A Text-to-Picture Synthesis System for Augmenting Communication*

Xiaojin Zhu, Andrew B. Goldberg, Mohamed Eldawy, Charles R. Dyer and Bradley Strock
Department of Computer Sciences
University of Wisconsin, Madison, WI 53706, USA
{jerryzhu, goldberg, eldawy, dyer, strock}@cs.wisc.edu

Abstract

We present a novel Text-to-Picture system that synthesizes a picture from general, unrestricted natural language text. The process is analogous to Text-to-Speech synthesis, but with pictorial output that conveys the gist of the text. Our system integrates multiple AI components, including natural language processing, computer vision, computer graphics, and machine learning. We present an integration framework that combines these components by first identifying informative and 'picturable' text units, then searching for the most likely image parts conditioned on the text, and finally optimizing the picture layout conditioned on both the text and image parts. The effectiveness of our system is assessed in two user studies using children's books and news articles. Experiments show that the synthesized pictures convey as much information about children's stories as the original artists' illustrations, and much more information about news articles than their original photos alone. These results suggest that Text-to-Picture synthesis has great potential in augmenting human-computer and human-human communication modalities, with applications in education and health care, among others.

Introduction

A picture is worth a thousand words. However, very few systems convert general text to pictorial representations that can be used in many circumstances to replace or augment the text. We present a novel *Text-to-Picture (TTP)* synthesis system which automatically generates pictures, that aims to convey the primary content of general natural language text. Figure 1 shows a picture automatically generated by our TTP system. Our system employs AI techniques ranging from natural language processing, computer vision, computer graphics, to machine learning. We integrate these components into a concatenative synthesizer, where the synergy of text unit selection, image parts generation, and layout optimization produces coherent final pictures. For example, we use 'picturability' to influence word selection, and use word importance to influence the layout of the picture. The details, as well as an appropriate evaluation metric, are presented. User study experiments show that participants' descriptions of TTP collages contain words that are a closer (or equivalent) match to the original text than their descriptions of original illustrations or photos that accompany the text.

Figure 1: A picture generated by our TTP system.

TTP has many applications when a text interface is not appropriate. One important application is literacy development. For children who are learning to read and for second language learners, seeing pictures together with text may enhance learning (Mayer 2001). Another application is as a reading aid for people with learning disabilities or brain damage. TTP can convert textual menus, signs, and safety and operating instructions into graphical representations. Importantly, TTP output can be created on demand by a user and does not depend on a vendor to produce it. Eventually, a person might carry a PDA equipped with TTP and optical character recognition so that the person could generate visual translations as needed during their daily activities. TTP naturally acts as a universal language when communication is needed simultaneously to many people who speak different languages, for example for airport public announcements (Mihalcea & Leong 2006). TTP can produce visual summaries for rapidly browsing long text documents.

The current work differs from previous text-to-scene type systems in its focus on *conveying the gist of general, unrestricted text*. Previous systems were often meant to be used by graphics designers as an alternative way to specify the layout of a scene. Such text-to-scene systems tend to emphasize spatial reasoning. Examples include NALIG (Adorni, Manzo, & Giunchiglia 1984), SPRINT (Yamada *et al.*

*We thank the anonymous reviewers for their constructive comments. Research supported in part by the Wisconsin Alumni Research Foundation.

1992), Put (Clay & Wilhelms 1996) and, notably, WordsEye (Coyne & Sproat 2001). WordsEye is able to produce highly realistic 3D scenes by utilizing thousands of predefined 3D polyhedral object models with detailed manual tags, and deep semantic representations of the text. Consequently, WordsEye works best with certain descriptive sentences, e.g., "The lawn mower is 5 feet tall. John pushes the lawn mower. The cat is 5 feet behind John. The cat is 10 feet tall." Other systems include (Brown & Chandrasekaran 1981; Lu & Zhang 2002). CarSim (Johansson *et al.* 2005) converts special-domain narratives on road accidents into an animated scene using icons. Blissymbols (Hehner 1980) and other graphic symbol systems create symbol-for-word strings rather than a coherent picture that conveys a global meaning.

