PSYCHOLOGY AND EDUCATION (2021) 58(5), ISSN 1553 - 6939
Article Received: 22th November, 2020; Article Revised: 26th March, 2021; Article Accepted: 26th April, 2021

Improving Pricing Intelligence by Multi-Modal Deep Learning Method

Ninad Madhab[1*], Alina Dash[2]
[1]Undergraduate Student, VSSUT Burla
[2]Assistant Professor, VSSUT Burla

ABSTRACT

Deep networks are successfully implemented when there is data of single modalities (e.g. Texts, Images etc) but when it comes to pricing comparison both data can be helpful for information gathering. Pricing intelligence involves of analysing pricing data, tracking, monitoring and for understanding the market and making educated change in pricing at various scale and speed. Often changes in product pricing has led retailers to continually monitor their relative price position and incorporate changes within an active strategy. The correct insights into which viable products are selling at a finest price, retailers can act in instantaneously with similar offers and discounts that get consumers enthusiastic or the idea about making the switch from a competitor. Retailers want to use automatic pricing intelligence, matching with a competitive approach. In this paper, we optimize and improvise the method of pricing intelligence by developing a deep learning model which considers both the products image and text. Therefore, we have used a novel method a shared classification layer to generate hierarchical universal embeddings, a multi-modal deep-learning method by which we have generated embeddings that comprises of both the product's text and image representation which can help in further downstream tasks for classification and product retrieval. It learns the semantic information along with cross-modal representation. A shared hidden layer is learnt by the model in which the distance between any two universal embeddings is like the distance between their corresponding class embeddings in the semantic embedding space. It also uses a classification objective with a shared classification layer to make sure that the image and text embeddings are in the same shared latent space.

Keywords
Pricing Intelligence, Multimodal Deep Learning, Universal Embedding Space, Shared Classification Layer, Smart Data Processing

Article Received: 10 August 2020, Revised: 25 October 2020, Accepted: 18 November 2020

Introduction

Pricing Intelligence is all about knowing the effect of market prices on buyers' purchasing decisions and, ultimately, businesses' performance. This data and knowledge are used to optimize a pricing strategy, making it more competitive, driving more sales and increasing profitability. It generally analyses the market and collects relevant data from competition, allowing to improve the pricing strategy, gain competitiveness and increase conversions. When we have access to updated and detailed data about the main competitors, the data can segment the information to optimize the pricing strategy. Tracing, monitoring and scrutinising pricing data to comprehend the market and making refined changes in pricing swiftly and scalable is developed but the major challenges comprise of classifying the products into right categories and matching the exact same products across different retailers for price comparison which includes data both in the form of images and text and as Often changes in product pricing has led retailers to continually monitor their relative price position and incorporate changes within an active strategy. The correct insights into which viable products are selling at a finest price, retailers can act in instantaneously with similar offers and discounts that get consumers enthusiastic or the idea about making the switch from a competitor.

This problem comes under the category of multi-label classification and product retrieval. [4] Sometimes product's textual data (product title & description) is helpful for pricing intelligence, and sometimes product's images help. Hence, we need a deep learning model which considers both the products image and text which we have tried to implement in this paper. Therefore, we have used the novel multi-modal deep-learning method, which generates the required embeddings that comprises of both the product's text and image representation which helps in further reducing the tasks for classification and product retrieval.

In this age of information, the data has been evolving from primarily text-based to being multimodal i.e. the text data is augmented with content from videos and images. There is a rapid growth of media types in different domains. It is essential and challenging to learn discriminative feature representations of all these modalities. Mostly creating modality-agnostic (universal) embeddings to share the same semantics (e.g., audio of a baby crying and a picture of baby) would allow to exploit the information which is complementary among themselves and also enriching the latent space resulting in ranking, hight benefit search, e-commerce space and ads. Moreover, such universal embeddings are highly relevant for cross-modal product retrieval. However, it is a challenging task to learn universal embeddings since features from different modalities can be inconsistent also known as 'media-gap'.

The novel method of adding a shared classification layer, unlike the previous ranking loss-based methods counters the above issues to learn hierarchical universal semantic embeddings. It projects text and image into a shared latent space [2, 5]. It allows the latent space that is shared to be completely independent of the semantic embedding space while retrieving the semantic structure. For improving pricing intelligence, it is important to achieve hierarchical precision and accurate classification results.

