Towards Adaptable and Interactive Image Captioning with Data Augmentation and Episodic Memory

Aliki Anagnostopoulou[1,2] **Mareike Hartmann**[3] **Daniel Sonntag**[1,2]
[1]German Research Center for Artificial Intelligence (DFKI), Germany
[2]Applied Artificial Intelligence (AAI), Oldenburg University, Germany
[3]Department of Language Science and Technology, Saarland University, Germany
{firstname.lastname}@dfki.de

Abstract

Interactive machine learning (IML) is a beneficial learning paradigm in cases of limited data availability, as human feedback is incrementally integrated into the training process. In this paper, we present an IML pipeline for image captioning which allows us to incrementally adapt a pre-trained image captioning model to a new data distribution based on user input. In order to incorporate user input into the model, we explore the use of a combination of simple data augmentation methods to obtain larger data batches for each newly annotated data instance and implement continual learning methods to prevent catastrophic forgetting from repeated updates. For our experiments, we split a domain-specific image captioning dataset, namely VizWiz, into non-overlapping parts to simulate an incremental input flow for continually adapting the model to new data. We find that, while data augmentation worsens results, even when relatively small amounts of data are available, episodic memory is an effective strategy to retain knowledge from previously seen clusters.

1 Introduction

Image Captioning (IC) is the task of generating a description in natural language for a given image (Stefanini et al., 2021). For the training of most state-of-the-art IC models, large amounts of annotated training data are required (Zhou et al., 2020). However, whenever models need to caption user-specific images without large-scale annotations, this is an impractical requirement. In this case, a potential solution can be found in an *interactive* framework, in which the model can be efficiently adapted to new data based on user feedback (Ling and Fidler, 2017; Shen et al., 2019). Additionally, interactivity renders AI/ML-systems more user-friendly and trustworthy (Bussone et al., 2015; Guo et al., 2022).

In interactive ML settings, training takes place with small amounts of data, and often in an incremental manner. These properties can lead to *overfitting*, on the one hand, which is the lack of generalization ability of the model, and *catastrophic forgetting*, on the other hand, which refers to the drop in performance on older tasks, when a model is trained on new data. For our interactive approach, we tackle these problems using a combination of methods previously proposed in the literature. To tackle overfitting, we apply data augmentation to each instance of user feedback to obtain larger batches of data, which the model is then updated on (Wang et al., 2021). Nevertheless, we find that this strategy fails to improve results in our image captioning task, indicating that the data augmentation methods we used are not suitable for this kind of task. In order to prevent catastrophic forgetting, we rely on continual learning methods. In the following, we present and test an IC pipeline that can be used in an interactive setting. Our work is guided by the following research questions:

1. How does data augmentation benefit a system which is trained incrementally with (simulated) user feedback? How does this system perform in few-shot scenarios?

2. How effective is an episodic memory replay module (de Masson d'Autume et al., 2019) for knowledge retention from previous trainings?

Our contributions are as follows:

- We propose a lightweight continual learning IC pipeline that leverages data augmentation, which can be used in an interactive machine learning setting.

- We adapt a continual learning method, namely sparse memory replay, proposed by de Masson d'Autume et al. (2019), for IC.

- We test a combination of data augmentation

Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP), pages 245–256
July 13, 2023 ©2023 Association for Computational Linguistics

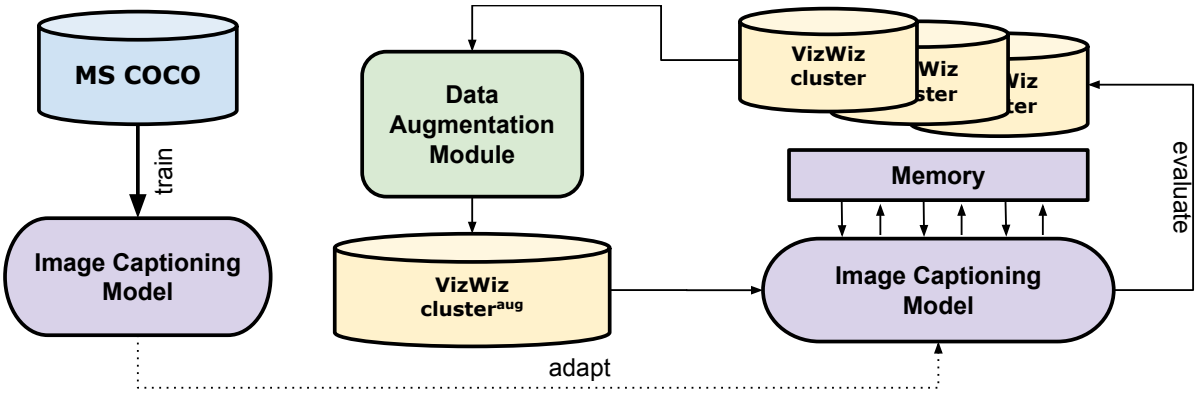


Figure 1: Our pipeline. Following the pre-training/fine-tuning paradigm, we first train our IC model on the MS COCO dataset. We then continue to train our model incrementally, by adding a new cluster each time from the VizWiz dataset, after applying DA methods on it to obtain more training data. During training on the VizWiz data for each cluster, an episodic memory module is activated, which is used to retrieve old data points from previously seen clusters.

methods for interactive IC in both image and text modalities.

