

Technical Disclosure Commons

Defensive Publications Series

June 26, 2017

SMART LIVE CHAT LIMITER

Kurt Wilms

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Wilms, Kurt, "SMART LIVE CHAT LIMITER", Technical Disclosure Commons, (June 26, 2017)
http://www.tdcommons.org/dpubs_series/565



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

SMART LIVE CHAT LIMITER

Media players may allow users to livestream media in real-time. Users of a livestream may include a streamer (e.g., a user of a user device that is capturing and/or transmitting the live media) and one or more viewers (e.g., users of user devices that are receiving and playing the live media). Viewers of the livestreaming media may post chat messages on a chat application included on a content sharing platform (that provides content for playback by a media player) during the livestreaming of the media. However, in some instances, a large amount (e.g., 500-1,000 messages per second) of chat messages may be received while the media is livestreaming. It may be difficult to display the chat messages in a comprehensible manner for viewers due to the sheer quantity and rapidity in which the chat messages are received. That is, conventionally, when the viewer is viewing the chat application, the messages, which may be arranged by time as they are received, are scrolling too fast to read. In addition, after the media has been livestreaming for a period of time, the chat messages may include low quality posts or spam. It may be undesirable to display such “junk” chat messages.

We present a mechanism that solves these issues by analyzing information related to each chat message and the poster of each chat message in real-time, and selecting the chat messages that are highly on-topic, relevant, and engaging to be shown by a chat application associated with a media player. The chat messages may be selected and displayed every few seconds (e.g., 1 or 2) to enable a viewer to follow along on the chat application. In some implementations, the chat application may be located on a screen proximate to the media player that is livestreaming the media.

In one implementation, the mechanism may use a model to output an aggregate score for each of the chat messages and poster of the chat messages. The model may be trained by

applying a machine learning technique (e.g., a neural network, a support vector machine [SVM], etc.) to training data. The training data may include pairs of training input such as information about the media (e.g., comments left on previous videos, biographic information, etc.) that is livestreaming, and target output such as one or more respective rankings of the information about the media. Subsequently, each new chat message and an identity of the poster of each new chat message may be input into the model. The model may assign a quality score, a topic score, and a poster score using pertinent signals.

For example, pertinent signals for the quality score may include types of words are in the text of the message (e.g., abusive type words, curse words, etc.), whether the words are arranged in a comprehensible manner or whether the words are gibberish, and the like. If the chat message includes abusive words or the words are gibberish, then those negative signals may result in the model assigning a low quality score to that chat message. However, if the words are arranged in a comprehensible manner and are not abusive or gibberish, then those positive signals may result in the model assigning a high quality score to that chat message.

Pertinent signals for the topic score may include frequent appearance of the words from a message on media that's livestreaming (e.g., if the media that is streaming is a video of a person drawing pictures, then check whether certain words like "drawing," "sketch," "paper," etc. appear in the chat messages). If the words of a chat message are relevant to the content of the livestreaming media, then that positive signal may result in the model assigning a high topic score to that chat message. However, if the words of a chat message are unrelated to the content of the livestreaming media, then that negative signal may result in the model assigning a low topic score to that chat message.

Pertinent signals for the poster score may include the popularity of the poster (e.g., how

many subscribers the person has on the media player), historical feedback of the poster (e.g., how many comments the poster has left on videos, how many chat messages has the poster typed in the past that have been flagged as inappropriate, how many comments has the poster typed in the past that have been liked), affinity of the poster to the streamer (e.g., how many videos the poster has watched are from the user that is streaming), account information of the poster (e.g., is the account brand new or has the account been existing for a threshold period of time), and the like. In one example, if a poster has an account with a content sharing platform for longer than a threshold period of time (e.g., longer than a year) and the poster has over a threshold number of subscribers (e.g., more than 1,000), then the likelihood is strong that the poster is popular among the users of the content sharing platform, and the model may assign a higher poster score to the poster's message. On the other hand, if the poster has a brand new account and has no subscribers, then the model may assign a low poster score to the chat messages received from such a poster.

The model may use the quality score, topic score, and poster score in a function to assign an aggregate score for the chat message. The mechanism may further include de-duplicating the messages after the aggregate scores have been assigned. De-duplicating the messages may enhance diversity of chat messages and/or posters. For example, if the chat messages from the same poster are continuously receiving the highest aggregate scores and the poster's chat messages have been recently displayed, then the mechanism may look to the chat message with the next highest aggregate score and determine if it is from a different poster. If it is determined that a different poster entered the chat message with the second highest score (as compared to the poster associated with the highest aggregate scoring chat message), then the mechanism may select the chat message with the second highest aggregate score to be displayed.

