Flipping The Script: Complementary Modes of Story Generation in Comics

Tony Veale[1]

[1]School of Computer Science, Belfield, Dublin D4, Ireland.

Abstract

An enduring Hollywood dictum tells filmmakers to "Show, Don't Tell." Cinema is primarily a visual medium, after all, and images stimulate our senses and our imaginations in different ways from words and music. The comic strip is another form of sequential art that follows the Hollywood dictum, though it is more apt to say that comics show *and* tell, for if they are well-chosen, words can reinforce and complement the images, and vice versa. This paper considers the automated generation of comic strip narratives that serve a serious purpose, such as the unbiased presentation of different sides of a contentious debate, or the teaching of a foreign language with a mix of text, image and audio. We explore two complementary approaches to the creation of comic narratives: in the first, the generation moves from plot to text to visually rendered story; in the second, which follows what is known as the "Marvel method" in the comics industry, a skeletal plot is first created and visualized, and only then is its textual substance (dialogue, descriptions, etc.) generated by an LLM with respect to this rendering.

Keywords

storification, comic strips, serious comics, the Marvel Method, Small Language Models (SLMs)

1. Introduction: Show *and* Tell

If a picture is worth a thousand words, as the old saw puts it, then what are we to make of generative AI's capacity for spinning complex and detail-rich images from prompts of just a few words? The tacit key to this equation is imagination, and the degrees of generative freedom it gives us. For just as words and images stimulate different senses, they can stimulate the imagination in different ways. Sequential art forms like cinema and the comic strip offer more than a series of still images; as the images guide the eye, the words can tell us what to think, and to feel, about what we are seeing [1]. In cinema, the projector moves still images at a fast enough speed to fool the eye into perceiving continuous motion; in a comic strip, the eye moves at its own pace, from one still image (or "panel") to the next [2]. The white-space gutters between panels also give more space to the imagination to insert itself into the interpretation process [3]. This additional time to think is well suited to serious comics, whose aim is to encourage readers to do just that, *think*, about a divisive topic or a learning goal. We present two kinds of serious comics here, and explore two complementary approaches to their production.

In: R. Campos, A. Jorge, A. Jatowt, S. Bhatia, M. Litvak (eds): Proceedings of Text2Story'25 Workshop, Lucca (Italy), 10 April 2025. ©2025 Copyright for this paper by its authors.Use permitted under Creative Commons License Attribution 4.0

tony.veale@UCD.ie (T. Veale)
http://Afflatus.UCD.ie/ (T. Veale)
0000-0003-2375-1811 (T. Veale)

CEUR Workshop Proceedings (CEUR-WS.org)

The first of these comic types concerns the balanced distillation of contentious debates on social media. The intent is to create a comic that has something for each side of a debate, which is to say, something to agree with and something to vehemently reject. By turning debates into two-sided comics, we aim to pop the filter bubbles that encourage opposing sides to ignore and talk past one another [4, 5]. The second type has a more overtly educational aim: to support the learning of a second language. Popular apps such as *Duolingo* gamify the learning of language as an interactive experience, and comics can provide a complementary form of learning material that can be enjoyed offline. The grounding of words in images is ideally suited to demonstrations of word meaning in context, especially if this context is a narrative one. A language-learning comic can illustrate meaning with apt emotions and actions, as single-panel lessons or as stories that extend over many panels. We will use large language models, or LLMs, to generate the text content for each of these comic types, and will explore two ways of linking this content to the most apt visual assets. The first of these is also the most obvious: once the text of a debate/story/lesson is produced by the LLM, it is segmented into a sequence of narrative "beats," and each beat is converted into a single panel. The LLM can help in this conversion if its propensity to hallucinate is managed. In the second approach, a skeletal plot is first generated using a symbolic story generator. This plot skeleton, or *fabula*, is then provided to the LLM as a guide to story generation. If a mapping from the plot primitives of the story generator to the visual assets of the comic generator already exists, then mapping the LLM's rich output into a suitable visual form becomes straightforwardly unambiguous.

