March 30, 2018

Automatic generation of native ad styles

Bo Lin

Automatic generation of native ad styles

ABSTRACT

Native advertisements mimic the look and feel of a publisher's content slots and are used to monetize content inventories. Currently, native ad styles are based on hand-written rules and heuristic which can result in ad styles that are not consistently high-quality. This disclosure describes the use of machine learning models to automatically generate native ad styles for online or mobile publishers. For a given content slot and elements therein, native ad styles are automatically generated that closely resemble publisher style, match page context, and perform well. Local and global features of the content slot and elements are used to generate the native ad. The techniques can automatically incorporate new abstractions.

KEYWORDS

- Native ads
- Content slot
- Document Object Model
- HTML
- Machine learning

BACKGROUND

A native ad is a form of advertisement or other branded content that follows the natural form and function of editorial content of a publication. Online publishers, e.g., publishers that target mobile devices, often, even exclusively, turn to native advertising to monetize their content inventories. Current methods of generating native ads rely on hand-written rules and heuristics, which don't always produce native styles of high quality. Techniques of

automatically generating native styles of high quality are of value to publishers and ad networks.

DESCRIPTION

Native ads typically have a defined set of components, e.g., headline, body, URL or advertiser-name, etc. The problem of automatic native ad generation can be described as follows: given an arbitrary content slot from a publisher page, generate a suitable set of native ad components for that content slot. Essentially, the automatic native ad generator identifies the native ad components that best fit a content slot. Once identified, styles from the content-slot elements are extracted and applied to the native ad component to generate native ads that match the content slots.

In the context of HTML pages, the problem of native ad generation is formulated as follows: given a HTML Document Object Model (DOM) element, identify the most suitable native ad component for the element. However, a given HTML DOM element may be mapped differently based on context. For example, suppose that text with the largest font-size in a certain content slot is mapped to the headline element of an ad. Further suppose that text with the second largest font-size is mapped to the body element of the ad. Then a piece of text with, e.g., 14-point font-size, can be mapped to headline in one context or content slot and to body in a different context or content slot. In this manner, an HTML DOM element can be considered suitable as body text in one context but as a headline in another context.

Thus, the problem of native ad generation is further clarified as follows. Given a context (content slot) and a HTML DOM element *within that context*, identify the most suitable native ad element. In some implementations, the elements of a given content slot are looped over, and a decision is made as to the mapping for each such element.

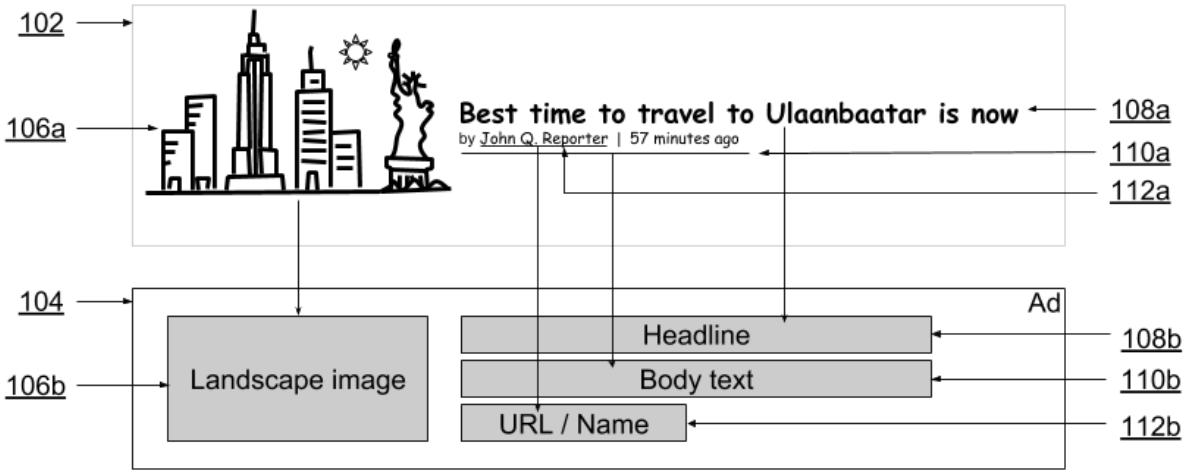

Fig. 1: Mapping elements from publisher content slot to a native ad

Fig. 1 illustrates an example mapping between elements of a content slot (102) and components of a native ad (104). In the example of Fig. 1, an image with a landscape orientation (106a) in the content slot is mapped to a landscape image (106b) in the native ad, the headline (108a) in the content slot is mapped to the headline (108b) in the native ad, the byline (110a) in the content slot is mapped to the body text (110b) in the native ad, and the author-name (112a) in the content slot is mapped to a URL or advertiser name in the native ad.