- Since we report negative results for the system using data augmentation methods on the user feedback, we additionally investigate why these methods do not work in our case, and we offer some possible explanations for the deteriorating performance.
- We propose a method based on nominal phrase similarity between captions of different images for splitting a dataset into different tasks suitable for evaluating task-incremental continual learning when only image captions are given.

For our simulated user feedback, we use a domain-specific dataset, namely VizWiz (Gurari et al., 2020; Simons et al., 2020), which consists of images taken by visually impaired people. We choose this dataset exactly because of this property: the quality of the images is lower than in most general-use IC datasets, resembling the image quality of user images.

2 Related work

Image captioning (IC) Deep-learning based IC models (Xu et al., 2015; Anderson et al., 2018) traditionally consist of two parts: an *encoder* and a *decoder*. The visual encoder breaks the image down into features or creates an intermediate representation. The decoder is a language model, which takes the encoder output as input and generates a caption. For *grounded* approaches, more supervision is required: image features, such as regions, are additionally inserted into the visual encoder (Lu et al., 2018). Following the trend in other deep learning tasks, recent approaches include large-scale vision-language pre-training, as well as generalized models that work for a variety of computer vision and vision-language tasks, including image retrieval, referring segmentation, and visual question answering (Zou et al., 2022; Li et al., 2022).

Interactive IC Interactive IC has not gained as much attention as other ML tasks. Jia and Li (2020) involve the human-in-the-loop by providing incomplete sequences as input, in addition to each image, during inference time. Biswas et al. (2020) extend the Show, Attend, and Tell architecture by combining high-level and low-level features, which provide explainability, as well as beam search during decoding time.

Data augmentation Data augmentation (DA) is widely applied to multiple tasks which include learning from large data, whenever there is a lack of annotated instances. It can additionally be used as a regularization technique to avoid overfitting by introducing noise into the dataset. In Computer Vision, transformations like cropping, warping, and horizontal/vertical flipping are often applied (Takahashi et al., 2019; Katiyar and Borgohain, 2021).

For text, augmentation methods need to be more elaborate, since image-inspired techniques often change the semantics of the text drastically. Popular methods include, but are not restricted to, EDA (Wei and Zou, 2019) (including random insertion, deletion, word swap), back-translation (Sennrich

et al., 2016; Turkerud and Mengshoel, 2021), synonym replacement and contextual augmentation (Kobayashi, 2018; Atliha and Šešok, 2020), often using a pre-trained language model (Devlin et al., 2019). For both modalities, retrieval-based augmentation from additional resources is possible as well (Li et al., 2021).

Continual Learning In cases where a model is trained repeatedly on new data, *catastrophic forgetting* (Kirkpatrick et al., 2017) can be observed. This refers to the degradation of model performance on older tasks when it is trained on new ones. In order to overcome this, continual learning methods are often applied. Methods such as weight regularization, encoder/decoder freezing, pseudo-labeling, and knowledge distillation, have been previously applied to IC models (Nguyen et al., 2019; Del Chiaro et al., 2020). In the natural language processing domain, de Masson d'Autume et al. (2019) use a combination of episodic memory replay during training and local adaptation of the model during inference.

3 Method

In this section, we describe the approach followed, including our benchmark strategy, our DA methods, as well as the episodic memory module. Our pipeline is illustrated in Figure 1.

3.1 Interactive IC pipeline

Architecture We experiment with a concrete implementation of the interactive approach outlined in Hartmann et al. (2022). We use a PyTorch implementation of *Show, Attend and Tell* (Xu et al., 2015). This architecture consists of a convolutional neural network (CNN) encoder, which is used to extract feature vectors from images, and a long-short-term memory (LSTM) decoder, which generates a caption conditioned on these vectors, with the use of attention. Following Dognin et al. (2022), we replace the ResNet encoder with a ResNext network (Xie et al., 2016).

For the decoder, an LSTM network is used. A problem arising from incremental training here is the expansion of the vocabulary. In order to tackle this problem, we rely on the subword vocabulary given by the BERT (Devlin et al., 2019) tokenizer provided by Huggingface[1]. By using a pre-trained subword tokenizer, we account for new words learned incrementally, without the need to expand the model size. The training strategy used is cross-entropy loss.

While current state-of-the-art architectures achieve better scores, we adapt this particular architecture because of its simplicity, and because its inputs are raw images, as opposed to image features like bounding boxes and labels from object recognition models, which further decreases preprocessing time. The pipeline can potentially be adapted to any IC model that takes images as input, rather than image regions and classes.

Data augmentation methods For our experiments, we use DA on Image (IMG), Text (TXT), and both modalities simultaneously (BOTH). For IMG, we use the Albumentations (Buslaev et al., 2020) library. We create a pipeline of different operations, including CLAHE, optical and grid distortion, blur, flip, and rotation. Our goal here is to introduce noise to the input data, in order to help the model generalize better to unseen data. For the

original captions:
a. 'a bottle of Hawthorne food supplement or drug',
b. 'Foreign currency is sitting on the table and it looks kind of exotic.'
c. 'White t-shirt with black line drawings of Mt Rushmore, motorcycles, rose and beer, with text.'

paws:
a. 'a bottle of Hawthorne food supplement or drug combination',
b. 'foreign currency sitting on the table and it looks sort of exotic.',
c. 'White tee with black line drawings by Mt Rushmore, motorcycles, roses and beer, with text.'

pegasus:
a. 'There is a bottle of Hawthorne food supplement.',
b. 'The table has foreign currency on it.',
c. 'The t-shirt has black line drawings of Mt Rushmore, motorcycles, rose and beer.'