The Text-to-Picture System

Let the input text be a word sequence $\mathcal{W}_{1:n}$ of length n. In our concatenative TTP synthesizer, we first use natural language processing techniques to select k keyphrases (important words or phrases found within $\mathcal{W}_{1:n}$) to "draw." Then for each selected keyphrase, we use computer vision techniques to find a corresponding image $\mathcal{I}_i$. (We use the word "picture" to denote the overall composed output, while "image" to denote the individual constituents.) Finally we use computer graphics techniques to spatially arrange all k images to create the output picture. To integrate these components together, we formulate the TTP problem as finding the most likely keyphrases $\mathcal{K}^*_{1:k}$, images $\mathcal{I}^*_{1:k}$, and placement $\mathcal{C}^*_{1:k}$ given the input text $\mathcal{W}_{1:n}$:

$$(\mathcal{K}^*_{1:k}, \mathcal{I}^*_{1:k}, \mathcal{C}^*_{1:k}) = \mathrm{argmax}_{\mathcal{K},\mathcal{I},\mathcal{C}}\ p(\mathcal{K}, \mathcal{I}, \mathcal{C}|\mathcal{W}_{1:n}). \quad (1)$$

In our implementation, the placement $\mathcal{C}_i$ of image i is specified by the center coordinates, but other factors such as scale, rotation and depth can be incorporated too. To make the optimization problem tractable, we factorize the probability into

$$p(\mathcal{K}, \mathcal{I}, \mathcal{C}|\mathcal{W}) = p(\mathcal{K}|\mathcal{W})p(\mathcal{I}|\mathcal{K}, \mathcal{W})p(\mathcal{C}|\mathcal{I}, \mathcal{K}, \mathcal{W}), \quad (2)$$

and approximate the joint maximizer of Eq. (1) by the maximizers of each factor in Eq. (2), as described below.

1. Selecting Keyphrases

Given a piece of text, e.g., a sentence or a whole book, the first question is, which keyphrases should be selected to form the picture? Formally, we solve the subproblem $\mathcal{K}^*_{1:k} = \mathrm{argmax}_{\mathcal{K}}\ p(\mathcal{K}|\mathcal{W})$.

Our approach is based on extractive *picturable* keyword summarization. That is, it builds on standard keyword-based text summarization (Turney 1999; Mihalcea & Tarau 2004), where keywords and keyphrases are extracted from the text based on lexicosyntactic cues. The central issue in keyword summarization is to estimate the importance of lexical units. We do so using an unsupervised learning approach based on the TextRank algorithm (Mihalcea & Tarau 2004). TextRank defines a graph over candidate words based on co-occurrence in the current text, and uses the stationary distribution of a teleporting random walk on the graph as the importance measure. Our novelty is that we include a special teleporting distribution over the words in the graph. Our teleporting distribution is based on "picturability," which measures the probability of finding a good image for a word. Our approach thus selects keyphrases that are important to the meaning of the text and are also easy to represent by an image.

The TextRank Graph Following Mihalcea and Tarau (2004), we define the TextRank graph over individual words. The ranking of these words will be used later to construct the final set of longer keyphrases. All nouns, proper nouns, and adjectives (except those in a stop list) are selected as candidate words using a part-of-speech tagger. We then build a co-occurrence graph with each word as a vertex. We represent this unweighted graph as a co-occurrence matrix, where entry ij is 1 if term i and term j co-occur within a window of size 5.

Teleporting Distribution based on Picturability We base each graph vertex's teleporting probability on whether we are likely to find an image for the corresponding word. We call this measure "picturability" and compute it using a logistic regression model. The picturability logistic regression model was trained on a manually-labeled set of 500 words, randomly selected from a large vocabulary. Five annotators independently labeled the words. A word is labeled as picturable ($y = 1$) if an annotator is able to draw or find a good image of the word. When shown the image, other people should be able to guess the word itself or a similar word. Words labeled as non-picturable ($y = 0$) lack a clearly recognizable associated image (e.g., "dignity").

