# Tails Tell Tales: Chapter-Wide Manga Transcriptions with Character Names — Supplementary Material

Ragav Sachdeva, Gyungin Shin, and Andrew Zisserman

Visual Geometry Group, Dept. of Engineering Science, University of Oxford

## 1 More on datasets

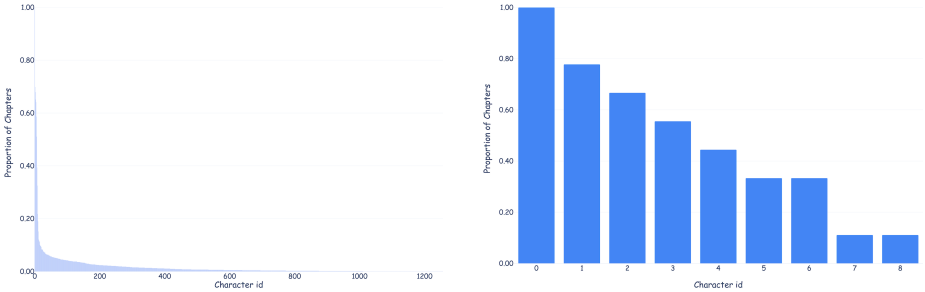
In this section we provide more details regarding the two dataset contributions—PopCharacters and PopManga-X.

### 1.1 PopCharacters

PopCharacters is a character bank dataset comprising principal characters from PopManga dataset, and containing information such as the characters' names, the manga series each character belongs to, a list of chapters that each character appears in, and a set of exemplar images for each character. In the following we provide details on the data curation process.

**Web-scraping.** Curating a character bank of principal characters for any arbitrary manga is a very tedious process. The only solution today is to read all chapters of the manga in question, manually keep track of all the characters that have been introduced and store this information in a dataset. This, of course, is a very expensive endeavour. Luckily, a lot of this heavy lifting has already been done by fans of most mangas in the PopManga dataset. Therefore, to compile the PopCharacters dataset, we semi-automatically scrape Fandom [9], a website for fans to catalogue details regarding their favourite manga. This results in 11K+ principal characters, across 76 series, with 16K+ thumbnail images, which forms the core the PopCharacters dataset.

**Analysis.** We make a few observations about the data scraped from Fandom. First, not all 84 manga series in PopManga have Fandom webpages with character information suitable for our purposes. Second, the downloaded thumbnails for characters are often not from the manga but instead from the anime adaptation of the series, and sometimes also from the live adaptation. These images have a significant distribution shift in terms of the appearance of the character and are not a good representation of the manga character. Third, out of the 11K+ characters scraped, around half do not have information on which chapters they appear in.




**(a)** One Piece **(b)** Vagabond

**Fig. 1:** Proportion of chapters for each character for two series: One Piece (a) and Vagabond (b). The proportion of chapters (y-axis) is defined as the number of chapters in which a character appears divided by the total number of chapters.

Given that the thumbnails scraped from Fandom are often not a good representation of the manga characters, we manually add a few 'exemplar' images for each character using crops from manga chapters. However, this is too expensive to do for each of the 11k characters. For instance, in manga series like One Piece, there are more than 1.2K characters that have been catalogued by fans. While each of them might play a significant role in the story, a handful of them appear far more frequently than others (see Fig. 1a). Since the purpose of this dataset is to help name the detected characters during inference, we limit the scope of 'exemplar mining' to characters that appear very frequently.

**Identifying frequently appearing characters.** We identify characters that occur frequently using the *list of chapters where a given character appears*. In particular, we consider the 'character appearance frequency' (which is the proportion of chapters the character appears in, defined as the number of chapters in which a character appears divided by the total number of chapters; e.g., if a certain character appears in Chapters 1, 2, 4, and 8 when there are 16 chapters for the series, its frequency is 0.25) as a quantitative measure. However, there is a notable challenge when using this statistic—as the number of chapters and characters are all different for different series, it is not obvious how to find a good character frequency threshold which well divides characters into the two groups and can be robustly used across the different series. For example, as shown in Fig. 1, two series with high and low number of characters reveal stark contrast in distribution in appearance frequency across different characters.

