Tabular text embedding for Vietnamese text-based person search

Phan Thi Hoai[1, 2], Nguyen Minh Phuc[1], Nguyen Huu Hieu[1]
Pham Thi Thanh Thuy[1*], Le Thi Lan[2]

[1]Faculty of Information Security, Academy of People Security, Ha Dong, Hanoi, Vietnam;
[2]SigM Lab, School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hai Ba Trung, Hanoi, Vietnam.
[*]Corresponding author: thanh-thuy.pham@mica.edu.vn
DOI: https://doi.org/10.54939/1859-1043.j.mst.93.2024.128-136

ABSTRACT

Vietnamese text-based person search is still a challenging problem with the limited dataset of Vietnamese descriptions. The current popular approach to this problem is Deep Neural Networks (DNNs), and recently, transformer networks have been more favored because of their outperformance over CNN and RNN networks for both vision and natural language processing tasks. However, DNN, or transformer networks, require a large amount of training data and computing time for efficient learning of visual and textual features. This brings a burden for implementing Vietnamese text-based person search by DNN, or transformer networks. Towards building a Vietnamese text-based person search system on a scarce resource dataset of Vietnamese descriptive sentences with low computing cost, in this work, we propose to apply the transformer-based architecture named TabTransformer for contextual embedding of the noun phrases chunked from the Vietnamese descriptive sentences. This is the first time the TabTransformer network has been deployed together with CNN and RNN architectures for Vietnamese text-based person search. The experimental results on a limited dataset of 3000VnPersonSearch show the better recognition accuracy of the proposed method compared to the baseline method by about 7.5% at Rank 1. In addition, the computing time of our method is more effective than the baseline method.

Keywords: Text-based person search; Tabular data; TabTransformer; CNN; Bi-LSTM.

1. INTRODUCTION

Text-based person search is a problem of searching person images from databases by the input queries of descriptive sentences. This problem has received much research attention recently for applications in tracing target objects (lost persons, criminals, and suspects) from surveillance cameras. Recently, transformer neural networks have become the state-of-the-art (SOTA) technique in the fields of NLP (Natural Language Processing) and image processing. ViT-Base (Vision Transformer) [2] is utilized for visual feature encoding and BERT [3] is applied for textual feature embedding. In general, despite achieving significantly superior retrieval performance, the transformer networks still have the disadvantage of having a high computational cost and requiring a huge amount of training data to be effective. For the text-based person search problem, low-resource languages like Vietnamese have little data for training such transformer networks. In order to effectively apply transformer architecture for the resource-scare language of Vietnamese, in this paper, we propose to use TabTransformer network [4] for textual embedding in Vietnamese text-based person search. TabTransformer is an innovative deep tabular data modeling framework. It is built on self-attention-based Transformers with the layers transform the embeddings of categorical features into robust contextual embeddings to achieve higher prediction accuracy. In this work, we deploy both global and local feature extraction for text-based person search. For the visual branch, a CNN architecture is proposed for global and local feature encodings. For the textual branch, the local attributes are represented by one-hot vectors and then encoded by Bi-LSTM network. The textual global feature vectors are learned by Tabtransformer, with the inputs being noun phrases (NPs) chunked from the sentences and represented as categorical features. A joint training step is finally done for cross-modal alignment using loss functions.

The contributions of our work are twofold: (1) Using a combination of CNN, RNN, and transformer-based methods for text-based person search. This combination aims to promote the advantages of each method for each branch of textual and visual processing in text-based person search; (2) Applying TabTransformer for robust contextual embeddings of descriptive sentences. To our knowledge, this is the first time TabTransformer is employed for textual feature encoding in text-based person search. The experiments are conducted on a modest dataset of 3000VnPersonSearch [5], but the results are promising. The accuracy at Rank 1 of the proposed method is 7.5% better than the baseline method (using one-hot encoding and Bi-LSTM for textual feature learning). In addition, the computing time of our method is more effective than the baseline method.

The remainder of the paper is organized as follows. Section 3 presents the proposed framework for text-based image retrieval. The experimental evaluations are discussed in section 4. Finally, the conclusions is shown in section 5.