2. XML: Comic Specification vs. Comic Rendering

From a publisher's viewpoint, comics are a malleable commodity. They can be re-printed, re-sized, re-coloured (or de-coloured, for black-and-white editions), re-structured (into smaller chapters, or longer compilations) and re-ordered (as a part of new or revised chronologies). It helps, therefore, to distinguish between the specification of a comic and its actual rendering. An XML schema can be used to express the composition of a comic in terms of its scenes, panels, and panel elements (figures, backdrops, balloons, speech, captions), and it may reference a set of pre-generated visual assets from which these elements can be rendered.

These assets — including images of characters in specific poses and expressing specific emotions, or background images for panels — can be created on demand by text-to-image diffusion models, or can form part of a pre-existing inventory of stock poses and backdrops that are recombined as needed by the renderer [6, 7, 8, 9, 10]. Veale [10] defined the *ComiXML* schema for this purpose, as part of a comics generator named *Excelsior*, that, in its earliest form, converted stories produced by the *Scéalextric* story generator [11, 12] into comic strips. This generator is grammar-based, and produces its skeletal plots by joining together triads of basic plot actions, of which Scéalextric defines over 800 (such as *love, kill, accuse*) for use in 2000 or so action triads (such as X *fall_in_love_with* Y : Y *cheat_on* X : X *accuse* Y). Excelsior simply maps these basic actions to visual assets for setting (such as *restaurant* or *hospital*) and for poses (for characters X and Y), stating their positions in a panel, so as to turn each action of a story into its own panel within a comic. As Scéalextric defines lines of dialogue for each participant in every action, this dialogue is directly transplanted from the story to the speech balloons of the comic.

