Text-to-Video: Story Illustration from Online Photo Collections

Katharina Schwarz[1], Pavel Rojtberg[2], Joachim Caspar[2],
Iryna Gurevych[2], Michael Goesele[2], and Hendrik P. A. Lensch[1]

[1]Ulm University [2]TU Darmstadt

Abstract. We present a first system to semi-automatically create a visual representation for a given, short text. We first parse the input text, decompose it into suitable units, and construct meaningful search terms. Using these search terms we retrieve a set of candidate images from online photo collections. We then select the final images in a user-assisted process and automatically create a storyboard or photomatic animation. We demonstrate promising initial results on several types of texts.

Three blind mice. Three blind mice. See how they run. See how they run. They all ran after the farmer's wife, Who cut off their tails with a carving knife, Did you ever see such a sight in your life, As three blind mice?

1 Introduction

Telling a story by natural language is a process everybody gets trained for all his life. Telling a story by images however is significantly harder, because the generation of images let alone video is a time consuming process and the outcome largely depends on skill. On the other hand, pictures have a similar expressive range as natural language, with regard to describing objects, actions, or evoking specific emotions. Our goal is to provide a framework that simplifies the process of telling a story with pictures but eliminates the need to create the images yourself. More precisely, we aim at generating visualizations driven by natural language, augmenting written text semi-automatically by a sequence of images obtained from online photo collections as shown in the example above.

The main task is to obtain a semantically close translation from natural language to image sequences. As currently neither the semantics of written text nor of images can be automatically extracted with sufficient success rates by freely available tools, we provide a user-in-the-loop solution: Rather than semantically analyzing the text we parse individual sentences to extract the functional description for the individual words. In order to determine semantically matching images we rely on the tagging of images in photo-community sites such as Flickr. From the extracted parse trees, we formulate optimized search queries to obtain

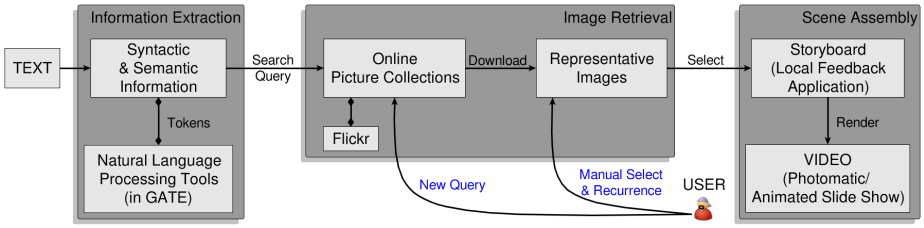


Fig. 1: System overview: information extraction, image retrieval and scene assembly.

as specialized images as possible for each part of a sentence. Typically, for each query a set of images is returned. The ultimate choice of which image to include in the final output is left to the user. We provide automatic means to select images with similar color distribution along the sentences. The images can now be presented in various formats, e.g., as storyboard or as slide animation. The quality of the image sequence largely depends on the available tagged imagery and the semantic complexity of the input text. Issues that are not correctly resolved by the parser will typically yield unsatisfying results but can often be corrected with little user intervention.

Figure 1 shows a high level overview of our system based on three parts. The information extraction part consists of automatically parsing and segmenting a given input text into parts of text, or *POTs* (Sec. 3). For each POT we construct an optimized search query (Sec. 4). The image retrieval part (Sec. 2) consists of automatically querying the online collection Flickr and retrieving a set of candidate images for each POT. Finally, the scene assembly part (Sec. 5) consists of automatically or semi-automatically picking the most representative images (*shots*) per POT. We present results on a nursery rhyme, fairy tale, screen play, short story, and a news article (Sec. 6), and close the paper with a short discussion (Sec. 7).

2 Image Retrieval

Online photo collections provide a tremendous amount of imagery (e.g., Flickr currently stores more than 4.5 billion images). They are widely used in the graphics community, e.g., as source for clip arts [8], to perform scene completion [6], to create photorealistic images from sketches [1] or computer-generated imagery [7], or to visualize and reconstruct scenes [12].

