# Supplementary material for "Unsupervised Discovery of Multimodal Links in Multi-image, Multi-sentence Documents"

**Jack Hessel  Lillian Lee  David Mimno**
Cornell University
{jhessel, llee}@cs.cornell.edu  mimno@cornell.edu

## 1 Data preprocessing details

**MSCOCO**. We downloaded the train/val 2017 images, and the train/val annotations from 2014 and 2017 from the MSCOCO website (but create our own training and validation splits). Then, we randomly designate half of the images as "true" images (which will eventually be paired with their true captions in documents) and half of the images as "fake" images, which will not be paired with their true captions in documents. Then, we randomly group all true images into groups of five, and all fake images into groups of five. Then, we pair each real-image set with a fake image set, and divide the resulting groups of 10 images into train/validation/test splits. Then, for each of the training/validation/testing document sets independently, for each document, we create (usually) 5 true versions of each document (for testing and validation, we only sample a single version of each document, and do not consider the alternate true captions provided by MSCOCO) because (in general) each MSCOCO image comes with 5 caption annotations. For each of these true versions, we randomly sample captions from a pool of all captions written on all images not in that document (but from the train/validation/test pools independently, so that there is no overlap between these sets, except in cases where captions happen to be identical). Then, we shuffle the sampled captions for each version. The result is 4968/1655/1655 train/validation/test documents, but each training "document" generally consists of 5 versions because MSCOCO images generally come with 5 captions each.

**Story-DII/Story-SIS**. We downloaded the Story-DII/Story-SIS train/validation/test splits along with all images from the Visual Storytelling Dataset website;[1] we preserve these splits for our train/validation/test sets. DII stories have multiple annotations per fixed image set, whereas SIS stories have multiple annotations per Flickr album, as human annotators were allowed to select images for their story from all the images within an album. We discard any story with any invalid or missing image (the FAQ page on the data download website mentions that images may be missing because users deleted them).

**DII-Stress**. We augmented the documents from Story-DII with 45 distractor captions (i.e., captions that were not written about any of the images in the document) selected uniformly at random. To preserve train/validation/test splits, we limit these uniform selections to within-split samples, i.e., training document distractor captions are sampled only from training documents.

**RQA**. We download the train and validation questions (29.6K/3.5K) and extract the "context" of each question, which consists of a list of recipe steps and their associated images; without filtering, there are 8.1K unique recipes in the training set, and 983 unique recipes in the validation data. We also download the training/validation images provided. We treat the provided validation split as the test data.

We concatenate the title and the body of the step (separating them with a space). We discard recipe steps that do not contain any tokens, and discard recipes for which there are no images that correspond to steps (e.g., if the only steps for which there were images contained empty text). Then, we reserve training recipes to act as our validation split. Then, we discard all recipes with fewer than 2 images/recipe steps. The result is 6502/946/878 training/validation/test recipes, with 69K total images. The sizes of the documents are: mean/median/max number of images: 11/8/93; and mean/median/max number of sentences: 7/6/20.

[1] http://visionandlanguage.net/VIST/

**DIY**. We downloaded all the submissions on pushshift.io's files page from Jan. 2013-Oct. 2018. We looped over all of them and found the ones available made to the subreddit "DIY," for 241K posts. Then, we discard posts with score less than 25. While the semantics of the Reddit "score" field have changed over time,[2] we intend for this filtration step to act as a basic spam filter. We only consider link submissions to imgur urls with "/a/" in the url, indicating that the imgur link is an album, rather than a single image. We then scrape the associated imgur album page and search for all "div" html fields that are "post-image-container," and extract both the image associated with that field and its associated caption, if it's not empty; users may leave image captions empty, but may not upload a caption without an associated image. We ignore imgur albums with no "post-image-container" fields. There are 13K documents after this step. We attempt to scrape all images for these documents, discarding gifs and invalid images for simplicity, resuling in 295K images.

