Building Multimodal Corpora Using Microtask Pipelines and Local Annotators

Helmiina Hotti, Raúl Vázquez, Anna-Kaisa Jokipohja, Timo Kalliokoski, Henna Paakki, Rosa Suviranta, Tuomo Hiippala
Multimodality Research Group, University of Helsinki
Helsinki, Finland
{helmiina.hotti, raul.vazquez, anna-kaisa.jokipohja, timo.j.kalliokoski, henna.paakki, rosa.suviranta, tuomo.hiippala}@helsinki.fi

Abstract

Multimodality, or how human communication and interaction combine multiple forms of expression, is studied across diverse fields of research. Many of these fields have underlined the need for large, richly annotated multimodal corpora to support empirical research. While language resources are increasingly annotated using microtask crowdsourcing, multimodal corpora remain largely reliant on expert annotators, which creates a bottleneck for scalability and broad applicability. This paper presents a novel hybrid approach to multimodal corpus annotation, leveraging the efficiency of microtask pipelines while preserving theoretical rigour. Our approach decomposes the annotation process into sequences of simple, well-instructed tasks, which are then performed by locally recruited non-expert annotators. We demonstrate the feasibility of this approach by presenting a pipeline for annotating the multimodal structure of school textbooks.

Keywords: multimodality, corpora, annotation, microtask, crowdsourcing

1. Introduction

Multimodality, or how human communication and interaction combine multiple forms of expression, is now actively studied across diverse fields ranging from linguistics (Levinson and Holler, 2014; Stöckl, 2020), semiotics (Bateman et al., 2017) and digital humanities (Smits and Wevers, 2023) to psychology (Perniss, 2018) and language resources (Lukin et al., 2024; Lai et al., 2024). Although these fields approach the phenomenon of multimodality from distinct theoretical perspectives, many of them share one key concern: the need for large corpora with rich annotations for supporting empirical research on multimodality (see e.g. Bateman et al., 2004; Allwood, 2008; Huang, 2021).

Multimodal corpora are actively developed among communities concerned with language resources (Carletta, 2007; Kipp et al., 2009), linguistics and multimodality research (Bateman, 2008; Hiippala et al., 2021; Rühlemann and Ptak, 2023). Many language resources are now created using microtask crowdsourcing (Poesio et al., 2017), in which the annotation work is broken down into small, piecemeal tasks suitable for non-expert annotators. However, the infrastructure available for building multimodal corpora is still largely developed with individual expert annotators in mind (Kipp, 2001; Wittenburg et al., 2006; Cardoso and Cohn, 2022). This presents a logjam that prevents creating large corpora needed by the diverse research communities studying multimodality.

In this paper, we present ongoing work on developing a novel approach to annotating multimodal corpora, which combines aspects of microtask crowdsourcing and traditional approaches to building multimodal corpora using expert annotators. We decompose the annotation process into sequences of simple tasks, coupled with carefully designed instructions, which are then performed by locally recruited non-expert annotators.

To create the infrastructure needed for supporting this effort, we repurpose an existing commercial annotation tool, Prodigy[1], which we then enhance with additional components for (1) combining the annotation tasks into pipelines, (2) cross-validating the annotations and (3) supporting annotator access to tasks. We demonstrate our proposed approach using a pipeline designed for annotating the multimodal structure of double-page spreads in primary school science textbooks written in English.

The paper itself is structured as follows. In Section 2 we review current challenges in annotating multimodal corpora and relate our approach to the state-of-the-art, followed by a discussion of designing the annotation tasks and pipeline, the role of instructions and the technical implementation in Section 3. In Section 4, we demonstrate the application of the pipeline to primary school science textbooks. Section 5 presents our conclusions and future work and Sections 6 and 7 address the limitations and ethical issues associated with our work.

[1] https://prodi.gy

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 6483–6495
11–16 May 2026.

2. Challenges in Building Multimodal Corpora

The phenomenon of multimodality is manifested in diverse combinations of 'modes' of expression. Face-to-face interaction involves the coordinated use of speech, gestures, posture, gaze and other embodied resources, whereas everyday media such as websites, newspapers, school textbooks and instruction manuals integrate written language, various forms of depiction, photographs, diagrams, maps and layout (Bateman et al., 2017).

Capturing how such multimodal ensembles are formed and what kind of variation they exhibit across cultures and situations has been a major concern among the research communities that study multimodality, and corpus-based approaches have been proposed as useful for improving our understanding in this area (see e.g. Allwood, 2008; Kong, 2013; Thomas, 2014; Bateman, 2014; Knight and Adolphs, 2020; Huang, 2021). However, a wider adoption of corpus approaches in research on multimodality remains challenging for several reasons, which are introduced below.

To begin with, much of the work on multimodal corpora assumes a strict division into linear and non-linear data (Bateman, 2012), that is, between communicative phenomena that unfold in time and those that are organized in space. More specifically, linear data allows aligning annotations against time, which provides a robust organising principle for describing the multimodality of communicative situations that unfold temporally (see e.g. Allwood et al., 2007; Cavicchio and Poesio, 2012). Non-linear data, in contrast, includes communicative artefacts that unfold in space along two or more dimensions, such as various types of documents (Bateman, 2008) or diagrams (Hiippala et al., 2021), whose multimodal structure may be described using coordinates that designate points in 2D/3D space.

Most tools developed for annotating multimodal corpora follow the linear/non-linear division (Cassidy and Schmidt, 2017), because they have been developed among communities that primarily work with just one type of data (see e.g. Kipp, 2001; Wittenburg et al., 2006; O'Donnell, 2008). However, real-life multimodal communication rarely respects this division: news broadcasts, for example, unfold in time while simultaneously organising e.g. camera feeds, written language and graphic elements in the 2D space on the screen (Hensellek, 2025). Although these limitations may be negotiated by cross-referencing linear and non-linear annotations (Thiele et al., 2024), corpus approaches to multimodality would benefit from tools that allow annotating data that unfolds in time, space or along both dimensions simultaneously.

Current tools are also primarily designed for expert annotators who work independently of each other. Although some tools have adopted a client–server design to enable collaborative annotation (Cardoso and Cohn, 2022), the most widely used annotation tools such as ANVIL (Kipp, 2001) and ELAN (Wittenburg et al., 2006) are distributed as standalone applications that do not support dividing the annotation workload among multiple annotators. This applies equally to more recent tools such as Multimodal Analysis Image (O'Halloran et al., 2011) and GRAPE-MARS (Fortanet-Gómez et al., 2025). The orientation towards individual experts presents a formidable challenge, because annotating multiple modes of expression for their features naturally increases the annotation workload. In short, relying on individual expert annotators prevents scaling up the size of multimodal corpora.

