November 16, 2018

Optimization of Optimal Article Content for the Internet for Article A/B Testing

N/A

OPTIMIZATION OF OPTIMAL ARTICLE CONTENT FOR THE INTERNET FOR ARTICLE A/B TESTING

Introduction:

The present disclosure provides systems and methods for optimizing content, and in particular a content optimization model to generate optimized content for the internet. Article content creation is a highly competitive field and requires a high amount of skill and talent to deliver the best content, layout, and formulation of articles for users. Those who create content (e.g., content creators) are often skilled in areas associated with content creation (e.g., writing, philosophy, history, political science, etc.), but can lack skill associated with content optimization (e.g., what works well for users on the internet, how to optimize page layout, etc.). The present disclosure can enable the automatic optimization of content. The systems and methods consistent with the present disclosure can include a machine-learned content optimization model that can optimize content and output the optimized content in response to receiving content data. By automating content optimization in this way, the present disclosure removes the need for pre-publication optimization work, enabling faster publication of content. This can be especially advantageous for content creators who want to be the originator (or breaker) of content.

Summary:

According to aspects of the present disclosure, a content optimization model can include a machine-learned article generation model and a machine-learned traffic allocation model. The machine-learned article generation model can be configured to output article data representing one or more article forms in response to receiving content data. The machine-learned traffic allocation model can be configured to output traffic allocation data representing an allocation of traffic to one or more article forms represented by the article data. The system can determine

and analyze one or more objective for each article form represented by the article data, in order to optimize content for an article. The example embodiment of the present disclosure described below are directed toward optimizing content for text and/or image based articles. However, one skilled in the art can recognize that the systems and methods described herein can be used to optimize other types of content (e.g., video creation, virtual reality segments, etc.).

Detailed Description:

Figure 1 depicts a block diagram of an example computing system 100 for optimizing article content for the internet according to an example embodiment of the present disclosure. The computing system 100 can include a computing device 102, a server computing system 130, and a training computing system 150 that are communicatively coupled over a network 180.

The computing device 102 can be any type of computing device, such as, for example, a personal computing device (e.g., laptop or desktop), a mobile computing device (e.g., smartphone or tablet), a gaming console or controller, a wearable computing device, an embedded computing device, or any other type of computing device. The computing device 102 includes one or more processors 112 and a memory 114. The one or more processors 112 can be any suitable processing device (e.g., a processor core, a microprocessor, an ASIC, a FPGA, a controller, a microcontroller, etc.) and can be one processor or a plurality of processors that are operatively connected. The memory 114 can include one or more non-transitory computer-readable storage mediums, such as RAM, ROM, EEPROM, EPROM, flash memory devices, magnetic disks, etc., and combinations thereof. The memory 114 can store data 116 and instructions 118 which are executed by the processor 112 to cause the computing device 102 to perform operations.

In some implementations, the computing device 102 can store or include one or more machine-learned models 120. For example, the machine-learned models 120 can be or can otherwise include models such as neural networks (e.g., deep neural networks) or other types of machine-learned models, including non-linear models and/or linear models. Neural networks can include feed-forward neural networks, recurrent neural networks (e.g., long short-term memory recurrent neural networks), convolutional neural networks or other forms of neural networks.

In some implementations, the one or more machine-learned models 120 can be received from the server computing system 130 over network 180, stored in the computing device memory 114, and then used or otherwise implemented by the one or more processors 112. In some implementations, the computing device 102 can implement multiple parallel instances of a single machine-learned model 120.

Additionally or alternatively, one or more machine-learned models 140 can be included in or otherwise stored and implemented by the server computing system 130 that communicates with the computing device 102 according to a client-server relationship. For example, the machine-learned models 140 can be implemented by the server computing system 140 as a portion of a web service. Thus, one or more models 120 can be stored and implemented at the computing device 102 and/or one or more models 140 can be stored and implemented at the server computing system 130.