Our study is organised as follows: In Section-2 we have scrutinised the related work. In Section-3 we created text and image embedding inputs and since the multimodal model takes in two input, image and text, the Image is

passed onto a pre-trained model which produces an embedding for an individual images and GloVe embeddings are used to obtain a representation of the Text. In Section-4 we implemented model to create final embeddings and in Section-5 we incorporated three losses into the architecture. In Section-6 we compare the Classification Task with other models to our model.

Pricing Intelligence

Dynamic pricing i.e. fluctuations in price of products because of the demand is ever longing concept from centuries. For eg. The matinee shows were less costly than other shows in evenings, the room tariff at a beach varied with seasons. However, in current scenario it has become more important to exactly calculate fair pricing algorithmically. Algorithmic dynamic pricing can be seen changing the outlook of transportation sector, entertainment, e-commerce and other many industries operating online and offline.

In current scenario, the prevailing software runs on cloud and AI tools at backend making pricing intelligence more accurate with additional provision of automation in sales. The dynamic pricing by AI has made good progress but still has not been fully integrated with the intricate sales strategy and is unable to use data which is available in different modalities.

Our approach is upfront: a model which can evaluate these data of different modalities and the prices routinely alter based on both text and image data to swiftly adjust to fluctuations in the marketplace and expand profitability. It is important to address this data issue since gradually the algorithms are becoming powerful and thus increasing the modalities of the data. Therefore, companies' product and service prices can repeatedly respond to demand and opposition in real time.

Multimodal classification

Multimodal classification, in previous works is mostly divided into early fusion, intermediate fusion and late fusion methods. Early fusion, also known as Data level fusion, is a traditional way of fusing data of multiple modalities before passing into the deep neural network. This process is also known as input level fusion. Various research conducted anticipated two possible methods for this technique. The first approach consisted of merging data by eliminating the correlation between them. The second approach suggested that data to be fused at the common space in lower layers of neural network. These approaches have many statistical solutions which can be used to achieve one or both methods, which includes canonical correlation analysis, principal component analysis (PCA) and independent component analysis [3, 6]. Intermediate fusion methods concatenate different modalities of data into a complex level of from the intermediate layers of the neural network through multiple layers. This is done through cross modal attention. Each layer runs linear and nonlinear functions which convert the scale of input data, swing, skew which provides a new depiction of the original data which was the input. In context of multimodal deep learning, it is a fusion of different modalities embeddings into one hidden layer such that the model learns a combined representation of each of the modalities. Late fusion method classifies different sources of data followed by fusion at a decision-making layer of deep neural network. The use of late fusion method is encouraged by the popularity of ensemble classifiers. It is simple in usage than earlier fusion methods and predominantly when the sources of data are significantly wide-ranging from one another in terms of data dimensionality, sampling rate and unit of measurement. Often this approach gives improved performance since errors from multiple models are dealt with independently — by simple methods such as by training another network on top of the classification scores and weighted average. We have used for multimodal classification the late fusion technique where the scores of classifications from text and images are fused. However, we have used a shared hidden layer for classification for all modalities unlike other fusion approaches where there is a separate classification layer for each modality. The shared hidden layer also helps in building a universal embedding space which clusters the embeddings to the same class.

Universal Embeddings

Embeddings that are pre-trained on a large corpus and can be plugged in a variety of downstream task models (sentimental analysis, classification, translation) to automatically improve their performance by incorporating some general word/sentence representations learned on the larger dataset. It's a form of transfer learning. Transfer learning has been recently shown to drastically increase the performance of NLP models on important tasks such as text classification. [2, 6]

Unlike others, we are using classification objective and a shared hidden layer to learn universal embeddings. The extracted text and image features from pretrained networks are passed through their respective towers and then L2 normalized to form the resulting embedding. Then, a shared hidden layer which maps text and images into a universal shared space classifies these normalized embeddings.

Semantic Embeddings

Existing image classifiers are restricted to a fixed set of output categories. And these categories are either entirely discrete, or if interrelated are interrelated in rigidly defined ways. Classification is brittle and the errors are often nonsensical and arbitrary. The previous works were focused upon finding a way to embed the ImageNet labels into a much larger, semantically, and syntactically structured space.

