Technical Disclosure Commons

Defensive Publications Series

November 04, 2016

Threshold-free Selection of Taxonomic Multilabels

Hsin-yi Lin

Brian Milch

Michel Adar

Scot Fang

Rene van de Veerdonk

THRESHOLD-FREE SELECTION OF TAXONOMIC MULTILABELS

In online search or content selection systems, significant computational resources are expended to classify or categorize electronic documents into topics, concepts, or entities. For example, in a hierarchical taxonomy of categories, such as a tree structure, a top-level label can be “health”, and sub-labels can include “sports”, “age”, “cancer”, or “heart”. A classifier can process, parse or otherwise analyze the document to assign one or more labels to the document based on the taxonomy. The classifier can generate a score for each of the labels, and provide the labels and the scores to other components or modules for further downstream processing. To keep downstream processes efficient without causing excessive processing of labels, the classifier may filter out the labels to return a subset of labels based on comparing a label’s score with a threshold. For example, if the score for the label exceeds the threshold, the classifier can return or provide the label for downstream processing. However, using a threshold-based technique to filter out labels may not account for the tree structure of the taxonomy, and it may also fail to take into account the likelihood dependencies between all parent nodes and child nodes.

The proposed technique solves this by (1) selecting a set of labels returned by the classifier that optimizes certain metrics, such as precision and recall metrics; and (2) using a greedy multi-label selection algorithm that optimizes the precision/recall in step (1). Using these techniques, the system can select a subset of labels to return or provide for further processing.

The greedy multi-label algorithm can select labels iteratively by optimizing the estimated expected value of the evaluation metric, e.g., eval(L, G). The evaluation metric can be any metric that evaluates predicted labels L and golden labels G. During prediction, the golden labels are unknown. Golden labels can refer to a set of ground-truth labels that are applied by human raters to an instance. The estimated expected value of the evaluation metric for the particular pair of predicted labels L and a set of labels C can be calculated by P(C)*eval(L, C), where the P(C) is the predicted probability of a document being a member of class C from a classifier. P(C) can be output from a predictive model. The system can select a set of labels C that optimizes P(C)*eval(L, C). Thus, a threshold is not required to select the set of labels C.

4820-4943-7493.1

This greedy multi-label technique can be more efficient, and thus reduce processor run-time, as compared to performing an exhaustive search on expected evaluation metrics of all possible pairs of predicted labels and possible golden labels. The greedy algorithm can iteratively add a single label each iteration to the selected set from the previous iteration. The greedy algorithm can further the one label that leads to the highest expected evaluation metric to the selected set of labels. The system can stop the selection when the expected evaluation metric starts to decrease or the maximum number of selected labels is reached. By not using a threshold, it is possible to select the number of labels that optimize an expected evaluation metric, as opposed to being fixed to a predetermined threshold.

The evaluation can be customized based on the business requirement. For instance, classifying documents into a taxonomic tree can use a soft evaluation metric instead of the commonly used exact match. A predicted label "/Arts & Entertainment/TV & Video/TV Shows & Programs" against golden label "/Arts & Entertainment/TV & Video/TV Shows & Programs/TV Sci-Fi & Fantasy Shows" receives partial credits of a match. Also, since a document is unlikely to belong to sibling labels under the same subtree, once a sibling label is selected, the other label should not increase the evaluation metric. This logics can be encoded in the evaluation metrics to facilitate the decision. Now the label selection we proposed also uses the information to select estimated optimal set of labels.

Figure 1 illustrates a system 1 that includes a data processing system 5 to perform the technique. The system 1 can include or interact with one or more computing devices 2, one or more content providers 3, and one or more content publishers 4 via a network 10. The data processing system 5 can include a taxonomic classifier 6, taxonomic post-processor 7 and a data repository 11. The taxonomic post-processor can include a goodness selection estimator 8 and an iterative selector 9. The data repository 11 can store a taxonomy 12, metrics 13, and algorithms 14. The data processing system 5, taxonomic classifier 6, taxonomic post-processor 7, goodness of selection estimator 8, iterative selector 9, computing device 2, content provider 3, and content publisher 4 can include one or more servers, processors, computing devices, memory, logic arrays, circuitry, software or hardware modules, logic elements, or digital logic blocks configured to facilitate socially sharing content items.