Also, another check may be used by the mechanism to ensure that the chat messages that are selected have aggregate scores that are at least above a threshold value. This prevents selecting chat messages with very low quality or that are off topic from being displayed. If there are no messages that are above the threshold aggregate score, then the mechanism may not display any messages for that time period.

Figure 1 depicts a flow diagram of a method for selecting a chat message to display in a chat application in real-time while media is livestreaming, in accordance with some implementations. First, at step 102, a model (e.g., machine learning model) is trained using training data related to the livestreaming media. The training data of the livestreaming media may include training inputs such as information about the chat messages (e.g., quality and/or topic) and posters (e.g., reputation, biographic, historical interaction on the media player, etc.) of the chat messages, and respective target outputs (e.g., scores of individual chat messages). The trained model may find patterns between input attributes (e.g., information about chat messages and information about the poster) to a target (e.g., scores). The trained model may receive new chat messages and output one or more scores for each received chat message, as described in more detail below.

Next, at step 104, chat messages may be received during a livestream of a media content item. As described above, a large number of chat messages may be received (e.g., upwards of 1,000) per second while a popular media content item is live streaming. The mechanism may input each chat message (e.g., actual text of the chat message) and information about the poster of each chat message into the trained model.

Next, at step 106, a quality score, a topic score, and a poster score may be determined for each message using the trained model. The scores may be any suitable indicator, value, number,

or the like that vary in a range from low (e.g., 0) to high (e.g., 1). As described above, signals related to each score may be used to determine the scores. For example, the quality score may be determined using signals including what the actual words are in the text of the chat message (e.g., are the words abusive, are the words in a comprehensible manner and not gibberish, etc.). The topic score may be assigned using signals including whether the words in the text of the chat message are related to the media content being livestreamed. Also, the poster score may be assigned using signals including the popularity of the poster on the content sharing platform (e.g., reputation), how many subscribers the poster has on the content sharing platform, how many chat messages or comments the poster has posted in the past that have been liked, how many chat messages or comments the poster has posted in the past that have been flagged as inappropriate, how long the poster's account has been active, how many of the streamer's videos has the poster watched (e.g., affinity to the streamer), and the like.

Next, at step 108, an aggregate score for each chat message may be determined based on the quality score, the topic score, and the poster score provided by the trained model. In other implementations, the model includes a function with variables for the quality score, the topic score, and the poster score that outputs the aggregate score. The aggregate score may be any suitable indicator, value, number, or the like that vary in a range from low (e.g., 0) to high (e.g., 1).

Next, at step 110, chat messages may be de-duplicated in view of the aggregate scores. De-duplicating the chat messages may enable diversification of chat messages being displayed and/or posters. For example, the mechanism may determine that the chat message with the highest aggregate score is from a poster whose chat messages have been selected and displayed more than a threshold number of times recently. Upon such determination, the mechanism may

inspect the chat message that was assigned the next highest score and determine who the poster of that chat message is. If the poster of the chat message with the second highest aggregate score is different than the poster of the chat message with the highest aggregate score, the chat message with the second highest aggregate score may be selected.

Next, at step 112, the chat message with the highest aggregate score may be selected. That is, the aggregate scores for each of the chat messages within the same time period (e.g., seconds) may be compared to each other and the chat message associated with the highest aggregate score may be selected. However, as discussed above, in some instances, the chat message with the highest aggregate score may not be selected when de-duplication is used to diversify the chat messages and/or posters.

Next, at step 114, a determination is made whether the aggregate score that is selected is above a certain threshold (e.g., 0.1, 0.2, 0.3, 0.4, 0.5, etc.). If the answer is yes, then, at step 116, the selected chat message is displayed on the chat application. If the answer is no, then, at step 118, the selected chat message is not displayed. This check may enable preventing chat messages from being displayed that are of low quality, off topic, or from spammers or ill-reputed posters.

The mechanism described herein allows selecting chat messages in real-time that are on-topic, relevant, and engaging to be shown in a chat application associated with a content sharing platform that enables the livestreaming of media. This mechanism solves the problem of filtering an overwhelming amount of messages in a way that is not duplicative. Because the chat messages that are selected are based on quality signals and topic signals, the selected chat messages may include interesting subject matter that is related to the media content being livestreamed. Also, since poster signals are used to select the chat messages, the selected chat

messages may be from posters that are highly respected in the content sharing platform community related to media that is livestreamed. Further, limiting the chat messages to be selected and displayed every second or every few seconds enables the viewer of the chat application to actually read the chat messages by slowing down the flow of chat messages across the chat application. As such, users are more likely to use and watch such a chat application while media are livestreamed, as it may enhance their livestreaming experience, and for many other reasons.