Our method is alike to the class level semantic embedding space in terms of distance but learns a universal embedding space. This has an added advantage of having different dimensions which can be compared to embedding spaces with class labels.

In summary, the main contributions of the paper are:

1. Implementing a multi-modal deep-learning method is required and using it we generated embeddings that comprises of both the product's text and image

representation which can help in further downstream tasks for classification and product retrieval.
2. Creating a universal embedding space with inclusion of semantic information of products.
3. Implementing a novel semantic embedding method that doesn't involve projection onto semantic embedding space.
4. Using a shared classification layer to learn universal embeddings for improvising and increasing pricing intelligence.
5. Implementing the three losses, for Class Level Similarity, Semantic Similarity, Cross Modal Gap.

Related Work

This section briefly scrutinises different literature corresponding to pricing intelligence, multimodal classification, universal embeddings, and semantic embeddings as our work is related to these areas. Research regarding multimodal classification was done by Wang, et al. (2019) [1] shows the reasons for difficulty in training a multimodal classification network. They had identified two main causes for this performance drop: first, due to enlarged capacity multi-modal networks are often prone to overfitting. Second, dissimilar modalities overfit and generalize at different rates, so training them jointly with a single optimization strategy is sub-optimal. They have also addressed these two problems with a technique we call Gradient Blending. Antol et al. in proceedings of the IEEE international conference on computer vision (2015) [2] researched on usage of both text and image data as their input proposed the task of free-form Visual Question Answering when given an image and a natural language question about the image, the achieved the task to provide an accurate natural language answer in real-world scenarios. Earlier research on word embeddings and semantic embedding by Berardi et al. (2015) [5] compared two popular word representation models, word2vec and GloVe, and trained them on two datasets with different stylistic properties. Their research results implied that the tested models were able to create syntactically and semantically meaningful word embeddings despite the higher morphological complexity of Italian with respect to English and moreover concluded that the formal properties of the training dataset have a related role in the kind of information apprehended by the produced vectors. Research on improving word embeddings by presenting a character-enhanced word embedding model by Chen et al. in Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)[7] were able to address the issues of character ambiguity and non-compositional words by propose multiple-prototype character embeddings and an effective word selection method. Meanwhile Wang et al. proposed deep experiments of recipe recognition on them with a large multimodal food dataset using visual, textual information and fusion in 2015 IEEE International Conference on Multimedia & Expo Workshops [8]. Earlier research which helped us to think through the product retrieval was Harley et al. who presented a new state-of-the-art for document image classification and retrieval, using features learned by deep convolutional neural networks (2015) [3]. Research by Ballesteros et al. on high-performance transition-based parser using long short-term memory (LSTM) recurrent neural networks to learn representations of the parser state which aided us to replace lookup-based word representations with representations constructed from the orthographic representations of the words using LSTMs. [4] Our procedure of common space and usage of semantic graphs is inspired from the research by Young et al. which computed the denotational similarities by constructing a denotation graph based on a large corpus of 30K images and 150K descriptive captions. [11] Our architecture is derived from the research by Collobert et al. (2008) which presented how both multitask learning and semi-supervised learning improve the generalization of the shared tasks, resulting in state-of-the-art-performance. [15]

Creating Text And Image Embeddings Inputs

As our method uses semantic information when learning universal embeddings that are derived from class labels, we create a large multimodal product classification dataset. We use our approaches on the Dataset consisting of 50k products accompanied along with its image, text and class name extracted from various e-commerce websites.

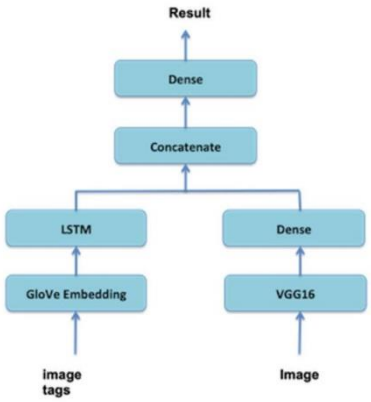


Fig 1: The architecture of the new deep learning model for multi-input models transfer learning.

