Semi-automated annotation of page-based documents within the *Genre and Multimodality* framework

Tuomo Hiippala
Centre for Applied Language Studies
University of Jyväskylä
P.O. Box 35, 40014 Finland
tuomo.hiippala@iki.fi

Abstract

This paper describes ongoing work on a tool developed for annotating document images for their multimodal features and compiling this information into a corpus. The tool leverages open source computer vision and natural language processing libraries to describe the content and structure of multimodal documents and to generate multiple layers of XML annotation. The paper introduces the annotation schema, describes the document processing pipeline and concludes with a brief description of future work.

1 Introduction

Multimodality – or how multiple modes of communication interact and co-operate – has become a concern within many fields that fall under the umbrella of digital humanities (Svensson, 2010; O'Halloran et al., 2014). Whereas gestures, gaze and postures accompany spoken language in face-to-face conversation, written language works together with photographs, diagrams, typography and other communicative resources in documents. Given the inherent complexity of multimodal phenomena, combined with the variation arising from contextual factors, corpus-based approaches have been suggested as necessary for bringing multimodality under increased analytical control (Allwood, 2008; Bateman, 2014b).

This paper contributes to the empirical study of multimodality in page-based documents by presenting a prototype tool for creating multimodal corpora from document images that were not born digital. The tool generates stand-off XML annotation following the *Genre and Multimodality* (GeM) model, which provides an annotation schema with multiple layers of description that attend to the content, layout, appearance and discourse relations in page-based documents (Bateman, 2008).

The GeM annotation schema, which is intended to "function as a tool for isolating significant patterns against the mass of detail that multimodal documents naturally present" (Bateman, 2014a, 33), has proven useful for comparing the multimodality of documents across cultures (Thomas, 2009; Kong, 2013) and describing their change over time (Hiippala, 2015b). Yet the GeM model has not been adopted widely, because applying the multi-layered annotation schema requires ample time and resources. This requirement arises from the aforementioned mass of detail that occurs in multimodal documents.

The tool presented in this paper attacks the bottleneck issues of time and resources by leveraging several open source computer vision and optical character recognition libraries for the semi-automatic annotation of multimodal documents. To support this task, the paper proposes a variant of the GeM annotation schema named *auto-GeM*. This variant of the annotation schema is geared towards generating machine-readable annotation, which may be studied using tools developed for the purpose (Hiippala, 2015a), while also providing ground truths for specific document genres, whose availability is considered a prerequisite for automating other parts of the annotation process.

The paper begins with a brief introduction to the GeM model and its annotation schema, relating the work on the prototype tool to previous attempts at automating parts of the annotation process. The document processing pipeline and the proposed *auto-GeM* annotation schema are then described in greater detail. Finally, the conclusion outlines current challenges and sketches future work on the tool.

Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH), pages 84–89,
Berlin, Germany, August 11, 2016. ©2016 Association for Computational Linguistics

2 The Genre and Multimodality model

The GeM model provides a multi-layered XML schema for stand-off annotation of multimodal documents (Henschel, 2003; Bateman, 2008).

The model has four layers of annotation: any document described using the GeM model is first segmented into *base* units. These units constitute the base layer. Recognized base units include, among others, sentences, headers, photographs, captions and illustrations (Bateman, 2008, 111). The base units are then picked up for description in the *layout* layer, which features three components that describe their grouping and logical organization (layout structure), determine their typographic and graphic features (realization information), and establish their position in the document layout (area model).

The *rhetorical* layer, in turn, describes the discourse relations holding between the content, extending Rhetorical Structure Theory to cover both verbal and visual base units (Mann and Thompson, 1988; Taboada and Mann, 2006). Finally, the *navigation* layer describes how documents support their use with pointers such as "see page 5" and their corresponding entries, such as page and section numbers.

Each document is thus described from four different perspectives, and the annotation for each layer is cross-referenced using unique identifiers. These identifiers help to track how content elements relate to each other across the layout, rhetorical and navigation layers. Unlike other frameworks developed for describing documents, such as the Text Encoding Initiative (TEI), which is slowly beginning to pay attention to layout, typography and materiality, but continues to be primarily concerned with the representation of documents built around linear written language (Muñoz and Viglianti, 2015), the GeM model is inherently geared towards describing all kinds of multimodal documents, whether they organize their content linearly or make extensive use of the two-dimensional layout space.