Figure 2: Generated data points generated based on the DA methods described in subsection 3.1. Top: image DA (combination of several DA methods). Bottom: text DA.

[1]We use `bert-base-uncased`.

	train	val	test	all	WT
1	3,332	954	2,476	6,762	10,047
2	1,535	302	488	2,325	4,988
3	5,668	1,402	2,199	9,269	13,497
4	333	83	113	529	2,931
5	6,160	1,516	2,474	10,150	12,407
all	17,028	4,257	7,750	29,035	21,955

Table 1: VizWiz cluster (task) statistics after filtering out bad quality images (according to the procedure mentioned in subsection 3.3). WT stands for word types.

TXT modality, we aim at generating meaningful captions. For this reason, we employ two paraphrasing models provided by Huggingface, namely `pegasus_paraphrase`, a PEGASUS (Zhang et al., 2019a) model fine-tuned for paraphrasing, and `paws_paraphrase`, a T5 (Raffel et al., 2020) model trained on the PAWS (Zhang et al., 2019b; Yang et al., 2019) dataset. The reason we use two different paraphrasing tools is that we found out that the quality of the generated samples is different. In addition, paraphrasing quality drops in each tool when the number of paraphrases increases. In order to introduce more variety without compromising the quality, we decide to utilize two paraphrasing tools. In the case of combined (BOTH) DA, IMG augmented images are combined with synthetically generated captions. In every case, we generate batches that are 10 times bigger than the initial ones. Examples of generated data points can be found in Figure 2.

Episodic memory for lifelong learning In order to help the model retain old knowledge when being adapted to new data, we implement a continual learning method, more specifically a sparse memory replay that operates during training. We adapt the method described by de Masson d'Autume et al. (2019): During training, some samples/experiences are written into the memory. Every training sample has a certain probability to be selected for memory writing. These experiences are then sparsely replayed (i.e. 1 sample from memory for every 200 new data points, see subsection 3.2) while the model is trained on new data. This way, the model retains information from previous training iterations with very low additional computational effort.

3.2 Procedure and training details

We follow the pre-training/fine-tuning paradigm, where we first train the model on a *supervised pre-training* task using a large, generic dataset, namely MS COCO (Lin et al., 2014) (details below). During (supervised) pre-training, we do not use any DA or continual learning method. After obtaining the best model, we continue with our incremental *model adaptation*, during which we apply DA and continual learning.

Training details For the supervised pre-training step, we train our model on MS COCO in two stages: during the first training, we freeze the encoder and only train the decoder. The encoder is then trained in the second stage. For the adaptation step, we train our models on each task once.

We train with a batch size of 32 and a learning rate of 4e-4 for the decoder. For our memory module, the replay frequency is 200, as mentioned in subsection 3.1; that means that for every 200 batches, one batch is drawn from the memory and added to the current training batch. The memory writing probability is 20%.

We use early stopping. During our initial experiments, we trained with higher (p=10) and lower (p=2) patience values for early stopping. During our initial experiments, lower patience seems to produce better results, hence we adopt this value for our adaptation training. During supervised pre-training, we used 20 as the default patience value.

3.3 Datasets

Supervised pre-training step We first train our model on the MS COCO dataset (Lin et al., 2014). It contains 328k images, and it is broadly used as a pre-training dataset for vision tasks, including object recognition, object segmentation, and IC. We use the 2014 release, which contains 82,783 training and 40,504 validation images. Each image is annotated with five captions, describing the content of each image. We make use of the Karpathy splits (Karpathy and Fei-Fei, 2017).

Adaptation After obtaining the best possible captioning model trained on MS COCO, we train our model incrementally using VizWiz (Gurari et al., 2020; Simons et al., 2020), a dataset consisting of images taken by visually impaired people. Since there are no test captions available, we use the validation set as our test set. A part of the training samples is used as our validation set.

DA	+ cluster 1 [3332]				+ cluster 2 [1535]				+ cluster 3 [5668]				+ cluster 4 [333]				+ cluster 5 [6160]			
	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH
1	18.8	6.4	15.8	15.3	12.4	2.2	11.3	4.4	15.9	2.4	13.0	7.3	12.7	1.9	9.8	3.9	11.8	2.8	9.7	7.1
2					26.0	6.9	19.8	16.4	25.0	5.5	18.7	11.3	18.7	4.6	13.0	7.2	22.6	3.5	14.9	13.8
3									27.7	4.2	24.5	16.3	21.1	2.3	16.4	4.9	22.4	2.9	16.9	11.9
4													26.7	4.6	20.5	13.1	20.4	3.4	15.4	10.6
5																	25.9	3.7	19.2	15.3
all	18.8	6.4	15.8	15.3	16.4	3.4	14.6	7.4	23.6	3.6	19.9	12.2	18.4	2.4	14.2	5.0	21.2	3.3	16.2	12.1

Table 2: CIDEr results on our experiments on VizWiz data clustered according to the procedure described in subsection 3.3. We start with the model resulting from the supervised pre-training step on MS COCO and continue to train this model incrementally on the VizWiz clusters (+cluster ...). We include the amount of (original) training data in brackets. DA: Data augmentation, NO: no DA, IMG: image DA, TXT: text DA, BOTH: image and text DA. The numbers in the left column stand for clusters evaluated on. 'all' refers to the micro avg.

