

Associated Press: Text-To-Image Recommendation System Update Report

I. Progress Summary

Our progress to date encompasses two areas of the project:

- (1) Modeling: The following models have been completed and integrated into the user interface. The details are outlined in Section II.
 - (a) Baseline KNN Model
 - (b) Tag-to-Tag based Image Recommendation
- (2) User Interface: Details on the current interactions available are detailed in Section III. We will provide a screen recording of the current recommendation workflow on the UI by Nov 18.

II. Model Updates

1) Baseline K-Nearest Neighbors Model

For our baseline image recommendation model, we implemented a variation of the K-Nearest Neighbors (KNN) algorithm. This model recommends images using the following steps.

Define the training set as the set of cleaned articles and tags. Given a test article and extracted tags (to be implemented using the AP Tagging API), the model calculates the “distance” between the test tags and each article in the training set. It then returns the images associated with the K most similar articles, where K defaults to 3.

The distance between two sets of tags is currently defined as a number between 0 and 1 representing the normalized exact overlap: $([\text{test_tags}] \cap [\text{train_tags}]) / \text{size}([\text{test_tags}])$. We have also implemented a distance measure that takes into account any tags that are synonyms to a lesser extent: $([\text{test_tags}] \cap [\text{train_tags}] + \text{eta} * \text{synonyms}([\text{test_tags}, \text{train_tags}])) / \text{size}([\text{test_tags}])$.

2) Tag-to-Tag Image Recommendation

In our current model iteration, demonstrated in the UI video, we leverage the AP Tagging Service output for image recommendation. This model is intuitive because with the current editorial workflow, images and articles submitted to the AP database undergo the same tagging taxonomy. The recommendation process essentially compares the set of input article tags with those for images. The metrics supporting the comparison / recommendation process are discussed below:

- TextRank on Article Tags + Tag Importance Ranking on Image Tags

In preparation for the recommendation, we performed TextRank on article tags using the full text, and a Tag Importance Ranking (TIR) framework adapted by Li et al. (2017) on image tags. On the high level, TextRank accounts for the occurrence and co-occurrence of the proposed tags in the article full text. TIR weights the proposed object tags given the image captions, and the proposed scene tags based on their respective grammatical positions (subject noun, or prepositional phrases) of the tag in the image captions. Our current design choice only evaluate Place, Organization, Person, and Subject tags; we defined object tags as tags that are likely to be tangible in an image (Person), and scene tags as tags capturing the background information of an image (Place, Organization, Subject).

During one recommendation process, TextRank importance scores are computed for the proposed article tags. We have implemented dot product, and cosine similarity distance metrics for comparing the article importance scores vector with those of images. We rank the distances in descending order and outputs a user-defined number of recommended images. Note that our current recommendation process eliminates all images that have already been associated with test articles.

- Word Mover's Distance and Soft Cosine Distance for Word Embeddings

On top of using exact tag, we also wanted to incorporate word embedding weights (e.g., word2vec or GloVe), so that semantically similar words can be weighted higher in the recommendation, for example, "campaign" and "election" share similar semantic context, but will be considered as different words if we are using exact word token to recommend images. Leveraging word embedding weights will allow us to consider "campaign" and "election" similar.

We used two approaches to incorporate word embedding weights, word mover's distance (WMD) and soft cosine distance (SCD). Looking at the recommendations of a few articles, word mover's distance performs good recommendations, however, it is not computationally efficient (~15 minutes to recommend images per article). Therefore, we decided to use soft cosine distance (SCD) which was shown to be much faster and performs similarly as WMD in other downstream tasks, accordingly to the package developer.