Moreover, the GeM model was designed for corpus-driven research from the outset, and several tools have been developed to support the analysis of corpora annotated using its schema. Thomas (2007) describes a concordancer for querying GeM annotation, while Hiippala (2015a) uses Python to parse GeM corpora, transforming the annotation into GraphViz DOT graphs (Gansner and North, 2000) to visualize descriptions of document structure stored in the layout, rhetorical and navigation layers.

Certain attempts have also been made to address the bottleneck issues of time and resources required for producing GeM-annotated corpora. Thomas (2009) explores the use of commercial optical character recognition (OCR) software for automatically producing GeM annotation by using XSLT and Perl to transform and enrich the OCR output. Using XML output from *ABBYY FineReader 8.0 SDK* for generating annotation for the base and layout layers, Thomas observes that OCR output proves useful for the time-consuming task of describing typographic features, but nevertheless requires extensive manual post-processing.

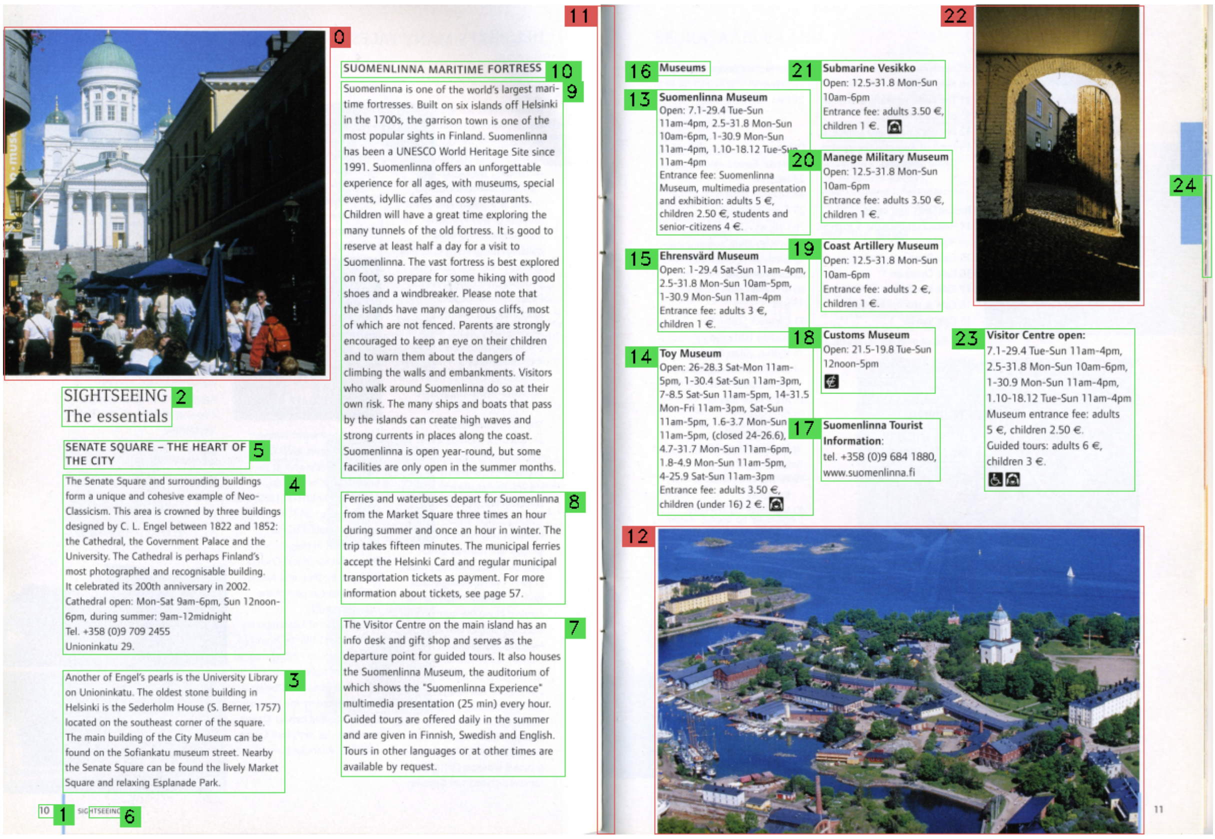
Thomas (2009, 245) concludes that producing GeM annotation for the layout areas missed by the OCR engine constitutes the most time-consuming post-processing task. Thomas et al. (2010) attempt to reduce the time spent on post-processing by using XSLT to transform the OCR output into the OpenDocument format, in which the output could be manually tweaked and improved. Despite integrating well into the document processing pipeline, the OpenDocument format loses most of the information pertaining to the document layout, which multimodal documents frequently exploit to provide cues about their use and organization (Waller, 2012).