Computing device 2 can include, for example, mobile computing devices, mobile telecommunications devices, smartphones, personal digital assistants, laptop computers, notebooks, tablet computers, smart watches, or wearable devices. The computing device 2 can include a display such as a liquid crystal display, light emitting diode (LED) based display, organic light emitting diode based display, bitmap display, pixel display, electronic ink display, or other display configured to visually output content including text, characters, strings, symbols, images, or multimedia content provided by a data processing system 5. The computing device 2 can include an input interface designed and constructed to receive input from a user. Input interfaces can include or provide, for example, touch input, keyboard, mouse, motion, sensor, location sensor, touchpad, trackpad, or a scroll wheel. The network 10 can include one or more of any type of computer network such as the Internet, cellular network, WIFI network, WiMAX network, mesh network, Bluetooth, near field communication, satellite network, or other data network that facilitates communications between the data processing system 5 and computing device 2.

A content provider 3 can refer to, or include, an advertiser or other provider of content items, such as online documents, blogs, media or advertisements. The content provider 3 can establish an advertisement campaign with advertisements and advertisement selection criteria, such as keywords and geographic location. The content provider 3 can further establish a budget or spend amount for the advertisement campaign. The content publisher 4 can refer to or include a web site operator, such as an entity that operates a web page. The web site operator or content publisher 4 can include at least one web page server that communicates with the network 10 to make the web page available to the computing device 2.

The taxonomic classifier 6 can receive an online document (e.g., web page, article, blog, news article, etc.) including various terms, keywords, phrases, text, metadata, or other content. The taxonomic classifier 6 can access a taxonomy 12 from data repository 12. The taxonomic classifier 6 can retrieve, from the data repository 12, a taxonomy including a tree structure of labels. The taxonomic classifier 6 can be configured with a multi-class classification framework to analyze the document and output a probability for each class. The taxonomic classifier 6 can output a probability for each category in the taxonomy.

The taxonomic post-processor 7 can receive or obtain, from the taxonomic classifier 6, the output probability for each category in the taxonomy. The taxonomic post-processor 7 can select one or more metrics 13 from data repository 11 to use to further process the output probability for each category in the taxonomy received from the taxonomic classifier 6. The taxonomic post-process 7 can provide the selected metrics to a goodness of selection estimator 8. The goodness of selection estimator 8 can include a script, function, or module that analyzes the scores for each category.

The goodness of selection estimator 8 can estimate a precision metric and a recall metric for each category. Precision (or positive predictive value) can refer to the fraction of retrieved instances that are relevant. In this case, precision can refer to the number of labels identified by the taxonomic classifier 6 that are relevant to the document. Recall (or sensitivity) can refer to the fraction of relevant instances that are retrieved. In this case, recall can refer to the fraction of labels that are scored correctly. The goodness of selection estimator 8 can combine the precision and recall metrics to generate a single metric for an expected value of goodness. The goodness of selection estimator 8 can combine this metric with the score received from the taxonomic classifier 6.

The data processing system 5 can determine this combined expected value of goodness for each score for each label returned from the taxonomic classifier 6. In some cases, this score can be deterministic (0 or 1). The iterative selector 9 can use this score to iteratively add categories to the set of predicted labels, given an input of probabilities of all the labels received from the taxonomic classifier 6 and using the goodness of selection estimator 8 to compute a score for the combination of labels. The iterative selector 9 can use various other techniques or algorithms to identify the subset of labels. In some cases, the iterative selector 9 can be configured with a greedy algorithm or technique to identify a final set of labels.

For example, the iterative selector 9 can identify all the probabilities for all the categories from the taxonomic classifier 8. The iterative selector 9 can launch the goodness of selection estimator 8 to determine an expected value of goodness score. This expected value of goodness score can include or be based on a precision score, recall score, a score based on a combination of the precision and recall score, a precision recall weighted sum, or some other score. The

iterative selector 9 can identify one or more labels that optimize this expected value of goodness score.

In the first iteration, the iterative selector 9 can select a top ranking label based on the score from the taxonomic classifier 6. The iterative selector 9 can launch the goodness of selection estimator 8 to determine an expected value of goodness for this first label. The iterative selector 9 can determine that the expected value of goodness for the first label is satisfactory if the score is greater than zero, a dynamic threshold, or a predetermined threshold. The threshold can close to zero in order to filter out labels that have a likelihood close to zero, as opposed to a greater threshold traditionally used for identifying a final sets of the labels.

The iterative selector 9 can then select a second label that has a second highest probability score based on the taxonomic classifier 6. The iterative selector 9 can execute the goodness of selection estimator 8 to determine an expected value of goodness for the combination of the first label and the second label. The iterative selector 9 can determine that the combination of the first label and the second label improves the expected value of goodness. Responsive to determining that the combination of the first label and the second label improves the expected value of goodness as compared to just the first label, the iterative selector 9 can add or select the second label for inclusion in the subset.