Image Embeddings Input Model

Since our model is a multimodal model it takes in two input, image, and text. The Image is passed onto a pre-trained VGG16 neural network which produces an embedding for an individual image. We extract pretrained VGG16 of size 224 from individual images. VGG16 is a convolutional neural network model proposed which is a dataset of over 14 million images belonging to 1000 classes. [7] Image embedding created from this part of the model are forwarded into the image tower for further processing. We used transfer learning and image augmentation to acquire desired results.

Text Embeddings Input Model

We use GloVe Embedding pretrained text model for representing text into embedding space. Text embedding created from this part of the model are forwarded into the text tower for further processing. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space [xi, xii]. We restrict the maximum length of words in a sentence tag and limit the size of tags vocabulary to prevent overkill. We tokenize the words from GloVe dataset converting all of them into sequences which will represent the encoded form and unique tokens. We also introduce padding to make sentences of equal size.

Model Implementation For Creating Final Embeddings

In this section, the output from VGG16 is passed onto an Image Tower in parallel to output from GloVe which is passed onto the
Text Tower. The L2 normalized output from both the towers are further passed onto a shared fully connected layer. The output of the shared fully connected layer is further used to calculate different losses. [15]

4.1 Image Tower Model

It consists of sequence of dense and activation layers, 5 hidden layers of 512 hidden units each. The final L2-normalized output is sent to a shared fully connected layer.

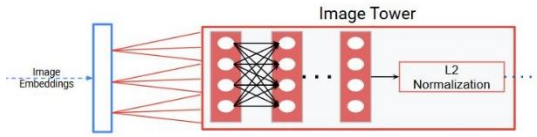


Fig 2. (i) Image embeddings are passed through their Image tower to get image embeddings in universal space.

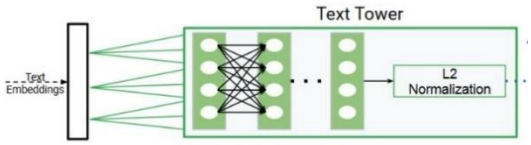
```
embedded_tag = layers.Embedding(max_words, embedding_dim)(tag_input)
encoded_tag = layers.LSTM(512)(embedded_tag)
z = layers.BatchNormalization()(encoded_tag)
z = layers.Dense(512, activation = 'tanh')(z)
z = layers.Dropout(0.15)(z)
z = layers.Dense(512, activation = 'tanh')(z)
z = layers.Dropout(0.15)(z)
z = layers.Dense(512, activation = 'tanh')(z)
z = layers.BatchNormalization()(z)
```

Fig 2. (ii) Code for image tower implementation

```
x = layers.Flatten()(vgg16)
x = layers.BatchNormalization()(x)
x = layers.Dense(256, activation='relu')(x)
x = layers.Dropout(0.15)(x)
x = layers.Dense(512, activation='relu')(x)
x = layers.Dropout(0.15)(x)
x = layers.Dense(512, activation='relu')(x)
x = layers.Dropout(0.15)(x)
x = layers.Dense(512, activation='relu')(x)
x = layers.Dropout(0.15)(x)
x = layers.Dense(512, activation='relu')(x)
x = layers.Dropout(0.15)(x)
x = layers.Dense(512, activation='relu')(x)
x = layers.BatchNormalization()(x)
```

Text Tower Model

It consists of sequence of dense and activation layers, 2 hidden layers of 512 hidden units each. The final L2-



normalized output is sent to a shared fully connected layer.

Procedure for Training

Between all hidden layers of image and text towers a dropout of 0.15 is used. The network was trained using the RMSProp optimizer with a learning rate of 1.6192e-05and momentum set to 0.9 with random batches of 1024 for 250,000 steps [4, 10]. To maximize the image and text classification accuracies on the validation set these hyperparameters are chosen. The inference can be drawn by training with different epochs of different batch sizes that the model doesn't easily underfit and overfit after concatenation of two towers. As can be seen from below Fig 4.