To solve this challenge, we take the following two-step approach. First, given a series, we classify whether it has a small number of characters ($<= 30$). If it does, we regard all of the characters as high-frequency characters. Then, for each series with many characters ($> 30$), we sort all the characters in decreasing order of their appearance frequency and select characters that account for up to 80% of the entire distribution.

**Exemplar mining.** After identifying the set of high-frequency characters, we use their scraped thumbnails as query images to retrieve high confident matches from the set of all character crops in PopManga. In the interest of diversity of appearance, we randomly select up to 20 retrieved candidate images (instead of considering top-20 most similar matches), which are then filtered manually. Our filtering criteria is: (i) remove false positives, (ii) remove low quality images (e.g. with significant occlusion by speech bubbles or other characters). In some cases there were a significant number of false positives, which does not reflect poorly on the embedding module but rather on the distribution shift of scraped thumbnails.

**Shortcomings.** Despite being the first of its kind in the research community, the PopCharacters dataset has a few shortcomings. First, unlike in movies, manga characters are much more likely to undergo radical appearance changes due to aging, magical abilities that involve transformations, or simply changes in style. In this version of the dataset, although we include images of characters undergoing appearance changes, we do not classify differences between images of the same character, which may limit its applicability. Second, the dataset has not been manually verified in its entirety and may contain some noise, incomplete or even incorrect information as a consequence of web-scraping. Having said that, the subset of the data that is used in evaluation (as character bank in 'chapter-wide character naming') has been been through human quality assurance to ensure robust and fair benchmarking.

### 1.2 PopManga-X

PopManga-X is the extended version of the PopManga dataset [46], wherein the test images now contain annotations for (i) speech-bubble tail bounding boxes, (ii) text-box to corresponding tail-box association, (iii) the name (identity) of each character box, and (iv) sub-classification of text boxes. Note that, to ensure a high quality of the data, PopManga-X has been reviewed multiple times, independently, by two human annotators with domain expertise. In the following we provide details on the data annotation process.

**Speech bubble tails.** In manga, speech bubbles (see Fig. 2) are often used to enclose dialogues, narrations etc. These speech bubbles often, not always, have tails indicating who the speaker is, or even who the speaker is not (in case of "negative tails" or tails pointing away from a character and towards the edge of the panel). We manually annotate these tail boxes by drawing tight bounding boxes around them. Additionally, we annotate the text-tail associations, which is a many-to-many relationship—for instance, a multi-part speech balloon may only have 1 tail but multiple text boxes, and, a speech bubble may have multiple tails indicating that it is being simultaneously said by multiple characters.


**Fig. 2: Examples of speech bubbles and their intentions.** We note that this list is not complete and also not universal. Manga artists typically have their own unique conventions (e.g., it is common to also have speech bubbles with no tails as 'normal speech' and not 'thinking'). Image taken from animeoutline.com.

**Character Name Annotation.** Previously in PopManga, character boxes had a per-page cluster ID indicating which character boxes on the page belong to the same character (*i.e.*, have the same identity). These cluster IDs were not globally unique across the entire chapter or the series. Towards the goal of character name aware transcript generation, we label each character box with a globally unique ID (name). This is done manually by a human by considering the context of the story and using reference images from PopCharacters (where available).

**Text Category Annotation.** Manga pages have all sorts of texts for the reader to enjoy. However, not all of it is essential to generate a transcript and in fact can actually be a nuisance, if inappropriately included in the transcript. We manually classify the text boxes in PopManga-X to record this information. Specifically, we identified the following 9 initial text categories. Fig. 3 shows a histogram for these text categories.

- Action/sound word: onomatopoeia or verbs describing action (e.g., "bang!" or "Slam!")
- Background information: narration or context
- Conversational text: conversations between characters
- Internal thought: texts for internal thoughts of characters
- Explicit interjection: interjections that are meant to be shown to other characters in the same scene
- Implicit interjection: interjections that are not supposed to be noticed by other characters or interjections in an internal thought
- Editorial note: book/chapter names, page number, or meta-level information that is not relevant to the story
- Scene text: texts that are part of objects such as signs
- Others.