For linguistic corpora, the volume of annotations has been scaled up using microtask crowdsourcing (Poesio et al., 2017). Microtask crowdsourcing, which decomposes larger tasks into piecemeal work and distributes this effort to workers on online platforms (Gadiraju et al., 2014; Bhatti et al., 2020), is now widely used in data-intensive fields, including natural language processing (Zhou et al., 2022; Mickus et al., 2024), computer vision (Kovashka et al., 2016) and digital humanities (Causer et al., 2018). Although there have been previous attempts to annotate multimodal corpora using microtask crowdsourcing (Kembhavi et al., 2016; Hiippala et al., 2022), annotation quality remains low even if the annotators are taught to perform the task and provided with carefully designed instructions (Suviranta and Hiippala, 2025).

Finally, multimodality is studied within diverse research communities, which approach the phenomenon from different theoretical and methodological standpoints. Some of the current tools have been designed to work with particular approaches only, as exemplified by Charon, which is explicitly designed for working with FrameNet (Belcavello et al., 2022). This needs to be accounted for when developing annotation tools, which should be able to implement annotation schemes that have been derived from different theories of multimodality (Cardoso and Cohn, 2022). To summarise, annotation tools should enable working with different types of data and annotation schemes.

3. The Way Forward: Microtask Pipelines for Local Annotators

We seek to address the challenges identified above by developing annotation pipelines that resemble those commonly used in microtask crowdsourcing, but which distribute the tasks to locally recruited annotators instead of an anonymous crowd recruited through an online platform (Poesio et al., 2017).

To reduce the need for expertise in applying complex multimodal annotation frameworks, we design small, generic microtasks that tap into everyday knowledge of multimodality, which are then joined together into a pipeline. Recruiting annotators locally enables easy communication and feedback between researchers and annotators, which is a common challenge in crowdsourcing (Nouri et al., 2020).

Realizing the full potential of microtasking requires designing effective annotation pipelines. This means that subsequent tasks are generated immediately after preceding tasks are completed (Mizusawa et al., 2018). To function efficiently and reliably, we suggest that the following criteria are necessary for microtask annotation pipelines:

1. ensure optimal **task scope**, which means that tasks are small enough to avoid ambiguity, but large enough to retain a meaningful context (Fort et al., 2011);
2. incorporate built-in redundancy as **validation mechanisms** to ensure consistency (Snow et al., 2008); and
3. provide a **flexible annotation infrastructure** that supports dynamic workflows, such as rerouting insufficiently annotated tasks for reannotation, and ingesting multiple types of data (Hiippala et al., 2022).

We seek to meet these criteria by decomposing the annotation process into separate tasks, which build on two fundamental task types: classification and segmentation. Combined with carefully designed instructions, these two fundamental task types allow annotating data *and* reviewing annotations created by others, thus building validation and quality control mechanisms directly into the annotation pipeline. For implementing the annotation tasks, we use Prodigy, a proprietary, server-based annotation tool developed by Explosion AI.

As Borisova et al. (2024) note in a recent review of annotation software, Prodigy is one of the few tools that supports annotating text, video, audio and images, and allows tasks and user interfaces to be created programmatically. While relying on proprietary software is not necessarily optimally aligned with principles of open science, we argue that doing so allows us to direct our resources towards building multimodal corpora rather than developing the entire infrastructure from scratch, which would also require a long-term commitment to maintain the software (see Section 6).

3.1. Annotator Selection

Over the past decade, the quality and reliability of crowdsourced data has plummeted (Marshall et al., 2023), partly due to language barriers and the increasing use of generative models to automate task completion (Veselovsky et al., 2023; He et al., 2024). Hence, we seek to retain the benefits of microtasking while ensuring quality by recruiting a small cohort of trusted annotators.

Building on the work of Suviranta and Hiippala (2025), we designed a structured screening process to introduce potential annotators to the tasks and evaluate their suitability. The screening includes classification and segmentation tasks, which resemble those used in the actual annotation pipeline described in Section 4.1. We assessed annotator performance using quantitative metrics (e.g., F1-scores for classification and Intersection over Union for segmentation) as well as qualitative reviews to identify consistent errors or misunderstandings that may indicate issues in instruction clarity. Beyond performance, we prioritize ongoing interaction with annotators, providing dedicated channels for reporting technical issues, clarifying instructions, and offering feedback. To ensure fair and predictable compensation, we pay the annotators 13 €/h for their work regardless of task type.

3.2. Instruction Design

Clear instruction design is critical for effective microtask annotation, as it directly influences both annotation quality and workflow efficiency (Gadiraju et al., 2017). Unclear instructions can confuse annotators, leading to errors, reduced motivation, and task abandonment, all of which disrupt the pipeline and hinder progress. By ensuring clarity, instructions minimize ambiguity, maintain annotator engagement, and promote a more reliable annotation process.

Our instructions are simple and self-contained, and specifically designed for each task. Instead of producing an exhaustive annotation manual that annotators need to study beforehand, each task may be performed by relying solely on the instructions provided. The instructions are always available to annotators through the task interface. Their design aims at maximal accessibility and comprehensibility, while exposing the annotators minimally to the theoretical motivations behind the annotation task. The benefits of requiring less theoretical knowledge are twofold: it expands the pool of potential annotators and makes the instructions more adaptable across different data types.

The instructions follow a standardized HTML template, which ensures visual and structural consistency across tasks. The template can be adapted for each task type in the pipeline (classification and segmentation), allowing it to be reused and adapted as needed. Each set of instructions combines textual descriptions with example images drawn from the data that is being annotated. Combining verbal

descriptions with visual examples into multimodal instructions promotes clarity and reduces the need for prior theoretical knowledge (Suviranta and Hiippala, 2025). Visual examples help the annotator easily understand which tools to use, how to perform the task, and what the output should look like.