2. THE PROPOSED FRAMEWORK

The proposed framework for learning pairs of visual and Vietnamese textual features is shown in figure 1. Feature learning takes place at both the global and local levels. At global feature learning, the input image is put into a CNN network to give out the image feature vector $\mathbf{f}_{img\_glob}$. This visual feature vector will be input for a global network of global branches to output a global image feature vector v_{glob}. At the global textual process, a description is pre-processed and chunked into noun phrases (NP). These NPs will be presented in tabular data and then passed to TabTransformer network to give out the feature vector of $\mathbf{f}_{txt\_glob}$. This textual feature vector is then put into the global branch to form a global textual feature vector of $\mathbf{t}_{glob}$.

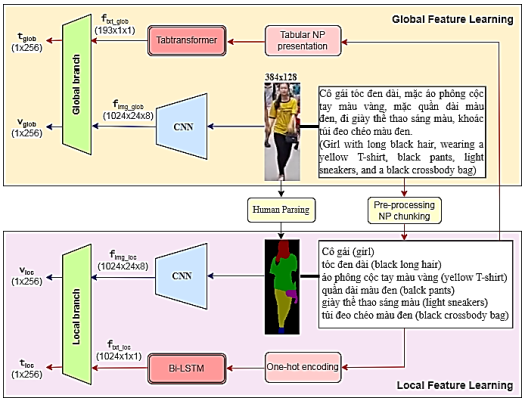

Figure 1. *Framework for the text-base image retrieval with TabTransformer.*

At local feature learning, a person's image is parsed to five visual attributes (head, upper body, lower body, shoes, and bag) by training a human parsing network of HRNet [6]. The image region of "head" relates to the description of hair, hat, glasses, and face; the upper body and lower body regions are equivalent to the top and below-clothes descriptions. Each visual attribute is encoded to feature vector of $\mathbf{f}_{img\_loc}$ by CNN network. This vector is then fed into Local network or Local branch to generate a local visual feature vector of $\mathbf{v}_{loc}$. For textual process at local feature learning, the noun phrases parsed from the description are one-hot encoded. They are the input for Bi-LSTM network to output textual feature vectors $\mathbf{f}_{txt\_loc}$ which finally go through Local branch to bring out a local textual feature vector of $\mathbf{t}_{loc}$.

The global and local features of images and texts are aligned and matched during training the vision-text matching model. The trained model will be used in the testing phase to find out which person images are most relevant to the textual queries. The details for the main processing blocks in the proposed framework are presented in the following subsections.

2.1. Global and local visual feature extraction

- CNN network

The architecture of the CNN network for global and local visual feature extraction is shown in figure 2. The input image has been normalized to 384 x 128 pixels to become the input for the CNN network. The CNN output is a feature map with a size of 1024x24x8 for global extrraction ($\boldsymbol{f}_{img\_glob}$) or local extrraction ($\boldsymbol{f}_{img\_loc}$). These output feature vectors are then put into the global and local branches to give out global and local visual vectors $\boldsymbol{v}_{glob}$ and $\boldsymbol{v}_{loc}$, respectively. At the local branch, the feature map is split into five attribute vectors corresponding to five human body parts: head, upper body, lower body, shoes, and bag. The R-CNN network [7] is used to parse five human body parts and annotate them under the supervision of the Human Parsing Network [6].