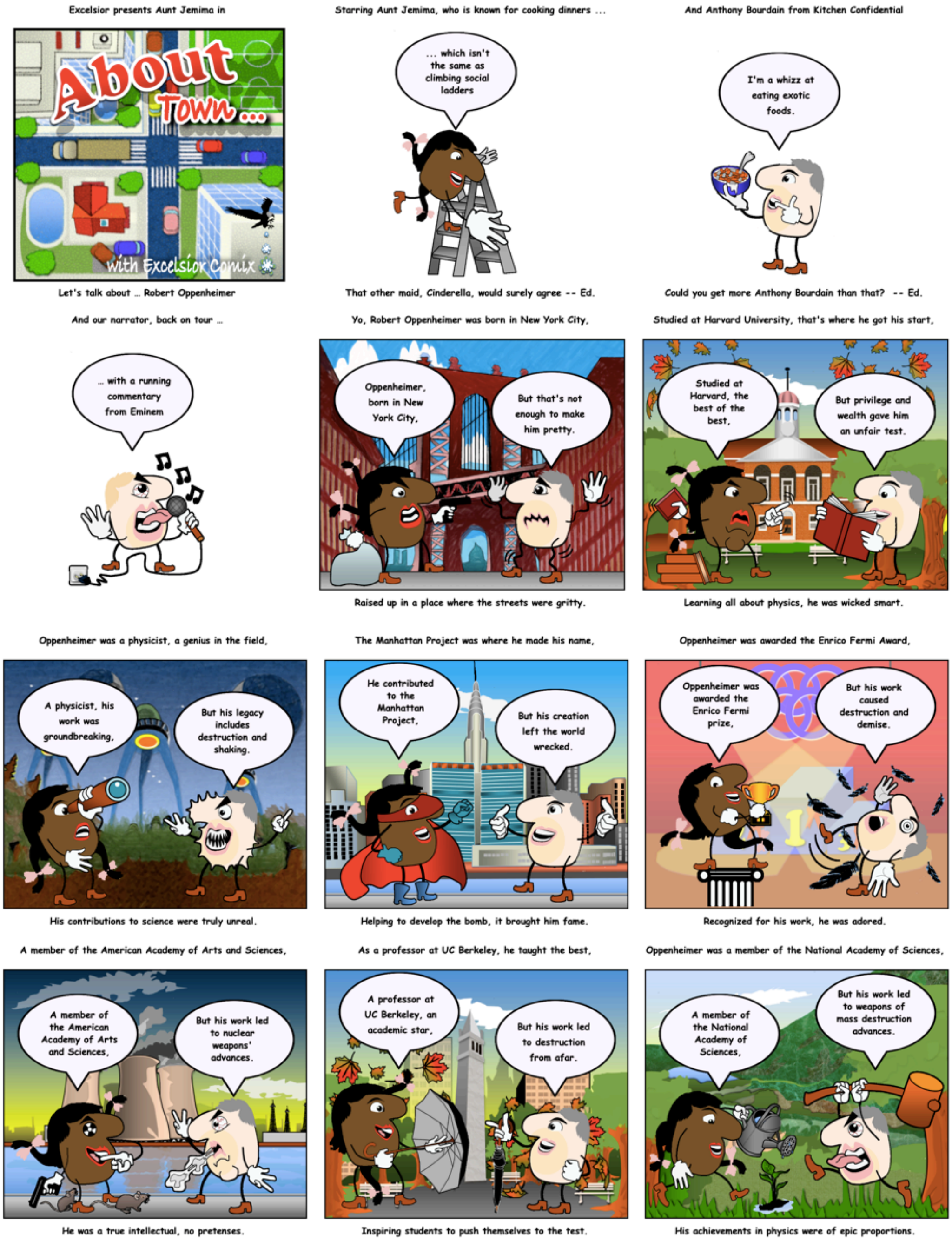
Excelsior presents Aunt Jemima in
About Town...
with Excelsior Comix
Let's talk about ... Robert Oppenheimer
Starring Aunt Jemima, who is known for cooking dinners ...
... which isn't the same as climbing social ladders
That other maid, Cinderella, would surely agree -- Ed.
And Anthony Bourdain from Kitchen Confidential
I'm a whizz at eating exotic foods.
Could you get more Anthony Bourdain than that? -- Ed.
And our narrator, back on tour ...
... with a running commentary from Eminem
Yo, Robert Oppenheimer was born in New York City,
Oppenheimer, born in New York City,
But that's not enough to make him pretty.
Raised up in a place where the streets were gritty.
Studied at Harvard University, that's where he got his start,
Studied at Harvard, the best of the best,
But privilege and wealth gave him an unfair test.
Learning all about physics, he was wicked smart.
Oppenheimer was a physicist, a genius in the field,
A physicist, his work was groundbreaking,
But his legacy includes destruction and shaking.
His contributions to science were truly unreal.
The Manhattan Project was where he made his name,
He contributed to the Manhattan Project,
But his creation left the world wrecked.
Helping to develop the bomb, it brought him fame.
Oppenheimer was awarded the Enrico Fermi Award,
Oppenheimer was awarded the Enrico Fermi prize,
But his work caused destruction and demise.
Recognized for his work, he was adored.
A member of the American Academy of Arts and Sciences,
A member of the American Academy of Arts and Sciences,
But his work led to nuclear weapons' advances.
He was a true intellectual, no pretenses.
As a professor at UC Berkeley, he taught the best,
A professor at UC Berkeley, an academic star,
But his work led to destruction from afar.
Inspiring students to push themselves to the test.
Oppenheimer was a member of the National Academy of Sciences,
A member of the National Academy of Sciences,
But his work led to weapons of mass destruction advances.
His achievements in physics were of epic proportions.

Figure 1: An *ExcelsiorLLM* comic on the topic of J. Robert Oppenheimer.

A ComiXML structure specifies every element of a comic, from the definition of its named characters (who have a gender and skin, hair and lip tones) to the order of panels within scenes and of scenes within the comic. Each panel, in turn, can place up to three figures against a backdrop image, and specifies the text to be uttered (or just thought) by each. A panel may also specify text captions to be placed above and below the action within. In total, Excelsior provides a stock of 500 or so poses and 500 or so backdrops for an XML specification to reference. The core tag set, which includes the tags $\langle panel \rangle$, $\langle scene \rangle$, $\langle figure \rangle$, and $\langle balloon \rangle$, is small but extensible. For example, the dialogue in language learning comics must be shown and spoken, so an audio tag can specify a sound file for use with a speech balloon.

