Synerise at SIGIR Rakuten Data Challenge 2020: Efficient Manifold Density Estimator for Multimodal Classification

Dominika Basaj
Synerise
Warsaw, Poland
dominika.basaj@synerise.com

Barbara Rychalska
Synerise
Warsaw, Poland
barbara.rychalska@synerise.com

Jacek Dąbrowski
Synerise
Warsaw, Poland
jack.dabrowski@synerise.com

Konrad Gołuchowski
Synerise
Warsaw, Poland
konrad.goluchowski@synerise.com

ABSTRACT

The recently proposed EMDE (Efficient Manifold Density Estimator) model proved competitive in multimodal settings, achieving state-of-the-art results in session-based and top-k recommendations. In this work we explore its application to Rakuten Data Challenge Task 1: Multimodal Classification. The aim of the challenge is to assign items to classes based on their titles, descriptions and photos. We achieve a competitive result in this task applying a new, trainable version of EMDE to separate pretraining of single modalities and selected combinations of modalities.

CCS CONCEPTS

• **Information systems → Multimodal Classification**.

KEYWORDS

Rakuten Data Challenge, deep learning, aneural networks, multimodal classification

ACM Reference Format:
Dominika Basaj, Barbara Rychalska, Jacek Dąbrowski, and Konrad Gołuchowski. 2020. Synerise at SIGIR Rakuten Data Challenge 2020: Efficient Manifold Density Estimator for Multimodal Classification. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'20)*. ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Rakuten Data Challenge Task 1: Multimodal Classification. The aim of the challenge is to assign items to the correct class of the Rakuten France product catalog. Each item is represented with textual data: title, (possibly missing) description, and visual data: item photo.

The released train set contains 84916 items. The test set is released in two stages: Stage 1 data contains 937 items. Stage 2 test set (used for determining the final results) contains 8435 items. Test sets are drawn from the same distribution as the train set. The difficulty in handling the challenge data consists in high levels of noise, missing data, various text languages (mainly French, but also English and German), and class imbalance.

The gold standard assignments for the test set are not public and the models are evaluated via an online leaderboard. The metric used for performance evaluation is macro-F1 score. The score is understood as the arithmetic average of per product type code F1 score. It is not weighted by class size, which means that even the smallest classes must be predicted accurately.

Our solution. Our contributions are as follows:

- We introduce a trainable version of EMDE [3].
- We achieve competitive results in this task with a system composed of two complementary stages: modality pretraining done with trainable EMDE, and modality fusion. We do not exploit any external data sources, relying only on the challenge train set.
- We analyze the effectiveness and challenges of our method.

Overall challenge results. Our approach takes the 5th place in Stage 1 out of 15 submissions, achieving 89.72% macro-F1, compared to the leading score of 91.94% macro-F1, and surpasses baseline performance measure of 87.05%. Our model proves to be robust as it scores 89.78% on much bigger test sample of 8435 items. We score a solid 4.42 pp. higher than the baseline Rakuten Institute of Technology model on the final leaderboard.

2 RELATED WORK

Multimodal learning involves two challenges: extracting useful information from each modality and adequately leveraging the knowledge derived from all modalities. The latter challenge is called multimodal fusion and can be performed in two ways: early or late fusion [5]. Early fusion consists in the creation of a joint multimodal representation which is fed to a separate classification model. Late fusion approaches use the decisions made by each per-modality classifier and combine them with mechanisms such as averaging [1]. [10] demonstrate that late fusion approaches can work best in the classification setting, which we empirically confirm.