Dataset processing We want to simulate a continual learning setting where we incrementally adapt the IC model to new sets of user-specific data. For this, we split VizWiz into non-overlapping clusters representing user-specific datasets. We follow the procedure for other continual learning datasets, where data is split according to classes/concepts, and each new class/concept represents a new task (Del Chiaro et al., 2020). As the VizWiz dataset does not contain object annotations for its images, we resort to splitting the data according to the objects mentioned in the captions, using the procedure described below. The resulting clusters resemble the user-specific data we might expect to receive from different users in a real-world setup: Whereas one user might be more interested in captioning screenshots or images of IT-related concepts, another user might be interested in captioning images of containers of food and drinks, etc. Example NPs for each cluster can be found in Appendix A.

We follow the steps below:

1. We collect all nominal phrases (NPs) in the entire caption corpus. We use TextBlob[2] for the extraction of the NPs.

2. From all the NPs, we choose so-called *keywords*, namely phrases that appear at least 15 times in the dataset.

3. Using GloVe (Pennington et al., 2014) embeddings, we extract word embeddings for each keyword. In case a keyword is phrasal, we average between individual word embeddings.

4. We cluster the keyword embeddings in 5 clusters, using K-means clustering (Hartigan and Wong, 1979).

5. We iterate over all captions for each image, looking for relevant keywords, and assigning them to clusters. In case one image corresponds to more than one cluster according to its keywords, we favor the smaller cluster.

VizWiz contains some images of bad quality: in some cases, the caption reads *'Quality issues are too severe to recognize visual content'*. In order to avoid the generation of these captions during inference, they can be removed from the training set (Çaylı et al., 2022). In our work, we exclude an image from training, if at least 3 out of the five captions in the image contain this caption; that means that more than 50% of the annotators could not describe the content of the image. If *Quality Issues* are brought up only once or twice, we remove this caption and duplicate one or two of the other captions, so that, in the end, each image is annotated with five captions. We do not remove *Quality Issues* images and captions from our test set. We exclude a total of 2,146 images.

While we technically do not use the complete dataset provided, it is justified by the fact that we test our pipeline in a low-resource scenario. Table 1 includes statistics over our tasks, including word type counts.

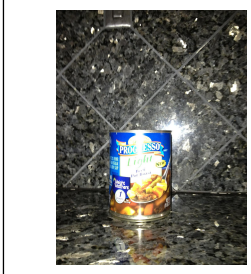
4 Evaluation & Results

In this section, we present the evaluation metrics we used, our procedure, as well as the results from our core experiments.

4.1 Evaluation metrics & splits

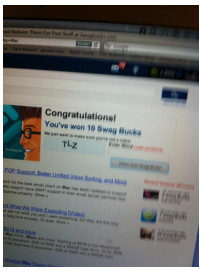
Since IC is a natural language generation task, results are evaluated using standard metrics for evaluating text generation tasks. These metrics measure similarity to the ground truths. The metrics most commonly used are BLEU (Papineni et al.,

[2] https://textblob.readthedocs.io/en/dev/



gold: a can of progresso light soup sitting on a counter
no: a can of campbell's cream of mushroom soup
txt: there is a can of soup on the counter.

gold: I see a red wine bottle with writing on it
no: a bottle of alcohol is on top of a table
txt: there is a bottle of wine.



gold: A computer screen wanting the user to fill a captcha field
no: a computer screen with a captcha on it
txt: there is a text on the screen.

gold: a package containing sugar free Hawaiian punch singles
no: a single packet of kool - aid drink mix
txt: there is a package of food.

Figure 3: Generated captions without DA and with TXT DA, compared with one of the gold captions.

2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), and SPICE (Anderson et al., 2016). For our hyperparameter tuning on the validation set, we use the BLEU metric. We report CIDEr scores in the main paper for brevity, scores for the other evaluation metrics can be found in Appendix B. We use the Pycocoevalcap[3] library for evaluation. In order to evaluate the continual learning abilities of our IC model, we report scores per cluster, as well as micro-averages over the clusters trained so far.

4.2 Results

We present our results in Table 2. The use of our DA methods does not improve the results. Especially when IMG DA is involved, performance drops dramatically compared to the NO DA baseline. This leads us to the conclusion that the DA operations we applied to the images were not suitable. Unexpectedly, we observe that TXT DA does not improve results compared to the NO DA baseline, which is in contrast to findings of previous work showing that caption augmentation is beneficial for low-resource IC (Atliha and Šešok, 2020). We analyze this in more detail in section 5.

5 Analysis

In this section, we take a closer look into the quality of the captions generated by our models. We focus on the NO and TXT models since they produce better results. We also conduct two ablation studies: one considers training without the use of the memory module, and the other one tests our method in a low-resource scenario.

[3] https://github.com/salaniz/pycocoevalcap.git

	NO	IMG	TXT	BOTH
no. of types	1,383	2,418	1,397	1,053
∅ (median)	10.0	10.0	8.0	10.0
∅ (mean)	10.229	10.464	7.949	9.894

Table 3: Statistics over captions generated with our models. ∅ : average caption length.