The computing device 102 can also include input component 122 that receives user input. For example, the input component 122 can be a touch-sensitive component (e.g., a touch-sensitive display screen or a touch pad) that is sensitive to the touch of a user input object (e.g., a finger or a stylus). The touch-sensitive component can serve to implement a virtual keyboard. Other example input components include a microphone, a traditional keyboard, camera device, or other means by which a user can provide user input.

The server computing system 130 includes one or more processors 132 and a memory 134. The one or more processors 132 can be analogous to the one or more processors 112 and the memory 134 can be analogous to the memory 114. The memory 134 can store data 136 and instructions 138 which are executed by the processor 132 to cause the server computing system 130 to perform operations.

In some implementations, the server computing system 130 includes or is otherwise implemented by one or more server computing devices. In instances in which the server computing system 130 includes plural server computing devices, such server computing devices can operate according to sequential computing architectures, parallel computing architectures, or some combination thereof.

As described above, the server computing system 130 can store or otherwise include one or more machine-learned models 140. Example machine-learned models include neural networks or other multi-layer non-linear models. Example neural networks include feed forward neural networks, deep neural networks, recurrent neural networks, and convolutional neural networks.

The computing device 102 and/or the server computing system 130 can train the models 120 and/or 140 via interaction with the training computing system 150 that is communicatively coupled over the network 180. The training computing system 150 can be separate from the server computing system 130 or can be a portion of the server computing system 130.

The training computing system 150 includes one or more processors 152 and a memory 154. The one or more processors 152 can be analogous to the one or more processors 112, and the memory 154 can be analogous to the memory 114. The memory 154 can store data 156 and instructions 158 which are executed by the processor 152 to cause the training computing system

150 to perform operations. In some implementations, the training computing system 150 includes or is otherwise implemented by one or more server computing devices.

The training computing system 150 can include a model trainer 160 that trains the machine-learned models 120 and/or 140 stored at the computing device 102 and/or the server computing system 130 using various training or learning techniques, such as, for example, backwards propagation of errors. In some implementations, performing backwards propagation of errors can include performing truncated backpropagation through time. The model trainer 160 can perform a number of generalization techniques (e.g., weight decays, dropouts, etc.) to improve the generalization capability of the models being trained.

The network 180 can be any type of communications network, such as a local area network (e.g., intranet), wide area network (e.g., Internet), or some combination thereof and can include any number of wired or wireless links. In general, communication over the network 180 can be carried via any type of wired and/or wireless connection, using a wide variety of communication protocols (e.g., TCP/IP, HTTP, SMTP, FTP), encodings or formats (e.g., HTML, XML), and/or protection schemes (e.g., VPN, secure HTTP, SSL).

Figure 1 illustrates one example computing system that can be used to implement the present disclosure. Other computing systems can be used as well. For example, in some implementations, the computing device 102 can include the model trainer 160 and the training dataset 162. In such implementations, the models 120 can be both trained and used locally at the computing device 102. In some of such implementations, the computing device 102 can implement the model trainer 160 to personalize the models 120 based on user-specific data.

According to aspects of the present disclosure, the computing device 102 can obtain content data for an article. The content data can include, for example, one or more content portions (e.g., titles, subtitles, introductions, images, conclusions, etc.). In some implementation, the content data can include more than one content portion of the same type. For example, content data can include a plurality of different titles (e.g., title 1, title 2, etc.) for a first article, a plurality of different subtitles (e.g., subtitle 1, subtitle 2, etc.) for the first article, a plurality of introductions for the first article, etc. In some implementations, the content data can include one or more different types of content portions for an article, and one or more content portions for each type. By collecting content data that includes multiple variants at or before the time of article creation, the present disclosure can help avoid having to rehire freelancers for additional content post-publication.

In some implementations, the device 102 can obtain the content data from one or more pre-published templates. The pre-published template(s) can each specify one or more content portions to include as part of an article. For example, a pre-published template can specify at least two different content titles, one subtitle, and at least one image. A user can fill the template with content data (e.g., a first and second title, a subtitle, and a plurality of images), and the device 102 can receive the template and extract the content data from the template.

