems.

Ruder et al. [10] provides an overview of multi-task learning techniques in deep neural networks, which can be applied to domain classification tasks when multiple related domains are involved.

Proposed System:

The proposed deep learning model consists of LSTM which is a Recurrent Neural Network. RNN is a type of neural network with a memory status for preprocessing sequence inputs.

RNN uses internal states(memory) to process the input sequence. This makes it applicable to tasks such as Natural Language Processing, speech recognition, time series analysis data processing. Long short-term memory (LSTM) has recently become popular in NLP for the superior ability to model and learn from sequential data. LSTM aims to solve the RNN problem called the gradient vanishing and exploding. LSTM replaces the hidden vectors from RNN with memory blocks equipped with gates.

This can maintain long-term memory by practicing gating weights and has proven to be very useful in achieving state of art. RNN is more suitable for NLP than CNN because of its timing and the ability to process variable-length input and explore long-term dependence.

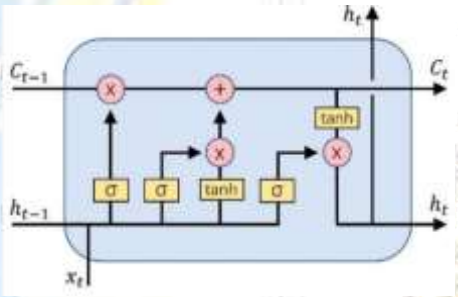

Figure.1 - The architecture of LSTM

In our model we go through several steps before acquiring the output

1) We first import the required libraries which are numpy, pandas, matplotlib, seaborn, keras from tensorflow.
2) Then we read our dataset which in this case is BBC-text.csv file.

Which contains multiple types of labels from the conversations

- Tech
- Business
- Sports
- Entertainment
- Politics

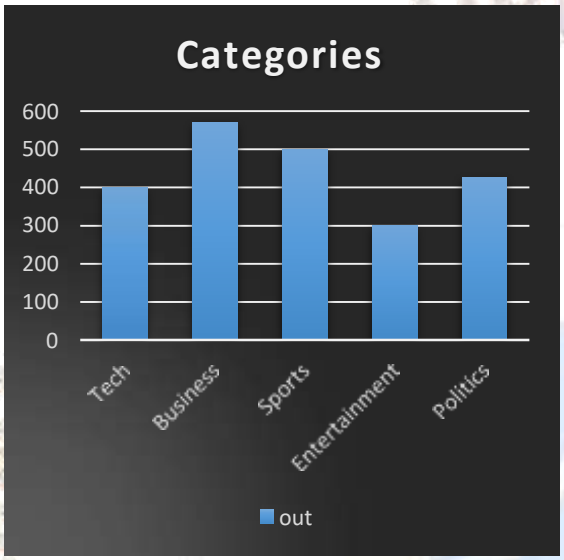

Figure.2: Domains

3) We perform text pre- processing in which we remove numbers, white spaces, special characters, capital alphabets by converting it into lower case etc. then we embed the words.
4) We tokenize the pre-processed data, in which the data is divided into multiple tokens.
5) We define our LSTM model and fit the data into it. We run epochs to find the accuracy and loss of the model.
6) With help of seaborn and matplotlib libraries we plot the accuracy and loss graphs.
7) We run the model and feed the model with random input and get the domain of that particular statement/sentence.

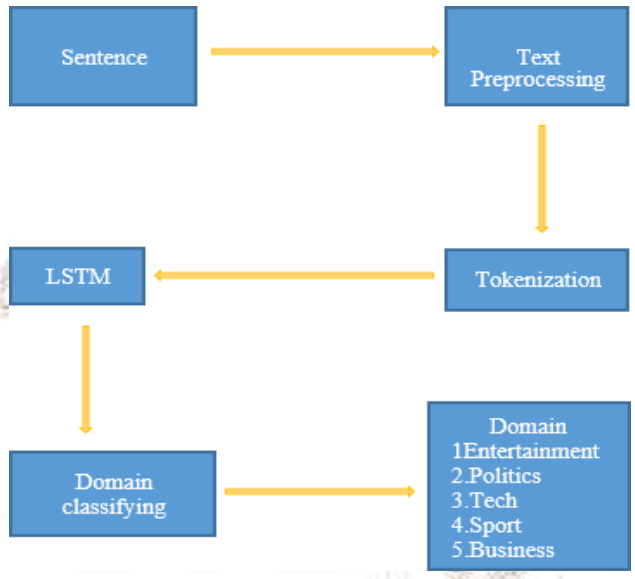

Figure.3: Architecture of the model

Results:

The Long-Short Term Memory Network (LSTM) application of this model to textual data is used to classify the domain of a text. A domain is classified as a result of input text being provided. Textual information, such as an article or a sentence, is present in the dataset used. Imagine, for example, that you type "Odisha train accident: Congress, NCP demand Railway Minister Ashwini Vaishnaw's resignation," and the model responds with the sentence "The domain is politics."

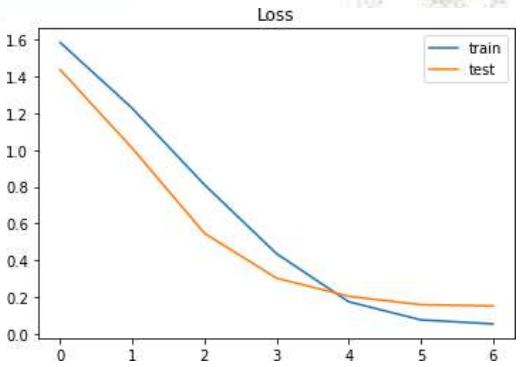

Figure.4: Graph- Loss Vs Epoch

We are validating the data loss that occurred during the operation in order to make sure that the proportion of data lost continuously lowers with each iteration. The data loss % is shown on the Y-axis, while the epoch value for each iteration is shown on the X-axis. The graph, as seen in Figure 4, shows a diminishing trend in the proportion of data loss with rising epoch values. This shows that the model is gaining knowledge and getting better with each iteration, which is essential for getting correct results with machine learning.