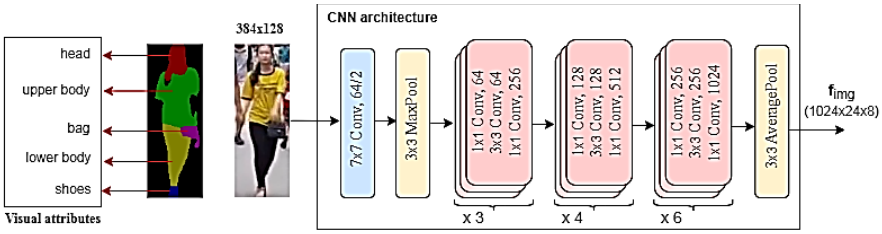

Figure 2. *CNN architecture for global and local visual feature extraction.*

- Global and local branches

The global and local branches are applied for both visual and textual processings. The global branch includes two convolutional layers with filter size (3×3) and number of filters is 2048. Finally, there is an average pooling layer (1×1) and a linear layer which outputs the 256-dimensional global feature vector for each image ($\boldsymbol{v}_{glob}$) or desriptive sentence ($\boldsymbol{t}_{glob}$). The local branch also includes two convolutional layers with filter size and number of filters $[3 \times 3;\ 256]$. Max pooling layer (1×1) followed by a linear layer is used to create 256-dimensional local feature vector for each image ($\boldsymbol{v}_{loc}$) or text attribute ($\boldsymbol{t}_{loc}$).

2.2. Global and local textual feature extraction

In order to extract global and local textual features, a pre-processing step of descriptive sentences is done. The main purpose of this step is to extract NPs from the sentences. These NPs will then be used for (1) local feature embedding by one-hot encoding and local feature learning by a Bi-LSTM network; (2) global feature learning by presenting NPs as tabular data and using a tabular transformer network for the presented NP tabular data.

2.2.1. Description pre-processing

In order to prepare input for textual processing networks, the descriptions are pre-processed through four steps: normalization, word tokenization, Part-of-Speech Tagging (POSTagging) and NP chunking. The first step helps to reduce the noise in input text, such as spelling mistakes, ununified foreign writing, etc. The second step relates to Vietnamese tokenizing. The third step of Part-of-Speech tagging is responsible for labeling word type in sentence, such as noun, verb, etc. The final step is NP chunking which forms noun phrases based on word label and a defined rule for matching. In this rule, we consider noun phrases to be formed by a combination of nouns (N) and adjectives (A). This is because the descriptions of the pedestrian's appearance mainly combine nouns and adjectives. The combination is done with the continuous series of N labels and A labels in the sentence until another word label appears.

2.2.2. Global textual feature extraction

In order to extract global textual feature, three main steps are excecuted: (1) presenting NPs as tabular data; (2) textual feature embedding by TabTransformer network; (3) applying global branch for the results of (2) to give out the global textual feature vector $t_{glob}(1 \times 256)$.

- NP presentation in Tabular form

In order to presenting tabular data for NPs, we group NPs into six groups according to six attributes of Person, Head, Upperbody, Lowerbody, Shoes, Other. The NPs contains the words such as "đàn ông" (man), "phụ nữ" (woman), "nam" (boy), "bé gái" (girl),… belong to the "Person" group. The "Head" group contains NPs of "mái tóc" (hair), "kính" (glasses), "mũ" (hat), etc. The NPs of "áo" (shirt), "áo phông" (T-shirt), "váy" (dress),…belong to the "Upperbody" group. The "Lowerbody" group includes the NPs of "quần" (pant), "quần bò" (jeans),…The "Shoes" group contains "giày" (shoes), "dép_lê" (sandal), etc. The NPs of "ví" (wallet), "balo" (backpack), "túi xách" (handbag) belong to the "Other" group.

From the analysis of descriptive sentences into NPs that indicate each attribute, statistics are made on the number of occurrences of NPs representing the attribute. It should be noted that the number of NPs for tabular representation is a subset of the NPs in the entire dataset. In the whole dataset, each ID has more than 2 images and each image is described by two descriptions. For tabular representation, we choose only one image and two corresponding descriptions for each ID. table 1 shows the occurrence frequency of some NP examples from the whole dataset (NPs-All) and from the sub-dataset extracted for tabular representation (NPs-Tab).

Table 1. *Occurrence frequency of some NPs in the whole dataset (NPs-All) and in sub-dataset for tabular representation (NPs-Tab).*

NPs	NPs-All number	NPs-Tab number
tóc ngắn màu đen (black hair)	489	246
áo sơ mi (shirt)	466	221
quần màu đen (black pants)	454	214
giày thể thao (sport shoes)	1458	618
Đồng hồ (watch)	140	69

The NPs are then presented in tabular form with seven columns and n rows (table 2).