5.1 Caption quality

In order to gain a better insight into our results, in particular the observation that TXT DA worsens results compared to the NO DA baseline, we compare the generated captions based on their average length and the number of unique word types contained in the captions. One aspect that strikes immediately when comparing captions generated with TXT DA vs NO DA is variation. While we find that NO captions and TXT captions share a similar amount of unique word types, their average length is different, with TXT captions being more than 2 words shorter than NO captions.

We include some examples of generated captions in Figure 3. While we see that the captions generated are not necessarily erroneous, captions generated with the models trained with TXT DA are less informative than the gold captions and captions generated without DA. Automated evaluation metrics often penalize changes in the length of the output. Captions generated by the TXT DA model tend to be more similar to the paraphrases generated by the PEGASUS paraphrasing model (which was used to generate data for the training of the TXT DA model), which are shorter and less informative. Hence, this paraphrasing tool is not suitable for this

	+ cluster 1				+ cluster 2				+ cluster 3				+ cluster 4				+ cluster 5			
DA	NO		TXT		NO		TXT		NO		TXT		NO		TXT		NO		TXT	
MEM	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-
1	27.1	27.1	20.9	20.8	16.5	15.6	14.3	9.7	22.8	22.9	17.7	16.3	19.3	20.8	13.1	15.2	17.6	19.0	13.3	13.5
2					26.0	27.0	22.2	20.1	25.2	24.9	18.7	17.2	19.3	17.2	16.0	15.3	23.3	23.0	18.3	15.2
3									32.4	31.3	28.1	24.1	24.2	23.8	17.9	18.4	25.1	24.2	18.1	19.1
4													25.3	23.7	17.5	20.9	18.5	18.9	13.5	12.2
5																	27.1	25.6	19.9	19.0
all	27.1	27.1	20.9	20.8	**21.0**	20.4	**18.5**	14.3	**29.7**	29.1	**24.8**	22.0	23.4	**23.5**	17.2	**18.1**	**24.9**	24.3	**18.6**	18.4

Table 4: CIDEr results on the validation set for NO and TXT augmentation with (+) and without (-) episodic memory replay. We mark in **bold** the cases in which episodic memory contributes to an improvement, and in **red** the cases in which it does not.

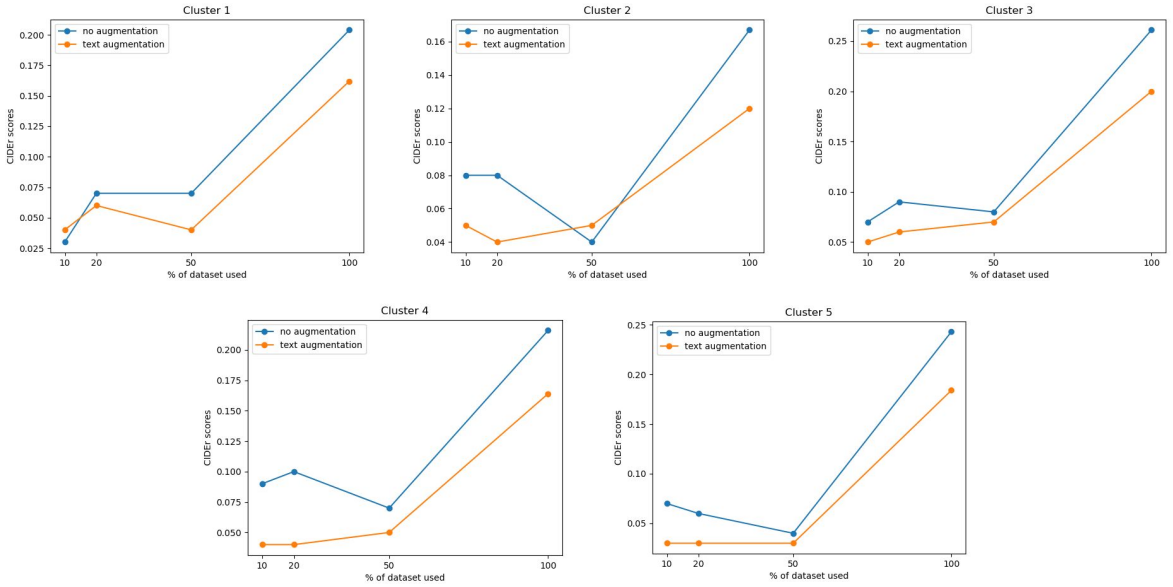


Figure 4: CIDEr results on the validation set for each task training with 10%, 20%, 50%, and 100% of the data.

particular task. In the future, we plan to compare more paraphrasing tools for DA on IC tasks.

To confirm our qualitative observations in a quantitative manner, we carried out a small manual analysis. We randomly sampled 100 captions generated with the TXT models and compared them to the gold captions. Our criterion was informativeness: we ranked each generated caption as *non-informative*, *partially informative*, or *very informative*. We find that 46 of them are very or partially informative, while for some of the rest, the lack of informativeness comes from the fact that the image quality is low (since seven of them contain severe quality issues).