In some implementations, the device 102 can obtain the content data from one or more content creators (e.g., via the input component 122). As an example, the content creator(s) can fill one or more pre-published templates via the input component 122. As another example, the content creator(s) can directly input data representing the content via the input component 122. As another example, the content creator(s) can provide the device 102 with data indicative of a location of the content data (e.g., a local location or remote network location).

In some implementations, the device 102 can obtain the content data from existing article data representing one or more existing or previously generated article forms. For example, when the machine-learned article generation model generates article data representing one or more article forms, the device 102 can store the article data in the data 116. The system 102 can later retrieve the stored existing article data and extract content data associated with one or more existing or previously generated article forms represented by the existing article data.

According to aspects of the present disclosure, the computing device 102 can obtain metric data. The metric data can include one or more measurements and/or one or more analytics of one or more objectives associated with an article form. The objective(s) can include, for example, time spent on an article, scroll path through a viewport, number of times the article was shared/reposted, number of pages reached, etc. For example, the device 102 can determine one or more objectives to monitor with respect to a first article form, and monitor the objective(s) to determine the metric data associated with the first article form.

According to aspects of the present disclosure, the computing device 102 can generate article data representing one or more article forms based at least in part on content data. In some implementations, the device 102 can include a machine-learned article generator model that can receive content data as an input, and output article data representing one or more article forms based on the content data in response to the input. The device 102 can obtain the content data as described above, and input the content data into the machine-learned article generation model to obtain the article data. In some implementations, the device 102 can train the machine-learned article generation model to generate the article data based on training data representing one or more previously successful article forms (e.g., one or more winning article forms) and/or associated metric data.

In some implementations, the machine-learned article generation model can generate the article data based on content data and layout data. The layout data can represent a plurality of different layout options (e.g., display attributes, display position, image insertion, pagination, slideshows, etc.) for an article. For example, the machine-learned article generation model can determine a first article form and a second article form based at least in part on the layout data. The first article form can include a title displayed at a first position and no subtitle, and the second article form can include a title displayed at a second position and a subtitle. The machine-learned article generation model can determine one or more layout templates based on the layout data (e.g., by selecting one or more content portions and associated layout options, or using on one or more predetermined layout templates). The machine-learned article generation model can fill a layout template with content selected from the content data to generate an article form corresponding to the layout template. For example, the machine-learned article generation model can select one or more content portions from the content data and arrange the content portion(s) according to a first layout template to generate a first article form.

In some implementations, the machine-learned article generation model can generate the article data based on content data and metric data associated with an existing article form. As an example, the device 102 can obtain existing article data and associated metric data. The device 102 can extract content data from the existing article data, and input the content data and the metric data into the machine-learned article generation model. As another example, the device 102 can obtain metric data associated with previously stored data representing an existing article form. The device 102 can retrieve/extract content data from the stored data, and input the content data and the metric data into the machine-learned article generation model. In response, the machine-learned article generation model can adjust the existing article data (e.g. one or more existing or previously generated article forms) based on the metric data, to generate article data representing one or more article forms. In this way, the device 102 can iteratively fine-tune one or more article forms for an article to optimize the content of the article. At each iteration, the machine-learned article generator model can dynamically change one or more content portions and associated layout based on how users are reacting. This real-time correction can be the difference between an article quickly disappearing and one going viral with high search engine optimization.

According to aspects of the present disclosure, the computing device 102 can determine network traffic allocation data for one or more article forms. In some implementations, the device 102 can include a machine-learned traffic allocation model that can receive article data as an input, and output network traffic allocation data representing a network traffic allocation for each article form in the article data in response to the input. The device 102 can obtain the article data when it is generated by the machine-learned article generation model, or the device 102 can obtain existing article data. The device 102 can input the article data into the machine-learned traffic allocation model to obtain network traffic allocation data corresponding to the input. In some implementations, the device 102 can train the machine-learned traffic allocation model to generate the network traffic allocation data based on training data representing one or more previously successful article forms (e.g., one or more winning article forms) and/or associated metric data.