Images on Flickr are attributed by titles, tags, and texts by users in an informal way yielding a so called *Folksonomy* [5] (as opposed to the more formal ontology). Flickr allows for a full text search in each of these three categories. Each query will potentially return a set of images. One can measure the precision of the answer with respect to a query by manually counting the number of matching images:

$$precision = \frac{|\{relevant\ \ images\}| \cap |\{retrieved\ \ images\}|}{|\{retrieved\ \ images\}|}$$

Search Tokens	Queries	Full Text Search	Title Search	Tag Search
Combined Nouns	25	21% (25)	21% (20)	36% (24)
Nouns & Adjectives/Adverbs	25	17% (25)	25% (19)	36% (21)
Nouns & Verbs	25	8% (25)	13% (17)	16% (11)
Nouns & Averbs	20	5% (12)	10% (10)	12% (5)
Nouns & Adverb Stems	20	8% (9)	3% (7)	14% (5)
Nouns & Verbs	20	8% (20)	13% (13)	15% (6)
Nouns & Verb Stems	20	10% (19)	14% (12)	25% (11)

Table 1: Compound query results for fairy tales: precision in percent and number of total successful queries in parenthesis. A query was counted as successful, if at least one matching image was retrieved.

girl ∨ ball girl ∧ ball girl ∧ ball ∧ beautiful

Fig. 2: Successive specialization of a shot for "the beautiful girl at the ball".

Basic Tokens and Stemming In order to obtain an image which matches the semantics of a POT, we need to assemble a compound query. By far the most frequent category of words in Flickr tags are nouns (about 90%) while verbs, adjectives and adverbs are found less often. Most often, querying for a single noun ignores the information of the remainder of the POT. By combining the nouns and attributes of one POT by conjunction, more and more specific images can be retrieved, such that they finally match the desired semantics (Fig. 2). Thus, combining nouns or adding adjectives, adverbs, or verbs can help retrieving a more specialized collection of images even though the precision for the query per se might drop (Table 1). In particular, the precision for queries including verbs is low. But most often the retrieved images show the action represented by a sentence much better. Fig. 3 demonstrates this with the retrieved images for the queries *cocoon* (noun), *cocoon emerges* (noun & verb), and *cocoon emerge* (noun & stemmed verb). As shown in the second half of Table 1, the precision for adverb or verb queries can also be improved by querying for the stem rather than the inflected form.

The highest precision is typically obtained searching in tags, but verbs and adverbs are rarely found in tags while they are more frequent in full text or title search. We will use these insights in Sec. 4 to assemble an optimal query for each POT.

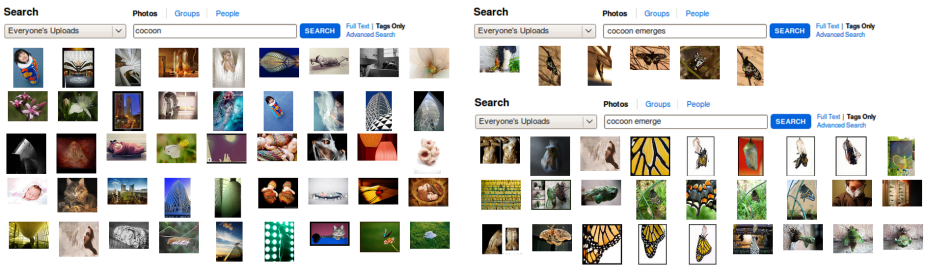


Fig. 3: Improving the correlation of shot and sentence action by combining nouns with verbs or verb stems.

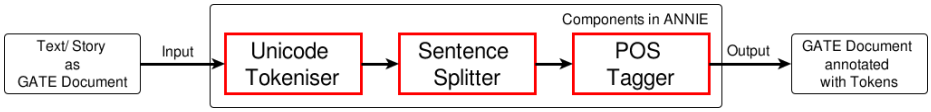


Fig. 4: GATE preprocessing with ANNIE components.

3 Information Extraction

Information extraction (IE) takes unseen documents as input and produces selectively structured output from the explicitly stated or implied data which is found in the text [2]. In order to form versatile queries for image retrieval, we extract syntactic and semantic information from the story using the General Architecture for Text Engineering (GATE) [3]. GATE is a modular infrastructure for developing and deploying NLP (Natural Language Processing) software components from various developers.

Preprocessing with ANNIE GATE is distributed with an IE system called ANNIE (A Nearly-New Information Extraction System). Figure 4 shows the components we use from ANNIE. Tokeniser and Sentence Splitter are required to annotate each word or symbol with a part-of-speech tag by the POS Tagger. The tokens determined by this pipeline form the basis for generating our queries.

Additional Tokens from Stemming Furthermore, we discovered the usage of verb stems as a significant improvement concerning the query answers. GATE provides the Snowball Stemmer as plugin, a tool based on the Porter stemmer for English [11]. It annotates each token with its stem. We apply this to verbs and adverbs only.

Text Segmentation Starting with sentences as the most general entities, they are split by punctuation marks or coordinating conjunctions in order to receive smaller segments, the POTs, which could be clauses, the entities of an enumeration or similar. Within these POT objects we query ANNIE tokens, especially nouns, verbs, adjectives and adverbs, as they contain the most significant information. For the same reason, we remove auxiliary verbs. The next step is to find an appropriate image (a shot) for each POT.