3 SYSTEM

After thorough experiments, we apply a two stage approach:

(1) **Multimodal Pretraining.** Multimodal pretraining is a separate training of each single modality on our classification

SIGIR eCom'20, July 30, 2020, Virtual Event, China

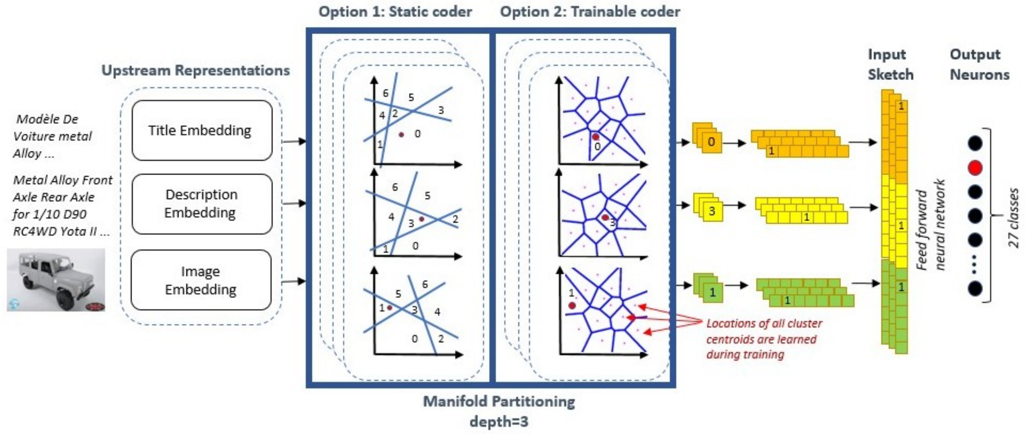
Option 1: Static coder
Option 2: Trainable coder
Upstream Representations
Modèle De Voiture metal Alloy ...
Metal Alloy Front Axle Rear Axle for 1/10 D90 RC4WD Yota II ...
Title Embedding
Description Embedding
Image Embedding
Locations of all cluster centroids are learned during training
Input Sketch
Output Neurons
Feed forward neural network
27 classes
Manifold Partitioning depth=3

Figure 1: Subsequent steps of the EMDE algorithm for each of the 3 modalities. Upstream representations depend on modality type and are obtained by EfficientNet (for images) , Camembert or byte-pair encoding with CNN on top (titles and descriptions). Input sketch depicted above serves as a representation of each modality in Multimodal Fusion stage. Figure adapted from [3] and enriched with trainable coder introduced in this work.

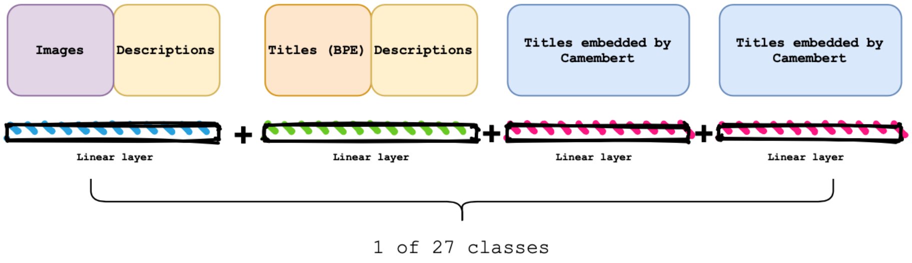

Figure 2: Overview of the Multimodal Fusion part of the system. Each modality is pretrained with a separate model in order to train it for the best performance. Colored blocks represent input sketches from Figure 1. Some of the sketches are concatenated (descriptions with images and with titles encoded by BPE), whereas titles embedded by Camembert are standalone. On top of concatenated sketches and standalone sketches we apply linear layers with 27 output classes. The outputs of these layers are summed, and *argmax* operation defines the correct class. The figure is best viewed in color, as the same color of linear layer indicates shared weights and the same color in top blocks indicate the same sketch representation.

task. Representations learned in this step are fused in a later stage. Each modality is pretrained with recently introduced EMDE [3], which is a multimodal density estimator based on intuitions from Local Sensitive Hashing [4] and Count-Min Sketch [2] algorithms. It recently achieved competitive results in recommender systems, reframing recommendations as a density estimation problem. In this paper, we introduce a trainable version of EMDE shown in Figure 1.

(2) **Multimodal Fusion.** For each per-modality model, we collect representations produced by EMDE encoders in the Multimodal Pretraining step, which we simply call an *input sketch.* We feed these sketches to separate linear layers, which are trained jointly (see Figure 2). Obtained per-class logits for each modality are summed and *argmax* operation on the vector of length 27 defines the final prediction of the classifier.

The details about each per-modality network are provided in section 3.4.