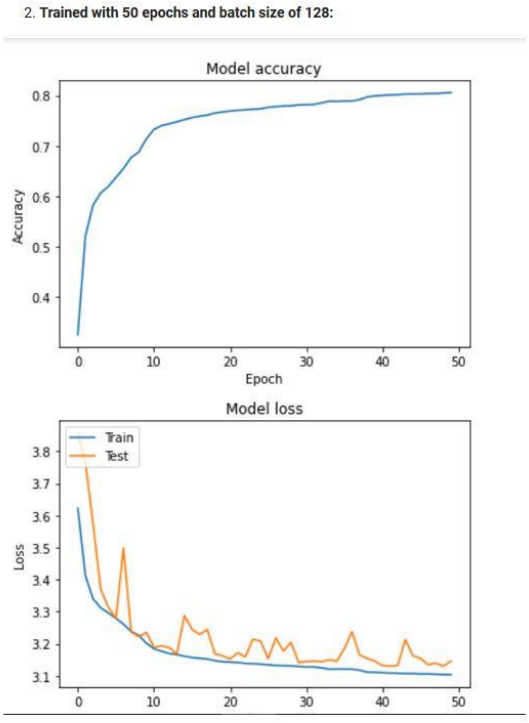


Fig 3. (i) Text embeddings are passed through their Image tower to get text embeddings in universal space.

Fig 4: Training the model with different epochs and different batch sizes to find best parameters

Semantic Graph

A semantic graph is constructed from the class names based on the embeddings extracted. Mostly, class names mostly contain

more than a single word (e.g. Open back), hence, we use sentence encoders to provide sentence level embedding. The cosine distance between universal sentence encoder embeddings is treated as weight of the edge and each class is treated as a vertex. It is used in calculating the loss. Let G= (V, E), where V={v_1,v_2,...,v_K}represents the set of K classes and E represent the edges between any two classes. Let ψ(·) represent the function that extracts embeddings of a class name. The adjacency matrix A={A_{ij}}from range i,j=1 to K of graph G contains non-negative weights associated with each edge, such that (Eq.1))

$$A_{ij} = d(\psi(v_i), \psi(v_j))$$

where *d* is cosine distance. [vii, viii]

The Losses Into The Model

This paper incorporates three losses, for Class Level Similarity, Semantic Similarity, Cross Modal Gap.

If we formulate the problem at hand, let the labelled set given as *D* containing image-text-labels as *(p,q,y)*, where *y* is class label to the image *p* and text *q*. Our objective is to learn a universal embedding space which can understand the sematic structure. Let *x, y* corresponds to the representation of modalities for either image *p* or text *q* with class *y*. Let *φI(·)* represent the function which projects an image to dense vector correspond to embedding in universal embedding space and *φT(·)* represents a function which projects a text to the shared space. *φ(·)* is the projection function which projects to the shared layer space. *φ(·)* corresponds to *φI(·)* if *x* is image and *φT(·)* if *x* is text. Assume *d(·)* to be the distance which measures the cosine distance between two embeddings. With these assumptions we try to calculate the losses.

Class level similarity

The embeddings from image tower and text tower is passed through a shared fully connected layer and the model is trained using SoftMax cross entropy loss. It refers to the distance between any embeddings representing the same class be on an average is less than the distance between them belonging to different classes. Eq(2) [3, 9]

$$d_{avg}(\varphi(x_i^a), \varphi(x_j^b)) < d_{avg}(\varphi(x_m^a), \varphi(x_n^b)) \quad (2)$$

where d_{avg} indicates the average distance, computed in different (suitable) choices of pairs.

The following loss function was used for class level similarity. Eq(3)

$$L_{classification} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} I[y_i = k]\log(p(k|x_i)) \quad (3)$$

For our model, we found the class level similarity comes out to be as (1) the Fig 5

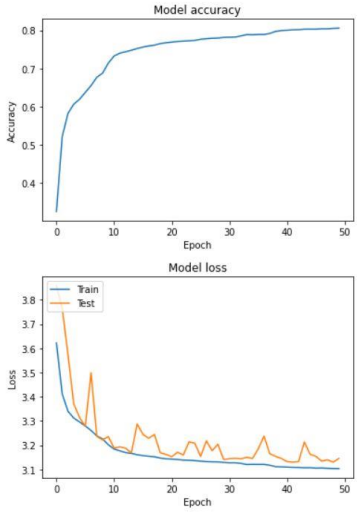


Fig.5: The embeddings from image tower and text tower is passed through a shared fully connected layer and the model is trained using SoftMax cross entropy loss