The iterative selector 9 can continue to select labels, compute the expected value of goodness for the combination, and then compare the expected value of goodness based on the combination of labels with the previous expected value of goodness to determine whether to add the label to the subset. Thus, the iterative selector 9 can identify a combination of labels with a high score.

The iterative selector 9 can stop at a maximum number of iterations, such as 5, 6, 7, 8, 9 or 10. The iterative selector 9 can start with a minimum number of iterations, such as 2, 3, 4, or 5. The iterative selector 9 can have a dynamic maximum number of iterations that can be based on the number of scores greater than a threshold for labels received from the taxonomic classifier 6.

Thus, the present system can filter out the labels received from a classifier to generate a subset of labels and provide the subset of labels for downstream processing, thereby improving

the efficiency of any downstream processing.

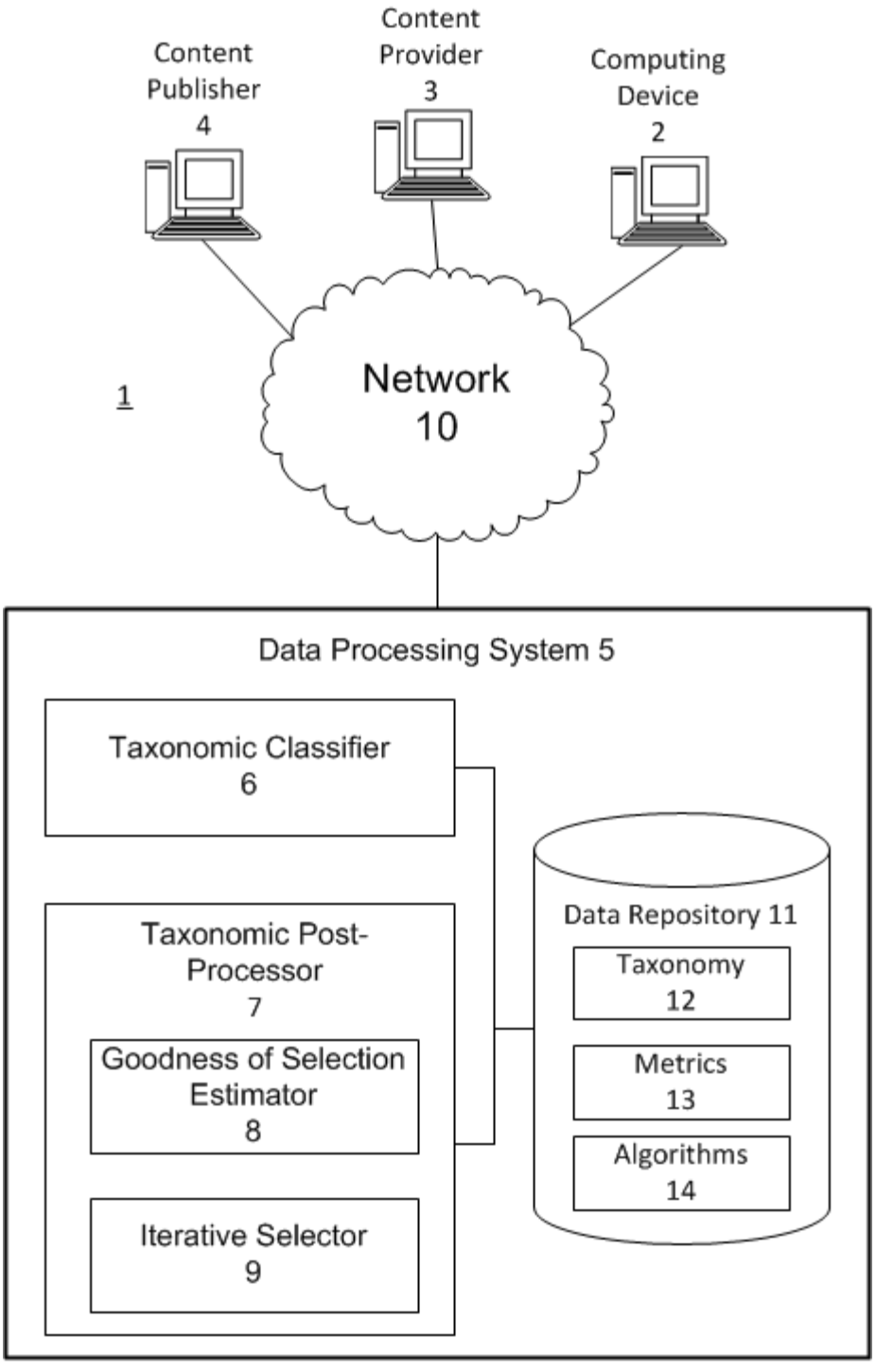

FIG. 1