Table 2. *Tabular presentation for noun phrases: the first six columns presents attribute groups that NPs belong to, with "?" being a missing attribute (not included in the description). The seventh column is the weight calculated for NPs in a row except for "Person" attribute. Each row corresponds to a class.*

Person	Head	Upper body	Lower body	Shoes	Other	Flnweight
cô gái (girl)	tóc đen dài (long black hair)	áo phông cộc tay màu vàng (yellow short sleeve t-shirt)	quần dài màu đen (long black pants)	giày thể thao (sneakers)	?	1,247
nam thanh niên (young man)	?	áo phông xanh (green t-shirt)	quần bò dài xanh đậm (dark blue long jeans)	dép lê màu đen (slipper)	?	53

Each row in the table shows the class ID. Seven columns correspond to seven features for each class in each row. The first six features are the six attribute groups of "Person", "Upper body", "Lower body", "Shoes", "Others". The last column is the Flnweight parameter which is the weight for each class in a row. It is calculated as a sum of the occurrences of NPs in a row except for "Person" attribute. For example, the descriptive sentence "nam thanh niên mặc áo phông xanh, mặc quần bò dài xanh đậm, đi dép lê màu đen" (young man wearing green T-shirt, long jeans, and slippers) has four NPs corresponding to four attribute groups: {Person: nam thanh niên (young

man); Upper body: áo phông xanh (green T-shirt); Lower body: quần bò dài xanh đậm (dark blue long jeans); Shoes: dép lê màu đen (black slippers)}. The "Head" and "Other" attributes are not included in the sentence, and they are presented by "?". The Flyweight for this example sentence is 53 = 22 + 1 + 30; with 22, 1, and 30 values being the total number of occurrences of "áo_phông_xanh" (green t-shirt), "quần bò dài xanh đậm" (dark blue long jeans), "dép lê màu đen" (black slipper) in the dataset, respectively.

- TabTransformer network

The tabular data of NPs will be used for global feature learning by the TabTransformer network. The architecture of the TabTransformer network is shown in figure 3.

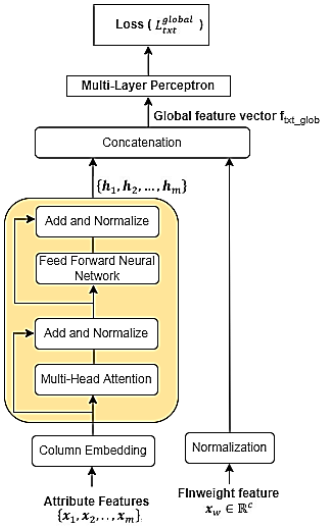

Figure 3. *TabTransformer network for global textual feature embedding.*

The main component of the TabTransformer network is Transformer network which includes multi-head self-attention layers followed by feed-forward layers. It aims to learn efficient global contextual embeddings from attribute features extracted from descriptive sentences. We have the input data as a set of attribute features $\boldsymbol{x}_{attr} \equiv \{\boldsymbol{x}_1, \boldsymbol{x}_2, .., \boldsymbol{x}_m\}$, which are represented in each column in table 1. A column embedding block is designed to embed each of the attribute features into a parametric embedding of dimension d. For each attribute feature x_i ($i \in \{1, .., m\}$), we have an embedding lookup table $\boldsymbol{e}_{\phi_i}(.)$. For ith feature of the d_i classes, the embedding table $\boldsymbol{e}_{\phi_i}(.)$ has $(d_i + 1)$ embeddings where the additional embedding corresponds to a missing value. The embedding for the encoded value $x_i = j \in [0, 1, 2, \ldots, d_i]$ is $\boldsymbol{e}_{\phi_i}(j) = \left[\boldsymbol{c}_{\phi_i}, \boldsymbol{w}_{\phi_{ij}}\right]$, where $\boldsymbol{c}_{\phi_i} \in \mathbb{R}^{\ell}, \boldsymbol{w}_{\phi_{ij}} \in \mathbb{R}^{d-\ell}$, ℓ is a hyper-parameter. $\boldsymbol{c}_{\phi_i}$ is the unique identifier that distinguishes the classes in column i from those in the other columns. Let $\boldsymbol{E}_{\phi}(x_{attr}) = \left\{\boldsymbol{e}_{\phi_1}(x_1), \ldots, \boldsymbol{e}_{\phi_m}(x_m)\right\}$ is the set of embeddings for all the attribute features. Next, these parametric embeddings $\boldsymbol{E}_{\phi}(x_{attr})$ are inputted into the first Transformer layer, and the output of this layer is put into the second Transformer layer, the result from the second Transformer layer will be the input for the third Transformer layer. Each Transformer layer converts input parametric embedding into contextual embedding. We denote the sequence of Transformer layers as a function f_{θ}. The function f_{θ} operates on parametric embeddings $\left\{\boldsymbol{e}_{\phi_1}(x_1), \ldots, \boldsymbol{e}_{\phi_m}(x_m)\right\}$ and returns the corresponding contextual embeddings $\{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_m\}$ where $\boldsymbol{h}_i \in \mathbb{R}^d$. The contextual embedding values $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_m\}$ are concatenated with normalized Flnweight value $\boldsymbol{x}_w \in \mathbb{R}^c$, with c denotes the number of classes to form a global feature vector $\boldsymbol{f}_{txt\_glob}$ of $(d \times m + 1)$ dimension. This global feature vector $\boldsymbol{f}_{txt\_glob}$ is the put into an MLP (Multi Layer Perceptron) to predict the target.