Similarity

We created a semantic graph, which consists of adjacency matrix which is a matrix of cosine distance between the text embedding representation of individual classes such that if you have 5 unique classes, your adjacency matrix will be a 5x5 matrix consisting of cosine distance of text embedding of one class name with all other classes names. This matrix is used in calculating Semantic Similarity loss. Formulating the mathematics behind it, the embeddings representing the two different but semantically similar classes would be on average closer to each other than the embeddings representing to two semantically different classes so, according to the equation *a* and *b* are more semantically similar classes than *a* and *c*. Eq. (4)[13, 15]

$$d_{avg}\left(\varphi(x_i^a), \varphi(x_j^b)\right) < d_{avg}(\varphi(x_m^a), \varphi(x_n^c)) \quad (4)$$

The following loss function was used for calculating Semantic Similarity: Eq(5)

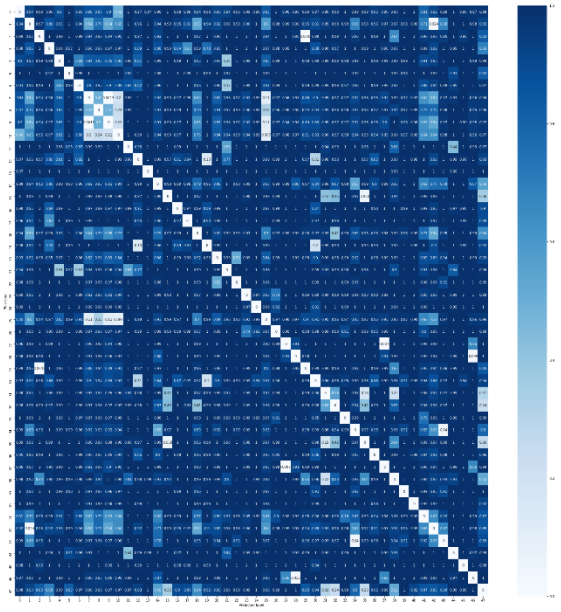
$$L_{graph} = \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\sigma_{mn}^{ij}\left(d\left(\varphi(x_m^i), \varphi(x_n^j)\right) - A_{ij}\right)^2$$

where

$$\sigma_{mn}^{ij} = \begin{cases} 1 \;\; if\; A_{ij} < \varsigma \; and \; d\left(\varphi(x_m^i), \varphi(x_n^j)\right) < \varsigma \\ 0 \; otherwise \end{cases} \quad (5)$$

(6)

Semantic similarity for our model can be inferred from the following Fig 6. Maximum all true labels are predicted correctly. All the deep coloured labels represent inference drawn by predicted labels with maximum confidence with the true labels.



Cross modal gap

The cross-modal gap loss, which is the distance between image and text embeddings corresponding to the same product instance. It becomes challenging for the universal embedding learning as the embed-dings from various modalities have different allocations. Preferably, the distance between paired text and image should be close to zero as they correspond to same instance. So, we would like to have Eq. (7)

$$d\left(\varphi_I(p_n), \varphi_T(q_n)\right) \approx 0 \quad (7)$$

For different points *n* in the data. Intuition behind Cross Modal gap Loss can be referred to the tensor outputs from the text tower and the image tower. These values can be converted to NumPy array and plotted, this plot may tell us the difference(loss) between the text embeddings and image embeddings. Our model when plotting to see the similarity in the normalized outputs for 19 classes resulted in the following Fig.7 Clearly, model needs to be trained more as the values are a lot different, but we can see the spikes to be somewhat similar which was desired. The following loss function was used for calculating Cross modal gap. Eq(8)

(8)

$$L_{gap} = \frac{1}{N}\sum_{n=1}^{N} d\left(\varphi_I(p_n), \varphi_T(q_n)\right)$$

Fig 6: Adjacency Matrix for showing essence of Semantic Similarity captured by the model.

Comparison Of Classification Task

Our objective of the model is classification and it is natural to apply the model for classification task. For a pair of image and text, our model returns separate classification scores for image and text. These SoftMax scores are then fused using weighted averaging. Fig.8 details the text, image and fusion classification accuracies of our model and other baselines. Our model achieved an accuracy of 90.3% on the dataset, surpassing other models. Furthermore, for fusing image and text channels, the earlier models used complex gated attention method. We only used a simple weighted averaging to fuse SoftMax scores from both channels. Reviews of other baselines similar to our model are:

CME: It maximizes the cosine similarity between similar image-text pairs and minimizes it between all dissimilar image-text pairs to learn Cross-Modal Embeddings. It uses an added classification loss to regulate semantic similarity.