3. Debate Comics: Why So Serious?

Serious comics do not operate in the realm of pure fiction. Although they aim for novelty of visual and linguistic presentation like any other comic, they must be constrained by shared knowledge of the domain in which they operate. A comic for language-learning can invent scenarios and stories for its lessons, but cannot invent new translations for the words and phrases that comprise them. A serious comic that seeks to distill a debate about a hot-button issue into a visual dialogue has considerable leeway in its choice of words and visual assets, but must aim to reflect what people are actually thinking. This requires knowledge of the domain.

This knowledge is a mix of the general and the specific, of common sense and topical facts. It is the former that allows a generator to grasp why certain facts can make one faction angry and bitter but another optimistic and gleeful, and it is these emotional perspectives on the facts that we want our comics to visualize. The system of [10, 13, 14] used different sources for each kind of knowledge. For its common-sense know-how it used an ontology of emotional frames and associated patterns of hashtag formation (such as *#cultOfX* and *#arrestX*). For its topical knowledge it looked to Twitter (now $\mathbb{X}$), either by trawling for a corpus of on-topic tweets or by directly searching for hashtags suggested by the ontology. Many of the tags suggested for topic x by the ontology will not be in use on Twitter within the one-week cut-off of its free search API, so [15] defined a tree-pruning approach to efficiently search for a large set of tags with the fewest API calls. The one-week search horizon is notionally a limitation, but it proves to be a feature, ensuring all matching tag uses are recent. These tags are linked to the ontology entries that suggested them, so they can be understood in terms of the ontology's own frames, specifically how one frame relates to others. The relations between ontology frames (e.g., between *disappointment* and *anger*, or between *joy* and *optimism*) allow the generator to construct a simple story arc in which debate figures move from one emotional state to the next. The ontology also associates dialogue templates (with slots for the topic x) with each frame, so it is a simple matter to generate the speech balloons for the resulting comic strip.

The split between a generic ontology and specific topics serves *Excelsior* well, allowing it produce comics for a wide range of contentious issues. Yet the generic nature of its framing and its dialogue can make those different comics look and sound very similar, and so its outputs soon appear formulaic and staid. An LLM can generate fresh dialogue for each panel, if it is prompted with the ontology's own framing as a guide. Alternatively, an LLM can generate all of the substance of a comic, from the talking points its characters will debate to the lines they will utter. A large LLM will have broad support in its training data for most topics, and topicality can be enhanced by priming the LLM with recent hashtags. This is the approach taken in *ExcelsiorLLM* [15]. The OpenAI LLM *GPT4o-mini* is first prompted to generate 10 neutral talking points on the topic of interest, as a numbered list of semantic triples. For instance, when the topic is Elon Musk, the triples include ⟨ *Elon Musk, is the CEO of, Tesla* ⟩. The LLM is then prompted to generate two perspectives on each triple, the benign view of a fan and the cynical view of a critic. So that these opposing views connect with a click, the LLM is also tasked with making them rhyme. Each talking point then receives its own panel, where the opposing views in each are counter-balanced as a rhyming couplet.

The LLM also suggests the visual assets to use in each panel, for the backdrop image (e.g., *car factory*) and for the poses of the opposing debaters. However, the LLM has a tendency to hallucinate poses and backdrops that do not exist in *Excelsior*'s repertoire, even when given a complete list of all its assets. To make sense of hallucinations, *ExcelsiorLLM* uses vector-space encodings [16] to match the LLM's inventions to the most similar asset in its visual inventory. The resulting comic of 10 panels, one for each talking point, is then introduced with a pithy remark by a famous figure (e.g., Borat, Eminem, Forrest Gump) for which Excelsior has a visual asset. This figure then narrates the comic to come, by providing captions above and below each panel. The LLM is a skilled mimic, and captures the distinctive linguistic voices of these figures rather well. A comic generated in this way for the topic *Robert Oppenheimer* is shown in Fig. 1.