4 Forming the Query

Shots for the individual POTs are obtained by submitting proper queries to the online image collections. Using the list of tokens extracted for each POT (Sec. 3), we formulate an appropriate query that results in a high precision and sufficiently many images to choose from. A conjunction of all n_t token T_i in a POT will produce the most specialized query. However, if too many constraints are added it might happen that only an unsatisfactory number of images is reported. We therefore assemble the query iteratively with the goal to find a sufficiently large set of images which are as specific as possible.

Following the evidence of Sec. 2, we create a priority list Q of compound queries for a shot. The first query is the conjunction of all n_t tokens, followed by a disjunction of conjunctions formed by all possible subsets containing $n_t - 1$ tokens, etc., until we end with a disjunction of all n_t individual tokens. For $n_t = 3$, we would assemble the following list of queries:
$Q = \{(T_1 \wedge T_2 \wedge T_3), ((T_1 \wedge T_2) \vee (T_1 \wedge T_3) \vee (T_2 \wedge T_3)), (T_1 \vee T_2 \vee T_3)\}$.

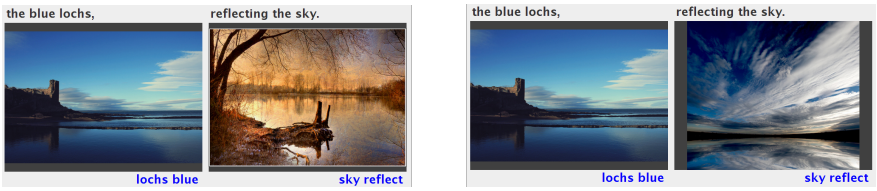
Due to their importance in the syntactic analysis and their frequency in the image tags, we treat nouns in a special manner. Queries combining multiple nouns yield the highest precision and therefore, we first treat the conjunction of all nouns as a single token in the algorithm outlined above, and in a second step create a list for each noun separately and append them.

Based on the priority list Q, the system issues a sequence of queries, accumulating the downloaded images. The process is stopped as soon as the number of downloaded images for one shot exceeds a user defined threshold, e.g., 30. For each entry of the list, we first perform a query on Flickr tags and then perform a title search. This way, we were able to download 30 images for almost all of our example shots and due to the structure of the priority queue ensured that the most specialized images are always at the top of the image set.

5 Scene Assembly

We now retrieved a set of candidate images per shot, sorted by relevance, from which we can automatically select the highest ranked image for each shot. Alternatively, this selection can be performed by the user. User selection will, for example, be necessary if the semantics, style, or composition of the highest ranked image does not match the user preference. To simplify the selection process, we provide the user with two sets of tools, one dealing efficiently with *recurring queries*, and another handling *color consistency* between neighboring shots.

Recurrence It is often desirable to use the same image for similar shots (see first and last image of the teaser figure). Our system therefore reuses by default the selection results for shots with the same query. If the user chooses a different image for one shot, all other shots in this category are updated accordingly. Beyond a literal match, the user can manually group multiple queries together which will then be represented by the same image.



original images color consistency enforced

Fig. 5: Color consistency enforced on the selection for one query. The queries are indicated in blue.

text type	prec. #1	prec. #10	# words	# queries	#user interactions	query time
nursery rhyme	40%	46%	53	8	5	00:02:33
fairy tale	80%	60%	538	73	45	00:33:18
screen play	80%	55%	230	41	28	00:20:59
news	60%	49%	147	19	12	00:12:02
short novel	40%	38%	967	150	113	01:27:08

Table 2: Results on different text types.

Color Consistency. Online images typically vary largely in style and color. Similarity across queries can be increased by sorting the image sets for neighboring shots by color similarity. Mehtre et al. [9] indicate that it is sufficient to perform a coarse comparison to exclude severe color miss-matches in image retrieval. We therefore compute for each image its mean RGB color vector. Given the current representative image I_A for shot A, we select for a neighboring query B the representative image I_B that minimizes the Euclidean distance of the mean color vectors. For the next query C, the comparison is carried out with respect to I_B, and so on. The user is free to indicate whether or not the color matching constraint should be applied and, if so, into which direction the color should be propagated. The improved results are shown in Fig. 5. Alternatively, color consistency can be achieved by processing the selected images (e.g., converting them to sepia or black-and-white representations, applying color style transfer techniques [10]).