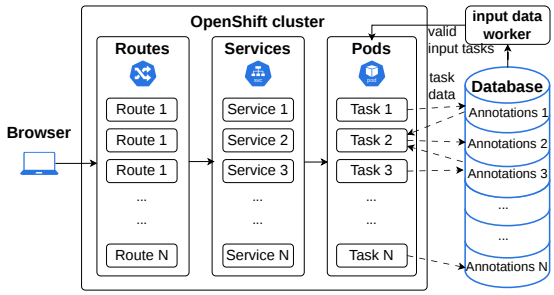


Figure 1: Architecture of the technical implementation. Prodigy instances are deployed in pods that interact with a relational database to poll input data and store annotations. The annotation interfaces are exposed to the internet with routes and annotators can access the interfaces on their browsers.

3.3. Technical Implementation

In this section, we present our annotation toolkit named RAMP (Reproducible Annotation Microtask Pipelines). The source code is openly available via our institutional repository.[2] The codebase includes the custom components needed for combining tasks into pipelines, a submodule containing definitions and instructions for the annotation interfaces presented in this paper as well as a deployment tool used to deploy the pipeline on an OpenShift cluster (see Figure 1).

We adopt a microservice-based deployment strategy using the container platform OpenShift. This setup allows treating each annotation interface as a separate, self-contained application that can be started, scaled, and terminated independently. To create an end-to-end pipeline consisting of several annotation interfaces, we chain together several Prodigy instances using custom components.

Each annotation interface is packaged into a container image that includes the necessary software and scripts to run a Prodigy instance configured for the task at hand. Containers running these images can be deployed independently with OpenShift, making it trivial to start or shut down multiple instances on demand. This poses an advantage for experimentation and iteration: new interfaces can be tested quickly without affecting ongoing work, and multiple pipelines can be active at once. For an overview of the microservice architecture, see Figure 1. Annotators access their annotation sessions in their browser with a personal annotator ID. Tasks are divided between annotators automatically by the routing mechanism in Prodigy, and each annotator can only access the specific tasks routed to them.

Prodigy offers different interfaces for annotating diverse data types (e.g. text, images and audio). Prodigy instances are defined with customizable "recipes", which consist of Python functions that describe a workflow. A recipe defines a source for the input data, task instructions and other interface details, and how the task results are routed and validated, among other possibilities. While Prodigy offers core functionalities out of the box, we developed several custom components to support the creation of complex pipelines (see Table 1).

Using the custom functionalities, we forward tasks from one interface to another based on conditions such as majority vote based on review or classification tasks. We found that adding such complex functionality directly in the Prodigy processes considerably slows down the interfaces, and therefore deploy a separate worker process to manage task forwarding (see Table 1). The worker is a separate Python process that continuously monitors the annotation database, filters out tasks that fail to meet overlap or other predefined criteria, and dynamically updates a list of valid input tasks for each annotation task accordingly.

To minimize the amount of *invisible labor* (see Section 7), we provide annotators with a simple website that shows which annotation instances currently have tasks available. When annotators enter their personal ID, they are immediately provided with links to the annotation tasks available to them.

4. Applying the Approach to Primary School Science Textbooks

We now demonstrate how our proposed approach can be applied to annotating the page-level multimodal structure of primary school science textbooks, which exemplify the kinds of materials commonly studied in multimodality research (Dimopoulos et al., 2003; Bezemer and Kress, 2010). More specifically, we design and implement a pipeline for performing **layout segmentation**, as defined in the *Genre and Multimodality* (GeM) model (Bateman, 2008, 116–117).

The GeM model defines multiple layers of annotation for capturing variation in how multimodal documents combine different modes of expression (Hiippala, 2017). In the GeM model, the purpose of layout segmentation is to establish an inventory of page-level analytical units, such as paragraphs, headings, diagrams, photographs and maps, which

[2] https://version.helsinki.fi/multimodality-research-group/ramp

Custom components	
Task routing	
input data worker	continuously update a list of input tasks for each annotation instance
review task router	route review tasks to annotators excluding the annotator of the original task
Task validation	
classification	only allow task submission if one option is selected
segmentation	only allow task submission if at least one segmentation has been added
segmentation review	only allow task submission if the original segmentations have not been modified
layers	only allow task submission if all segmentations are assigned to layers
Other functions	
majority vote	calculate majority vote for a set of examples
separate bounding boxes	create classification tasks from segmentation tasks, including one bounding box per task

Table 1: Overview of the custom components developed for building annotation pipelines using Prodigy

Text Elements (Task 7)		**Visual Elements (Task 8)**	
Category	Description	Category	Description
TEXTBLOCK	Continuous unit of text	PHOTO	Photographic images
HEADING	Standalone text elements that introduce or label the content that follows	DEPICTION	Non-photographic pictorial representations, such as drawings or illustrations
MARGIN	Content in headers, footers, or margins	DIAGRAM	Maps and schematic representations, such as charts and graphs
LIST	Sequences of items marked using bullet points, numbers, or other symbols	INTERACTIVE	Interactive content, such as textbook exercises
CAPTION	Descriptions of visual elements	FORMULA	Mathematical expressions or equations
		COMPOSITE	Elements that combine e.g. graphic shapes and written language, such as speech bubbles
		SHAPE	Abstract graphical elements, such as lines or icons
		TABLE	Tabular data presentations

Table 2: Textual and visual layout units in the primary school textbooks

may be then picked up for further annotation in subsequent layers (see also Bateman et al., 2001). We refer to these as **layout units**.

The GeM model defines the layout units individually for each medium under analysis: the assumption is that primary school science textbooks and technical manuals, for instance, operate with different sets of layout units. This complicates the use of document analysis systems for layout segmentation, as they typically operate with fixed categories defined by the training data. The layout units we identified as pertinent for describing the school textbooks are provided in Table 2.

4.1. Annotation Pipeline

Our pipeline for textbooks adopts a *cumulative, stepwise* approach, progressively refining the annotations from detection and segmentation to classification and layering. The input data consists of images of double-page spreads from primary school science textbooks, which regularly use the entire space provided by the entire page spread to organise content presented in written language, photographs, illustrations, diagrams and other modes of expression (Bezemer and Kress, 2008; Danielsson and Selander, 2016).

Figure 2 illustrates the entire annotation pipeline. To avoid unnecessary processing pages with textual layout units only, **Task 1** involves a binary choice about whether a page contains visual layout units or not (see Table 2). Each page is annotated by three annotators, and the system uses majority voting to resolve inconsistencies. Pages without visual layout units are forwarded to **Task 4** for segmentation of textual layout units, whereas pages with visual layout units are forwarded for segmentation in **Task 2**. The segmentation tasks **2** and **4** are followed by review tasks **3** and **5**, in which three annotators cross-check the annotations created by others, which are then returned for revision or passed along the pipeline.