We represent a word using 253 candidate features, derived from the log-ratios between 22 raw counts. We obtain the raw counts from various Web statistics, such as the number of hits from image and Web page search engines (e.g., Google, Yahoo!, Flickr) in response to a query of the word. We perform forward feature selection with $L2$-regularized logistic regression. The log-ratio between *Google Image Search hit count* and *Google Web Search hit count* dominated all other features in terms of cross-validation log likelihood. With the practical consideration that a light system should request as few raw Web counts as possible, we decided to create a model with only this one feature. Intuitively, 'number of images vs. Web pages' is a good picturability feature that measures image frequency with respect to word frequency. The resulting picturability logistic regression model is

$$p(y = 1|x) = \frac{1}{1 + \exp(-(2.78x + 15.40))} \quad (3)$$

where $x = \log\left((c_1 + 10^{-9})/(c_2 + 10^{-9})\right)$ is the log ratio between smoothed counts c_1 (Google Image hits) and c_2 (Google Web hits), and 10^{-9} is a smoothing constant to prevent zero counts. For example, the word 'banana' has 356,000 Google Image hits and 49,400,000 Web hits. We find that $p(y = 1|\text{'banana'}) = 0.84$, meaning 'banana' is probably a picturable word. On the other hand, the word 'Bayesian' has 17,400 Google Image hits and 10,400,000 Web hits, so $p(y = 1|\text{'Bayesian'}) = 0.09$, indicating it is not so picturable.

We use Eq. (3) to compute a picturability value for each candidate word in the TextRank graph. These values are normalized to form the teleporting distribution vector $\mathbf{r}$.

Determining the Final Keyphrases To obtain the ranking of words, we compute the stationary distribution of the teleporting random walk

$$\lambda P + (1 - \lambda)\mathbf{1}\mathbf{r}^\top,$$

where P is the graph-based transition matrix (i.e., row-normalized co-occurrence matrix) and $\mathbf{r}$ is the teleporting distribution defined above. This is the same computation used by PageRank. λ is an interpolation weight, which we set to 0.75, and $\mathbf{1}$ is an all-ones vector. The stationary distribution indicates the centrality or relative importance of each word in the graph, taking into account picturability. We select the 20 words with the highest stationary probabilities, and form keyphrases by merging adjacent instances of the selected words (as long as the resulting phrase has a picturability probability greater than 0.5). Next, we discard phrases lacking nouns, multiple copies of the same phrase, and phrases that are subsumed by other longer phrases. The end result is a list of keyphrases that appear important and are likely to be representable by an image. Finally, each extracted keyphrase $\mathcal{K}_i$ is assigned an importance score $s(\mathcal{K}_i)$, which is equal to the average stationary probability of the words comprising it.

2. Selecting Images

The goal of this stage is to find one image to represent each extracted keyphrase. Our algorithm handles each keyphrase independently: $\mathcal{I}_i^* = \mathrm{argmax}_{\mathcal{I}_i}\ p(\mathcal{I}_i|\mathcal{W}, \mathcal{K}_i^*)$, $i = 1 \dots k$. Our image selection module combines two sources to find such an image. First, we use a manually labeled clipart library. Second, if the keyphrase cannot be found in the library, we use an image search engine and computer vision techniques. Combining the two sources ensures accurate results for common keyphrases, which are likely to exist in the library, and good results for other arbitrary keyphrases. We focus on the second source below.

Image search engines are not perfect, which means many images returned do not visually represent the keyphrase well. In particular, the first image returned by an image search engine is often not a good image to depict the keyphrase. Our approach to selecting the best image from search results, which is similar to the method by Ben-Haim *et al*. (2006), consists of the following steps. First, the top 15 images for this keyphrase are retrieved using Google Image search. Next, each image is segmented into a set of disjoint regions using an image segmentation algorithm (Felzenszwalb & Huttenlocher 2004). Parameters for the algorithm were set so that, on average, each image is segmented into a small number of segments so that over-segmentation of the object of interest is less likely.

For each region extracted in each image, we next compute a feature vector to describe the appearance of that region. Color histograms have been shown to perform well for databases of arbitrary color photographs (Deselaers, Keysers, & Ney 2004). We compute a vector of color features

Figure 2: The image selection process on three retrieved images for the word "pyramids." Segmentation boundaries are overlaid on the images. The region closest to the centroid of the largest cluster is indicated by the arrow, and that image is selected as the best for the word.

to describe each region. Specifically, the color histogram in LUV color space of all pixels in a region is computed. The L component is then quantized into 5 bins, and the UV pairs of values are quantized into 25 bins, resulting in a feature vector of size 30.