3.1 Non-Trainable Efficient Manifold Density Estimator

Efficient Manifold Density Estimator (EMDE) introduced in [3] is a probability density estimator inspired by Count-Min Sketch algorithm (CMS) and local sensitive hashing (LSH). Input data represented by vectors embedded on a manifold obtained by metric representation learning methods is partitioned via a data-dependent LSH method (DLSH). A region of the partitioning corresponding to a region of the vector manifold is analogous to a hash-bucket in CMS. While a single region is large (typically 64-256 regions form a single partitioning covering the whole manifold), multiple independent partitionings allow to obtain a high resolution map of the manifold via intersection or ensembling. In [3] the DLSH method is neither differentiable nor end-to-end trainable, due to the highly discontinuous binary-unary conversion being a key operation during assignment of inputs to region indices.

3.2 Trainable Efficient Manifold Density Estimator

In this paper we introduce a trainable version of EMDE. It retains the ability of static EMDE to model data density. However, instead of a static assignment of inputs to specific regions of the manifold, we propose to use trainable centroids. Our algorithm proceeds as follows:

1. Randomly initialize a trainable vector of $\cdot depth \cdot width$ centroids of dimensionality *inner_dim* with $\mathcal{N}(0,\ 1)$.
2. Feed item modality embedding (e.g. embedding vector of image) to a simple linear layer to obtain a projected representation x of size $depth \cdot width \cdot inner\_dim$.
3. Apply batch normalization to vectors x.
4. Compute $(x - centroids)^2$, then sum across $inner\_dim$, to get squared euclidean distances $d(x, centroids)^2$.
5. Apply $Softmax(-d(x, centroids)^2)$ across $width$ to obtain $depth$ soft assignments to $width$ regions each.

The parameter *inner_dim* represents the dimensionality of centroids and hence that of inner space in which squared centroid distances are calculated. We empirically find that on all modalities tested the optimal value of *inner_dim* is 8. The parameters *depth* and *width* correspond to N (number of independent manifold partitionings) and K (number of regions produced by each partitioning) parameters from the original static EMDE.

We find that batch normalization of projected inputs combined with $\mathcal{N}(0,\ 1)$ initialization of centroids is crucial in achieving good performance, preventing an otherwise degenerate initialization where all centroids would be very far from the inputs.

The ability to train the network in an end-to-end fashion allows to use EMDE to learn the compressed manifold map in a gradient-guided way, leading to significant improvements in accuracy.

Listing 1: Pytorch pseudocode of differentiable EMDE

```
class DifferentiableEMDECoder(Module):
  def __init__(self, input_dim, width, depth, inner_dim=8):
    self.lin = Linear(input_dim, depth*width*inner_dim)
    self.bn = BatchNorm1d(depth*width*inner_dim)
    self.centroids = Parameter(torch.randn(depth*width*inner_dim))

  def forward(self, x):
    x = self.bn(self.lin(x))
    d = (x - self.centroids)**2
    d = d.view(-1, self.depth * self.width, self.inner_dim).sum(-1)
    d = d.view(-1, self.depth, self.width)
    soft_assignments = F.softmax(- d, -1)
    return soft_assignments.view(-1, self.depth * self.width)
```

3.3 Data Preparation

In the challenge, we have two types of modalities: textual and visual.

Textual modalities. Exploratory analysis of product titles and descriptions proves that these modalities are noisy and apart from French, there are also other languages involved. Although we test several preprocessing techniques including removal of HTML tags and extra entities (eg. *&amp*), we observe that its impact on the final performance depends on the embedding technique that we choose. We decide to use features extracted by Camembert [6], which is a powerful BERT-based language model trained for French, for both titles and descriptions. However, apart from that, we additionally encode titles with byte-pair-encoding with vocabulary of 2000 codes. We treat these as two separate modalities and we train per-modality classifier for each of them separately. When we embed titles with Camembert or BPE we do not apply any preprocessing, as we observed a substantial decrease in the performance of the final model if we do so (by 1 pp.). It leads us to believe that the noise introduced by HTML tags is representative for a particular product class. However, for the embedding of the descriptions, we remove HTML residues in order to get clean, contextual sentences.