**Table 1: Text categories.** The terms "bkg info" and "conv. text" refer to background information and conversational text, respectively.

| | text category |
|---|---|
| Non-essential | action/sound word, editorial note, scene text, others |
| Essential | bkg info, conv. text, interjections (explicit and implicit), internal thought |

We then group these 9 categories into two: essential or non-essential for dialogue as shown in Tab. 1. As a result, there are 13k+ and 7k+ texts for dialogues and non-dialogues, respectively.

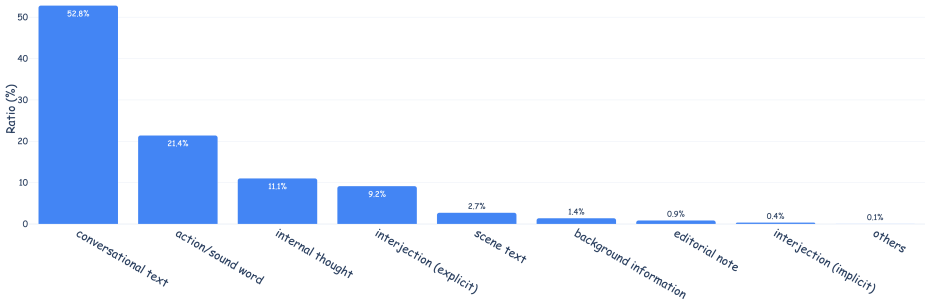



**Fig. 3: Histogram of the text labels.**

### 1.3 Ethical and legal considerations

In this research, we adhered to U.K. copyright law, which permits web scraping of publicly available content for non-commercial academic research purposes. However, researchers planning to use this data should ensure that their actions comply with the legal frameworks of their respective countries. We also recommend following any specific guidelines provided by publishers, which may impose further restrictions on the use of copyrighted materials in academic work.

## 2 More on semi-supervised training

The training datasets used to train the detection and association model is a mixed bag of unlabelled (most), partially labelled (some) and comprehensively labelled (few) images. We utilise Algorithm 1 to train our model in a semi-supervised way.

**Algorithm 1** Model Training Procedure

**Input:** Dataset $D_l$ labeled subset, $D_u$ unlabeled subset
**Initialise:** Model parameters, $\theta_{init}$
**Phase 0: Mine *partial* pseudo-labels** (no tails etc.)
**for** each $x_i$ in $D_u$ **do**
    $\hat{y}_i \leftarrow$ predict($x_i$) using Magi [46]
**end for**
**Phase 1: Warm-up**
**for** each $x_i$ in $D_u$ **do**
    Train model, using partial pseudo-labels $\hat{y}_i$
**end for**
Update model parameters $\theta_{init} \rightarrow \theta_{interim}$
**Phase 2: SSL training**
**repeat**
    **Phase 2a: Train on labelled data**
    **for** each epoch **do**
        Train model on $D_l$
    **end for**
    Update model parameters $\theta_{interim} \rightarrow \theta_{tuned}$
    **Phase 2b: Mine *complete* pseudo-labels**
    **for** each $x_i$ in $D_u$ **do**
        $\hat{y}_i \leftarrow$ predict($x_i$) using model $\theta_{tuned}$
    **end for**
    **Phase 2c: Re-train on pseudo labels**
    Initialise model with $\theta_{init}$
    **for** each epoch **do**
        Train on $D_u$ using complete pseudo labels $\hat{y}$
    **end for**
    Update model parameters: $\theta_{init} \rightarrow \theta_{interim}$
**until** fixed number of cycles
**Phase 3: Fine-tuning**
**for** each epoch **do**
    Train model on $D_l$
**end for**
Update model parameters $\theta_{interim} \rightarrow \theta_{tuned}$

## 3 More on edge prediction evaluation

The detection and association model is designed to output three kinds of edges: (i) character to character, (ii) text to character, and (iii) text to tail. In this section we provide more details regarding how our model's edge predictions are evaluated and the design decisions.

**Character-character edges.** The important thing to note about character-character edges is that they are transitive in nature. Therefore, the evaluation setting is that of cluster prediction and the metrics used—AMI, NMI, R-P, P@1,

MAP@R, MRR—are the ones commonly used in clustering literature. Their implementation is taken from [33]. These metrics better reflect the task at hand than directly measuring the edge prediction quality, as is done below for other two types of edges.