According to aspects of the present disclosure, the computing device 102 can determine a winning article form from a plurality of article forms associated with an article, based at least in part on the metric data. In some implementations, the device 102 can determine one or more objectives to monitor with respect to each article form in the article data, and direct network traffic to each article form based on the network traffic allocation data associated with the article data. The device 102 can monitor the objective(s) to determine metric data associated with each article form. The device 102 can determine when one or more article forms perform well, based on the metric data. For example, the device 102 can determine that an article form performs well when the objective(s) associated with the article form exceed one or more threshold performance values. The device 102 can feedback the metric data to automatically tune the well-performing article forms to further optimize the content. The device 102 can determine a winning article form when the objective(s) associated with an article form exceed a winning threshold performance value. In some implementations, once the winning article is determined, 95% of network traffic can be allocated to the winning article form with a small 5% holdback to one or more other article forms on the off chance that an article form performs better on a longer time horizon.

Drawings:

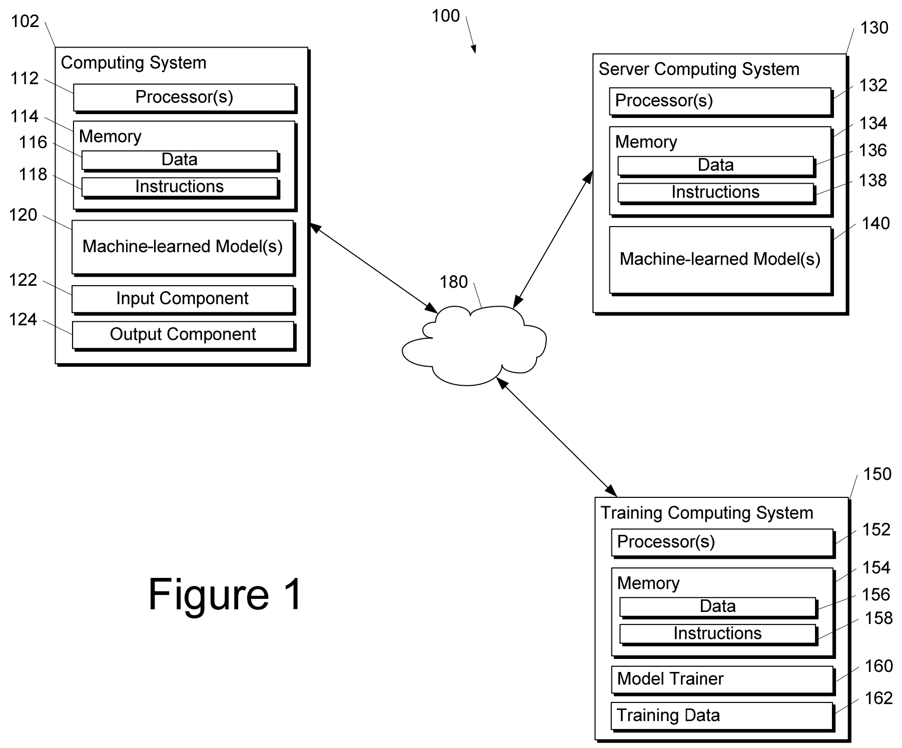

Figure 1

Abstract:

The present disclosure describes systems and methods for generating optimized content for the internet. According to aspects of the present disclosure, a computing system can obtain content data and/or metric data. The computing system can generate article data representing one or more article forms based at least in part on content data, and determine network traffic allocation data for the one or more article forms. The computing system can monitor one or more objectives associated with each article form of an article to optimize content for the article, and/or select a winning article form for the article. Keywords associated with the present disclosure include: computing systems; content creation; content optimization; content layout; machine-learned model; split-testing; optimization objectives.