An approach to data tag optimization based on self-attention generative adversarial nets

Yuan Chen a, Yintao Hou b, Bing Zhang c
Department of Information and Communication, National Defense University of Science and Technology, Wuhan, Hubei, China

a jsq_cy@163.com, b dongpo004@163.com, c yita1008@163.com

Abstract. In order to solve the problem of low performance caused by errors and omissions in tags in the data resources, this paper proposes a tag optimization method based on self-attention generative adversarial nets (SAGAN). First, use TF-IDF training to calculate the similarity between texts, then use the consistency between the similarity between texts and tag similarity to define the objective function. Second, add correction terms to reduce the bias of user-provided tags before and after optimization. Finally, we propose to apply Self-Attention GAN to further improve the performance of this optimization objective function, and the results are compared with the original tags. Compared with the original tag, the optimized tagging performance has been improved. The test results show that the tag is optimized and improved, and the performance problem in tag recommendation is solved.

1. Introduction

At present, network provides a large number of user-labeled tags for file information, making semantic tag-based data retrieval a popular and practical method in many application scenarios [1], such as tag-based contents. Similarity calculation, recommendation of similar content lists based on user query, etc., can meet the needs of different groups in different environments.

At the same time, the National Strategy for Information Sharing and Safeguarding (NSISS) identifies data tagging as a Priority Objective (PO) critical to the ability to both locate information and enable automated access control decisions and explore valuable information among data and push specific information to specific man[1]. They use manual tags as an important mechanism for retrieving contents and navigation. Due to the randomness and ambiguity of network user tags, data retrieval or recommendation systems usually have the problem of “cold start”. Therefore, the use of statistical learning algorithms for data tag optimization is particularly important in the present.

Since Goodfellow proposed Generative Adversarial Networks, various GAN-based derivative models have been proposed. GAN has become a hot research direction in the field of artificial intelligence. The research focuses on applying GAN’s derivative model, such as infoGAN[2] to train this network to find the mapping between the data features and the semantic features of the data, thus achieving the labeling of the content tags.

Despite of these great progresses, there still exist some challenging cases, such as ambiguities caused by tag missing and tag misunderstanding. In this work, we propose an approach to data tag optimization base on SAGAN[3] which combine a self-attention mechanism into convolutional GANs. The self-
attention mechanism is complementary to convolutions and helps with modeling long range, multi-level dependencies across data regions. Armed with self-attention, the tag optimization method can get fine scores at most factors are carefully coordinated with fine performance in experiment.

2. Data “tags” are metadata

Data “tags” are metadata—“data about data” applied to resources. A “tag” is an assertion describing some aspect of a resource, pairing a semantic label (or “tag name”) with a corresponding tag value. The tag consists of both the name and the value, illustrated in Figure 1.

A “tag” consists of both the "tag name"and the "tag value"

Tag

Author/Organization= "State Dept"

Tag Name Tag value

Figure 1. Data Tag Concept

For example, if the Tag Class is “Author,” one agency may use a single tag to convey both the author’s organization as well as the specific author, such as “Author=FBI/Agent Smith”. Another agency may use two tags: “AuthorAgency=FBI” and “Author=Agent Smith”. But some descriptions in the tags are inaccurate or wrong, or even missing. In this case, we think that the labels need to be optimized, which can be done through semantic similarity or classification.

Tagging is a process in which a user can give meaningful terms to a resource to facilitate the easy discoverability of the resource. Tags are the nonhierarchical keywords of a resource, i.e., bookmarking, picture, or file. Tagging allows the user to categorize the web resources, such as web pages, blog posts, photos, multimedia images, and so on, based on their content. For example, if many users use the same word to tag an item, the tag will become more popular. Tagging sites are constructed with the data that are produced by users designed for individual management and resource discovery. Thus, the main objective of the tagging system is to structure and manage the web content and to discover the relevant content shared by other users. In web 2.0 applications, a large number of tagging systems is available, e.g., Delicious, Flickr, BibSonomy, Technorati, Last fm, and so forth.[4]

In order to facilitate users to use the function of tags, many websites have the function of tag recommendation. The tag recommendation function mainly has methods based on graph theory and text content. In the graph-based method, collaborative filtering and Folk Rank algorithm similar to PageRank technology are the two main algorithms. However, in the Web 2.0 site, the information is updated quickly, and these graph-based methods seem powerless in the face of words that do not appear in historical data. Among text-based methods, the K nearest neighbor method has attracted much attention. After clustering the text using K-nearest neighbors, the most frequent words in the same category are selected as labels. The disadvantage of this method is that it cannot refine the labels, the granularity is coarse, and the social label recommendation effect is not ideal.