**Text-tail edges.** The text-tail edges represent the relationship between text boxes and tail boxes, *i.e.*, whether a given tail corresponds to a given text box. This can be a many-to-many relationship, *i.e.*, 1 or more text boxes can have a 0 or more tails. For instance, a multi-part speech bubble has 2+ text boxes which may only have a single tail box, or a single text box may have many tails indicating the case where multiple speakers simultaneously say something. To evaluate all these cases in a unified fashion, we treat it as a binary classification problem, and compute the average precision metric, as shown in Fig. 4.

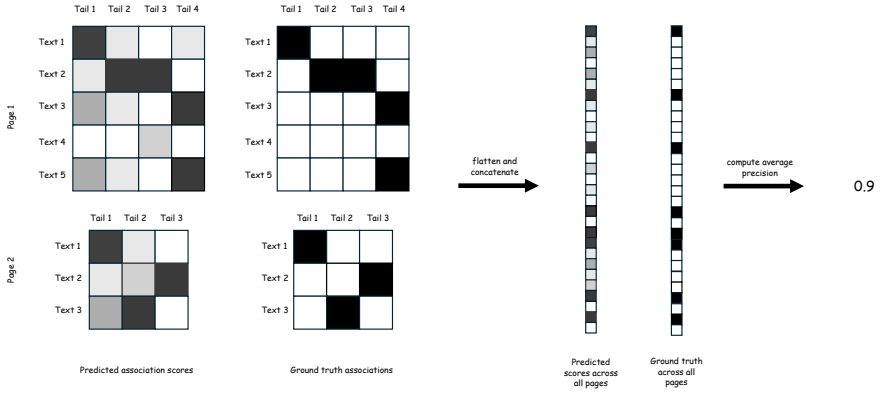



**Fig. 4: Text-tail (and text-character edge) evaluation process.**

**Text-character edges.** The text-character edges embed the speaker diarisation information, *i.e.*, which text box is said by which character. The important thing here is to note the distinction between (i) text box to character box association, and (ii) text box to character identity association. While the model outputs the former, we actually care about the latter in terms of generating the transcript. In other words, 'text box 1 is associated to character box 1' is no different than 'text box 1 is associated to character box 2', *if* character box 1 and 2 belong to the same character. This text to character *identity* evaluation setting (instead of text to character *box*) is even more important when the speaker of a text box is not present in the same panel as the text, but is present in preceding and subsequent panels. In such cases, it is almost arbitrary as to which particular character box is the right association; however, there is only a single correct character identity that must be associated with this text.

To evaluate the text-character identity predictions, we consider the text box to character box predictions by the model, and max-pool the scores for character boxes that have the same identity. Afterwards, the evaluation process is the same as for text-tail edges, as denoted in Fig. 4.

## 4 More on character naming evaluation

As mentioned in the paper, the purpose of this evaluation is to measure the efficacy of the method in forming chapter-wide character clusters. When evaluating a specific chapter, we sample exactly 1 exemplar image (from PopCharacters dataset) for each character that appears in this chapter. This exemplar is used as a reference image when assigning/matching crops to this character, and it was arbitrarily chosen by a human beforehand for each character in the test set and fixed for the purposes of evaluation.

The advantage of using a single exemplar image per character is that it keeps the evaluation fair as some characters have more exemplars than others. However, this introduces another variable – "the choice of exemplar image" which can significantly impact the performance results (some exemplars are better representations of the character than others). To demonstrate just how significant the gap can be, we report the character naming results in Tab. 2 when "optimal exemplars" are used. Specifically, instead of arbitrarily choosing and fixing an exemplar image from PopCharacters dataset, for a given character, we consider all possible crops of this character in the chapter being evaluated (using ground truth information) and use the crop which has the highest average similarity to all the other crops of this character, as the "optimal exemplar". For the sake of evaluation, this has a few benefits – (i) it eliminates the "choice of exemplar" variable from the evaluation process, (ii) sampling the exemplar from within the chapter increases the likelihood of the exemplar being visually representative of the character, thus reducing the effect of edge cases, and (iii) it makes the evaluation setting agnostic to the external character bank which is desirable for future benchmarking and comparison.