4. Flipping the Script in Language Learning

Vector encodings map the gist of a talking point or a line of dialogue to a visual asset — talking points to backdrops, dialogue to poses — but such mappings are coarse and ignore the peculiarities of a text or an asset. Consider the Excelsior pose for *sympathetic*, in which a caring figure offers a tissue to dry one's eyes. If dialogue is generated that exudes a sense of concern, this pose will rank high as a match candidate, but the dialogue will not have been produced with knowledge of the pose's visual details. Conversely, if the pose is chosen before any dialogue is generated, the LLM can weave a sympathetic text that reinforces the image, as in "Don't cry. Here, take one of *these*." Of course, it helps if the other figure seems to be crying when this line is said, so an LLM should ideally know all of the poses (and their visual details) in a panel before it generates any dialogue for that panel.

Flipping the script in this way, so that visual decisions are made before textual ones, is the essence of what is called "The Marvel Method" [17]. As used at Marvel Comics from the 1950s, the method allows a writer and an artist to sketch a rough outline of a plot, which the artist then elaborates into a visual narrative. All panel, pose and viewpoint decisions are made by the artist, before the writer ever writes a single caption or a line of dialogue for the fully rendered story. This gives the artist considerable freedom, and also allows the writer to refer to specific visual choices when creating dialogue and exposition. To use the Marvel method for automated comic generation, we will need a means of generating high-level plots, of rendering those outlines as visual sequences, and of layering a rich text of dialogue and exposition over each sequence. We can use *Scéalextric* or a similar system to generate high-level plots, and the mapping of *Scéalextric* actions onto Excelsior assets to generate a visual rendering of these action sequences. By prompting the LLM with a combination of plot points (what is happening to whom, because of whom, and where) and of verbal action descriptions (what those happenings look like to an observer), it can generate captions and dialogue that speak directly to the final visualization.

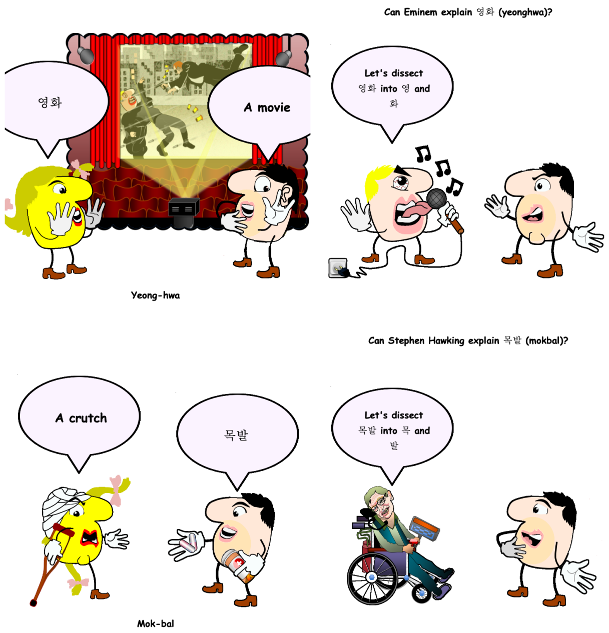
영화
A movie
Yeong-hwa
Can Eminem explain 영화 (yeongwha)?
Let's dissect 영화 into 영 and 화
Can Eminem explain 영화 (yeongwha)?
영화
Shadow (영) picture (화).
A crutch
목발
Mok-bal
Can Stephen Hawking explain 목발 (mokbal)?
Let's dissect 목발 into 목 and 발
Can Stephen Hawking explain 목발 (mokbal)?
목발
Wood (목) leg (발).

Figure 2: Two vignettes from a comic for learning Korean, with LLM-generated language insights.