5.2 Ablation study: Training without episodic memory replay

In order to investigate the effect of the sparse episodic memory replay on the continual learning abilities of the model, we train models in the same settings as in our core experiments, except for the use of sparse episodic memory replay. Results for these experiments are shown in Table 4. We observe that, in general, there is an improvement in performance in almost all cases, both in models trained with NO DA and in models trained with TXT DA. The only exception is the model after training with cluster 4, which is significantly smaller than the rest of the other clusters (approx. 1/3 the size of the next smaller cluster). This shows that, while the episodic memory module positively influences performance when at least 1000 samples are present, it is not as effective with very few samples.

5.3 Ablation study: Training with parts of the dataset

In an interactive setup, we cannot assume large amounts of annotated data provided by the user, hence we evaluate our models after training on only 10%, 20%, and 50% of the data of each clus-

ter. Training data points for each cluster are chosen randomly - for this reason, we present average scores over 3 trainings with the same settings. Our training takes place without memory since in most cases, the amount of data is too small for the memory to be activated. The results for models trained on reduced amounts of data for each cluster are shown in Figure 4.

It seems that TXT DA does not improve results even in a low-resource scenario - the curves for NO and TXT DA are similar for the larger clusters (1, 3, 5). For task 2, a slight improvement in performance can be observed when training with 50% of the data. This, in turn, leads to an additional observation, namely the fact that almost all our NO DA models deter when trained with half of the data of each cluster. This might be attributed to the data distribution of the clusters with which we trained.

6 Conclusion

We have presented a pipeline for interactive IC, which combines simple methods for incremental model improvement. This framework allows incremental adaptation of a pre-trained IC model to new data that is entered by users. The user input is transformed into a larger data batch using various data augmentation methods (for image, text, and both modalities). We additionally adapted a continual learning method for IC, which prevents catastrophic forgetting after repeated updates. In order to simulate incremental user input, we split the relatively small, domain-specific VizWiz dataset into non-overlapping clusters based on nouns mentioned in the image captions. VizWiz is a good test bed for our pipeline, as it contains real-world images with varying quality.

We analyzed the effectiveness of DA in our experiments, and we noticed a lower performance of our models when trained with augmented data. The drop in performance resulting from the application of DA methods was evident in our low-resource experiments as well. We concluded that, especially for IC, IMG DA must be applied carefully. The same applies to TXT DA: since brevity is penalized in this task, the DA outputs should be of similar length and descriptiveness as the gold captions. We confirmed that sparse memory replay does enable the models to retain knowledge learned from previous datasets while adapting to new data.

In the future, we plan to experiment with more elaborate joint DA methods for IC. Apart from evaluating the approach with respect to model performance using automated performance metrics, we intend to evaluate its usefulness and usability for end-users in a human study. Since prompting using large models is a popular paradigm recently, we intend to experiment with models like CLIP (Radford et al., 2021) as well, additionally assessing the trade-off between initial training cost and adaptation cost. Last but not least, applying active learning methods to select the best sample(s) for the episodic memory module can potentially increase the effectiveness of the continual learning method used in our pipeline.

Limitations

Despite the promising results of our IML pipeline for image captioning, our work has some limitations. Firstly, the experiments were conducted on a domain-specific dataset, VizWiz, and may not generalize to other datasets or domains. Secondly, our approach may not be suitable for scenarios where user feedback is sparse or unreliable, as the effectiveness of IML heavily depends on the quality and quantity of the feedback. Thirdly, our use of episodic memory to retain knowledge from previously seen clusters may not scale well to smaller datasets and other methods may be required. Lastly, our approach does not address the challenge of bias in the data, which can lead to biased models.

Ethical Statement

As of now, we do not see ethical concerns with the study presented in this paper. We used a dataset that is publicly available. The study is currently not applied to human subjects with personal data; in this case, the use of user feedback in the training process could potentially introduce biases if the feedback is not diverse or representative of the population. Lastly, our approach may be used to develop image captioning models that generate harmful or inappropriate content, such as captions that perpetuate harmful stereotypes or stigmatize certain groups of people.

Acknowledgments

We thank the reviewers for their insightful comments and suggestions. The research was funded by the XAINES project (BMBF, 01IW20005) and by the No-IDLE project (BMBF, 01IW23002).

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: semantic propositional image caption evaluation. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, volume 9909 of *Lecture Notes in Computer Science*, pages 382–398. Springer.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE CVPR conference*, pages 6077–6086.

Viktar Atliha and Dmitrij Šešok. 2020. Text augmentation using bert for image captioning. *Applied Sciences*, 10(17):5978.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. ACL.

Rajarshi Biswas, Michael Barz, and Daniel Sonntag. 2020. Towards explanatory interactive image captioning using top-down and bottom-up features, beam search and re-ranking. *KI - Künstliche Intelligenz, German Journal on Artificial Intelligence - Organ des Fachbereiches "Künstliche Intelligenz" der Gesellschaft für Informatik e.V. (KI)*, 36:1–14.

Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. 2020. Albumentations: Fast and flexible image augmentations. *Information*, 11(2).

Adrian Bussone, Simone Stumpf, and Dympna O'Sullivan. 2015. The role of explanations on trust and reliance in clinical decision support systems. In *2015 International Conference on Healthcare Informatics*, pages 160–169.

Özkan Çaylı, Volkan Kılıç, Aytuğ Onan, and Wenwu Wang. 2022. Auxiliary classifier based residual rnn for image captioning. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1126–1130.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Riccardo Del Chiaro, Bartłomiej Twardowski, Andrew D Bagdanov, and Joost Van de Weijer. 2020. Ratt: Recurrent attention to transient tasks for continual image captioning. *arXiv preprint arXiv:2007.06271*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the NACCL: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. ACL.