TABLE 1

HIE: It stands for Hierarchy-based Image Embeddings. It maps images onto class embeddings whose pair-wise dot products correspond to a measure of semantic similarity between classes. We have extended it also to text. [1, 3]

Model	**Image**	**Text**	**Fusion**
Separate Models	73.4	85.6	90
CME	73.4	77.8	87.1
HIE	73.1	81.4	87
Our Model	**74.6**	83.9	**90.3**

Conclusion

For improvising the pricing intelligence, we incorporated a novel model with a shared classification layer to learn the universal embedding space. It also includes the semantic information and categorises products of different modalities into semantic similarity. The previous methods used for pricing intelligence and also to handle multimodal data maps text and image embeddings to a constant class label embedding space. Unlike these our model created a universal embedding space with inclusion of semantic information of products. The shared classification layer used for both products' text and image embeddings and incorporating the instance losses reduced the modalities gap and lead to strong cross-modal performance. Besides, our model achieved an accuracy of 90.3% on the dataset of 50k products accompanied along with its image, text and class name extracted from various e-commerce websites surpassing others. Furthermore, our future tasks remain to extend our model in designing proper product retrieval and dispatching systems for smart data processing in Smart Cities and building Smart solutions as majority of the data either from cameras or sensors comprises of both image and text data.

References

[1] Wang, Weiyao, Du Tran, and Matt Feiszli. "What Makes Training Multi-Modal Networks Hard?." arXiv preprint arXiv:1905.12681 (2019).

[2] Antol, Stanislaw, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. "Vqa: Visual question answering." In Proceedings of the IEEE international conference on computer vision, pp. 2425-2433. 2015.

[3] Harley, Adam W., Alex Ufkes, and Konstantinos G. Derpanis. "Evaluation of deep convolutional nets for document image classification and retrieval." In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 991-995. IEEE, 2015.

[4] Ballesteros, Miguel, Chris Dyer, and Noah A. Smith. "Improved transition-based parsing by modeling characters instead of words with LSTMs." arXiv preprint arXiv:1508.00657 (2015).

[5] Berardi, Giacomo, Andrea Esuli, and Diego Marcheggiani. "Word Embeddings Go to Italy: A Comparison of Models and Training Datasets." In IIR. 2015.

[6] Bojanowski, Piotr, Armand Joulin, and Tomas Mikolov. "Alternative structures for character-level RNNs." arXiv preprint arXiv:1511.06303 (2015).

[7] Chen, Xinxiong, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. "Joint learning of character and word embeddings." In Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.

[8] Wang, Xin, Devinder Kumar, Nicolas Thome, Matthieu Cord, and Frederic Precioso. "Recipe recognition with large multimodal food dataset." In 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 1-6. IEEE, 2015.

[9] Botha, Jan, and Phil Blunsom. "Compositional morphology for word representations and language modelling." In International Conference on Machine Learning, pp. 1899-1907. 2014.

[10] Chrupała, Grzegorz. "Normalizing tweets with edit scripts and recurrent neural embeddings." In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 680-686. 2014.

[11] Young, Peter, Alice Lai, Micah Hodosh, and Julia Hockenmaier. "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions." Transactions of the Association for Computational Linguistics 2 (2014): 67-78.¬¬¬

[12] Abdel-Hamid, Ossama, Abdel-rahman Mohamed, Hui Jiang, and Gerald Penn. "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition." In 2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP), pp. 4277-4280. IEEE, 2012.

[13] Weston, Jason, Samy Bengio, and Nicolas Usunier. "Wsabie: Scaling up to large vocabulary image annotation." In Twenty-Second International Joint Conference on Artificial Intelligence. 2011.

[14] Baroni, Marco, and Alessandro Lenci. "Distributional memory: A general framework for corpus-based semantics." Computational Linguistics 36, no. 4 (2010): 673-721.

[15] Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning." In Proceedings of the 25th international conference on Machine learning, pp. 160-167. 2008.