Next, we search for any image duplicates using `findimagedupes` (https://gitlab.com/opennota/findimagedupes) with a neighbor threshold of 3. We discard any documents with any duplicate images. Then, we discard all documents without at least 2 image captions with at least 5 tokens, and discard documents without at least 2 valid images. Because a small number of documents are quite long, we discard documents with more than 40 images or more than 40 captions.[3] We split the remaining documents into 6.8K/1K/1K train/validation/test documents. Between these documents, there are 154K unique images. The sizes of the documents are: mean/median/max number of images: 17.4/16.0/40; mean/median/max number of sentences: 16.4/15.0/40.

**WIKI**. We downloaded the English-language subset of the ImageClef 2011 Wikipedia retrieval data as a starting point (https://www.imageclef.org/wikidata). This dataset contains the full text of Wikipedia articles, alongside a list of images in each article. We then stripped out wiki formatting, and used Spacy's (https://spacy.io/) English-sentence tokenizer to split documents into sentences (the resulting sentence tokenization is imperfect, but sufficient). We keep only the first 100 identified sentences in a document. We discarded documents with fewer than 10 sentences, and documents with fewer than 3 images. The result is 16K articles, for which we used a 14K/1K/1K train/validation/test split. For the results discussed in the paper, we explore same-document predictions on training documents using a model checkpoint with low validation error. The sizes of the documents are: mean/median/max number of images: 6/5/108, mean/median/max number of sentences: 72/86/100.