In a language learning setting, the *Marvel Method* can be used on a small-scale, for vignettes, and on a larger scale, for complete narratives. A vignette is a single scene, extending over one or more panels, in which a character illustrates with their actions a word or a phrase and its translation. Two are shown in Fig. 2 above. To produce a vignette, the system first selects from its inventory a pair of related visual assets at random, a pose and a backdrop. A description of each is provided to the LLM, which is prompted to choose an apt word or phrase in this context that a language student might learn. The LLM is then asked to generate the target language translation of this text, while a text-to-speech (TTS) service is used to obtain audio files for both source and target texts. We use OpenAI's TTS service, a multilingual, multi-voice service which returns *.mp3* files via its web endpoint. The vignette is rendered over a small number of panels, in which a speaker is introduced (in the relevant pose), the setting is established, and the speaker speaks the source language text, both as a text balloon and in an audio format. Another figure, in a pose reacting to that of the first speaker, then utters the text in translation, again as a balloon text and as an audio recording. The pacing of this sequence can be compressed into a single panel or spread over several, to give the reader more time to digest and learn the content.

To produce a complete story using the Marvel Method, we start with the plot. The *Scéalextric* system defines an inventory of 800 primitive actions from which a large corpus of reusable plots can be woven for the generic characters X and Y. As outlined in [12], these two-character plots revolve around love, hate, competition, deception, betrayal and forgiveness, and typically involve an ironic twist or a sudden reversal of fortune. For consistency, we use the same recurring characters, named "Alfie" and "Betty," across all stories in our language comics, as this also allows the audio recordings of the LLM's dialogue (which can refer to others by name) to be cached and reused whenever possible. Fixing the gender of Alfie and Betty also allows the system to choose appropriate voices from the TTS service. As with the vignettes, the LLM is tasked with adding meat to these skeletal plots, by generating exposition and dialogue in response to descriptions of how each story beat, a single action involving Alfie and Betty, will be rendered by Excelsior. In effect, the LLM is tasked with doing to Excelsior's panel layouts what writer and editor Stan Lee would do to the layouts of artists Jack Kirby and Steve Ditko at Marvel: to write each story anew to suit what has already been visualized.

4.1. Alfie-Betty Pruning

The commercial LLM *GPT4o-mini* [18] satisfies all our text generation needs, from dialogue and exposition to the translation of words and phrases, while OpenAI's TTS service is an effective source for their audio equivalents. The former charges for compute on a per token basis, the latter per character, and while the charges are modest they are cumulative. So just as a large set of vignettes is generated in advance, we also pre-generate 500 stories with their translations and audio files, allowing new language comics to be generated by recombining these elements.

To get the most from our compute while ensuring our stories are as diverse as possible, we whittle our 500 plots from an initial pool of 12,000 as generated using the *Scéalextric* story grammar. Looking at the larger pool, we see that many stories have a similar shape, and send their characters along similar trajectories to the same, or very similar, ends. The same reversals of fortune also occur again and again, making the unexpected quite expected in the aggregate. Some actions, such as X *become_dependent_upon* Y, occur in as many as 1 in 7 plots, while this action precedes Y *is_overworked_by* X in one third of those cases. However, an analysis of which actions and action pairs recur most often allows us to prune the most heavily rutted paths, leaving the 500 most diverse plots. After pruning, no single action occurs in more than 6% of these, none occurs as the final action in more than 4%, and no action pairing occurs in more than 5%. Although we can hope that an LLM may weave different surface narratives from similar plots, diversity is improved if we can feed the LLM a broad range of plots to begin with.

4.2. All Generators Great and Small

As our LLM of choice, *GPT4o-mini* satisfies a wide-range of needs from the cloud. Since we produce our language-lesson vignettes and stories offline, a commercial cloud-based LLM of this size is both practical and cost-effective. However, there are deployment scenarios where a smaller LLM, an "SLM," should be used locally and perhaps even shipped with the comics generator. One such scenario relates to the use of visual stories in games. As with comics, games have their own visual assets that a new story must accommodate, and although many games run in the cloud, a games producer will likely prefer to not incur the costs of a commercial LLM across hundreds of thousands of users. To this end, we have experimented with story-generation via the Marvel Method on an SLM that is fine-tuned on example narative outputs [19].