Building on the previous work, this paper proposes several improvements to annotating multimodal documents semi-automatically within the framework proposed by the GeM model. Firstly, preferring open source libraries over commercial software enables a top-down approach, that is, attending to the key features of the layout first. Secondly, controlling the design of the entire document processing pipeline removes the need for interim formats, generating the annotation only after major corrections have been applied, propagating these modifications across all annotation layers. These improvements have been implemented in the prototype annotator, which is introduced in the following section.

3 The prototype annotator

3.1 System design

The prototype annotator is provided as an interactive Jupyter/IPython notebook to help novice users to deploy and use the tool (Pérez and Granger,



SUOMENLINNA MARITIME FORTRESS

Suomenlinna is one of the world's largest maritime fortresses. Built on six islands off Helsinki in the 1700s, the garrison town is one of the most popular sights in Finland. Suomenlinna has been a UNESCO World Heritage Site since 1991. Suomenlinna offers an unforgettable experience for all ages, with museums, special events, idyllic cafes and cosy restaurants. Children will have a great time exploring the many tunnels of the old fortress. It is good to reserve at least half a day for a visit to Suomenlinna. The vast fortress is best explored on foot, so prepare for some hiking with good shoes and a windbreaker. Please note that the islands have many dangerous cliffs, most of which are not fenced. Parents are strongly encouraged to keep an eye on their children and to warn them about the dangers of climbing the walls and embankments. Visitors who walk around Suomenlinna do so at their own risk. The many ships and boats that pass by the islands can create high waves and strong currents in places along the coast. Suomenlinna is open year-round, but some facilities are only open in the summer months.

SIGHTSEEING
The essentials

SENATE SQUARE – THE HEART OF THE CITY

The Senate Square and surrounding buildings form a unique and cohesive example of Neo-Classicism. This area is crowned by three buildings designed by C. L. Engel between 1822 and 1852: the Cathedral, the Government Palace and the University. The Cathedral is perhaps Finland's most photographed and recognisable building. It celebrated its 200th anniversary in 2002.
Cathedral open: Mon-Sat 9am-6pm, Sun 12noon-6pm, during summer: 9am-12midnight
Tel. +358 (0)9 709 2455
Unioninkatu 29.

Ferries and waterbuses depart for Suomenlinna from the Market Square three times an hour during summer and once an hour in winter. The trip takes fifteen minutes. The municipal ferries accept the Helsinki Card and regular municipal transportation tickets as payment. For more information about tickets, see page 57.

Another of Engel's pearls is the University Library on Unioninkatu. The oldest stone building in Helsinki is the Sederholm House (S. Berner, 1757) located on the southeast corner of the square. The main building of the City Museum can be found on the Sofiankatu museum street. Nearby the Senate Square can be found the lively Market Square and relaxing Esplanade Park.

The Visitor Centre on the main island has an info desk and gift shop and serves as the departure point for guided tours. It also houses the Suomenlinna Museum, the auditorium of which shows the "Suomenlinna Experience" multimedia presentation (25 min) every hour. Guided tours are offered daily in the summer and are given in Finnish, Swedish and English. Tours in other languages or at other times are available by request.