The feature vectors in all images are now clustered in feature space. Assuming there are several regions that correspond to the keyphrase and their appearances are similar, we expect to find a compact cluster in feature space. We use the Mean Shift clustering algorithm (Comaniciu & Meer 2002). Assuming that regions corresponding to background parts of an image are not as similar to one another as the regions that correspond to the keyphrase, we treat the largest cluster as the one that is most likely to correspond to the keyphrase. Once the largest cluster is found, we find the region whose feature vector is closest to the centroid of this cluster. The image which contains this region is then selected as the best image for this keyphrase. Figure 2 shows an example of the result of this algorithm.

3. Picture Layout

The third and final stage takes the text, the keyphrases, and their associated images, and determines a 2D spatial layout of the images, $\mathcal{C}_{1:k}^* = \mathrm{argmax}_{\mathcal{C}}\ p(\mathcal{C}|\mathcal{W}, \mathcal{K}^*, \mathcal{I}^*)$, to create the output picture.

Our problem of composing a set of images is similar to the problem of creating picture collages, e.g., (Wang *et al.* 2006). However, our goal is to create a layout that helps to convey the meaning of the text by revealing the important objects and their relations. Since we are interested in handling unrestricted text, we do not assume the availability of semantic knowledge or object recognition components, relying instead on the structure of the text and general layout rules that make the picture intuitively "readable." To this end, we first scale all the images to make them roughly the same size. To determine the best locations for the images, we define a good layout to have the following three properties:

1. *Minimum overlap*: Overlap between images should be minimized,
2. *Centrality*: Important images should be near the center,

3. *Closeness*: Images corresponding to keyphrases that are close in the input text should be close in the picture.

Finding the best positions for all the images is formulated as an optimization problem to minimize the objective:

$$\lambda_1 \sum_{i=1}^{k} \sum_{j<i}^{k} \frac{o(\mathcal{I}_i, \mathcal{I}_j)}{A_{total}} + \lambda_2 \sum_{i=1}^{k} s(\mathcal{K}_i) d(\mathcal{I}_i) + \lambda_3 \sum_{i=1}^{k} \sum_{j<i}^{k} q(i,j)$$

where λs are weights, $o(\mathcal{I}_i, \mathcal{I}_j)$ is the area of overlap between pictures $\mathcal{I}_i$ and $\mathcal{I}_j$, A_{total} is the sum of the areas of all images, $s(\mathcal{K}_i)$ is the importance of keyphrase $\mathcal{K}_i$, $d(\mathcal{I}_i)$ is the distance of image $\mathcal{I}_i$ from the center of the picture, and $q(i,j)$ is an indicator function defined as

$$q(i,j) = \begin{cases} 1 & \text{if the closeness constraint is violated} \\ 0 & \text{otherwise.} \end{cases}$$

The closeness constraint is violated if two keyphrases, $\mathcal{K}_i$ and $\mathcal{K}_j$, are close in the text but their corresponding images, $\mathcal{I}_i$ and $\mathcal{I}_j$, are not touching in the picture. Two keyphrases are said to be close if they are less than 7 words apart and no other keyphrase separates them in the input text.

To solve this highly non-convex optimization problem, we use a Monte Carlo randomized algorithm to construct multiple candidate pictures and then pick the one that minimizes the objective function. At each step of the algorithm for constructing a candidate picture, one image is selected and its position in the picture is determined. When all images have been selected, the candidate picture is complete.

The most important image is always placed first at the center of the picture. To select the next image to add to the picture, we make a random decision between selecting an image based on importance or based on obeying closeness constraints. To select an image based on importance, a random image is selected from the remaining images, where the probability of selecting image $\mathcal{I}_i$ is $\frac{s(\mathcal{K}_i)}{\sum_j s(\mathcal{K}_j)}$ and the summation is over all remaining images $\mathcal{I}_j$. Recall that $s(\mathcal{K}_i)$ is image $\mathcal{I}_i$'s associated keyphrase importance. To choose an image based on closeness constraints, an image is selected, uniformly at random, from the remaining images that are close to one of the images already placed. A local gradient descent move is used to remove any overlap between images.

The process of creating a candidate picture is repeated a large number of times (currently 1000), and the best picture (with the lowest objective function) is selected as the final result. Branch-and-bound was also implemented so that a partial picture is immediately rejected if the objective function exceeds that of the best picture found so far. Figure 3 shows an example of the picture layout optimization procedure.