When the tag recommendation algorithm cannot meet the needs of Web 2.0 tags, the optimization of tags appears to be very important. LIU D and others retagging the Web 2.0 photo sharing site. They proposed a Retagging scheme, which uses non-negative matrix factorization to re-recommend tags based on the visual similarity of pictures and the semantic similarity of tags, which can effectively reduce tags that are not related to the content of the picture, and image data collected on the photo sharing site Flickr. The set proves the effectiveness of this scheme. With reference to the above schemes, this study is based on the two measures of content similarity and semantic similarity, and performs text mining based on user-labeled tags and resource profiles.
3. Data tag optimization

3.1. Data preprocessing
Given a set \( X = \{x_1, x_2, x_3, ..., x_n\} \), \( X \) is the collection of all data. Tag set \( T = \{t_1, t_2, t_3, ..., t_m\} \) is all after filtering Collection of tags.

The matrix \( W, W_{i,j} \), is introduced as the content similarity of the data introduction. Similarity between texts is
\[
W_{i,j} = \exp\left(-\frac{||x_i - y_j||^2}{\sigma^2}\right) \tag{1}
\]
Among them, \( \sigma \) is a parameter, and the values of \( x_i \) and \( x_j \) are obtained by constructing a vector space model (VSM) of text based on the TF-IDF algorithm.

3.2. Tag optimization
Analogous to the similarity calculation method proposed by LIN D et al.\[5\], set the final objective function is:
\[
\min L = \sum_{i,j=1}^{n} (W_{i,j} - \sum_{k,l=1}^{m} Y_{k,l} S_{k,l} Y_{j,l})^2 + c \left( \sum_{j=1}^{n} \sum_{l=1}^{m} (Y_{j,l} - \alpha_j \cdot e^{y_{j,l}})^2 e^{-y_{j,l}} \right)
\text{s.t. } Y_{j,l}, \alpha_j \geq 0, i, j = 1, 2, ..., n, k, l = 1, 2, ..., m \tag{2}
\]
Among them, \( c \) is the weighting factor of the correction term, \( Y, \alpha_j \) are unknown parameters.

The meaning of the first half of the objective function is that the difference between the similarity between the text and the semantic similarity of the tag is the smallest, which is that blog with similar content have similar standards. At the same time, for media, the tags found by this method have semantic similarity. In other words, semantically similar tags focus on the same media. The objective function is a correction term in the second half. Considering that user-defined tags are very important in tag selection, it improve the accuracy of tag recommendations. The final activation score level of the node after reaching the destination is computed from the starting activation level and the number of links passed by the node before reaching the destination. The distance taken to reach the destination is needed to determine the activation score from the specified distance from the node. In order to reduce the spreading throughout the network during activation process, special attention is provided on those nodes that are connected to a large number of nodes in the network. In the association IR, activation process is carried out based on the specified inference rules.

3.3. Implement attention mechanism
Self-attention exhibits a better balance between the ability to model long-range dependencies and the computational and statistical efficiency. The self-attention mechanism calculates response at a position as a weighted sum of the features at all positions, where the weights or attention vectors are calculated with only a small computational cost.

We call our particular attention "Scaled Dot-Product Attention" (Figure 2). The input consists of queries and keys of dimension \( d_k \), and values of dimension \( d_v \). We compute the dot products of the query with all keys, divide each by \( \sqrt{d_k} \), and apply a softmax function to obtain the weights on the values.
In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix $Q$. The keys and values are also packed together into matrices $K$ and $V$. We compute the matrix of outputs as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$  

The two most commonly used attention functions are additive attention [5], and dot-product (multiplicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of $\frac{1}{\sqrt{d_k}}$. Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.

While for small values of $d_k$ the two mechanisms perform similarly, additive attention outperforms dot product attention without scaling for larger values of $d_k$. We suspect that for large values of $d_k$, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients[6]. To counteract this effect, we scale the dot products by $\frac{1}{\sqrt{d_k}}$.

As mentioned above, although data tag optimization has been significantly improved by deep learning, still, all the difficulties lie in occlusion, overlapping with other people, or clutter background. In such cases, the model may similar features which belong to the background. Recent works try to improve the performance of data tag optimization, which are shown to be efficient, since such processes are indeed to learn structural constraints of human body joints. As stated in [7], the self-attention mechanism is powerful to model.

4. Experiment

We used web crawler technology to crawl about 6850 blog tag data released in 2009-2019 from Sina Blog. Stored in the database, containing 5,750 Chinese blog profiles and about 8,000 tags. Filter these tags to filter out labels for noun attributes. In the end, 2990 related independent tags are obtained. In the content similarity calculation, the parameter is set to the average of the Euclidean distances between all pairs of text. The size of the parameters will affect the results of the model. Too large values will cause the correction algorithm to have too little effect on the results, and the optimization effect will not be good. Too small a value will cause the original site's blog tags to be ignored. After experiments, the results can take into account the user's initial label settings, and reflect the superiority of the optimization algorithm.
4.1. Training

4.1.1. Training Data and Batching. We trained on the standard words and sentences are encoded using byte-pair encoding, which has a shared source target vocabulary of about 3900 tokens. Sentence pairs were batched together by approximate sequence length. Each training batch contained a set of sentence pairs containing approximately 2500 source tokens and 2500 target tokens.