Pierre Dognin, Igor Melnyk, Youssef Mroueh, Inkit Padhi, Mattia Rigotti, Jarret Ross, Yair Schiff, Richard A. Young, and Brian Belgodere. 2022. Image captioning as an assistive technology: Lessons learned from vizwiz 2020 challenge. *J. Artif. Int. Res.*, 73.

Lijie Guo, Elizabeth M. Daly, Oznur Alkan, Massimiliano Mattetti, Owen Cornec, and Bart Knijnenburg. 2022. Building trust in interactive machine learning via user contributed interpretable rules. In *27th International Conference on Intelligent User Interfaces*, IUI '22, pages 537–548, New York, NY, USA. Association for Computing Machinery.

Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. 2020. Captioning images taken by people who are blind. *CoRR*, abs/2002.08565.

J. A. Hartigan and M. A. Wong. 1979. A k-means clustering algorithm. *JSTOR: Applied Statistics*, 28(1):100–108.

Mareike Hartmann, Aliki Anagnostopoulou, and Daniel Sonntag. 2022. Interactive machine learning for image captioning.

Zhengxiong Jia and Xirong Li. 2020. Icap: Interactive image captioning with predictive text. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, ICMR '20, pages 428–435, New York, NY, USA. Association for Computing Machinery.

Andrej Karpathy and Li Fei-Fei. 2017. Deep visual-semantic alignments for generating image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):664–676.

Sulabh Katiyar and Samir Kumar Borgohain. 2021. Image captioning using deep stacked lstms, contextual word embeddings and data augmentation. *arXiv preprint arXiv:2102.11237*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the NACCL: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. ACL.

Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, et al. 2022. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. *arXiv preprint arXiv:2205.12005*.

Guodun Li, Yuchen Zhai, Zehao Lin, and Yin Zhang. 2021. Similar scenes arouse similar emotions: Parallel data augmentation for stylized image captioning. *arXiv preprint arXiv:2108.11912*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. ACL.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Huan Ling and Sanja Fidler. 2017. Teaching machines to describe images via natural language feedback. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5075–5085.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2018. Neural baby talk. In *Proceedings of the IEEE CVPR conference*, pages 7219–7228.

Giang Nguyen, Tae Joon Jun, Trung Tran, Tolcha Yalew, and Daeyoung Kim. 2019. Contcap: A scalable framework for continual image captioning. *arXiv preprint arXiv:1909.08745*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 311–318, Philadelphia, Pennsylvania, USA. ACL.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. ACL.

Tingke Shen, Amlan Kar, and Sanja Fidler. 2019. Learning to caption images through a lifetime by asking questions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10393–10402.

Rachel N. Simons, Danna Gurari, and Kenneth R. Fleischmann. 2020. "i hope this is helpful": Understanding crowdworkers' challenges and motivations for an image description task. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW2).

Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. 2021. From show to tell: A survey on image captioning. *CoRR*, abs/2107.06912.

Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. 2019. Data augmentation using random image cropping and patching for deep cnns. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(9):2917–2931.

Ingrid Ravn Turkerud and Ole Jakob Mengshoel. 2021. Image captioning using deep learning: Text augmentation by paraphrasing via backtranslation. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 01–10.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *2015 IEEE CVPR*, pages 4566–4575.

Zijie J. Wang, Dongjin Choi, Shenyu Xu, and Diyi Yang. 2021. Putting humans in the natural language processing loop: A survey. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 47–52, Online. ACL.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on EMNLP and the 9th IJCNLP*, pages 6382–6388, Hong Kong, China. ACL.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2016. Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd ICML*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France. PMLR.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In *Proc. of EMNLP*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13041–13049.

Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, Nanyun Peng, Lijuan Wang, Yong Jae Lee, and Jianfeng Gao. 2022. Generalized decoding for pixel, image, and language. *CoRR*, abs/2212.11270.

A Example NPs for VizWiz clustering

We include example nominal phrases (NPs) from our VizWiz clustering. We follow the procedure described in the main body of the paper. For each cluster, we include 20 NPs. While there is no perfect separation in object categories, we do notice certain semantic similarities between the NPs in most clusters:

cluster 1	cluster 2	cluster 3	cluster 4	cluster 5
gift card	ac	kitchen counter top	ingredients	dark surface
button	labrador	top portion	small packet	glass cup
camera lens	quaker	small dog	crock pot	light fixture
nutrition information	stouffer	bottom	large bottle	wooden countertop
apple	dr	left side	nutritional	beige carpet
video games	packet	small	kitchen appliance	black
electrical outlet	screenshot	eye drops	ingredients label	lamp
tv screen	sainsbury	math problems	lotion bottle	wire
cable box	barcode	paper money	milk chocolate	concrete floor
computer tower	coke	led	liter bottle	interior
tv	nokia	person 's knee	dark chocolate	plastic container
cd case	samsung	's chicken	medicine bottle	marble counter
silver device	tan	brand name	frozen dinner box	glass container
keys	unopened	side view	dinner table	shorts
image quality	container/ box / bottle	counter top	water bottle	styrofoam
design	sprite	sunny day	small jar	couch cushion
entertainment center	the/this	remote control	spice	plastic wrapping
book page	roni	body	coffee pod	glass door
background	k-cup	room area	brownie mix	clear plastic bag
laptop monitor	upc	left side	ice cream	flat horizontal surface

Table 5: First 20 NPs for each cluster from the VizWiz Dataset

B Results for BLEU-4, METEOR, ROUGE, SPICE metrics

In the main paper, we only include CIDEr scores for our main experiments. Here we present results in four additional metrics: BLEU-4 (Table 6), METEOR (Table 7), ROUGE-L (Table 8), and SPICE (Table 9). The tables can be found on the next page.