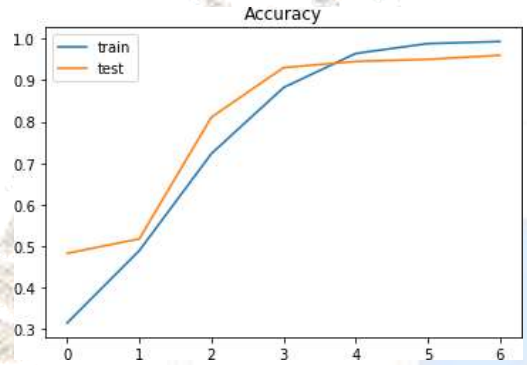

Figure.5: Graph- Accuracy Vs Epoch

A further way to confirm accuracy is to see if, after each iteration, the model accurately reproduces the input. The epoch value (each iteration) is used to establish the X-axis in this case, whereas the Y-axis is precisely specified. According to the graph, the accuracy of the model rises together with the number of iterations. For the model to learn and get better over time, it's crucial to keep an eye on the accuracy throughout training.

The model we proposed classifies the data and gives domains such as business, entertainment, politics, education, health, sports, and technology as outputs.

```
1/1 [==============================] - 0s 46ms/step
hobbit picture  four years away  lord of the rings ...
Domain: entertainment

1/1 [==============================] - 0s 47ms/step
game firm holds  cast  auditions video game firm b ...
Domain: tech

1/1 [==============================] - 0s 54ms/step
clarke plans migrant point scheme anyone planning  ...
Domain: politics

1/1 [==============================] - 0s 55ms/step
radcliffe will compete in london paula radcliffe w ...
Domain: sport

1/1 [==============================] - 0s 56ms/step
serena becomes world number two serena williams ha ...
Domain: sport

1/1 [==============================] - 0s 79ms/step
ultimate game  award for doom 3 sci-fi shooter doo ...
Domain: tech

1/1 [==============================] - 0s 79ms/step
algeria hit by further gas riots algeria suffered  ...
Domain: business

1/1 [==============================] - 0s 53ms/step
fast lifts rise into record books two high-speed l ...
Domain: business

1/1 [==============================] - 0s 69ms/step
muslim group attacks tv drama 24 a british muslim  ...
Domain: entertainment
```

Figure.6: Result of Domain Classification of text

Text	Domain	Time Taken(ms/step)
Hobbit picture four years away lord of rings…	Entertainment	46
Tim Cook reveals he uses ChatGPT after iOS 17…	Technology	47
Unemployment, price rise real issues, …	Politics	54
Live Cricket Score: WTC Final India vs Australia…	Sports	55

Conclusion:

In this study, we created a model that categorises the text's domain. The text is automatically categorised by the model into a certain domain or topic. We evaluated the performance of our model by measuring its accuracy against the number of epochs. The results showed that the accuracy of the model increased as the number of epochs increased, indicating that our model is effective in classifying text into different domains.

Acknowledgement:

We are grateful to our project mentor Prof. Sujit Das and Dr. S. Satyanarayana for their guidance, inspiration and constructive suggestions. Prof. Sujit Das' constant cooperation and encouragement made it possible, while Dr. S. Satyanarayana's unwavering encouragement and support ensured the success of the project. They also provided helpful suggestions on the topic we selected, assuring the success of the project.

Also, we would like to extend our profound gratitude to Dr. Thayyaba Khatoon, the department chair, for providing the rest of us with such a fantastic opportunity to learn about ourselves and the outside world while working on current real-world applications of deep learning. We also thank our parents and other members of our extended families for their contributions in terms of money and spirit, which helped to make the project a success.

References:

1. Liu, Y., Li, S., & Zhou, G. (2019). Domain classification for text mining. In Proceedings of the 2019 3rd International Conference on Education and Multimedia Technology (pp. 104-108). IEEE.
2. Akhtar, M. T., Raza, M. T., & Awan, M. A. (2019). Deep learning for domain-specific sentiment classification: A comprehensive review. Journal of Network and Computer Applications, 131, 42-69.
3. Kowsari, K., Brown, D. E., Heidarysafa, M., Jafari Meimandi, K., Gerber, M. S., Barnes, L. E., ... & Hsu, W. (2019). Text classification algorithms: A survey. Information, 10(4), 150.
4. Wang, P., Xu, J., Xu, B., Liu, C., & Zhang, H. (2019). Domain-specific sentiment classification using a multi-task learning framework. Knowledge-Based Systems, 175, 99-107.
5. Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345-1359.
6. Jindal, N., & Liu, B. (2008). Opinion spam and analysis. In Proceedings of the International Conference on Web Search and Data Mining (pp. 219-230). ACM.
7. Wang, Y., Cui, L., Su, H., & Song, J. (2018). Domain-specific word embedding via softmax-based attentive model. Expert Systems with Applications, 103, 156-167.
8. Yang, J., Li, Y., Zhou, C., Li, Y., & Zhou, M. (2018). Adversarial training for unsupervised domain adaptation of deep neural networks. In Proceedings of the 27th International Joint Conference on Artificial Intelligence (pp. 1463-1469). AAAI Press.
9. Ren, Y., & Qi, G. J. (2019). Label efficient learning of transferable representations across domains and tasks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 11329-11338).
10. Ruder, S., Piotr, B., & Rosasco, L. (2018). An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098.