In **Task 6**, the annotators check the annotations for completeness, and segment and categorise any

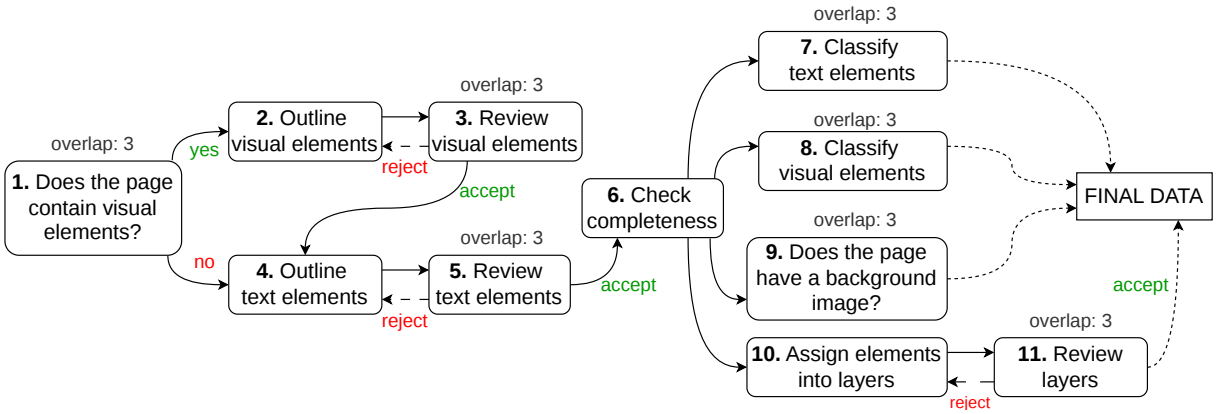
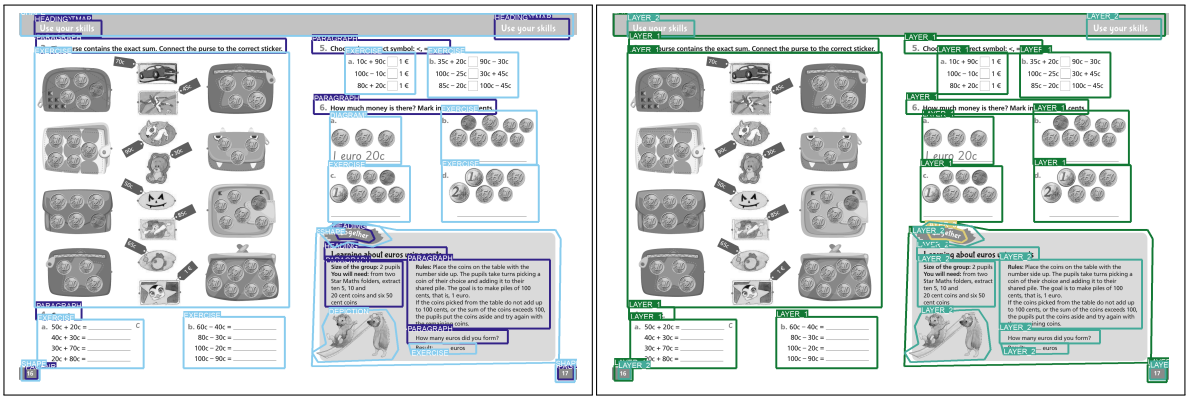


Figure 2: An annotation pipeline for describing the page-level multimodal structure in elementary school textbooks. Tasks marked with 'overlap' indicate that each data point in that interface is annotated by multiple annotators. The workflow incorporates built-in redundancy and iterative validation: tasks voted as incorrect by a majority of annotators performing review tasks are automatically re-queued for reannotation, ensuring consistent quality control. Tasks are run in parallel whenever possible to ensure efficiency. For the final data, results are aggregated from tasks 7-11, as indicated by the dotted lines.



(a) Layout segmentation

(b) Layout layering in depth

Figure 3: Example annotations for a double-page passed through the annotation pipeline

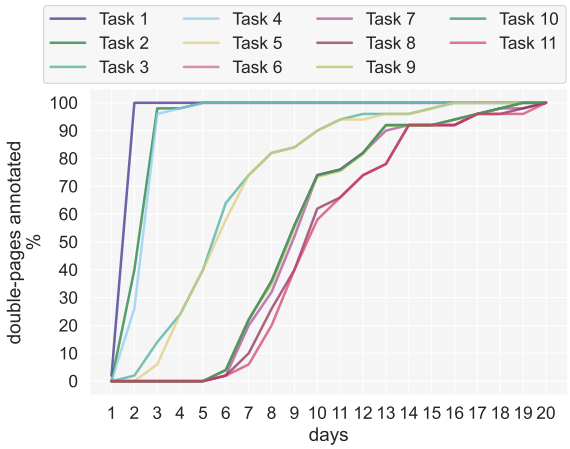
missing layout units. This safeguards against oversight in prior steps, after which the layout units are forwarded for further description. In **Tasks 7** and **8**, each textual and visual layout unit is classified into one of the categories provided in Table 2. The annotators are presented with a single layout unit per task to keep the tasks simple and reduce cognitive load. Each layout unit is classified by three annotators.

With all layout units segmented and classified, the pipeline proceeds to collect additional information about the use of layout space. In **Task 9**, the annotators define whether the page has a background image, which provides a foundation for describing how layout units are organised in depth. In **Task 10**, the annotators assign the layout units into layers to capture whether some layout units are contained within one another or overlap each other. The layer assignments are reviewed by three annotators in **Task 11** and returned for revision if necessary.