Images. Although noisy, images are not transformed in any way in our final solution. We experimented with background removal [7] and cropping the white frames around items, but we have not observed any decisive performance gains from these attempts. As we fine-tune EfficientNet [9] for image extraction, we normalize images in a standard way, with mean and standard deviation calculated on ImageNet dataset for each channel.

Train/Valid Split. We do not use any additional data sources apart from the challenge train set itself. We create our own validation set by sampling out 4000 examples from the train set and we maintain this split throughout all the fine-tuning procedures.

3.4 Multimodal Pretraining

As stated before, we train a separate classifier for each modality: title, description and image. It allows us to learn representative features that serve as an input to the final model, depicted in Figure 1 as *input sketches*. The general architecture for each pretrained network consists of the following blocks:

1. Input embedded by state-of-the-art feature extractor (upstream representation in Figure 1).
2. Trainable layer like CNN encoder for titles and descriptions that learns from the embedded input. In case of images we simply extract features from EfficientNet-b2.
3. Trainable Efficient Manifold Density Estimator, described in 3.2.
4. Final linear layer for classification.

Input Embeddings. For textual modalities we use both Camembert [6] and byte-pair encoding [8], which gives us two trained networks for the same modality contributing to the performance of the overall solution (Table 3). BPE is a static representation, however we could possibly fine-tune the whole Camembert network in the process of training per-modality classifier. Nevertheless, because of number of parameters within Camembert, we decide to treat it as a pure feature extractor and to limit the length of each description to 200 characters. We do not shorten the length of the titles. Images are encoded with pretrained EfficientNet-b2 network [9].

Trainable layer. Each extracted representation serves as an input to a trainable layer. For both titles and descriptions, we use

Table 1: Learning rates for each pretrained network and final network with fusion. Decay is reported in brackets ($decay^{epoch}$)

Image	Title BPE	Title Cam	Description Cam	Final fusion
0.0001 (0.9)	0.001 (0.9)	0.005 (0.7)	0.01 (0.7)	0.001 (0.9)

Table 2: Ablation Study displaying macro-F1 scores

3 modalities	Without double title layer	Final fusion
87.5%	88.8% (+1.3pp)	**89.72%** (+2.22pp)

Table 3: F1-macro score of each pretrained modality.

Image	Title BPE	Title Cam	Description Cam	Final Fusion	Baseline (Static EMDE)
68.61%	82.62%	84.39%	82.58%	**89.72%**	86.08%

Table 4: Confusion matrix between worst predicted classes on our validation set. X axis are predicted classes, Y axis are the target classes.

Class	10	1140	1280	1281	2403	2705
10	124	0	1	1	6	**10**
1140	1	113	4	2	0	1
1280	0	**17**	198	**10**	1	0
1281	2	4	**27**	70	0	1
2403	**16**	0	0	0	197	1
2705	2	0	0	0	0	124

a stack of convolutional layers with varying kernel sizes, which are treated as ngrams. For descriptions, we use kernel sizes of 2 to 5, and effectively 4 CNN layers. For encoding titles with BPE we train 5 layers with kernel sizes of 1 to 5, and for titles with features extracted with Camembert, we find out that the best performance is obtained with 4 layers, with increasing kernel sizes of 1 to 5. For images, on top of the encoded representation there is an average pooling layer and batch normalization layer.

Training. We train each modality with Adam optimizer with varying learning rates decreasing exponentially with every epoch. Details of our configuration are presented in Table 1. The best accuracy of each pretrained model is presented in Table 3.

3.5 Multimodal fusion and results

Our final results are presented in Table 3. The trainable EMDE achieves 89.72% F1-macro score and surpasses the baseline static EMDE (86.08%) by a significant margin. For multimodal fusion, we remove the last linear layer and feed representation obtained with EMDE (input sketches) to the linear layer in the final fusion classifier. We observe that joint training of linear layers as shown in Figure 2 with input obtained by EMDE (in multimodal pretraining step) gives us better performance than simple ensembling of probabilites from all per-modality networks.