Evaluation Metric

To assess the system's performance, an evaluation measure was used to gauge the amount of information conveyed by the picture produced. The user is shown the generated picture alone without the original text, and is asked to write down the meaning of the picture in text. Such *user-generated text*, u, is automatically compared to the original

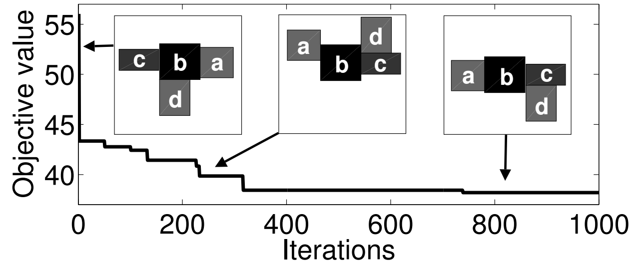


Figure 3: The minimum value of the objective function as a function of the number of candidate pictures generated. At selected points, the best layout found is shown. Closeness constraints were (a,b), (b,c) and (c,d). Darker images represent more important keyphrases.

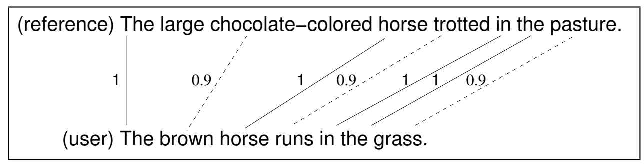


Figure 4: An example TTP alignment for evaluation.

reference text, r, used to generate the picture. The assumption is that the closer u is to r, the better the TTP system, because the user gets more correct information out of the picture. This procedure is similar to the game of Pictionary.

The key to this measure is an appropriate similarity function to compare u and r. For example, as shown in Figure 4, assume the TTP system generates a picture from the reference sentence $r =$"The large chocolate-colored horse trotted in the pasture," and, during evaluation, the user produces the sentence $u =$"The brown horse runs in the grass." Note several words are different but similar (i.e., substitutions). Insertions and deletions can occur too.

Intuitively, we want two things simultaneously. On one hand, the user sentence u should have all the words in the reference sentence r (so important concepts are covered). This can be captured by *recall* (R). The standard ROUGE measure for text summarization (Lin & Hovy 2003) is such a recall-based measure. On the other hand, the user sentence u should not have too many irrelevant words (otherwise u can always be the entire vocabulary, which would perfectly cover r). This can be captured by *precision* (P). The standard BLEU measure for machine translation (Papineni *et al.* 2002) is such a precision-based measure.

Since both recall and precision are important for evaluating TTP systems, we combine them and compute the standard F-score, $F = 2PR/(P + R)$. In order to compute precision and recall, we need to (i) handle (near) synonyms, and (ii) define an alignment between the reference and user text.

We address the synonym issue by defining a substitution function that takes a pair of words and returns a similarity measure between them. For example, the substitution function returns 1 if the two words are identical or share the same stem (e.g., *run* vs. *ran*). The function returns a score less than 1 if the two words are synonymous (e.g., *pasture* and *grass*, *mare* and *horse*). Several WordNet-based similarity measures exist (e.g., (Pedersen, Patwardhan, & Michelizzi 2004)). The results reported here use a similarity that decays exponentially (by a factor of 0.9) as the number of levels between the two words in the WordNet lattice increases. Words more than three levels apart receive a substitution score of 0.

Using the substitution function, a greedy alignment algorithm was defined. That is, among all reference-user word pairs, the pair with the highest substitution score is picked. All pairs containing either one of the two words are removed, and the procedure is then repeated until word pairs are exhausted. In the example in Figure 4, the result of greedy alignment is shown with the assumed substitution score of 1 for identical words and 0.9 for synonyms. Let $a(w)$ be the substitution score attached to word w after alignment, and $|u|$ and $|r|$ be the lengths of u and r, respectively. The 'soft' precision, recall, and F-score, which use substitution and alignment, are $P = \sum_{i=1}^{|u|} a(u_i)/|u|$, $R = \sum_{i=1}^{|u|} a(u_i)/|r|$, $F = 2PR/(P+R)$. For the example in Figure 4, $P = 6.7/7$, $R = 6.7/9$, and the final F-score is 0.84. Note that the actual evaluation measure ignores stop words in both sentences.