4.1.2. Hardware and Schedule. We trained our models on one machine with NVIDIA GPUs. For our base models using the hyper-parameters described throughout the paper, each training step took about 0.4 seconds. We trained the base models for a total of 20,000 steps or 24 hours, step time was 1.0 seconds.

4.1.3. Optimizer. We used the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.98$ and $\epsilon = 10^{-9}$. We varied the learning rate over the course of training, according to the formula:

$$lr_{rate} = a_{model}^{-0.5} \times \min(\text{step}_{num}^{-0.5}, \text{step}_{num} \times \text{warmup}_{steps}^{-1.5})$$ (4)

4.1.4. Regularization. We employ three types of regularization during training: to the output of each sub-layer, before it is added to the sub-layer input and normalized. In addition, we apply dropout to the sums of the the content of the experiment includes comparing the effects of the method without the correction term, the benchmark method and the method.

4.1.5. Performance Evaluation Metrics. This section presents the three most widely used metrics for evaluating the performance. These standard information retrieval metrics include Precision, Recall, and F-measure.[8]

$$\text{Precision} = \frac{|\text{relevant tags}| \cap |\text{retrieved tags}|}{|\text{retrieved tags}|}$$ (5)

$$\text{Recall} = \frac{|\text{relevant tags}| \cap |\text{retrieved tags}|}{|\text{relevant tags}|}$$ (6)

$$\text{F} \cdot \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$ (7)

4.2. Results

Manual tagging is used to select blog tags. According to the above algorithm, five optimal tags were obtained, and the volunteers voted. Finally, the correct tags were determined based on the results they jointly selected. Select 1000 groups of blogs that have become popular in recent years to form an evaluation set. The comparison of the precision, recall and F value of the three methods is shown in the figure 3.

As can be seen from Figure 3, in terms of accuracy, the method without the correction term does not refer to the tags given by users, and recommends tags directly through the blog profile, resulting in lower accuracy; in the benchmark method, users on the website gave the tag is more random and the accuracy is not high; this research method combines the blog profile and the tag given by the user, and the accuracy has been significantly improved. In terms of recall, the method without the correction term is recommended only through the blog profile, and it is almost impossible to recommend all the correct tags; in the benchmark method, the tags given by users can cover most of the correct tags; this research method combines the previous Both methods have also improved the recall rate. Due to environmental constraints, we have not introduced the attention mechanism into infoGAN, which will be our next research direction.
5. Conclusion

Aiming at the problem of missing and wrong tags, this study uses two measures based on content similarity and semantic similarity to optimize tags. This algorithm takes into account the original tags and data content provided by the user, so that the tag content is closer to the data content, and the accuracy of the tag improved after optimization. Experiments show that the algorithm can not only add missing tags, but also correct wrong tags, making the optimized tags more consistent with the data content.

We propose to apply self-attention GAN to further improve the performance of tag optimization objective function, and the results are compared with the original tags. Compared with the original tag, the optimized tagging performance has been improved. The test results show that the tag is optimized and improved, and the performance problem in tag recommendation is solved.

![Comparison and Performance Evaluation](image)

Figure 3. Comparison and Performance Evaluation

References

[1] Direct of national intelligence, PO3-Data-Tagging-Functional-Requirements, https://www.dni.gov/files/ISE/documents/DocumentLibrary/PO3-Data-Tagging-Functional-Requirements.pdf

[2] Chen X, Duan Y, Houthooft R, et al. Infogan: Interpretable representation learning by information maximizing generative adversarial nets[C]. Advances in neural information processing systems, 2016: 2172-2180.

[3] Zhang H, Goodfellow I, Metaxas D, et al. Self-attention generative adversarial networks[J], 2018.

[4] Vairavasundaram S, Varadharan V, Vairavasundaram I, et al. Data mining - based tag recommendation system: an overview[J], 2015, 5(3): 87-112.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in neural information processing systems, 2017: 5998-6008.

[6] Qian S, Peng F, Lu J J O S U. Tag optimization based on semantic similarity[J], 2015(2): 7.

[7] Cao Z, Wang R, Wang X, et al. Improving Human Pose Estimation with Self-Attention Generative Adversarial Networks[C]. 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2019: 567-572.

[8] Maimon O, Rokach L. Data mining and knowledge discovery handbook[J], 2005.

[9] Page L, Brin S, Motwani R, et al. The PageRank citation ranking: Bringing order to the web[R]. Stanford InfoLab, 1999.

[10] Chen P, Shao X. Audio auto-tagging based on generative adversarial networks[J], 2018, 10(6): 754.

[11] Chen X, Duan Y, Houthooft R, et al. SAGAN: Interpretable representation learning by information maximizing generative adversarial nets[C]. Advances in neural information processing systems, 2016: 2172-2180.

[12] Bai J, Bu Y. An Improved Algorithm for Semantic Similarity Based on HowNet[C]. 2018 2nd
International Conference on Data Science and Business Analytics (ICDSBA), 2018: 65-70.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.

[14] Dosovitskiy A, Tobias Springenberg J, Brox T. Learning to generate chairs with convolutional neural networks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1538-1546