10 SIGHTSEEING

Museums

Suomenlinna Museum
Open: 7.1-29.4 Tue-Sun 11am-4pm, 2.5-31.8 Mon-Sun 10am-6pm, 1-30.9 Mon-Sun 11am-4pm, 1.10-18.12 Tue-Sun 11am-4pm
Entrance fee: Suomenlinna Museum, multimedia presentation and exhibition: adults 5 €, children 2.50 €, students and senior-citizens 4 €.

Ehrensvärd Museum
Open: 1-29.4 Sat-Sun 11am-4pm, 2.5-31.8 Mon-Sun 10am-5pm, 1-30.9 Mon-Sun 11am-4pm
Entrance fee: adults 3 €, children 1 €.

Toy Museum
Open: 26-28.3 Sat-Mon 11am-5pm, 1-30.4 Sat-Sun 11am-3pm, 7-8.5 Sat-Sun 11am-5pm, 14-31.5 Mon-Fri 11am-3pm, Sat-Sun 11am-5pm, 1.6-3.7 Mon-Sun 11am-5pm, (closed 24-26.6), 4.7-31.7 Mon-Sun 11am-6pm, 1.8-4.9 Mon-Sun 11am-5pm, 4-25.9 Sat-Sun 11am-3pm
Entrance fee: adults 3.50 €, children (under 16) 2 €.

Submarine Vesikko
Open: 12.5-31.8 Mon-Sun 10am-6pm
Entrance fee: adults 3.50 €, children 1 €.

Manege Military Museum
Open: 12.5-31.8 Mon-Sun 10am-6pm
Entrance fee: adults 3.50 €, children 1 €.

Coast Artillery Museum
Open: 12.5-31.8 Mon-Sun 10am-6pm
Entrance fee: adults 2 €, children 1 €.

Customs Museum
Open: 21.5-19.8 Tue-Sun 12noon-5pm

Suomenlinna Tourist Information:
tel. +358 (0)9 684 1880,
www.suomenlinna.fi

Visitor Centre open:
7.1-29.4 Tue-Sun 11am-4pm, 2.5-31.8 Mon-Sun 10am-6pm, 1-30.9 Mon-Sun 11am-4pm, 1.10-18.12 Tue-Sun 11am-4pm
Museum entrance fee: adults 5 €, children 2.50 €.
Guided tours: adults 6 €, children 3 €.

11

Figure 1: Detected and classified bounding boxes on a document image, labelled with identifiers.

2007). The notebook calls its functions from an external module, *generator*, which contains the main functions for processing and annotating document images. The annotator is available at www.github.com/thiippal/gem-tools.

To process the documents and to generate a description using the *auto-GeM* annotation schema, the annotator relies on several open source libraries: OpenCV[1] for computer vision, Tesseract[2] for OCR and NLTK[3] for natural language processing. The integration of these libraries into the document processing pipeline is described in the following section.

3.2 Document processing pipeline

The high-resolution document image, preferably of 300 DPI resolution, is first resized into a canonical width of 1200 pixels, while naturally maintaining the original aspect ratio of the document. A smaller size allows more efficient processing in OpenCV, which is first used to convert the document image from colour to grayscale. Next, bilateral filtering is applied to the grayscale image to reduce noise while preserving the edges of document elements. The filtered image is converted into a binary image, calculating the threshold using Otsu's method.

At this stage, the user is required to define a kernel size for performing a series of morphological operations on the thresholded image. The kernel height should correspond roughly to the x-height of the font used for body text in the document image, which is a prerequisite for detecting text paragraphs correctly. The following morphological operations involve applying a morphological gradient to establish the outlines of document elements, followed by an erosion to separate the elements clearly from each other. The user can set the number of iterations performed for the erosion in the notebook.

To help the user to fine-tune the annotator parameters, such as kernel size and erosion iterations, each step involving image processing is documented in an HTML-file using the *visual-logging*[4] module. This log is provided with the output.