2.2.3. Local textual feature extraction

In order to extract local textual features from the descriptive sentences, we need to encode the NPs extracted from the sentences into feature vectors of integer numbers. A dictionary of NPs is built from the experimental dataset of 3kVNPersonSearch. Each NP is assigned a number in ascending or descending order in the dictionary, and it is be encoded to one-hot vector according to the size of the built-in dictionary. This one-hot vector is then put into Bi-LSTM network to output a local feature vector $\boldsymbol{f}_{txt\_loc}$ of $(1024 \times 1 \times 1)$. The local branch is applied to $\boldsymbol{f}_{txt\_loc}$ to give out a feature vector $\boldsymbol{t}_{loc}$ of (1×256).

2.3. Joint training of image and text

The joint training phase is done in an end-to-end manner, with the loss function calculated as in Eq. (1). It is the combination of six loss functions that are deployed for each individual branch of vision and text and for joint learning of both vision and text.

$$L = L_{img} + L_{img}^{attr} + L_{txt}^{global} + L_{txt}^{local} + \ + L_{match}^{global} + L_{match}^{local} \tag{1}$$

where L_{img} is cross-entropy loss used for training CNN networks to output a probability over the c classes for each image in the visual model as shown in Eq. (2):

$$L_{img} = -\frac{1}{B}\sum_{i=1}^{B} log\frac{e^{W_c^T v_i + b_c^i}}{\sum_{j=1}^{C} e^{W_j^T v_i + b_j^i}} \tag{2}$$

B is batch size; v_i is the ith visual feature of the input image, c is the class of the input image, W_k, W_j are weight matrixes, C is the sum of classes. L_{img}^{attr} is also a cross-entropy loss function for the the task of visual attribute segmentation. It is used to classify the visual attributes as indicated in Eq. (3):

$$L_{img}^{attr} = -\frac{1}{B}\sum_{i=1}^{B} log\frac{e^{W_m^T v_{a_i} + b_m^i}}{\sum_{j=1}^{C} e^{W_j^T v_{a_i} + b_j^i}} \tag{3}$$

where v_{a_i} is the ith visual attribute feature, and m is the total number of visual attributes. L_{txt}^{global} is utilized to learn all the TabTransformer parameters include φ for column embedding, θ for Transformer layers, and ψ for the MLP (Multi-layer Perceptron) layer, which supports multi-class classification by applying Softmax as the output function. It is a cross-entropy loss function (CE) presented in Eq. (4) as follows:

$$L_{txt}^{global} = CE\left(g_\psi\left(f_\theta\left(\boldsymbol{E}_\phi(\boldsymbol{x}_{attr})\right), \boldsymbol{x}_w\right), y\right) \tag{4}$$

where y is target label returned by a MLP layer for each $\boldsymbol{f}_{txt\_glob}$ from the Tabtransformer network.

