Article De-duplication Using Distributed Representations

Shumpei Okura
Yahoo Japan Corporation
Tokyo, Japan
sokura@yahoo-corp.jp

Yukihiro Tagami
Yahoo Japan Corporation
Tokyo, Japan
yutagami@yahoo-corp.jp

Akira Tajima
Yahoo Japan Corporation
Tokyo, Japan
atajima@yahoo-corp.jp

ABSTRACT

In news recommendation systems, eliminating redundant information is important as well as providing interesting articles for users. We propose a method that quantifies the similarity of articles based on their distributed representation, learned with the category information as weak supervision. This method is useful for evaluation under tight time constraints, since it only requires low-dimensional inner product calculation for estimating similarities. The experimental results from human evaluation and online performance in A/B testing suggest the effectiveness of our proposed method, especially for quantifying middle-level similarities. Currently, this method is used on Yahoo! JAPAN's front page, which has millions of users per day and billions of page views per month.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; I.2.6 [**Artificial Intelligence**]: Learning

Keywords

De-duplication, News recommendation, Neural network

1. INTRODUCTION

In news distribution systems, we often have multiple articles about the same event that have been provided at about the same time. In this case, if articles are presented to users in their interest level order, these articles tend to be displayed close to each other, and it is of concern that their satisfaction will be decreased by continuously looking at similar articles. Therefore, for example, it would be effective to select only the representative article and not display the other similar ones.

Though very similar articles can be detected using the co-occurrence of the words in them, however, it is difficult to measure correct similarity in the case that abbreviations are used or they are written in different styles.

On the other hand, additional information attached to the article, such as categories and tags, is also useful for detecting of similarity. This additional information is more robust and stable than word-level information, but the granularity is not fine enough for this purpose.

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2889355.

In this paper, we propose a method for converting bag of words vector for an article into a low-dimensional vector by considering the similarity of its category. This vector is useful in quantifying similarity that is vaguer than the co-occurrence of words and is more specific than categories.

Whereas unsupervised methods such as Paragraph Vector [1] can be used for this purpose, our method generates vectors in a supervised way so that the inner product of vectors represents their similarities, in order to be suitable for fast calculation on on-the-fly systems.

We now explain the results of using these vectors for de-duplication of articles in the news distribution system for Yahoo! JAPAN's front page.

2. METHODS

Generating distributed representation. We propose a method for generating distributed representation vectors based on the denoising auto-encoder [3] with weak supervision. The traditional denoising auto-encoder is formulated as follows:

$$\begin{aligned} \tilde{\boldsymbol{x}} &\sim C(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \\ \boldsymbol{h} &= f(\boldsymbol{W}\tilde{\boldsymbol{x}} + \boldsymbol{b}) \\ \boldsymbol{y} &= f(\boldsymbol{W'}\boldsymbol{h} + \boldsymbol{b'}) \\ \theta &= \underset{\boldsymbol{W},\boldsymbol{W'},\boldsymbol{b},\boldsymbol{b'}}{\arg\min} \sum_{\boldsymbol{x}\in X} L(\boldsymbol{y}, \boldsymbol{x}) \end{aligned}$$

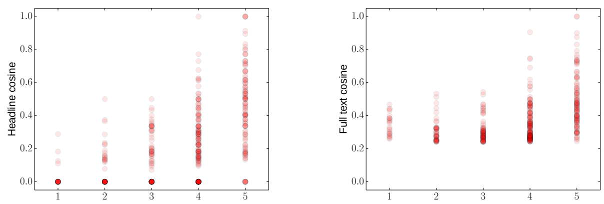
where $\boldsymbol{x} \in X$ is the original input vector, f is the activation function, L is the loss function, and C is corrupting distribution.

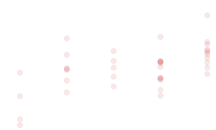
Usually, $\boldsymbol{h}$ is used as a representation vector corresponding to $\boldsymbol{x}$. However, $\boldsymbol{h}$ holds only the information of $\boldsymbol{x}$. We want to interpret that $\boldsymbol{h_1}^{\mathrm{T}}\boldsymbol{h_2}$ is larger if $\boldsymbol{x_1}$ is more similar to $\boldsymbol{x_2}$. To that end, we use a triplet $(\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}) in X^3$ as input for training and modify the objective function to preserve their categorical similarity as follows:

$$\begin{aligned} \boldsymbol{h_n} &= f(\boldsymbol{W}\tilde{\boldsymbol{x}}_{\boldsymbol{n}} + \boldsymbol{b}) - f(\boldsymbol{b}) & (1) \\ \phi(\boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{h_3}) &= \log(1 + \exp(\boldsymbol{h_1}^{\mathrm{T}}\boldsymbol{h_3} - \boldsymbol{h_1}^{\mathrm{T}}\boldsymbol{h_2})) & (2) \end{aligned}$$

$$\boldsymbol{\theta} = \underset{\boldsymbol{W},\boldsymbol{W'},\boldsymbol{b},\boldsymbol{b'}}{\arg\min} \sum_{(\boldsymbol{x_1},\boldsymbol{x_2},\boldsymbol{x_3})\in T} \sum_{n=1}^{3} L(\boldsymbol{y_n}, \boldsymbol{x_n}) + \alpha\phi(\boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{h_3})$$

where $T \subset X^3$, such that $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ in the same category/similar categories and $\boldsymbol{x_1}$ and $\boldsymbol{x_3}$ in different categories. By Eq.(1), $\boldsymbol{h}$ satisfies the property $\boldsymbol{x} = \boldsymbol{0} \Rightarrow \boldsymbol{h} = \boldsymbol{0}$.