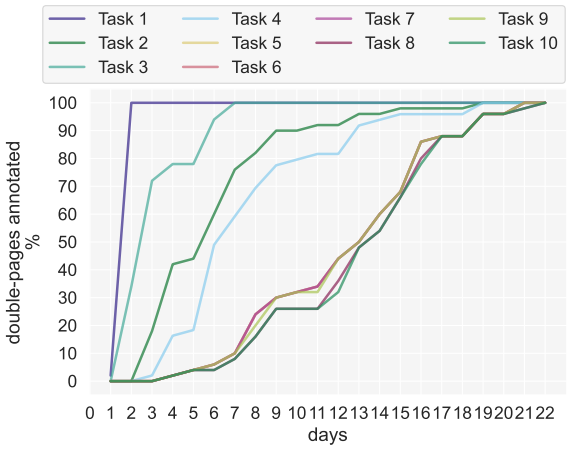
Figure 3 illustrates the output from the annotation pipeline for layout unit segmentation in Figure 3a (Task 6) and their spatial organisation in depth in Figure 3b (Task 11). This example illustrates our pipeline's flexibility: the same annotation infrastructure can be used for many different types of annotation. While we present one possible pipeline configuration, tasks can be split, merged, or reordered as necessary.

4.2. Observations from Ongoing Annotation

Figure 4 shows the progression for two annotation pipelines, which both seek to annotate 50 double-page spreads for layout units. In Pipeline 1, page spreads are annotated from scratch, as illustrated in Figure 2. In Pipeline 2, layout units predicted by the layout segmentation model DocLayout-YOLO (Zhao et al., 2024) are taken as the starting point for annotation, which the annotators review and



(a) Pipeline 1: Fully manual annotation



(b) Pipeline 2: Segmentation predicted by DocLayout-YOLO (Zhao et al., 2024) as the starting point

Figure 4: Annotation progress over time, measured by percent of annotated double-pages. Total number of double-pages in each pipeline is 50.

correct before moving on to classification and layering. These pipelines were run concurrently to examine how using automatic segmentation affects annotation efficiency. It is important to note that our annotators do not work on annotation tasks full time. When agreeing to participate in an annotation pipeline, the annotators commit to actively checking for available tasks daily, to ensure that tasks can advance through the pipeline.

In a typical annotation workflow, tasks in the beginning of the pipeline are completed in bulk before the subsequent annotation interfaces become active, as data points advance through the pipeline. Tasks requiring consensus through majority voting take longer to produce output: if a task is routed to be annotated by three annotators, just one inactive annotator can block the task from moving forward. The rate at which data points transition to later annotation stages is further influenced by two key factors: the *complexity* and *ambiguity* of the multimodal structure of the double-page in question. Highly complex layouts, characterized by a dense interplay of visual and textual elements, may increase the number of accidental annotation errors, while the ambiguity in defining the boundaries of layout units can produce multiple different, yet potentially valid interpretations.

The annotation tasks allowing for the most variability in interpretation, namely the segmentation tasks, are positioned early in the pipeline. Different interpretations of valid segmentations frequently lead to task rejection during review. This is in part due to different interpretations of how the instructions describe the desired level of granularity. High rejection rates can create bottlenecks in the pipeline, as tasks cannot advance until corrected and reapproved.

Improving the annotation instructions based on these findings and annotator feedback is a continuous effort, made possible by our deployment strategy that allows for improving the details of the pipeline with little to no disruption in the annotators' work. To exemplify, we have clarified the desired level of granularity for layout segmentation and added more extensive examples of desired annotations. This also highlights a key advantage of using trusted annotators over crowdsourcing: local, hired annotators are not only motivated to produce high-quality annotations, but also actively provide feedback, which enables an adaptive annotation process.

Certain configuration choices for annotation instances can also have a considerable effect in how the pipeline functions on a technical level. One such decision is whether to enable *work stealing*, a strategy provided by Prodigy that reassigns tasks from the queue of one annotator to another when the latter runs out of work. While work stealing can help mitigate delays caused by inactive annotators, we observed a higher-than-expected number of duplicate tasks as a result of tasks being completed by both the original annotator and the annotator who received the "stolen" task.

The impact of duplicates is cumulative: one duplicate early on in the pipeline produces duplicate tasks for that data point in all following stages. This causes a considerable amount of extra work for the annotators, slowing down the overall progression of the pipeline. For this reason, we decided to disable work stealing and implement other practices to ensure the progression of the pipelines. One such practice is dividing the data to small batches and deploying a separate pipeline for each batch. For each annotation batch, we only include annota-

tors that have confirmed to be available for active annotation work for the duration of the pipeline.

The expected duration of a pipeline depends on the amount of data and nature of the annotation tasks. The two pipelines described in this section were completed in approximately three weeks. The total number of completed tasks in Pipelines 1 and 2 (see Figure 4) are 6303 and 6331, respectively. These numbers highlight the amount of work needed for building richly annotated multimodal corpora: the layout annotation of 50 double-page spreads requires thousands of completed microtasks.

5. Closing Remarks and Future Work

Our microtask pipeline demonstrates that complex multimodal annotations can be decomposed into sequential, non-expert tasks without compromising theoretical rigour. By prioritizing modularity and validation, the pipeline addresses longstanding scalability challenges in multimodal annotation. Additionally, configurable and programmatically created annotation interfaces allow for reproducibility. To facilitate the adoption of the annotation infrastructure by other researchers, we have published the code with example configurations[3].

Looking ahead, we will test the pipeline's adaptability by extending it for other types of multimodal data beyond primary school textbooks, including manuals, magazines and news broadcasts. Our current technical implementation and instruction design make the workflow fully customizable, allowing adjustments to the task design, input data types and the workflow order. This versatility allows us to apply what we learn from the initial annotation of the textbooks to future pipelines to establish an iterative, scalable, always improving annotation workflow.

The workflow we present in this paper can be further scaled to larger datasets by implementing human-in-the-loop approaches. We plan to take advantage of suitable layout segmentation models and the inherent layout information embedded in PDF documents as a basis for layout segmentation. As described in Section 4.2, we have piloted using model-predicted layout segmentations as the starting point for annotation pipelines, but did not see improvement in efficiency as compared to fully manual annotation (Figure 4). In Pipeline 2, the majority of annotation effort was spent reviewing and correcting the model-predicted annotations, and the pipeline was completed two days later than Pipeline 1. This demonstrates the insufficient annotation quality of the out-of-the-box layout segmentation model (Zhao et al., 2024).

Without fine-tuning, layout segmentation models such as DocLayout-YOLO (Zhao et al., 2024) and LayoutLM (Xu et al., 2020) typically rely on predefined categories designed for generic document analysis. Employing human annotators to review and modify these preliminary annotations produces training data, which can in turn be used to fine-tune the models for the segmentation of theoretically derived layout units. After fine-tuning these models with human annotations, we hope to see an increase in annotation efficiency overall, as the starting point for the annotation improves in quality. Combining high-quality human annotations and the use of computational models bridges the gap between scalability and annotation reliability.