	+ cluster 1				+ cluster 2				+ cluster 3				+ cluster 4				+ cluster 5			
DA	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH
eval on 1	14.4	6.0	11.1	12.4	9.1	2.6	8.7	4.6	11.6	2.8	9.2	6.3	10.5	2.6	7.5	4.5	7.6	2.9	6.0	6.1
eval on 2					16.9	7.4	15.1	13.7	17.8	5.6	15.9	10.6	16.6	5.7	12.5	9.7	16.2	4.8	12.5	13.6
eval on 3									16.0	4.9	13.8	11.3	13.7	3.5	10.5	6.4	13.8	3.8	10.8	10.6
eval on 4													16.9	4.6	13.5	9.1	12.4	3.3	9.8	8.4
eval on 5																	15.1	4.4	11.3	12.1
micro avg	14.4	6.0	11.1	12.4	10.4	3.5	9.8	6.3	14.0	4.0	11.8	8.9	12.5	3.4	9.4	6.0	12.4	3.8	9.6	9.9

Table 6: BLEU-4 results on our experiments on VizWiz data clustered according to the procedure described in our main paper. We start with the model resulting from the supervised pre-training step on MS COCO and continue to train this model incrementally on the VizWiz clusters (+cluster ...). We include the amount of (original) training data in brackets. DA: Data augmentation, NO: no DA, IMG: image DA, TXT: text DA, BOTH: image and text DA. The numbers in the left column stand for clusters evaluated on. 'all' refers to the micro average score.

	+ cluster 1				+ cluster 2				+ cluster 3				+ cluster 4				+ cluster 5			
DA	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH
eval on 1	13.5	9.3	12.4	12.9	10.8	6.7	10.5	7.8	12.4	6.8	10.7	8.8	11.2	6.5	9.7	7.7	10.8	7.3	9.4	9.3
eval on 2					15.8	10.3	15.0	13.7	16.0	9.3	14.6	11.9	14.8	9.4	13.3	11.2	15.8	8.9	13.7	13.8
eval on 3									15.2	8.6	13.7	12.0	13.9	7.7	12.0	9.4	14.1	8.5	12.2	11.9
eval on 4													15.1	9.0	13.0	11.5	13.8	8.0	12.0	11.4
eval on 5																	15.4	9.3	13.3	13.0
micro avg	13.5	9.3	12.4	12.9	11.6	7.3	11.2	8.7	13.9	7.8	12.3	10.5	12.8	7.3	11.0	8.8	13.6	8.4	11.7	11.5

Table 7: METEOR results, as above.

	+ cluster 1				+ cluster 2				+ cluster 3				+ cluster 4				+ cluster 5			
DA	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH
eval on 1	34.0	26.0	31.0	31.5	28.3	21.0	28.8	23.1	31.3	22.2	29.0	26.9	30.8	21.6	26.9	24.8	29.9	23.6	27.0	28.4
eval on 2					39.0	30.8	39.1	36.1	42.4	29.3	39.5	34.9	39.7	29.7	35.8	33.6	42.3	29.8	37.2	39.9
eval on 3									39.8	27.2	37.2	34.3	37.4	25.6	33.2	29.8	38.7	27.8	34.5	35.3
eval on 4													38.0	27.1	34.4	32.2	36.4	26.8	34.0	33.7
eval on 5																	40.7	29.3	35.7	37.5
micro avg	34.0	26.0	31.0	31.5	30.1	22.6	30.5	25.3	35.9	25.0	33.5	30.8	34.5	24.1	30.5	27.8	36.7	27.0	32.6	34.1

Table 8: ROUGE-L results, as above.

	+ cluster 1				+ cluster 2				+ cluster 3				+ cluster 4				+ cluster 5			
DA	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH	NO	IMG	TXT	BOTH
eval on 1	7.5	5.3	7.2	7.2	5.0	1.7	5.0	2.1	6.2	1.5	5.1	3.1	5.0	1.5	4.6	1.8	4.6	1.5	4.0	3.3
eval on 2					9.6	3.8	8.8	6.6	8.8	2.5	8.2	5.1	7.9	3.2	7.4	4.5	8.6	2.2	7.6	6.6
eval on 3									8.3	2.4	7.3	5.6	6.8	1.7	5.7	2.7	7.0	1.9	5.8	4.8
eval on 4													8.0	2.9	7.1	4.5	7.0	1.7	6.1	5.3
eval on 5																	8.5	2.6	7.5	6.2
micro avg	7.5	5.3	7.2	7.2	5.8	2.0	5.6	2.9	7.3	1.9	6.3	4.3	6.1	1.8	5.4	2.5	6.8	2.0	5.9	4.9

Table 9: SPICE results, as above.