**Download**. All datasets are available for download: www.cs.cornell.edu/~jhessel/multiretrieval/multiretrieval.html

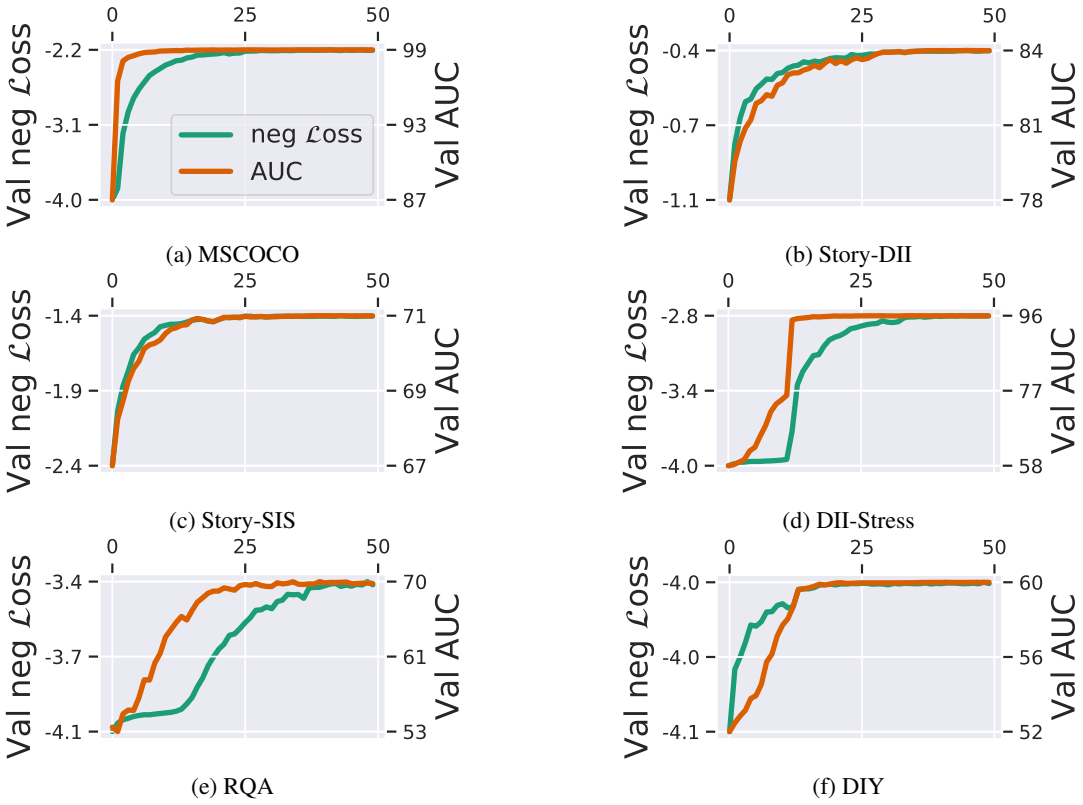
## 2 WIKI Fine-tuning Details

We experiment with fine-tuning the parameters of our image model for the organically-multimodal data, as an alternative to extracting features from a pretrained network. However, given that hundreds of images and sentences need to fit in GPU memory for each batch (we worked with a single GPU with 12GB of RAM), we needed to switch our CNN from DenseNet169 to one with a smaller memory footprint; we chose NASNetSmall. But even so, we still require a word-embedding matrix and a 1024-dimensional GRU in memory. Hence, additionally, at training time, for documents with more than 10 images/sentences, we randomly downsample images/sentences to a set of 10 (though at validation and test time, longer documents are kept intact). This subsampling process ensures that at most 110 images are in GPU memory at a time (for 10 negative samples per positive sample). When training the CNN, we also perform random data augmentation to help regularize. We first resize images to 256 by 256, and, at training time, perform the following data augmentation: random horizontal flipping, up to 20 degree random image rotation, and a random crop to 224 by 224. At validation/test time, we use a center crop (with no rotations or flips).

We trained models with AP using fixed, NASNetSmall pre-extracted features, and compared those models to ones where we fine-tuned the additional 5M CNN parameters. The resulting *test* AUC/negative-loss ($-\mathcal{L}$) values are:

[2] Other confounding factors: Reddit has become more popular over time, DIY has likely changed in popularity, etc.

[3] At this step, its possible for there to be more captions than images in a document, e.g., because we discard animated gifs that may have been associated with captions.




(a) MSCOCO (b) Story-DII

(c) Story-SIS (d) DII-Stress

(e) RQA (f) DIY

Figure S1: Inter-document objective (AP, $b = 10$, hard negative mining) and intra-document AUC during 50 epochs of training for all datasets we consider with ground-truth, intra-document annotations. While there are some interesting discontinuities, e.g., in DII-Stress's training curves, in general, for a fixed neural architecture/similarity function, better retrieval performance, as measured by the negative-loss computed over the validation set, equates to better intra-document performance, as measured by AUC.

| | RQA | | DIY | | WIKI | |
|---|---|---|---|---|---|---|
| | AUC | $-\mathcal{L}$ | AUC | $-\mathcal{L}$ | AUC | $-\mathcal{L}$ |
| Fixed CNN | **67.6** | **-.37** | **60.9** | **-.37** | N/A | -.26 |
| Finetuned CNN | 65.7 | -.40 | 57.9 | -.39 | N/A | **-.21** |

Thus, we did not observe intra-document performance increases with fine-tuning for DIY and RQA for the experiment settings we consider. However, on WIKI, for negative-training-loss (the only metric we can compute on this no-ground-truth dataset), fine-tuning performed better. [4] Since Figure S1 demonstrates that, for a fixed architecture and for datasets where AUC can be computed, AUC and (the negative of) training loss rise together, we expect that fine-tuning is beneficial for WIKI.

## 3 Additional Results

Tables containing our full results begin on the next page. Compared to the results presented in the paper, here we explicitly compare additional hyperparameter configurations. Specifically: we show results for $b = 10, 20, 30$ negative samples (the main paper just shows $b = 10$) and compare using hard negative mining vs. not using hard negatives (the main paper just shows hard negative mining results, e.g., "AP+hard neg" in these tables is the same as the "AP" described in the main paper). In general, hard negative mining improves performance, and the number of negative samples doesn't greatly affect performance in the range we examined.