Experimental Results

User studies were conducted to assess the TTP system's performance in two scenarios: children's book illustration and news article visual summarization. In the first scenario, TTP was used to produce pictures to represent short texts that originate from single pages of illustrated children's books. Our hope is that TTP-generated pictures convey as much information content as the original illustrations presented in the children's book. In the second scenario, we examine TTP's ability to present a visual summary of a news article, which is more descriptive than the original news photograph. We hope to show that, while the news photograph often lacks enough details for a viewer to determine the main idea in the article, combining the photograph with a TTP-generated composite picture allows the user to understand the gist of the article.

Children's Book Illustration For the first user study, the TTP system was used to illustrate 20 randomly selected texts from a large pool of children's books. These texts range from 16 to 140 words and span one or more sentences. Figure 5 shows the TTP output produced for one example text. Note each text also has an original illustration, so there are 40 pictures in all (20 TTP, 20 illustrations). Users were asked to write a short text description (i.e., the user text) of each of the 40 pictures, so we can compare whether the TTP picture or the illustration is better at presenting the meaning of the original story (i.e., reference text). Astute users may be able to figure out which illustration and TTP picture present the same story, and thus may have more information when describing the latter of the pair. To counteract this phenomenon, we displayed all the TTP-generated pictures (in random order) before all the illustrations (in different random order). This actually gives the book illustrations an advantage, since users might have remembered TTP pictures of the same stories shown before (and thus are able to mention details not explicitly illustrated).

```
The girl
loved the
dog.  The
girl loved
the dog's
soft eyes
and warm
nose and big
paws.  The
girl wished
she had a
dog.
```

Figure 5: A TTP picture for the above text. Note the monkey image obtained from image search represents, incorrectly, the keyphrase "soft eyes."

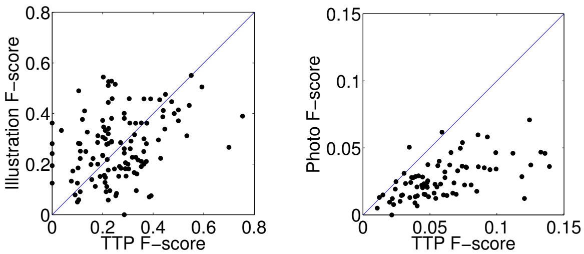
Six participants provided 40 short text descriptions each, ranging from a few words to a few sentences. For example, the responses for the TTP picture in Figure 5 were:

"A girl's pet puts its paw on her nose."
"The dog walked up to the girl and sniffed her."
"The dog bit the girl in her nose and ran away."
"The girl's nose smelled the dog and monkey as they walked away."
"The girl walked her dog and saw a hairy man with a big nose."
"The girl monkey nose smells dog paw prints."

Note that the actual book illustration shows only a girl sitting on a sofa hugging a large dog. While the responses for that picture (e.g., "The girl and her giant dog hugged on the couch.") tend to be accurate descriptions, they also differ greatly from the true text of the story.

Post-study, we compared each of the user texts to the corresponding reference text using the F-score introduced earlier. The scatter plot in Figure 6(a) shows the relationship between F-score based on TTP pictures (x-axis) and F-score based on original illustrations (y-axis). Each point represents one user's score for one of the 20 stories. Just over half (53%) of the points fall below the diagonal. If we average out individual user differences by combining the points for the same stories, 70% of the aggregate points fall below the diagonal (i.e., TTP helps recreate the reference text better in 14 of the 20 cases). Averaged over all stories and all users, the F-score based on TTP pictures is 0.98 times the average F-score based on the hand-drawn illustrations, suggesting that the TTP provides users with a picture that conveys almost (but not quite) as much information as the text's original illustration.

News Article Visual Summarization In the second study, 10 Associated Press news articles (350–850 words, plus one photograph) were randomly selected from different domains. Here, the goal is to investigate TTP's ability to augment a simple news photo with more information. Thus,



(a) Children's books (b) News articles

Figure 6: Scatter plots comparing TTP pictures (x-axis) vs. children's book illustrations or news photographs (y-axis).

we first show each real photograph, followed by the photograph next to the TTP-generated picture. Note that in such a long article, there will be many potentially picturable items, but the keyphrase extraction algorithm selects the ones most central to the text's meaning. To evaluate the difference in information provided by the original and combined pictures, the F-score was computed using the user's text and the corresponding full article text. Given the length of the full text compared to a typical user response, we expect these scores to be low, but we care only about the difference between the picture sources. Eight participants provided 20 user texts each. Figure 6(b) plots F-score using photograph+TTP pictures (x-axis) versus F-score based on original photographs alone (y-axis), where each point represents a single user on a single article. 94% of the points lie below the diagonal, and if we average over users, 100% of the aggregate points lie below the diagonal. The overall average F-score based on TTP-augmented pictures is 2.21 times the average F-score based on the original news photographs alone. This indicates that TTP renders a visual representation that is far superior in conveying the news article than its original photograph. This is to be expected, as the photos typically show only a single person or scene, whereas the articles discuss many entities that the TTP pictures capture.