In mapping a content slot to a native ad, e.g., as in Fig. 1, a rule-based model typically extracts features for each element, e.g., font-size, font-weight, line-height, whether it is an anchor tag, etc. The rule-based model uses human specified rules to make the classifications based on the features. Rules are constructed such that an element is not analyzed for features in isolation. Rather the element is compared with other elements within the context (content slot). Thus, the rule-based model uses information from the context in addition to the features from the element itself while making classifications.

In this manner, a mapping such as “map the text in the content slot with *heaviest* font weight and certain *relative* text length to the headline element of the native ad” is constructed.

These two types of features, e.g., features of an element and features in relation to the context, are referred to as local features and global features respectively. Rule-based models typically include fallback rules, e.g., if no *body* element is found in the content slot, then use the headline element of the content slot for the body of the native ad, if no URL element is found in the content slot, use the body element, etc.

This disclosure describes machine learning techniques to learn model parameters needed to classify content-slot elements in order to enable content-slot-to-native-ad mapping. HTML DOM elements of a content slot are analyzed in context, e.g., within the content slot that contains the elements to determine local and global features (204). The trained machine learner generates as output the most suitable native ad elements.

The features used as input by the machine-learning model typically relate to text elements, e.g., headline (title), body, URL (advertiser name), since image elements are easily classified by tag and content. Local features of textual HTML DOM elements (e.g., features extracted from the element itself) include, e.g., font size, line height, text length, whether the element is a headline tag, whether the element is an anchor or cite tag, etc. Global features of textual HTML DOM element (e.g., features that depend on the context) include, e.g., is the element the longest text, does the element have the biggest font weight, etc. The local and global features are represented as points in a high-dimensional mathematical space when fed as input to the machine-learning model.

In some implementations, the model comprises multiple layers of a neural network, e.g., a convolutional or recurrent neural network. The model computes a score for each content-slot element that indicates the mapped native ad component. Training data used to train the machine-learning model is manually labeled and can be obtained by parsing a sample set of

websites of premium publishers (with consent). Optionally, the machine-learning model can also take into consideration the location of the element and/or the sequence of elements.

The amount of training data should be at least of the same scale as the complexity of the neural network, sometimes measured with its Vapnik-Chervonenkis (VC) dimension, in order to ensure good generalization. As training data and resources increase, model complexity can grow and quality of inferences can improve further. The machine-learning model can be implemented as a lightweight model that is deployable as part of the client-side (e.g., publisher-side) code for style extraction or at a server.

The models outputs scores for each possible native ad component, e.g., in the format [score_title, score_body, score_URL]. In this manner, the machine-learning model disambiguates mapping of n content-slot element to one of title, body, or URL.

Example: A content-slot element with the following local and global features:

```
font size: 20
line height: 1.2
text length: 38
whether the element is a headline tag: false
whether the element is an anchor or cite tag: true
is the element the longest text: true
does the element have the biggest font weight: false
```

is mapped by the machine-learning model to the following vector:

```
[score_title, score_body, score_URL] = [0, 0, 1]
```

The vector indicates that the content-slot element is best mapped to the URL of the native-ad component.

The trained model maps content-slot elements to native-ad components with high accuracy, e.g., exceeding 90%, which is higher in comparison with the around 60% accuracy typically achievable by rule-based models (as judged by human evaluators in a controlled

experiment). The higher mapping accuracy indicates the high level of native ad match to publisher content. Furthermore, model training using labeled training data ensures that the model captures human knowledge embedded in the labels. While rule-based models can be modified to incorporate new abstractions, this is labor intensive and expensive. In contrast, using machine learning techniques as described herein enables new and non-obvious abstractions to be incorporated automatically. Furthermore, the techniques work not only on text elements, but are also easily extended to other types of elements, including but not limited to text, image, video, etc.

CONCLUSION

This disclosure describes the use of machine learning models to automatically generate native ad styles for online or mobile publishers. For a given content slot and elements therein, native ad styles are automatically generated that closely resemble publisher style, match page context, and perform well. Local and global features of the content slot and elements are used to generate the native ad. The techniques can automatically incorporate new abstractions.