[4] Fine-tuning NASNetSmall also beat using DenseNet169 extracted features.

| | MSCOCO | | Story-DII | | Story-SIS | | DII-Stress | |
|---|---|---|---|---|---|---|---|---|
| | AUC | p@1/p@5 | AUC | p@1/p@5 | AUC | p@1/p@5 | AUC | p@1/p@5 |
| Random | 49.7 | 5.0/4.6 | 49.4 | 19.5/19.2 | 50.0 | 19.4/19.7 | 50.0 | 2.0/2.0 |
| Obj Detect | 89.5 | 67.7/45.9 | 65.3 | 50.2/35.2 | 58.4 | 40.8/28.6 | 76.9 | 25.7/17.5 |
| NoStruct | 88.3 | 53.4/35.8 | 76.6 | 60.4/46.2 | 64.9 | 43.3/33.8 | 84.2 | 21.4/15.6 |
| NoStruct+ hard neg | 51.8 | 8.3/5.9 | 75.9 | 63.0/45.0 | 63.3 | 45.1/31.9 | 51.9 | 4.3/3.1 |
| DC | 98.8 | 92.0/78.6 | 81.8 | 69.1/53.7 | 68.0 | 49.7/37.6 | 93.8 | 58.3/40.1 |
| DC+ hard neg | 98.9 | 93.1/79.9 | 82.9 | 71.9/55.7 | 68.8 | 52.2/38.7 | 95.0 | 65.2/44.9 |
| TK | 98.8 | 92.1/78.6 | 81.8 | 69.6/53.8 | 68.0 | 49.7/37.6 | 94.4 | 60.2/42.2 |
| TK+ hard neg | 98.9 | 93.9/80.0 | 82.8 | 71.5/55.7 | 68.8 | 51.8/38.5 | 95.2 | 65.2/45.3 |
| TK+ hard neg+ $\frac{1}{2}k$ | 99.0 | 95.0/81.4 | 81.9 | 71.4/54.5 | 67.6 | 51.5/37.8 | 94.7 | 64.5/43.4 |
| AP | 98.5 | 87.6/75.3 | 81.7 | 68.3/53.5 | 67.3 | 47.1/36.6 | 93.5 | 58.3/39.7 |
| AP+ hard neg | 98.7 | 91.1/77.9 | 82.6 | 70.7/55.0 | 68.6 | 50.6/38.3 | 95.4 | 65.4/45.5 |
| AP+ hard neg+ $\frac{1}{2}k$ | 98.9 | 94.1/80.7 | 81.5 | 72.2/54.2 | 67.4 | 51.9/37.7 | 94.6 | 64.7/43.7 |

Table S1: Results for crowdlabeled data with ground-truth annotation with $b = 20$ negative samples.

| | MSCOCO | | Story-DII | | Story-SIS | | DII-Stress | |
|---|---|---|---|---|---|---|---|---|
| | AUC | p@1/p@5 | AUC | p@1/p@5 | AUC | p@1/p@5 | AUC | p@1/p@5 |
| Random | 49.7 | 5.0/4.6 | 49.4 | 19.5/19.2 | 50.0 | 19.4/19.7 | 50.0 | 2.0/2.0 |
| Obj Detect | 89.5 | 67.7/45.9 | 65.3 | 50.2/35.2 | 58.4 | 40.8/28.6 | 76.9 | 25.7/17.5 |
| NoStruct | 87.5 | 50.8/34.7 | 76.6 | 59.9/46.2 | 64.9 | 43.4/33.7 | 84.1 | 21.3/15.6 |
| NoStruct+ hard neg | 52.0 | 10.3/6.0 | 75.9 | 63.0/45.0 | 63.0 | 44.5/31.5 | 51.8 | 4.0/2.9 |
| DC | 98.8 | 92.0/78.7 | 82.2 | 70.5/54.6 | 68.0 | 49.7/37.7 | 93.9 | 58.6/40.3 |
| DC+ hard neg | 98.9 | 93.4/79.9 | 82.8 | 71.3/55.5 | 68.8 | 52.1/38.6 | 95.0 | 63.8/44.5 |
| TK | 98.8 | 91.6/78.7 | 81.8 | 69.5/53.9 | 68.0 | 49.9/37.7 | 94.4 | 60.5/42.4 |
| TK+ hard neg | 98.9 | 93.3/80.0 | 82.8 | 71.4/55.7 | 68.8 | 51.0/38.6 | 95.2 | 65.3/45.7 |
| TK+ hard neg+ $\frac{1}{2}k$ | 99.0 | 95.2/81.5 | 82.1 | 73.1/55.1 | 67.7 | 51.9/37.8 | 94.7 | 64.2/43.6 |
| AP | 98.5 | 87.3/75.4 | 81.7 | 67.7/53.4 | 67.3 | 47.1/36.6 | 93.4 | 57.2/39.8 |
| AP+ hard neg | 98.7 | 91.2/78.0 | 82.6 | 71.1/55.0 | 68.5 | 50.3/38.2 | 95.3 | 65.3/45.6 |
| AP+ hard neg+ $\frac{1}{2}k$ | 98.9 | 94.1/80.5 | 81.6 | 72.8/54.4 | 67.4 | 51.8/37.8 | 94.4 | 64.3/43.2 |