The loss functions L_{match}^{global} and L_{match}^{local} for global and local feature matching are calculated by Eq. 5 as follows:

$$L_{match} = \frac{1}{B}\sum_{i=1}^{B}\{log[1+P] + log[1+Q]\} \tag{5}$$

where $P = e^{-\tau_p(cos\gamma_i^+ - \alpha)}$ and $Q = e^{-\tau_n(cos\gamma_i^- - \beta)}$; τ_p and τ_n parameters adjust the slope of gradient ($\tau_p = 10$ and τ_n= 40); $cos\gamma_i^+$ is cosine similarity of a positive pair $\left(v_{glob}^i, t_{glob}^+\right)$ for global features or a local negative pair $\left(v_{loc}^i, t_{loc}^+\right)$; i indicates the identity index of a class, t^+ and t^- or $v^+$and v^- are the positive (+) and negative (−) samples of the descriptions respecting the

image v^i or the images respecting the description t^i, respectively. The positive samples are the descriptions of the images, or the images belong to the descriptions of the same class. The negative samples are equivalent to those of different classes.

3. EXPERIMENT AND RESULT

3.1. Dataset and experimental settings

3.1.1. Dataset

The 3000VnPersonSearch [5] dataset is used to evaluate the model. The dataset includes 3,000 person IDs, 6,302 person images, and 12,604 description sentences. The image data set is collected from surveillance cameras in crowded areas during the day and night, with the main recording taking place during the day and in normal weather conditions with stable lighting. The person detection is done on the recorded videos by a manual tool called LabelImg and mostly by an automatic method called YOLOv8. Each person's image will be described by two different people to ensure the diversity of descriptive language, and the description focuses on the person's appearance. Figure 4 shows some samples in the 3000VnPersonSearch dataset.

ID	Image	Descriptions
101		Phụ nữ mặc áo đỏ, quần dài đen tóc ngắn ngang vai (Woman wear red shirts and black pants with shoulder-length hair)
		Cô gái mặc áo cộc tay đỏ, tóc màu đen ngắn ngang vai, quần màu đen, chân đi dép quai hậu (The girl wear a red sleeveless shirt, shoulder-length black hair, black pants, and sandals)
1007		Nam tóc ngắn màu đen, đeo khẩu trang màu xám, mặc áo phông màu cam cổ màu đen cộc tay, mặc quần đùi màu lông chuột, đi giầy thể thao màu đen đế giầy màu trắng, tay phải cầm tay một người khác mặc áo màu vàng. (Man with short black hair, gray mask, orange T-shirt with black collar, short-sleeved, gray shorts, black sneakers with white soles, right hand holding another person wearing a shirt yellow)
		Nam mặc áo màu đỏ cổ áo màu đen, đeo khẩu trang màu ghi, mặc quần đùi màu đen, đi giày màu xám (Man wear red shirt with black collar, gray mask, black shorts, gray shoes)

Figure 4. *Some samples in the 3000VnPersonSearch dataset.*

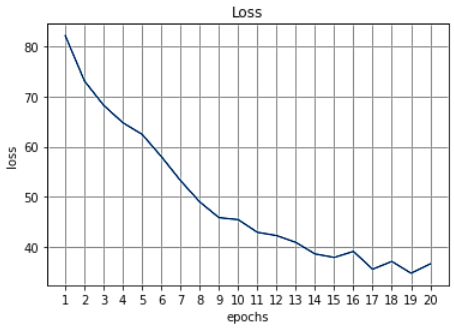

Figure 5. *The loss values at different epochs.*

The dataset is split for the experiment in the ratio of 2,370 IDs with 10,080 images, 20,160 descriptive sentences for the training set; the validation set contains 314 IDs including 258 images, 2,516 descriptions; and the test set is 316 IDs with 1,262 photos and 2,524 descriptions.

3.1.2. Configuration parameters and evaluation metric

- Configuration parameters

All the images are resized to 384 × 128. The weight decay is set as 4 x 10^{-5}, the batch size B = 64, and the learning rate is initialized at 1x10^{-4}. Based on the observations of loss values during training, the average loss for epochs from 17 onwards almost changes very little (figure 5), so the experiments will use the training epoch parameter = 20.

