Product Attribute Extraction from Text

Supriyo Roy
Dept.of Mechanical engineering
Indian Institute of Technology Bombay
Mumbai, India
193109013@iitb.ac.in

Mudit Sand
Dept.of Mechanical engineering
Indian Institute of Technology Bombay
Mumbai, India
203100068@iitb.ac.in

Manas Pandey
Dept.of Mechanical engineering
Indian Institute of Technology Bombay
Mumbai, India
203100065@iitb.ac.in

***Abstract*— Product attribute from the text is still done manually in most of the firms despite having great progress in the field of artificial intelligence. The reason behind this is text is generally hard to analyze with these algorithms because algorithms work on the numeric data. So various techniques to convert words into numerical vectors like word2vec and glove are developed. In this project we tried to develop a model to extract some features from the product description present in a dataset.**

Keywords— NLP, LSTM, GLOVE, CNN, Word Embedding

I. INTRODUCTION

The attributes of a product are very important entities in the commercial market as they affect the ultimate Purchase Decision. Product attributes are what helps us in how we differentiate between different products. Attributes include things like colour, size, mass and any other feature which are relevant to the product. But, the data we get regarding a product is often present as unstructured text with the attributes and its value buried within it. If we extract attributes and their value from the unstructured text of product description manually, it takes a lot of time and is quite cumbersome.

It is important to extract product attributes value pairs from product description in an efficient manner as they are an important matter in real-world business environments. From a manufacturer point of view, attributes help in understanding customer behaviour and also define a product's competitive set to help them outstand their rivals. From a customer perspective, product attributes are the building blocks of purchase choice.

There are various challenges faced while extracting product attributes from text. The design and layout of the text changes frequently and it is difficult to scrape the data from a single algorithm as the same model may not be able to keep up with the changes. Product categorization using text data for eCommerce is a very challenging extreme classification problem with several thousands of classes and several millions of products to classify. The attributes themselves might be described differently like the same brand name can be written in different ways (GE and General Electric) or say dimensions (7” or 7 or 7 feet).

Salient features of our work includes, using GLOVE for mapping the word embeddings and deploying two different approaches to extract the attribute from the texts one using CNN and another using BiLSTM model.

We are not able to get the results we thought we would achieve but still our models are able to extract certain features from the text.

II. BACKGROUND AND PREVIOUS WORK

There has been various research carried out regarding product attribute extraction. Younghoon Lee et al. [1] have done work on extraction and polarization of product attributes using an explainable neural network. The focus of the research was not only on attribute extraction but also on their relative importance. The extraction was done using a convolution neural network and transfer learning. A novel approach was used, consisting of variants of the Gradient-weighted class activation mapping (Grad-CAM) algorithm (an explainable neural network framework). The weight of each aspect was calculated to make a sentimental prediction model [1].

Abhinandan et al. [2] did a comparison between hierarchical models and flat models to show that in some cases flat models perform better. Two models (multi-CNN and multi-LSTM) that extract features from individual pieces of unstructured data were explored. Structured attributes and their values together were used in a disorderly fashion along with convolutional filters so that the regulation of the attributes and the differing attributes by product categories did not pose any problem [2].

Pasawee et al. [3] have done work on multi-label product categorization using multi-modal fusion models. Multi-modal algorithms using images, descriptions, and titles to categorize e-commerce products on Amazon was deployed. For baseline models, a modified CNN architecture was used to classify the description and title. The idea behind the research was that each modality complemented the deficiencies of the other modalities, confirming that raising the number of modalities can be an efficient method for advancing the performance of multi-label classification problems.

Shubhabrata et al. [4] have done work on the extraction of missing attributes value. Most of the past work done on this domain employed extraction of missing attribute values with the possible set of values known already, but their work was focused on discovering new values based on product profile information such as title and description. They restructured the problem as a sequence tagging task and offered a joint model exploiting recurrent neural networks to obtain context and meaning, and Conditional Random Fields (CRF) to enforce tagging consistency. An attention mechanism was provided to interpret the explanation for the model’s decision. A sampling strategy exploring active learning was also employed [4].

Petar et al. [5] have done work on machine learning for product matching and categorization. They applied standard classification techniques with neural language models and deep learning techniques. In order to train feature extraction models to be able to extract attribute-value pairs from product description structured product data was used as guidance [5].