Label	Criteria
5	Almost same
4	For same event
3	For related event
2	For similar genre
1	Not similar

Figure 1: Editor label vs. each similarity.

This means that an article that has no available information is not similar to any other articles. The notation ϕ is penalty function for article similarity corresponding to categorical similarity, and α is a hyper parameter for balancing.

We use the elementwise sigmoid function $\sigma(x)_i = 1/(1 + \exp(-x_i))$ as f, the elementwise cross entropy as L, and masking noise as C. We train the model $\boldsymbol{\theta}$ using mini-batch stochastic gradient descent.

Deduplicating articles. We use a greedy algorithm to determine whether to display an article. An ordered list of candidate articles is given by another ranking system. We basically display articles in given order except for certain articles shown below.

To decide whether to display an article, we calculate the similarities between the current article and all articles displayed previously. If the maximum value of similarities is greater than the threshold, we do not display that article.

This calculation must be done in a short time because it is required for each request from front page. When we use word-based similarity, we can quickly estimate similarities by calculating auxiliary values such as b-Bit Minwise Hashing [2], in advance. On the other hand, estimating our similarities requires only simple calculation of the low-dimensional inner product of $\boldsymbol{h}$s.

3. EXPERIMENT

Training. For training, We used about 400k articles that were posted to Yahoo! JAPAN's front page in March 2015. We used the top 10k nouns frequently used, except for stop words, as the vocabulary. Input vector $\boldsymbol{x} \in X$ was a binary vector that had 10k dimensions corresponding to each word in the vocabulary. The representation vector $\boldsymbol{h}$ had 500 dimensions. The corruption rate for C was 0.3.

Offline evaluation. For qualitative evaluation of representation vectors, we prepared about 400k articles that were posted in September 2015. We then made pairs of articles posted on the same day, because the articles that were posted on different days were not displayed at same time. However, most pairs made in this way were unrelated. Therefore, we used the top 0.2% pairs in order of cosine similarity of words in the full text for the following evaluation.

We asked editors to label articles from 1 to 5 based on the criteria (see Figure 1) for each pair of articles. The annotation results for 400 pairs are shown in Figure 1.

Text-based cosines tend to have high values for pairs labeled 4 and 5, but the labels and scores had less correlation for pairs labeled 1-3. In particular, similarity of many pairs was just zero based on headline cosines. On the other hand, our vector cosine values increased gradually with labels, especially for pairs labeled 1-3. For each similarity score, we showed AUC of binary classification problems of estimating the "label $\geq n$" in Table 1. Our method produced significantly good results for $n = 2, 3$, as we observed above.

ID	Similarity	≥ 2	≥ 3	≥ 4	$= 5$
1	headline cosine	0.832	0.806	**0.778**	0.811
2	full text cosine	0.521	0.646	0.719	**0.813**
3	vector cosine	**0.940**	**0.829**	0.749	0.803

Table 1: AUC for editor label.

Online evaluation. We verified the effectiveness of our method by A/B testing on Yahoo! JAPAN's front page for smartphones. For each user, a list of candidate articles were extracted in relevance order against his/her historical activities. While users can see up to 200 articles if they scroll down to the bottom, most see the top 10-20 articles only, resulting in various depth values among sessions. We observed CTR (#clicks/#imps), depth (#imps/#sessions) and module CTR (#clicks/#sessions) by changing skip conditions.

The experimental results are summarized in Table 2. Conditions 1 and 3 involved almost the same amount of skipping; more articles were skipped in 4 and less in 2. Conditions 2-4 increased mCTR by 2-3% compared to Condition 1. This means users were able to find more interesting articles in each session with our method.

By adopting a low threshold for Conditions 2-4, we can see that mCTR increased slightly, even though the depth decreased. This means the threshold worked effectively to adjust the de-duplication strength, and strong de-duplication was able to efficiently provide information. This suggests that the similarity estimation for pairs labeled 2 and 3 for offline evaluation is important for effective de-duplication.

ID	Skip condition	CTR[%]	depth[%]	mCTR[%]
1	headline cosine > 0.40	$+0.00$	$+0.00$	$+0.00$
2	vector cosine > 0.60	-2.78	$\mathbf{+5.25}$	$+2.32$
3	vector cosine > 0.50	-0.60	$+3.31$	$+2.69$
4	vector cosine > 0.45	$\mathbf{+1.36}$	$+1.61$	$\mathbf{+2.99}$

Table 2: A/B testing results.

In response to the results of these experiments, on all the traffic of Yahoo! JAPAN's front page on smartphone, we have incorporated this method instead of the traditional word co-occurrence-based method.

4. REFERENCES

[1] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014.

[2] P. Li and C. König. b-bit minwise hashing. In *WWW*, 2010.

[3] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.