As evident from Tab. 2, the character naming results are significantly better when "optimal exemplars" are used. Of course during inference we would never have such knowledge about optimal exemplars, and therefore these results are difficult to achieve in practice. An ideal case scenario during inference is that several diverse exemplars are available for each character and their average embedding is used as the representation of the character. To keep things simple, we have not deeply investigated this.

**Table 2: Character Naming Results.** We report the accuracy results, which have an upper bound of 1.0.

| embedding model | method | notes | exemplars | PopManga-X (Test-S) | PopManga-X (Test-U) |
|---|---|---|---|---|---|
| Magi | K-means [28] | nclusters= $k+1$ | random, fixed | 0.3800 | 0.3993 |
| Magiv2 | K-means [28] | nclusters= $k+1$ | random, fixed | 0.4126 | 0.4221 |
| Magi | iForest [26] + K-means [28] | nclusters= $k$ | random, fixed | 0.4710 | 0.4859 |
| Magiv2 | iForest [26] + K-means [28] | nclusters= $k$ | random, fixed | 0.5298 | 0.5096 |
| Magi | Constraint Optimisation (Ours) | Predicted per-page constraints | random, fixed | 0.6637 | 0.7058 |
| Magiv2 | Constraint Optimisation (Ours) | Predicted per-page constraints | random, fixed | 0.7273 | 0.7530 |
| Magi | Constraint Optimisation (Ours) | Predicted per-page constraints | optimal | 0.8164 | 0.8375 |
| Magiv2 | Constraint Optimisation (Ours) | Predicted per-page constraints | optimal | **0.8735** | **0.8770** |
| Magi | Constraint Optimisation (Ours) | GT per-page constraints | random, fixed | 0.7445 | 0.7975 |
| Magiv2 | Constraint Optimisation (Ours) | GT per-page constraints | random, fixed | 0.7987 | 0.8786 |
| Magi | Constraint Optimisation (Ours) | GT per-page constraints | optimal | 0.8579 | 0.8786 |
| Magiv2 | Constraint Optimisation (Ours) | GT per-page constraints | optimal | **0.9219** | **0.9302** |

## 5 More on transcript generation

After detection (characters, texts, panels and tails) and association (character-character, text-character, text-tail), and chapter-wide character naming, generating the transcript is relatively straightforward. As mentioned in the paper, this is a four-step process: (i) filtering non-essential texts, (ii) text ordering, (iii) OCR, and (iv) generating the transcript using the predicted text-character associations and character names. The implementation for the ordering algorithm and the OCR model have been directly taken from [46] as the purpose of this work is not to improve on these.

Beyond that we highlight a few design decisions that can be made while generating the transcripts to make them more robust to model's mistakes and ensure narrative consistency. First, low-confidence speaker predictions for essential texts can be rendered as '<unsure>' in the transcript, rather than confusing the reader. Second, given that we have detected tail boxes, and matched them to their corresponding text boxes, with our method it is possible to indicate the speakers in the transcript only for the texts that have tails. In other words, for texts without tails, it might be reasonable to just include them in the series of dialogues without indicating the speaker and let the reader infer the speakers from the context. This has two benefits: (i) it is in-line with the manga artists' intention (the fact that they chose to not draw an explicit tail for some texts), and (ii) texts without tails are usually where the model makes more mistakes.

**On using LLMs to enhance the transcripts.** In this work we also investigated using LLMs to enhance the quality of the generated transcripts. While the LLMs do a very good job at fixing OCR mistakes and can be employed successfully for that purpose as a post-processing step, we were mainly interested in exploring if LLMs can leverage conversational history and context to fix speaker prediction mistakes. We discovered that text-based speaker diarisation is a very challenging problem where the ambiguity in predicting who the speaker is increases drastically as the number of speakers increases. Often in two-person conversations, it is possible to deduce a change in speaker based on the conversation pattern; however, with three or more potential speakers, many of whom can possibly be 'other', the problem becomes very challenging, and

we did not have much success with using LLMs. We also investigated training a vision-language model that leverages both vision and language cues for this task, but had limited success which we attribute to two reasons: (i) lack of large-scale ground truth annotations (our training was largely done on pseudo-annotations mined for large-scale data which is quite noisy), and (ii) the inherent imbalance in the data (most texts in fact can be attributed to the correct speaker simply by picking the nearest one, therefore the signal for language during training is rather weak).