Overall, these experiments show that our TTP system conveys as much or more of the content of the text through the generated pictures, than the original illustrations or photos that accompany the text.

Conclusions

We presented a general-purpose Text-to-Picture synthesis system, built upon a synergy of AI components using natural language processing, computer vision and graphics, and machine learning. Two user studies quantitatively demonstrate the TTP system's ability to generate pictures that convey the gist of input text. The current work is a first step towards automatically producing pictures that realistically depict arbitrary text. Future work includes incorporating context to produce scenes, performing deeper semantic analysis, and depicting actions with animation. We plan to investigate several TTP applications, including literacy development for children and rehabilitation for brain-injured patients.

References

Adorni, G.; Manzo, M. D.; and Giunchiglia, F. 1984. Natural language driven image generation. In *Proc. COLING*, 495 – 500.

Ben-Haim, N.; Babenko, B.; and Belongie, S. 2006. Improving web-based image search via content based clustering. In *Proc. CVPR Workshops*.

Brown, D. C., and Chandrasekaran, B. 1981. Design considerations for picture production in a natural language graphics system. *Computer Graphics* 15(2):174 – 207.

Clay, S. R., and Wilhelms, J. 1996. Put: Language-based interactive manipulation of objects. *IEEE Computer Graphics and Applications* 16(2):31 – 39.

Comaniciu, D., and Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence* 24(5):603–619.

Coyne, B., and Sproat, R. 2001. WordsEye: An automatic text-to-scene conversion system. In *Proc. SIGGRAPH 2001*, 487–496.

Deselaers, T.; Keysers, D.; and Ney, H. 2004. Features for image retrieval: A quantitative comparison. In *Proc. 26th DAGM Symposium*, 228–236.

Felzenszwalb, P. F., and Huttenlocher, D. P. 2004. Efficient graph-based image segmentation. *Int. J. Computer Vision* 59(2):167–181.

Hehner, B. 1980. *Blissymbolics for use*. Blissymbolics Communication Institute.

Johansson, R.; Berglund, A.; Danielsson, M.; and Nugues, P. 2005. Automatic text-to-scene conversion in the traffic accident domain. In *Proc. 19th IJCAI*, 1073–1078.

Lin, C.-Y., and Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proc. HLT-NAACL 2003 Conf.*, 71–78.

Lu, R., and Zhang, S. 2002. *Automatic Generation of Computer Animation: Using AI for Movie Animation*. Lecture Notes in AI, vol. 2160. Berlin: Springer-Verlag.

Mayer, R. 2001. *Multimedia Learning*. Cambridge University Press, Cambridge, UK.

Mihalcea, R., and Leong, B. 2006. Toward Communicating Simple Sentences Using Pictorial Representations. In *Proc. Conf. Association for Machine Translation in the Americas (AMTA)*.

Mihalcea, R., and Tarau, P. 2004. TextRank: Bringing order into texts. In *Proc. Conf. Empirical Methods in Natural Language Processing*, 404–411.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proc. 40th ACL Meeting*, 311–318.

Pedersen, T.; Patwardhan, S.; and Michelizzi, J. 2004. WordNet::Similarity - Measuring the relatedness of concepts. In *Proc. 19th AAAI Conf.*, 1024–1025.

Turney, P. 1999. Learning to extract keyphrases from text. Technical Report ERB-1057, Institute for Information Technology, National Research Council of Canada.

Wang, J.; Sun, J.; Quan, L.; Tang, X.; and Shum, H.-Y. 2006. Picture collage. In *Proc. Computer Vision and Pattern Recognition Conf.*, 347–354.

Yamada, A.; Yamamoto, T.; Ikeda, H.; Nishida, T.; and Doshita, S. 1992. Reconstructing spatial image from natural language texts. In *Proc. COLING, Vol. 4*, 1279–1283.