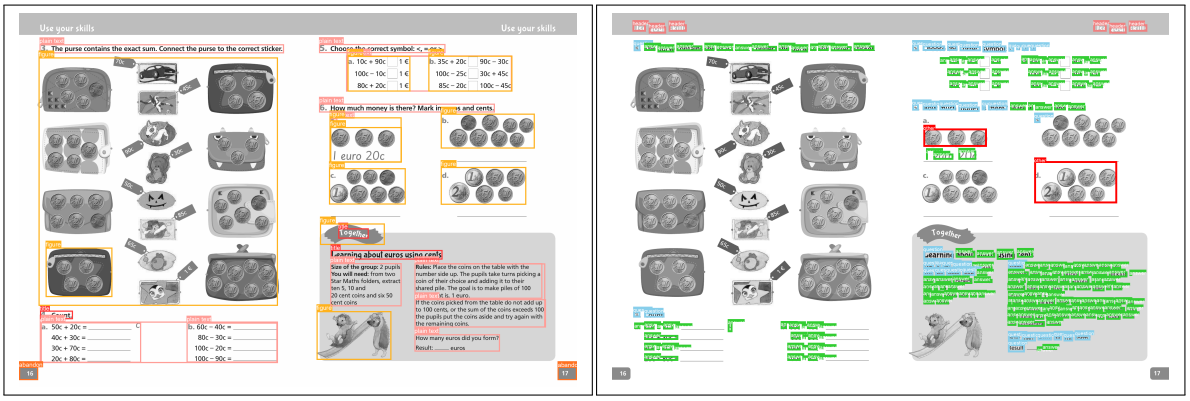
Figures 5a-5b illustrate how the layout analysis models DocLayout-YOLO (Zhao et al., 2024) and LayoutLM v2 (Xu et al., 2020) work out of the box. The model outputs show a considerable difference in granularity: LayoutLM segments individual words and ignores most visual elements while the granularity of DocLayout-YOLO seems to be a more suitable basis for layout unit segmentation. Compared to the human annotations (Figures 3a-3b), automatic annotations show inconsistencies in granularity (e.g. leaving out parts of exercises) and fail to recognize and segment some layout units, such as the colored backgrounds of the page header and bottom-right section of the right page. The models only supporting rectangular bounding boxes poses a challenge for precision; using polygons in addition to rectangular boxes allows for a more accurate description of how the layout space is used.

6. Limitations

While Prodigy offers a user-friendly foundation for simple annotation tasks, the design and deployment of complex annotation pipelines demand significant technical expertise. Composing sophisticated workflows necessitates advanced programming skills, including the ability to develop custom components and interpret source code. Furthermore, while Prodigy can be straightforwardly deployed on a single server for small-scale projects, the scalable and efficient deployment described in this paper requires specialized knowledge in distributed systems and containerization. The implementation of highly customized functionality, as in our case with dynamic task routing, may also push the boundaries of Prodigy's out-of-the-box capabilities. This entails creative problem-solving and extensive customization, further raising the barrier to entry for groups lacking expertise in software development and engineering.

Using proprietary software as the core of the

[3] https://version.helsinki.fi/multimodality-research-group/publications/lrec-2026



(a) DocLayout-YOLO (Zhao et al., 2024) (b) LayoutLM v2 (Xu et al., 2020)

Figure 5: Examples of predicted segmentations for a textbook double-page

annotation pipeline has both benefits and drawbacks. Expanding on existing annotation software accelerated the development of the pipeline, as developing a fully custom annotation system from scratch would have required substantial time and resources, diverting focus from research objectives. Additionally, leveraging Prodigy's actively maintained software minimizes the time needed to maintain the infrastructure in the future. We recognize the limitations of proprietary software, including its closed-source nature and licensing costs. However, Prodigy's provision of free research licenses mitigates financial barriers[4]. While we remain open to adopting open-source alternatives should they emerge, our current approach balances efficiency, reliability, and accessibility within the constraints of academic research.

The current solution of using majority vote as the basis of quality control is not optimal due to the simplicity of the method. We are planning to implement more sophisticated algorithms for aggregating task results to further improve the reliability of the annotations. One such option is Dawid-Skene (Dawid and Skene, 1979), an aggregation algorithm that takes each annotator's reliability into consideration when determining the true label. Dawid-Skene is widely used in crowdsourcing, where annotator reliability is pronounced as compared to working with trusted annotators. Other options include GLAD (Whitehill et al., 2009) and MACE (Hovy et al., 2013), which extend Dawid-Skene by incorporating features such as task difficulty and annotator bias into the algorithm. Implementing new aggregation methods into the pipeline presented in this paper is straightforward, once the required information about annotator behavior is available. The implementation of such functionality will be addressed in future work.

[4] https://prodi.gy/industries/research-education

7. Ethical Considerations

Annotation work often includes significant amounts of "*invisible labor*" (Crain et al., 2016; Daniels, 1987) outside of the formal annotation process. This type of extra work (e.g. looking for available annotation tasks) is typically unrecognized and uncompensated in crowdsourcing (Toxtli et al., 2021). Invisible labor affects the workers' perceptions of fairness and reduces the annotation efficiency. Reducing the time annotators spend looking for available tasks helps the annotators focus on the annotation itself, improving quality and annotator satisfaction (Toxtli et al., 2021). We provide our annotators with a website that displays readily available tasks (Section 3.3). Moreover, we aim to mitigate common issues identified in microtask crowdsourcing (Fort et al., 2011) by hiring trusted annotators to support a mutual relationship with the workers. Prior research has identified the lack of such relationships as a key contributor to worker dissatisfaction and the erosion of perceived fairness in microtask crowdsourcing, ultimately leading to abandoning the annotation task altogether (Fieseler et al., 2019). To ensure fair and predictable compensation, all annotators employed for this project are paid 13 €/h for their work, including the time spent checking for available tasks.

Acknowledgements

This research is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101122047). We also thank CSC – IT Center for Science Ltd. for computational resources and everyone who participated in annotating the data.

8. Bibliographical References

Jens Allwood. 2008. Multimodal corpora. In Anke Lüdeling and Merja Kytö, editors, *Corpus Linguistics: An International Handbook*, pages 207–225. Mouton de Gruyter, Berlin.