For the TabTransformer, the hidden (embedding) dimension, the number of multi-head self-attention layers, and the number of attention heads for each layer are fixed to 32, 6, and 8, respectively. The MLP layer sizes are set to $\{4 \times l, 2 \times l\}$, where l is the size of its input. The number of attribute features across the dataset ranges from 2 to 136. The dictionary size of NPs-Tab for one-hot vector representation is 2708. The experiments are implemented on Intel(R) Xeon(R) CPU @ 2.00GHz, RAM 12.7GB, GPU 15GB.

- Evaluation metric

In this study, the effectiveness of text-based person search is assessed using top-k ranking. The proposed system produces a list of person images that were found for each query sentence, sorted by confidence score. If the corresponding person's image appears in the top-k images, a relevant person retrieval has been accomplished. The Top-1, Top-5, and Top-10 accuracies are reported in our experiments.

3.2. Experimental results

3.2.1. The effectiveness of the text embeddings by TabTransformer

We take contextual embeddings from different layers of the Transformer and compute a t-SNE plot to visualize their similarity in function space. More precisely, for the experimental dataset, we take its test data, pass their categorical features into a trained TabTransformer, and extract all contextual embeddings (across all columns) from a certain layer of the Transformer. The t-SNE algorithm is then used to reduce each embedding to a 2D point in the t-SNE plot. Figure 6 shows the 2D visualization of "shoes" attribute features that are embedded by Transformer network. It can be seen from Fig. 6 that the clustering results are promising for text embeddings by TabTransformer. This demonstrates the effectiveness of TabTransformer network for representing textual features. It helps to boost the classification performance of the proposed framework for Vietnamese text-based person search.

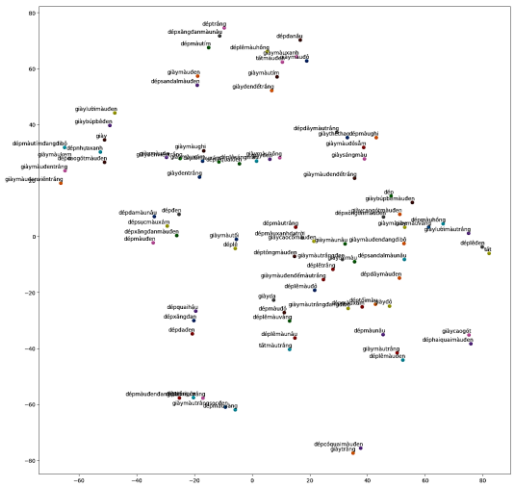

Figure 6. *T-SNE plots of "Shoes" features learned by Transformer network in the test set.*

3.2.2. The effectiveness of text-based person search

Two experimental scenarios are done to show the performance of text-based person searches by our proposed solution. In the first scenario, the person search model is trained without using TabTransformer to extract textual global features. The NPs are not represented as tabular data; they are encoded as one-hot vectors. These one-hot vectors are put into the Bi-LSTM network for textual feature learning. In the second scenario, the NPs are in tabular form and put into the TabTransformer network to get the global feature vector in the text processing branch.

Table 2. *The results of the texted-based person search for two experimental senarios: scenario 1 with NPs are encoded to one-hot vector, scenario 2 with NPs are represented in tabular data and embedded by TabTransformer network.*

Experimental scenarios	**Top-k ranking**		
	R@1	**R@5**	**R@10**
Scenario 1	46,197%	73,851%	83,677%
Scenario 2	53,724%	82,448%	89,540%

The comparative results are shown in table 2 at rank 1, rank 5, and rank 10. It can be seen that our proposed framework (Scenario 2) gains higher classification performance than the baseline method (Scenario 1). At Ranks 1, 5, and 10, the classification accuracy results for scenario 2 are 7.527%, 8.597%, and 5.863% higher than the ones for scenario 1, respectively. In addition, the computing time of our method (Scenario 2) for training with 20 epochs and batch size = 64 is 3h25m. Meanwhile, the computing time of the baseline method (Scenario 2) is 3h32m.