ABSTRACT

A mechanism for intelligently filtering a large number of chat messages received when media is livestreamed on a content sharing platform to enable displaying of a manageable number of chat messages every second or every few seconds. The mechanism analyzes information related to each chat message and the poster of each chat message in real-time and selects the chat messages that are the most on-topic, relevant, and engaging to be shown via a chat application associated with content sharing platform. The mechanism may use a model to output an aggregate score for each chat message in view of a quality score, topic score, and poster score. The chat message with the highest aggregate score may be selected to be displayed. The chat messages may be de-duplicated to enhance diversification of chat messages and/or posters. Further, when chat messages are not assigned an aggregate score higher than a threshold value, the mechanism may not display any chat messages.

Keywords: video, media, chat message, livestream, machine learning model, poster, chat application, media player, filter

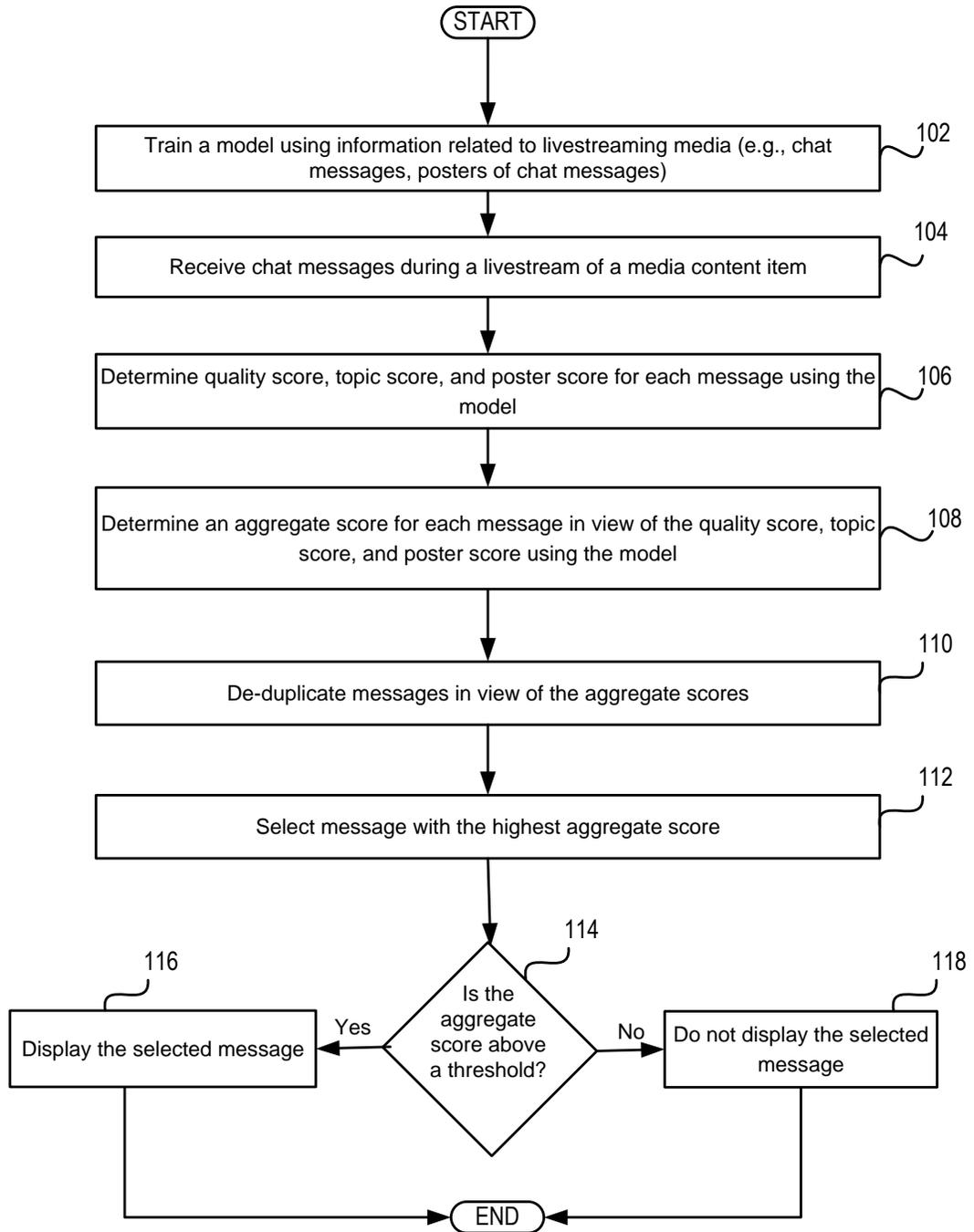


FIG. 1