In the following weeks, we plan to:

1. Shorten the image candidate list to calculate WMD distance
 - a. Currently WMD is slow because we need to calculate similarity of the entire image corpus (>70k images). Shortening the candidate list before the computation can help to shorten the computation time.
 - b. We plan to shorten the list by:
 - i. Only uses images that has ≥ 1 tag overlap with the article OR
 - ii. Only uses images that has all geographic tags OR
 - iii. Only uses images that has all geographic direct tags
2. Weigh the distance based on tag importance
 - a. Currently the distance between image and article tag vectors are calculated equally among tags. However, we can leverage the tag importance calculation to weight the important tags higher
 - b. For example, if "Obama" is considered as an important tag in this article, then we want to recommend an image that "Obama" is also important in the image, instead of using the less important tags of the article, for example, "United States", to recommend images
 - c. We can use TextRank and image tag retrieval to calculate the importance score of the tags

3) Text-to-Text Image Recommendation

On top of using tags, we also leveraged the longer text of articles and images. We used headline from articles and image summary, because both are textual features with no missing values in our provided datasets.

We used similar distance metrics as mentioned above, word mover's distance (WMD) and soft cosine distance (SCD). Looking at the recommendations of a few articles, word mover's distance performs good recommendations, however, it is not computationally efficient (~15 minutes to recommend images per article). Therefore, we decided to use soft cosine distance (SCD) which was shown to be much faster and performs similarly as WMD in other downstream tasks, accordingly to the package developer.

IV. User Interface Updates and Demo

Here is an example of model 2 in action. The user inputs an article about Kim Jong Un and Donald Trump, titled 'Kim-Trump border meeting: History or just a photo-op?' in this scenario. The first four recommended images (based on their captions and their tags) are shown below. The tags within the article are ranked and shown to the user as well.

Article Input

Kim-Trump border meeting: History or just a photo-op?


TOKYO (AP) — It sure looked historic: President Donald Trump and North Korean leader Kim Jong Un strode toward each other Sunday from opposite sides of a strip

Search

Tags: Importance Score


Donald Trump: person: 0.01402
North Korea: place: 0.00995
World War II: subject: 0.00501
Pyongyang: place: 0.00458
United States: place: 0.00316
Japan: place: 0.0018
Panmunjom: place: 0.00125
Asia: place: 0.00114

Recommended Images




North Korea US

Accept Reject




North Korea

Accept Reject



Trump North Korea

Accept Reject



Trump North Korea

Accept Reject

Since this is an existing AP article, we can check the true images associated with that article. These are shown below:



For our next steps on the UI, we would like to incorporate a greater amount of user feedback. That is, we want users to select whether they accept or reject a particular image recommended to them. Based on their selection, we will present a second set of images. An example of how the interaction works is as follows:



Trump North Korea

Accept

Reject

Additionally, we display the tags and their importance scores (according to textrank). But we want users to have more control over the recommendations given to them, so we want to allow users to deselect tags that they feel are not important in order to get finer-grained image recommendations.

Project Timeline

Nov 12 - Nov 19:

- Provide working UI demo (will send over screen recording by Nov 18)
- Explore Tagging API for qualitative evaluation of model on external usage

Nov 19 - Nov 26:

- Improving recommendation quality
- Implement quantitative evaluation metrics
- Training complete
- Start blog post draft

Nov 26 - Dec 3 (Thanksgiving):

- Poster draft

- Blog intermediate draft
- Final presentation draft
- Model optimization
- Finalize UI/API endpoint
- Get code base ready for shipping

Dec 3 - Dec 12:

- Prepare IACS Showcase (Dec 9)
- Prepare final presentations (Dec 11)
- Finalize blog post (Dec 12)
- Finalize deliverables to AP (Dec 12)

Deliverable

- Recommendation model
 - Trained on data from AP text and AP images
 - It accepts input of article text and recommends images to the text
- API endpoint
 - This is a wrapper for our models and can be called by our UI and potentially from AP's other client-facing UIs
- UI
 - It takes article text as input and outputs recommended images
 - It supplements the recommendation model by providing a nice interface to show images recommended as well as tags/weights
- Poster
 - We are going to present a poster at the IACS Showcase early Dec
- Blog Post
 - A medium blog post to illustrate our workflow, modeling and results