Rayid et al. [6] have done work on text mining for product attribute extraction. The objective of the research was to augment the database of the products in such a way that each product is represented as an attribute-value pair. The algorithms employed by them were single-view and multi-view semi-supervised learning and also unlabeled data was used (less expensive and easy to obtain)[6].

Rezk et al. [7] have done work on accurate product attribute extraction on the field. They applied a bootstrapping approach for attribute extraction from semi-structured text. An initial labelled dataset was created using the same semi-structured text [7].

III. Datasets

Dataset for our work is taken from the link cited in reference no.[8], this dataset contains product description and their respective brand. We also create our own code to scrap the data into json format in which title and different attributes can be extracted as per the requirement and the same can be used if we plan to extend our work in future.

IV. Procedure and experiments

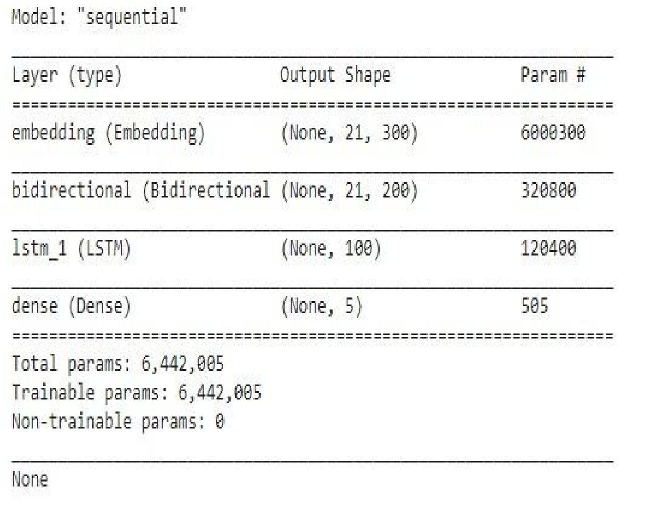
In our project, we have extracted product attributes(brand name) from the title of the product. So, to use any data driven models to extract attributes from a sentence just like a product title, first we have to convert all these words to vectors. To convert the words we can use word2vec, Glove Embedding etc. Glove Embedding is used in this project. It is based on matrix factorization techniques on the word-context matrix. In word2vec, words are taken as training data to model a neural network and embedding captures whether words appear in similar context but in glove embedding, words co occurrence over the whole text is focused.

‘Glove.6b.300d.txt’ is used for converting words. Here, this glove embedding can map the words to vectors of length 300. For word embedding,first a zero matrix of size (number of words, embedding size) is created then find the word from the glove file and add the word vector to the matrix.

For the prediction we use two deep learning model, Bi directional Long Short Term Memory (Bi-LSTM) and Convolutional Neural Network (CNN)

Bi-LSTM is a sequence processing model, consisting of two LSTM layers one is forward and another one is backward. Bi-LSTM increases the amount of available information in the network. So, we can achieve high accuracy using Bi-LSTM.

For our prediction model, 2 layers of Bi-LSTM model with 100 nodes per layer respectively are used. Optimized the hinge Loss using Nadam optimizer and iterated the model for 3 epochs.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 21, 300)           6000300
_________________________________________________________________
bidirectional (Bidirectional (None, 21, 200)           320800
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               120400
_________________________________________________________________
dense (Dense)                (None, 5)                 505
=================================================================
Total params: 6,442,005
Trainable params: 6,442,005
Non-trainable params: 0
_________________________________________________________________
None
```

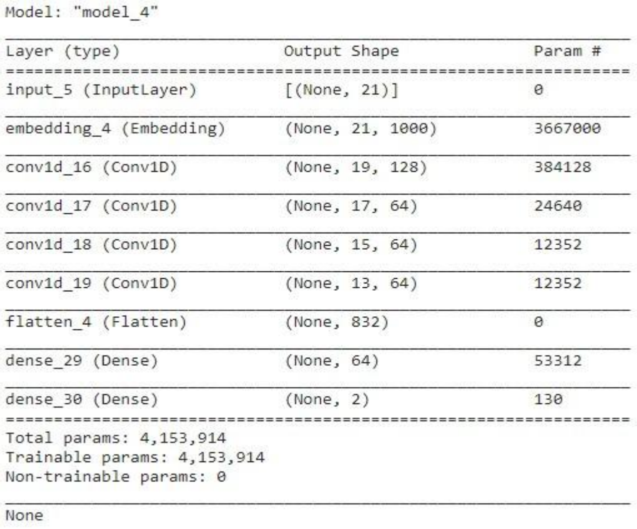
Fig.1 Bi- LSTM model summary

Another model we have tried to implement is Convolution Neural Network. CNN is generally used for image classification, but CNN can be applied for different NLP tasks. By varying the size of the kernels and layers, we can detect the pattern of the word vectors and we can predict our product attribute.

To convert words to vectors in CNN model, we have written a separate code to create a word embedding file by taking around 1500 titles from the training dataset.

We have taken the first 1500 data for training from the dataset mentioned above as our word embedding file is created using these data.

Used 4 convolution layers with ReLU activation function and 5 dense layers with 64 and 32 nodes per layer. Optimized the categorical cross entropy loss using stochastic descent gradient optimizer with learning rate 1e-5 and momentum 0.9 and used 5 epochs.