6 Results

At this stage, each POT has been assigned a shot. The resulting image set can simply be represented as a storyboard (see teaser image and Fig. 6) or presented as in the accompanying video as an animated slide show, where text and image transitions are achieved by a constant motion (see the paper web page).

We have applied our system on various text types: the nursery rhyme “Three Blind Mice”, the fairy tale “Cinderella”, a part of the screenplay to “Braveheart”, a news article about the Wii controller, and the short novel “Animal Farm”. In general, we observed high context-sensitive precision in our tests, considering the actual meaning of the sentence rather than just the queried tokens. This can be seen in Table 2, which shows our results for a range of text types. For each query, 30 images were downloaded and the context-sensitive precision for the

Fig. 6: Automatically generated visual story (top row) and manually improved version (bottom row) of a news article about the Wii controller.

Fig. 7: Semantically mismatching images due to metaphorical character names in the "Animal Farm" novel.

first and the first 10 images was evaluated. Because of the very good sentence related precision of the first reported images, the required user interaction to construct a semantically close storyboard is moderate.

We think that the variation in the text types correlates with the typical spectrum of submitted Flickr images. The content of the selected nursery rhyme and short novel is slightly more abstract than the content of the other categories.

In Fig. 6, we compare completely automatically generated results against the manually optimized selection. While quite a number of shots received a decent representative automatically, a few clicks were necessary to obtain the final selection where semantic errors and deviations in style have been removed.

In general, we were surprised by how often the retrieved images for our generated queries match the intention of the original text. However, Fig. 7 clearly shows the limits of our approach, namely dealing with word sense ambiguities. The images retrieved for the two pig characters *Snowball* and *Napoleon* from the novel "Animal Farm" do not depict pigs, but the literal or most frequent meaning of the words. A solution for this problem could be the usage of sophisticated lexical semantic analysis, such as word sense disambiguation and named entity recognition.

So far, our parsing is limited to group only tokens that are adjacent in a sentence. In the future, we would like to use dependency graphs [4] to assemble better queries for the non-connected parts of the sentence.

We expect further improvements if anaphora could be automatically resolved or if spatial relations in the scene could be considered in the query.

7 Conclusion

Our system can be seen as one of the first steps towards creating a movie based on a textual input. After parsing the text, the system automatically generates queries and retrieves images from the community site Flickr. Most often, a set of representative images is found automatically. After a few user interactions, a reasonable storyboard is produced.

By enhancing the semantic analysis of the text, e.g., by using WordNet or by considering syntactic as well as semantic relations between objects, the quality of the (semi-)automatically generated storyboards might be further increased. Additionally, an automatic classification of the retrieved images into sets of similar images might facilitate the manual selection process. One natural extension to the presented system is to animate the retrieved images to better visualize the action in a story – as a next step towards creating full-fledged movies.

Acknowledgments. We would like to thank the Flickr users for the images used in our research. This work was funded in part by the DFG Emmy Noether fellowships Le 1341/1-1, GO 1752/3-1, the DFG Research Training Group 1223, and the Lichtenberg Program No. I/82806 of the Volkswagen Foundation.

References

1. T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2Photo: Internet image montage. *ACM Trans. Graph.*, 28(5), 2009.
2. Jim Cowie and Wendy Lehnert. Information extraction. *Commun. ACM*, 39(1):80–91, 1996.
3. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proc. 40th Anniv. Meeting of the Assoc. for Comp. Ling.*, 2002.
4. M.-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. In *LREC-06*, 2006.
5. Guy and E. Tonkin. Folksonomies, tidying up tags? *D-Lib Magazine*, 1 2006.
6. J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Trans. Graph.*, 2007.
7. M. K. Johnson, K. Dale, S. Avidan, H. Pfister, W. T. Freeman, and W. Matusik. CG2Real: Improving the realism of computer generated images using a large collection of photographs. Technical Report MIT-CSAIL-TR-2009-034, 2009.
8. J.-F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J. Winn, and A. Criminisi. Photo clip art. *ACM Trans. Graph.*, 2007.
9. B. M. Mehtre, M. S. Kankanhalli, A. D. Narasimhalu, and G. C. Man. Color matching for image retrieval. *Pattern Recogn. Lett.*, 16(3):325–331, 1995.
10. L. Neumann and A. Neumann. Color style transfer techniques using hue, lightness and saturation histogram matching. In *Comp. Aesthetics in Graphics, Visualization and Imaging*, pages 111–122, 2005.
11. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
12. N. Snavely, I. Simon, M. Goesele, R. Szeliski, and S. M. Seitz. Scene reconstruction and visualization from community photo collections. *Proc. of the IEEE, Special Issue on Internet Vision*, 2010.