Jens Allwood, Loredana Cerrato, Kristiina Jokinen, Constanza Navarretta, and Patrizia Paggio. 2007. The MUMIN coding scheme for the annotation of feedback, turn management and sequencing phenomena. *Language Resources and Evaluation*, 41:273–287.

John A. Bateman. 2008. *Multimodality and Genre: A Foundation for the Systematic Analysis of Multimodal Documents*. Palgrave Macmillan, London.

John A. Bateman. 2012. Multimodal corpus-based approaches. In Carol A. Chapelle, editor, *The Encyclopedia of Applied Linguistics*, pages 3983–3991. Wiley-Blackwell, Oxford.

John A. Bateman. 2014. Using multimodal corpora for empirical research. In Carey Jewitt, editor, *The Routledge Handbook of Multimodal Analysis*, 2 edition, pages 238–252. Routledge, London and New York.

John A. Bateman, Judy L. Delin, and Renate Henschel. 2004. Multimodality and empiricism: preparing for a corpus-based approach to the study of multimodal meaning-making. In Eija Ventola, Cassily Charles, and Martin Kaltenbacher, editors, *Perspectives on Multimodality*, pages 65–89. Benjamins, Amsterdam.

John A. Bateman, Thomas Kamps, Klaus Reichenberger, and Jörg Kleinz. 2001. Towards constructive text, diagram, and layout generation for information presentation. *Computational Linguistics*, 27(3):409–449.

John A. Bateman, Janina Wildfeuer, and Tuomo Hiippala. 2017. *Multimodality: Foundations, Research and Analysis – A Problem-Oriented Introduction*. De Gruyter Mouton, Berlin.

Frederico Belcavello, Marcelo Viridiano, Ely Matos, and Tiago Timponi Torrent. 2022. Charon: A FrameNet annotation tool for multimodal corpora. In *Proceedings of the 16th Linguistic Annotation Workshop (LAW-XVI)*, pages 91–96, Marseille, France. European Language Resources Association.

Jeff Bezemer and Gunther Kress. 2008. Writing in multimodal texts: A social semiotic account of designs for learning. *Written Communication*, 25(2):166–195.

Jeff Bezemer and Gunther Kress. 2010. Changing text: A social semiotic analysis of textbooks. *Designs for Learning*, 3:10–29.

Shahzad Sarwar Bhatti, Xiaofeng Gao, and Guihai Chen. 2020. General framework, opportunities and challenges for crowdsourcing techniques: A comprehensive survey. *Journal of Systems and Software*, 167:110611.

Ekaterina Borisova, Raia Abu Ahmad, Leyla Garcia-Castro, Ricardo Usbeck, and Georg Rehm. 2024. Surveying the FAIRness of annotation tools: Difficult to find, difficult to reuse. In *Proceedings of the 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 29–45, St. Julians, Malta. Association for Computational Linguistics.

Bruno Cardoso and Neil Cohn. 2022. The Multimodal Annotation Software Tool (MAST). In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC 2022)*, pages 6822–6828, Marseille, France. European Language Resources Association (ELRA).

Jean Carletta. 2007. Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation*, 41(2):181–190.

Steve Cassidy and Thomas Schmidt. 2017. Tools for multimodal annotation. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 209–227. Springer, Dordrecht.

Tim Causer, Kris Grint, Anna-Maria Sichani, and Melissa Terras. 2018. 'Making such bargain': Transcribe Bentham and the quality and cost-effectiveness of crowdsourced transcription. *Digital Scholarship in the Humanities*, 33(3):467–487.

Federica Cavicchio and Massimo Poesio. 2012. The Rovereto Emotion and Cooperation Corpus: a new resource to investigate cooperation and emotions. *Language Resources and Evaluation*, 46(1):117–130.

Marion G. Crain, Winifred R. Poster, and Miriam A. Cherry, editors. 2016. *Invisible Labor: Hidden Work in the Contemporary World*. University of California Press, Oakland.

Arlene Kaplan Daniels. 1987. Invisible work. *Social Problems*, 34:403–415.

Kristina Danielsson and Staffan Selander. 2016. Reading multimodal texts for learning – a model for cultivating multimodal literacy. *Designs for Learning*, 8(1):25–36.

Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, pages 20–28.

Kostas Dimopoulos, Vasilis Koulaidis, and Spyridoula Sklaveniti. 2003. Towards an analysis of visual images in school science textbooks and press articles about science and technology. *Research in Science Education*, 33:189–216.

Christian Fieseler, Eliane Bucher, and Christian Pieter Hoffmann. 2019. Unfairness by design? the perceived fairness of digital labor on crowdworking platforms. *Journal of Business Ethics*, 156:987–1005.

Karën Fort, Gilles Adda, and K. Bretonnel Cohen. 2011. Amazon Mechanical Turk: Gold mine or coal mine? *Computational Linguistics*, 37(2):413–420.

Inmaculada Fortanet-Gómez, Noelia Ruiz-Madrid, Edgar Bernad-Mechó, and Julia Valeiras-Jurado. 2025. GRAPE-MARS: una nueva herramienta para el análisis multimodal en la investigación sobre segundas lenguas. *TEISEL: Tecnologías Para La Investigación En Segundas Lenguas*, 4.

Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. 2014. A taxonomy of microtasks on the web. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, pages 218–223, New York. ACM.

Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. 2017. Clarity is a worthwhile quality: On the role of task clarity in microtask crowdsourcing. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, pages 5–14, New York, NY, USA. Association for Computing Machinery.

Zeyu He, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Shaurya Rohatgi, and Ting-Hao Kenneth Huang. 2024. If in a crowdsourced data annotation pipeline, a GPT-4. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

Jan Hensellek. 2025. Cluttered screens: an eye-tracking study of visual attention allocation among viewers of TV news. *Visual Communication*, 24(3):570–594.

Tuomo Hiippala. 2017. An overview of research within the *Genre and Multimodality* framework. *Discourse, Context & Media*, 20:276–284.

Tuomo Hiippala, Malihe Alikhani, Jonas Haverinen, Timo Kalliokoski, Evanfiya Logacheva, Serafina Orekhova, Aino Tuomainen, Matthew Stone, and John A. Bateman. 2021. AI2D-RST: A multimodal corpus of 1000 primary school science diagrams. *Language Resources and Evaluation*, 55(3):661–688.