4. CONCLUSIONS

In this paper, we explore the tabular data representation for NPs that are extracted from the descriptive sentences of text-based person searches. The TabTransformer network is proposed for learning global textual features from the tabular data of NPs. The experimental results show the efficiency of our proposed solution for global textual feature embeddings of input sentences in text-based person search. In future work, we will continue to deploy the proposed solution for the remaining processing branches of the proposed framework (visual global and local feature encoding and local textual feature learning). The experiments on a larger dataset will also be implemented and compared to other transformer-based methods.

REFERENCES

[1]. Li, Shuang, Tong Xiao, Hongsheng Li, Bolei Zhou, Dayu Yue, and Xiaogang Wang. *"Person search with natural language description."* In Proceedings of the IEEE conference on computer vision and pattern recognition, (2017).

[2]. Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. *"An image is worth 16x16 words: Transformers for image recognition at scale."* arXiv preprint arXiv:2010.11929, (2020).

[3]. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. Bert. *"Pre-training of deep bidirectional transformers for language understanding"*. arXiv preprint arXiv:1810.04805, (2018).

[4]. Huang, Xin, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. *"Tabtransformer: Tabular data modeling using contextual embeddings. arXiv 2020."* arXiv preprint arXiv:2012.06678, (2012).

[5]. Pham, Thi Thanh Thuy, et al. *"Towards a large-scale person search by vietnamese natural language: dataset and methods."* Multimedia Tools and Applications 81.19: 27569-27600, (2022).

[6]. Yan, Shuanglin, Neng Dong, Liyan Zhang, and Jinhui Tang. *"Clip-driven fine-grained text-image person re-identification."* arXiv preprint arXiv:2210.10276, (2022).

[7]. Jiang, Ding, and Mang Ye. *"Cross-Modal Implicit Relation Reasoning and Aligning for Text-to-Image Person Retrieval."* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2787-2797, (2023).

TÓM TẮT

Biểu diễn văn bản dạng bảng cho tìm kiếm người dựa trên ngôn ngữ tiếng Việt

Tìm kiếm người dựa trên văn bản tiếng Việt vẫn là một bài toán đầy thách thức với bộ dữ liệu mô tả tiếng Việt còn hạn chế. Cách tiếp cận phổ biến hiện nay cho vấn đề này là DNN và gần đây, mạng Transformer đã được ưa chuộng hơn vì hiệu suất vượt trội so với mạng CNN và RNN cho cả nhiệm vụ xử lý ngôn ngữ tự nhiên và thị giác máy tính. Tuy nhiên, DNN hoặc mạng Transformer yêu cầu một lượng lớn dữ liệu huấn luyện và năng lực tính toán để học hiệu quả các đặc trưng ảnh và ngôn ngữ. Điều này đặt ra gánh nặng cho việc triển khai tìm kiếm người dựa trên văn bản tiếng Việt bằng DNN hoặc Transformer. Hướng tới xây dựng hệ thống tìm kiếm người dựa trên văn bản tiếng Việt trên nguồn dữ liệu hạn chế gồm các câu mô tả tiếng Việt với chi phí tính toán thấp, trong bài báo này chúng tôi đề xuất áp dụng kiến trúc dựa trên Transformer có tên TabTransformer để nhúng ngữ cảnh các cụm danh từ được tách ra từ câu mô tả tiếng Việt. Đây là lần đầu tiên mạng TabTransformer được triển khai cùng với kiến trúc CNN và RNN cho việc tìm kiếm hình ảnh dựa trên câu mô tả tiếng Việt. Kết quả thử nghiệm trên tập dữ liệu hạn chế 3000VnPersonSearch cho thấy độ chính xác nhận dạng của phương pháp đề xuất tốt hơn so với phương pháp cơ sở khoảng 7.5% ở Rank 1. Ngoài ra, thời gian tính toán của phương pháp đề xuất hiệu quả hơn phương pháp cơ sở.

Từ khóa: Tìm kiếm người dựa trên truy vấn văn bản; Dữ liệu dạng bảng; TabTransformer; CNN; Bi-LSTM.