TinyLlama [20] is a small language model of just 1.1 billion parameters that builds on the foundations of the larger Llama 2 model from Meta [21] (which comes in 7, 13 and 70 billion parameter varieties). This SLM (v1.1) has been pre-trained on 2 trillion tokens of filtered web text and code. We use a 4-bit quantized version of *TinyLlama* from *unsloth* for fast fine-tuning with *LoRA* (low-rank adaptation), and fine-tune the SLM for three different tasks: generating exposition for a skeletal Scéalextric plot; generating dialogue for the characters X and Y to go with this exposition and plot; and generating skeletal plots for itself, given only the types of X and Y (e.g., "a bully and a victim"). A training set of 3009 examples per task, or 9027 in total, is provided for fine-tuning. For the first two of these tasks, *GPT4o-mini* is used to generate the exposition for the skeletal plots (task 1) and to provide dialogue to go with this exposition (task 2), so we are effectively training *TinyLlama* to take the place of *GPT4o-mini*. For the third task, only *Scéalextric* data is used to fine-tune the creation of new plots. When tuned for one epoch, the SLM's outputs are comparable to those of *GPT4o-mini* for the same tasks, which is perhaps unsurprising, as the former is a concentrated distillation of the latter with fewer parameters.

5. Conclusions: Twist and Shout

If a source of data can be turned into a narrative (e.g., see [22]), that narrative can be turned into a comic. But thinking of narratives in terms of comics is useful even if one never intends to visualize them as such, because comics force us to be explicit about scenes, beats, settings and character placement. A comic gives an explicit temporal and visual structure to a narrative, and allows it to be edited, sampled and restructured like the work of sequential art that it is. In the case of language-learning comics, where the story serves an educational purpose, structuring the narrative with the ComiXML schema is especially useful. Language lessons are not one-off events, and are most useful when repeated. But this repetition should not be entirely predictable, and an XML schema allows us to shuffle the ordering of vignettes and stories in a comic, simply by shuffling the scenes without shuffling the panels within. This makes a comic a dynamic entity that preserves its lesson content while changing its layout each time it is read.

Reorganization at the XML level is also well-suited to experimenting with, and customizing, the presentation of information within a comic. In language stories, for instance, we have several options as to how target translations can be woven into the narrative. We can do this *within* panels, so that each panel contains both the source and the target form of a line of dialogue; or we can do it *across* panels, so that every other panel is a translation of the one that went before it; or we can do it across *replays*, so that readers are shown the whole story first in the source language and then again in the target language. It may be a matter of personal taste, or a matter of pedagogical effectiveness, as to which of these is preferable, and the XML schema allows us to edit and rework a language lesson with ease.

A comic's XML schema affords more flexibility in how its content is consumed, while the *Marvel Method* offers more flexibility in how that content is created. Since the value of a comic lies in its tight integration of text and visuals, so that one leans on and reinforces the other, it is vital that decisions in one modality are informed by, and inform, decisions in the other. A comic strip is a representation of a narrative, and that representation may evolve over time as visual assets are updated or replaced, dialogue is re-generated (perhaps in a new language or a different register) and exposition is rewritten (perhaps in a new authorial style, such as that of H.P. Lovecraft, or from a different stance), or as one LLM (or SLM) is replaced by another. In the case of debate comics, the Marvel Method allows the visual rendering to remain unchanged as the text is re-generated, perhaps to suit a shift in public opinion or a newly trending hashtag. A comic is more than a finished product; it can also be the starting point of a dynamic story.

References

[1] W. Eisner, Comics & Sequential Art, Poorhouse Press, Tamarac, Florida, 1985.

[2] N. Cohn, The Visual Language of Comics: Introduction to the structure and cognition of sequential images, Bloomsbury, London, UK, 2013.