Tuomo Hiippala, Helmiina Hotti, and Rosa Suviranta. 2022. Developing a tool for fair and reproducible use of paid crowdsourcing in the digital humanities. In *Proceedings of the 6th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 7–12, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.

Lihe Huang. 2021. Toward multimodal corpus pragmatics: Rationale, case, and agenda. *Digital Scholarship in the Humanities*, 36(1):101–114.

Aniruddha Kembhavi, Michael Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Proceedings of the 14th European Conference on Computer Vision (ECCV 2016)*, pages 235–251, Cham. Springer.

Michael Kipp. 2001. ANVIL: A generic annotation tool for multimodal dialogue. In *Proceedings of EUROSPEECH-2001*, pages 1367–1370.

Michael Kipp, Jean-Claude Martin, Patrizia Paggio, and Dirk Heylen, editors. 2009. *Multimodal Corpora: From Models of Natural Interaction to Systems and Applications*. Springer, Berlin.

Dawn Knight and Svenja Adolphs. 2020. Multimodal corpora. In Magali Paquot and Stefan Th. Gries, editors, *A Practical Handbook of Corpus Linguistics*, pages 353–371. Springer, Cham.

Kenneth C. C. Kong. 2013. A corpus-based study in comparing the multimodality of Chinese- and English-language newspapers. *Visual Communication*, 12(2):173–196.

Adriana Kovashka, Olga Russakovsky, Li Fei-Fei, and Kristen Grauman. 2016. Crowdsourcing in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 10(3):177–243.

Kenneth Lai, Richard Brutti, Lucia Donatelli, and James Pustejovsky. 2024. Encoding gesture in multimodal dialogue: Creating a corpus of

multimodal AMR. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5806–5818, Torino, Italia. ELRA and ICCL.

Stephen C. Levinson and Judith Holler. 2014. The origin of human multi-modal communication. *Philosophical Transactions of the Royal Society B*, 369:20130302.

Stephanie M. Lukin, Claire Bonial, Matthew Marge, Taylor A. Hudson, Cory J. Hayes, Kimberly Pollard, Anthony Baker, Ashley N. Foots, Ron Artstein, Felix Gervits, Mitchell Abrams, Cassidy Henry, Lucia Donatelli, Anton Leuski, Susan G. Hill, David Traum, and Clare Voss. 2024. SCOUT: A situated and multi-modal human-robot dialogue corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14445–14458, Torino, Italia. ELRA and ICCL.

Catherine C. Marshall, Partha S.R. Goguladinne, Mudit Maheshwari, Apoorva Sathe, and Frank M. Shipman. 2023. Who broke Amazon Mechanical Turk? an analysis of crowdsourcing data quality over time. In *Proceedings of the 15th ACM Web Science Conference 2023*, WebSci '23, page 335–345, New York, NY, USA. Association for Computing Machinery.

Timothee Mickus, Elaine Zosa, Raul Vazquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1979–1993, Mexico City, Mexico. Association for Computational Linguistics.

Ken Mizusawa, Keishi Tajima, Masaki Matsubara, Toshiyuki Amagasa, and Atsuyuki Morishima. 2018. Efficient pipeline processing of crowdsourcing workflows. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1559–1562, New York, NY, USA. Association for Computing Machinery.

Zahra Nouri, Henning Wachsmuth, and Gregor Engels. 2020. Mining crowdsourcing problems from discussion forums of workers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6264–6276, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Michael O'Donnell. 2008. The UAM CorpusTool: Software for corpus annotation and exploration. In Carmen M. Bretones Callejas, editor, *Applied Linguistics Now: Understanding Language and Mind*, pages 1433–1447. University of Almeria, Almeria.

Kay L. O'Halloran, Sabine Tan, Bradley A. Smith, and Alexey Podlasov. 2011. Multimodal analysis within an interactive software environment: critical discourse perspectives. *Critical Discourse Studies*, 8(2):109–125.

Pamela Perniss. 2018. Why we should study multimodal language. *Frontiers in Psychology*, 9:1109.

Massimo Poesio, Jon Chamberlain, and Udo Kruschwitz. 2017. Crowdsourcing. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 277–295. Springer, Dordrecht.

Christoph Rühlemann and Alexander Ptak. 2023. Reaching beneath the tip of the iceberg: A guide to the Freiburg Multimodal Interaction Corpus. *Open Linguistics*, 9(1):20220245.

Thomas Smits and Melvin Wevers. 2023. A multimodal turn in digital humanities: Using contrastive machine learning models to explore, enrich, and analyze digital visual historical collections. *Digital Scholarship in the Humanities*, 38(3):1267–1280.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii. Association for Computational Linguistics.

Hartmut Stöckl. 2020. Linguistic multimodality – multimodal linguistics: a state-of-the-art sketch. In Janina Wildfeuer, Jana Pflaeging, John A. Bateman, Ognyan Seizov, and Chiao-I Tseng, editors, *Multimodality: Disciplinary Thoughts and the Challenge of Diversity*, pages 41–68. De Gruyter, Berlin.

Rosa Suviranta and Tuomo Hiippala. 2025. Can digital humanities use microwork crowdsourcing in a fair manner? the effect of pedagogical training and multimodal instructions on annotation quality. *Digital Scholarship in the Humanities*.

Leandra Thiele, Florian Schmidt-Borcherding, and John A. Bateman. 2024. All eyes on the signal? Mapping cohesive discourse structures with eye-tracking data of explanation videos. *Frontiers in Communication*, 9.

Martin Thomas. 2014. Evidence and circularity in multimodal discourse analysis. *Visual Communication*, 13(2):163–189.

Carlos Toxtli, Siddharth Suri, and Saiph Savage. 2021. Quantifying the invisible labor in crowd work. *Proceedings of the ACM on Human-Computer Interaction*, 5(319).

Veniamin Veselovsky, Manoel Horta Ribeiro, and Robert West. 2023. Artificial artificial artificial intelligence: Crowd workers widely use large language models for text production tasks.

Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Joshua Bergsma, and Javier Movellan. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems*, pages 2032–2040.

Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. 2006. ELAN: a professional framework for multimodality research. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1556–1559, Genoa, Italy.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, New York, NY, USA. Association for Computing Machinery.

Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. 2024. DocLayout-YOLO: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception.

Xiang Zhou, Yixin Nie, and Mohit Bansal. 2022. Distributed NLI: Learning to predict human opinion distributions for language reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 972–987, Dublin, Ireland. Association for Computational Linguistics.