```
Model: "model_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_5 (InputLayer)         [(None, 21)]              0
_________________________________________________________________
embedding_4 (Embedding)      (None, 21, 1000)          3667000
_________________________________________________________________
conv1d_16 (Conv1D)           (None, 19, 128)           384128
_________________________________________________________________
conv1d_17 (Conv1D)           (None, 17, 64)            24640
_________________________________________________________________
conv1d_18 (Conv1D)           (None, 15, 64)            12352
_________________________________________________________________
conv1d_19 (Conv1D)           (None, 13, 64)            12352
_________________________________________________________________
flatten_4 (Flatten)          (None, 832)               0
_________________________________________________________________
dense_29 (Dense)             (None, 64)                53312
_________________________________________________________________
dense_30 (Dense)             (None, 2)                 130
=================================================================
Total params: 4,153,914
Trainable params: 4,153,914
Non-trainable params: 0
_________________________________________________________________
None
```

Fig.2 CNN model summary

V. Result

In Bi-LSTM model, optimizing hinge loss using Nadam, we are getting Hinge Loss as 0.7997 and 90.01% accuracy on training data and 0.7995 hinge loss and 90.73% accuracy on validation data after 3 iterations.

In CNN model, optimizing Categorical cross entropy loss using Stochastic Gradient Descent, we are getting loss as 134.7831 and accuracy

96.39% on training data and cross validation loss 179.1233 and accuracy 98.01%.

VI. Conclusion

The Bi-LSTM and CNN model employed were able to predict some of the features from the unstructured text data. Some unwanted attributes were also predicted. With further augmentation in the model by optimizing the hyper-parameters this can be overcome.

VII. Statement of contribution

Project Concept : - Manas, Mudit, Supriyo
Code writing and debugging - Mudit
Concepts utilized - Manas, Supriyo
Report writing - Manas, Supriyo
Video making - Mudit, Manas

References

[1] Lee Y, Park J, Cho S. Extraction and prioritization of product attributes using an explainable neural network. Pattern Analysis and Applications. 2020 Nov;23:1767-77.

[2] Krishnan A, Amarthaluri A. Large Scale Product Categorization using Structured and Unstructured Attributes. arXiv preprint arXiv:1903.04254. 2019 Mar 1.

[3] Wirojwatanakul P, Wangperawong A. Multi-Label Product Categorization Using Multi-Modal Fusion Models. arXiv preprint arXiv:1907.00420. 2019 Jun 30.

[4] Zheng, Guineng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. "Opentag: Open attribute value extraction from product profiles." In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1049-1058. 2018.

[5] Ristoski, Petar, Petar Petrovski, Peter Mika, and Heiko Paulheim. "A machine learning approach for product matching and categorization." *Semantic web* 9, no. 5 (2018): 707-728.

[6] Ghani, Rayid, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. "Text mining for product attribute extraction." *ACM SIGKDD Explorations Newsletter* 8, no. 1 (2006): 41-48.

[7] Rezk, Martin, Laura Alonso Alemany, Lasguido Nio, and Ted Zhang. "Accurate Product Attribute Extraction on the Field." In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1862-1873. IEEE, 2019.

[8] https://raw.githubusercontent.com/maciej-cecot/brand-detection/master/models/lstm_brand_detection_model/helper_files/train.csv

[9] https://github.com/maciej-cecot/brand-detection (Use this github repos as reference code)