[3] S. McCloud, Understanding Comics: The Invisible Art, Harper Collins, New York, 1993.

[4] P. Barberá, J. Jost, J. Nagler, J. Tucker, Tweeting from left to right: Is online political communication more than an echo chamber?, Psychological Science 26 (2015) 1531–1542.

[5] S. Zannettou, "I won the election!": An empirical analysis of soft moderation interventions on Twitter, in: Proc. of the 15th International AAAI Conference on Web and Social Media (ICWSM-2021), 2021, pp. 865–876.

[6] D. Kurlander, T. Skelly, D. Salesin, Comic chat, in: Proceedings of SIGGRAPH'96, the 23rd annual conference on Computer graphics and interactive techniques, ACM, 1996, pp. 225–236.

[7] T. Alves, A. McMichael, A. Simões, M. Vala, A. Paiva, R. Aylett, Comics2D: Describing and creating comics from story-based applications with autonomous characters, in: Proc. of the 20th Annual Conference on Computer Animation and Social Agents, 2007.

[8] J. Walsh, Comic book markup language: An introduction & rationale, Digital humanities quart. 6 (2012).

[9] R. Pérez y Pérez, N. Morales, L. Rodríguez, Illustrating a computer generated narrative, in: Proceedings of the ICCC-2012, 3rd International Conference on Computational Creativity, Dublin, Ireland, 2012, pp. 103–110.

[10] T. Veale, Two-fisted comics generation: Comics as a medium and as a representation for creative meanings, in: Proceedings of ICCC-22, the 13th International Conference on Computational Creativity, Bolzano, Italy, 2022, pp. 59–66.

[11] T. Veale, Déjà vu all over again: On the creative value of familiar elements in the telling of original tales, in: Proceedings of ICCC-17, the 8th International Conference on Computational Creativity, Atlanta, Georgia, 2017.

[12] T. Veale, Appointment in Samarra: Predestination and bicamerality in lightweight storytelling systems, in: Proceedings of ICCC-18, the 9th International Conference on Computational Creativity, Salamanca, Spain, 2018.

[13] T. Veale, Have I got views for you! generating 'fair and balanced' interventions into online debates, in: Proceedings of ICCC-23, the 14th International Conference on Computational Creativity, Waterloo, Canada, 2023.

[14] T. Veale, The funhouse mirror has two sides: Visual storification of debates with comics, in: Proceedings of the Text2Story workshop at ECIR 2023, Dublin, Ireland, 2023.

[15] T. Veale, From symbolic caterpillars to stochastic butterflies: Case studies in reimplementing creative systems with LLMs, in: Proceedings of ICCC-24, the 15th International Conference on Computational Creativity, Jonkoping, Sweden, 2024.

[16] R. Greene, T. Sanders, L. Weng, A. Neelakantan, New and improved embedding model, Open AI, 2022. URL: https://api.openai.com/v1/embeddings.

[17] S. Howe, Marvel Comics: The Untold Story, Harper, New York, NY, 2012.

[18] J. Menick, K. Lu, S. Zhao, E. Wallace, H. Ren, H. Hu, N. Stathas, F. P. Such, GPT-4o mini: advancing cost-efficient intelligence, Open AI, 2024.

[19] Z. Xie, T. Cohn, J. H. Lau, Can very large pretrained language models learn storytelling with a few examples?, ArXiv abs/2301.09790 (2023).

[20] P. Zhang, G. Zeng, T. Wang, W. Lu, TinyLlama: An open-source small language model, 2024. URL: https://arxiv.org/abs/2401.02385. arXiv:2401.02385.

[21] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, et al., Llama 2: Open foundation and fine-tuned chat models, 2023. URL: https://arxiv.org/abs/2307.09288. arXiv:2307.09288.

[22] B. Santana, R. Campos, E. Amorim, A. Jorge, P. Silvano, S. Nunes, A survey on narrative extraction from textual data, Artif. Intelligence Review (2023). doi:10.1007/s10462-022-10338-7.