[1] www.opencv.org

[2] www.github.com/tesseract-ocr

[3] www.nltk.org

[4] www.github.com/dchaplinsky/visual-logging

Next, connected-components labelling is performed to filter out remaining noise, before detecting contours using OpenCV. Contour detection is performed twice: during the first pass, each detected contour is filled with solid colour. The second pass retrieves the contours of filled elements: this procedure suppresses unwanted contours nested within photographs and other graphical elements. In initial testing, this procedure provided better results for grid-based layouts than applying a Non-Maximum Suppression algorithm. This, however, is likely to be largely dependent on the kind of document genre described.

The annotator then sorts the detected contours and feeds them to a Random Forest classifier, which classifies the regions of interest defined by the contours into two categories: text or graphics. The model, trained using Haralick textures and colour statistics extracted from 400 photographs and 400 text blocks, achieves on a high precision (1.00) and recall (0.99) on the testing data.

Finally, the classified contours are drawn on the resized image and displayed to the user in the notebook, as shown in Figure 1. The user is then asked to enter the identifiers of any false positives among the detected regions of interest. In Figure 1, these include regions labelled 11 and 24. The regions marked by the user are removed from the list of contours. At this stage, the user can also manually draw any regions of interest that evaded detection, such as the page number on the lower right-hand corner of Figure 1. For this purpose, the annotator uses the OpenCV HighGUI module.

When the user is finished, the contours are projected on the original high-resolution image to extract regions of interest, which are assumed to correspond roughly to layout units defined within the GeM model, that is, to text paragraphs, images, headers, captions and the like. Regions classified as text are then thresholded, resized to double their original size and fed to Tesseract for OCR. NLTK's Punkt tokenizer (Kiss and Strunk, 2006) is subsequently used for segmenting the layout units into sentences.

Three kinds of description are then created for each layout unit: basic layout segmentation, position in the document layout, and visual appearance. The base layer annotation is generated simultaneously using the segmentation produced by the Punkt tokenizer. Each region of interest is also extracted from the original high-resolution image and stored into the corpus, anticipating their use as training data for machine learning algorithms and for visualizing parts of the original document image in concordancer output (Thomas, 2007).

3.3 The auto-GeM annotation schema

The annotator generates *auto-GeM* annotation for the base and layout layers as described below. The base layer is first extracted from the layout layer, generating annotation for the minimal units of analysis defined within the GeM model. Within the base layer, each base unit is stored within a `unit` element and provided with a unique identifier in the `id` attribute to handle cross-references across annotation layers.

```
<unit id="u-1.4">Another of En-
gel's pearls is the University
Library on Unioninkatu.</unit>
```

The base units are picked up for description in the layout layer, in which they are combined into larger layout units, such as text paragraphs. In the layout layer annotation, the layout units are stored under the parent element `segmentation`. The following example shows the annotation for one child element, `layout-unit`, which represents a text paragraph consisting of multiple base units:

```
<layout-unit id="lay-1.4" src=
"lay-1.4.png" location="sa-1.4"
xref="u-1.4 u-1.5 u-1.6 u-1.7"/>
```

The `src` attribute refers to the image that contains the region of interest described by the layout unit, whereas the `location` attribute designates the position of the layout unit by referring to the `sub-area` element. The `xref` attribute refers to the base units that constitute the layout unit in question.

The `sub-area` element, positioned under the parent element `area-model`, contains a bounding box with relational coordinates, which can be projected on images of different sizes or used to render an abstract representation of the physical layout.

```
<sub-area id="sa-1.4" bbox=
"0.0490168139071 0.800747198007
0.231689940154 0.946865919469"/>
```

Finally, under the `realization` element, the `text` element characterizes the layout unit in terms of realization information, identifying the layout unit as consisting of written language.

```
<text xref="lay-1.4"/>
```

For graphic elements, the corresponding element `graphics` features additional attributes, `width` and `height`, which store relational values indicating the size of the graphic element in relation to the entire layout.

In comparison to the original GeM schema proposed in Bateman (2008), the coverage of the document structure in the *auto-GeM* schema is currently limited. Whereas the original GeM annotation schema can provide a rich description of the document layout and its appearance, but requires investing a considerable amount of time and resources in the annotation process, the tool described in this paper can be used to generate the base layer and parts of the layout layer much more efficiently. Given this trade-off and the current state of development, the prototype tool is likely to be most effective for generating a baseline for manual annotation. Future work will seek to bridge the gap between the original GeM schema and its proposed *auto-GeM* variant.