Table S2: Results for crowdlabeled data with $b = 30$ negative samples.

| | RQA | | DIY | |
|---|---|---|---|---|
| | AUC | p@1/p@5 | AUC | p@1/p@5 |
| Random | 49.4 | 17.8/16.7 | 49.8 | 6.3/6.8 |
| Obj Detect | 58.7 | 25.1/21.5 | 53.4 | 17.9/11.8 |
| NoStruct | 60.5 | 34.3/26.8 | 56.9 | 13.8/12.2 |
| NoStruct+ hard neg | 60.1 | 35.0/26.7 | 56.3 | 15.0/12.5 |
| DC | 67.1 | 43.8/34.9 | 59.5 | 19.3/15.2 |
| DC+ hard neg | 63.4 | 36.6/31.0 | 59.3 | 21.0/16.0 |
| TK | 65.2 | 41.6/33.1 | 60.0 | 20.4/15.5 |
| TK+ hard neg | 67.9 | 45.2/36.0 | 60.5 | 20.3/16.2 |
| TK+ hard neg+ $\frac{1}{2}k$ | 67.7 | 44.4/35.0 | 56.1 | 14.8/12.0 |
| AP | 66.9 | 37.8/34.2 | 59.1 | 16.9/13.9 |
| AP+ hard neg | 69.4 | 45.9/37.8 | 61.9 | 23.3/17.9 |
| AP+ hard neg+ $\frac{1}{2}k$ | 68.5 | 44.9/36.4 | 59.6 | 21.7/15.7 |

Table S3: Results for organically-multimodal data with ground-truth annotation with $b = 20$ negative samples.

| | RQA | | DIY | |
|---|---|---|---|---|
| | AUC | p@1/p@5 | AUC | p@1/p@5 |
| Random | 49.4 | 17.8/16.7 | 49.8 | 6.3/6.8 |
| Obj Detect | 58.7 | 25.1/21.5 | 53.4 | 17.9/11.8 |
| NoStruct | 60.4 | 34.5/26.7 | 56.9 | 13.3/11.9 |
| NoStruct+ hard neg | 59.7 | 31.8/27.0 | 55.9 | 14.7/12.4 |
| DC | 66.7 | 42.7/34.1 | 59.5 | 18.9/14.7 |
| DC+ hard neg | 63.5 | 37.6/30.6 | 59.4 | 20.8/16.4 |
| TK | 65.3 | 41.2/32.8 | 60.1 | 20.0/15.9 |
| TK+ hard neg | 68.0 | 44.0/36.2 | 60.5 | 21.4/16.1 |
| TK+ hard neg+ $\frac{1}{2}k$ | 67.8 | 43.2/35.1 | 57.3 | 19.1/13.5 |
| AP | 66.5 | 41.0/33.8 | 59.2 | 15.7/14.0 |
| AP+ hard neg | 69.3 | 47.5/37.4 | 61.9 | 24.4/17.8 |
| AP+ hard neg+ $\frac{1}{2}k$ | 68.7 | 45.2/36.2 | 59.4 | 22.0/15.7 |

Table S4: Results for organically-multimodal data with $b = 30$ negative samples.