Interestingly, we observe that not all combinations of modalities are worthwhile to be fused. Most importantly, we see that descriptions bring most success when concatenated together with images and titles and fed into linear layers with output of 27 neurons, one for each class (see Figure 2). This probably allows to gracefully handle the cases where descriptions are missing. What is also interesting is that combining features from titles obtained by Camembert with other modalities (descriptions or images) decreases the performance of the system, whereas combining features from the network where BPE served as an input increases the overall performance. That is why we decide to concatenate descriptions with titles (BPE), but at the same time we collect input sketch from the classifier with titles embedded by Camembert. In the second stage of multimodal fusion we treat it as a standalone feature. We also notice that this modality itself is so powerful that we can increase the performance of the final classifier by doubling it in the late fusion phase (see Figure 2). This hardly straightforward way of fusing pretrained modalities gave us the best performance on our valid set, which was very close to macro-F1 score calculated by the leaderboard. As shown in Table 2, simple joint training of concatenated titles, descriptions and images features is significantly worse (2.22 pp.) than such a sophisticated multimodal fusion.

Error analysis. To discover possible sources of errors, we analyzed the confusion matrix between target and predicted classes. Confusions between worst predicted classes are shown in Table 4. Most often confusions arise between classes belonging to Child (1280, 1281) and Entertainment (1140) meta-class. The wrongly classified examples are indeed not easy to handle as manual analysis proves that they do not exhibit any abnormal characteristics in any of the modalities. Bad performance of these classes is probably caused by the fact that all of them contain toys and games varied by a child's age and the mode of play (party games vs educational games, etc.), where the line separating these types can be very thin. Another triple with a lot of confusion are classes 10, 2403, and 2705 (Books). Books are hard to classify because their titles can be abstract and not so characteristic of the content. In this case, a possible solution would be to have a more specialized classifier putting larger weight on the image modality, as the style of the cover is often decisive.

4 SUMMARY

In this paper we present a 2-stage system which achieves competitive results in SIGIR Rakuten Data Challenge Task 1: Multimodal Classification. The system uses an improved, trainable EMDE version to create informative multimodal features for later fusion. Non-obvious interactions between various modalities contributed to the increase in the overall performance of the classifier. Thanks to the performance on the leaderboard, we prove that our solution is robust and competitive.

5 ACKNOWLEDGEMENT

Barbara Rychalska and Dominika Basaj were financially supported by grant no 2018/31/N/ST6/02273 funded by National Science Centre, Poland.

REFERENCES

[1] P. K. Atrey, M. A. Hossain, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. 2010. Multimodal fusion for multimedia analysis: a survey.

[2] Graham Cormode and S. Muthukrishnan. 2004. An Improved Data Stream Summary: The Count-Min Sketch and Its Applications. In *LATIN 2004: Theoretical Informatics*, Martín Farach-Colton (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 29–38.

[3] Jacek Dąbrowski, Barbara Rychalska, Michał Daniluk, Dominika Basaj, Piotr Babel, and Andrzej Michałowski. 2020. An efficient manifold density estimator for all recommendation systems. https://arxiv.org/abs/2006.01894

[4] Piotr Indyk and Rajeev Motwani. 2000. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing* 604-613 (10 2000). https://doi.org/10.1145/276698.276876

[5] Kuan Liu, Yanen Li, Ning Xu, and Prem Natarajan. 2018. Learn to Combine Modalities in Multimodal Deep Learning. arXiv:1805.11730 [stat.ML]

[6] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7203–7219. https://www.aclweb.org/anthology/2020.acl-main.645

[7] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar Zaiane, and Martin Jagersand. 2020. U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection. *Pattern Recognition* 106, 107404.

[8] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1715–1725. https://doi.org/10.18653/v1/P16-1162

[9] Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *CoRR* abs/1905.11946 (2019). arXiv:1905.11946 http://arxiv.org/abs/1905.11946

[10] Tom Zahavy, Alessandro Magnani, Abhinandan Krishnan, and Shie Mannor. 2016. Is a picture worth a thousand words? A Deep Multi-Modal Fusion Architecture for Product Classification in e-commerce. arXiv:1611.09534 [cs.CV]