4 Conclusions and future work

This paper described the ongoing development of an annotation tool for describing the multimodal content and structure of page-based documents that were not born digital. The tool is intended to speed up the process of creating multimodal corpora for empirical research and generating rich descriptions to be used as ground truths for machine learning tasks.

Future work on the tool will involve covering the entire scope of the original GeM model in the *auto-GeM* variant, while taking the automation process further. This includes:

- enriching the realization information with a description of typographic properties, such as font size and family, while also describing the types of graphic elements more accurately,
- determining and suggesting optimal kernel size and iteration parameters to the user,
- enhancing the classification of graphical document elements using emerging multimodal resources such as Elliott and Kleppe (2016),
- captioning photographs using the method proposed in Karpathy and Fei-Fei (2015),
- representing the logical organization of the content by constructing a hierarchical XY-tree from the detected bounding boxes,
- creating an interface for annotating the rhetorical structure, which will undoubtedly require the most manual input from the user,
- detecting and annotating pointers and entries in the document image to provide a representation of the navigation structure.

Additional user-configured parameters will also be included in future versions, in order to ensure that the tool can meet the demands of different document genres. To tackle the problem of variation, test corpora representing various different document genres are also being planned at the moment.

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.

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

John A. Bateman. 2014a. Developing a GeM (Genre and Multimodality) model. In Sigrid Norris and Carmen D. Maier, editors, *Interactions, Images and Texts: A Reader in Multimodality*, pages 25–36. De Gruyter Mouton, Berlin and New York.

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

Desmond Elliott and Martijn Kleppe. 2016. 1 million captioned Dutch newspaper images. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. May 23–28, Portorož, Slovenia.

Emden R. Gansner and Stephen C. North. 2000. An open graph visualization system and its applications to software engineering. *Software - Practice and Experience*, 30(11):1203–1233.

Renate Henschel, 2003. *GeM Annotation Manual*. University of Bremen, University of Stirling, second edition.

Tuomo Hiippala. 2015a. *gem-tools*: Tools for working with multimodal corpora annotated using the Genre and Multimodality model. DOI: 10.5281/zenodo.33775

Tuomo Hiippala. 2015b. *The Structure of Multimodal Documents: An Empirical Approach*. Routledge, New York and London.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*. June 7–12, Boston, MA.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

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.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Trevor Muñoz and Raffaele Viglianti. 2015. Texts and documents: New challenges for TEI interchange and lessons from the Shelley-Godwin archive. *Journal of the Text Encoding Initiative*, 8. DOI: 10.4000/jtei.1270

Kay L. O'Halloran, Alvin Chua, and Alexey Podlasov. 2014. The role of images in social media analytics: A multimodal digital humanities approach. In David Machin, editor, *Visual Communication*, pages 565–588. De Gruyter Mouton, Berlin.

Fernando Pérez and Brian E. Granger. 2007. IPython: A system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29.

Patrick Svensson. 2010. The landscape of digital humanities. *Digital Humanities Quarterly*, 4(1).

Maite Taboada and William C. Mann. 2006. Rhetorical structure theory: looking back and moving ahead. *Discourse Studies*, 8(3):423–459.

Martin Thomas, Judy Delin, and Robert H. W. Waller. 2010. A framework for corpus-based analysis of the graphic signalling of discourse structure. In *Proceedings of Multidisciplinary Approaches to Discourse (MAD 2010)*, Moissac, France, March 17-20.

Martin Thomas. 2007. Querying multimodal annotation: A concordancer for GeM. In *Proceedings of the Linguistic Annotation Workshop*, pages 57–60, Prague, Czech Republic, June 28–29. Association for Computational Linguistics.

Martin Thomas. 2009. *Localizing pack messages: A framework for corpus-based cross-cultural multimodal analysis*. Ph.D. thesis, University of Leeds.

Robert H. W. Waller. 2012. Graphic literacies for a digital age: The survival of layout. *The Information Society*, 28(4):236–252.