Mobile App for Text-to-Image Synthesis

Ryan Kang[1], Athira Sunil[2], and Min Chen[3(✉)]

[1] Tableau Software, Seattle, WA 98103, USA
rkang@tableau.com
[2] eBay Inc., San Jose, CA 95125, USA
atsunil@ebay.com
[3] University of Washington Bothell, Bothell, WA 98011, USA
minchen2@uw.edu

Abstract. Generating visual representation of textual information is a challenging yet interesting topic with many potential applications. In this paper, we propose a novel approach to visualize natural language sentences using ImageNet to enhance language education. Currently the focus is to assist English language learners in building their vocabulary of common nouns and developing an in-depth understanding of the various prepositions of locations. To achieve this goal, real-world images representing nouns are obtained from ImageNet and their foreground objects of interest are extracted using image segmentation. The objects are then re-arranged on a canvas based on their spatial relationship specified in the sentence. To demonstrate the effectiveness of the proposed approach, we have developed a mobile application that uses the RESTful API to retrieve the images from the web service that operate the image generation program. The prototype mobile application can create visual representations of natural language sentences and a text description of the spatial relationship of objects to assist in learning new vocabulary and spatial prepositions during language education.

Keywords: Text-to-Image · Image-to-Text · Mobile application · ImageNet · WordNet · RESTful API · Speech recognition · English language education

1 Introduction

Creating the visual representation of natural language sentences is a challenging task with the potential to spark new and innovative applications. For instance, the technology can be used for automatic generation of text illustration, language translation using images as intermediaries, and more descriptive and intuitive sentence-based image search. Recently, there have been significant efforts to study the relationship between natural language sentences and their image representations. This study is generally done in two directions: an image can be given as input and a sentence can be produced as output; or an animation or scene can be generated from a textual description. In this paper, we propose an approach to visualize natural language sentences or a text representation that describe the spatial relationship between two objects to assist in language education.

Y. Yin et al. (Eds.): MobiCASE 2019, LNICST 290, pp. 32–43, 2019.
https://doi.org/10.1007/978-3-030-28468-8_3

Several studies have shown that multimedia visual aids can be effectively used in English language learning to enhance and facilitate the comprehension of grammar and language [1]. Visualization of textual information can also help in memorizing new vocabulary and structures. The objective of this study is to assist English language learners in studying common nouns and prepositions of location through automatic illustration of sentences or automatic text generation to describe the spatial relationship between two objects. Our mobile application enables speech recognition to generate text from an English spoken sentence consisting of common nouns and prepositions of location. Then, it sends a RESTful API request to the web service for images to illustrate the spatial relationship of the two noun objects in the sentence. The generated images are based on the real-world images containing the nouns from the ImageNet [2] database. The mobile application also enables users to select, drag, drop, and move two random images, on which it produces text descriptions of the spatial relationships between the two images accordingly.

The contributions of our work can be summarized as follows. Firstly, the system uses ImageNet [1], a large-scale image database organized according to the nouns in the WordNet [2] hierarchy with an average of over five hundred images per node of the hierarchy. By using ImageNet, the system can illustrate a rich set of common nouns in real-world contexts to build the vocabulary of users in a more meaningful manner. WordNet is also used for word tokenization and to collect the *Synset*: a set of synonyms that share a common meaning. Secondly, the mobile application integrates the iOS speech recognition engine to enhance the user's experience in language learning by converting the user's voice to text and then use the text as a parameter to send the request to the web service to generate images. When the mobile application receives the images, it shows the visual representation of the sentence to the user. Thirdly, the system allows the users to select noun images they prefer to be used in visualization to make the learning experience more engaging. Our mobile application provides two different approaches that can be used in English language learning.

2 Literature Review

Several studies have focused on learning the relationship between images and their sentence based semantic descriptions. Many works have studied the task of generating textual descriptions of images. Significant works have been also made in the multimedia and computer vision communities to improve image search using textual queries. Many papers have explored the visual meaning of different parts of speech. Comparatively fewer studies have addressed the idea of scene creation from natural language sentences.

One study [4] proposed an application to improve the user experience when reading news articles through automatic generation of an audio-visual presentation of the article. The application focused on retrieving an image from Flickr to illustrate a given sentence in the news article through the relation of neighboring sentences and image tags. Another study [5] proposed a system that can automatically add objects to an image when the background image and labels of objects (e.g. car) to be added are provided as input. The system estimated the position, scale, and appearance of objects

and automatically added them to images without direct user input. The study in [6] focused on learning the visual features that correspond to semantic scene phrases derived from sentences. The work used abstract scenes generated from clip art to study semantic scene understanding. WordsEye [7] is an online application for converting text into representative 3D scenes. The system relies on its collection of 3D models and poses to depict entities and actions. All the models have associated shape displacements, spatial tags, and functional properties to be used in the scene generation process. Another work [8] proposed a system to create scenes from natural language sentences with a focus on the development of a hierarchical syntactic parser for sentence analysis and the correlation of words in the sentences with an image patch of the closest concept within a small number of choices.

3 Implementation

3.1 System Architecture

Figure 1 shows the high-level architecture of the system. Our system consists of the following four major components:

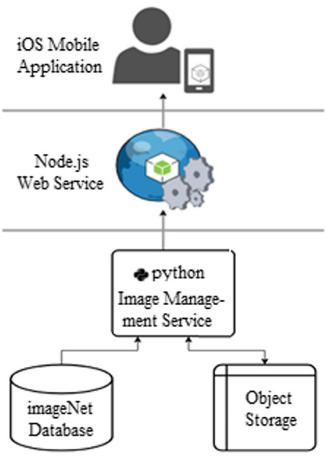

Fig. 1. Architecture diagram

- The ImageNet dataset: it contains the URLs to download the images and their annotations, and to obtain the set of cognitive synonyms (synsets) from the WordNet dataset for the nouns in the sentence.
- The image management component: it builds a database containing the attributes of the images in ImageNet, word tokenization, image segmentation, and output image generation.
- The web service component: it handles the request to start a job to generate the image and send back the generated image.
- The mobile application: it handles all the interactions with the user.

3.2 Data Set

The real-world images of nouns used in the visualization of sentences are obtained from the ImageNet database. A noun may have different meanings in different contexts. The WordNet database is used to obtain the specific meaning of a noun in a given sentence and retrieve images from ImageNet that visually represent the concept.

ImageNet. ImageNet is a large-scale image dataset organized according to the WordNet hierarchy. Currently, 21841 synsets of nouns in WordNet are indexed in ImageNet with an average of over 500 images per synset. ImageNet provides URLs of web images for each synset to download and all images in ImageNet are quality-controlled and human-annotated. ImageNet also provides annotations of object bounding boxes for each unique image in over 3000 popular synsets. Our system uses images with annotated and verified bounding boxes.

WordNet. WordNet is a large lexical database for English in which nouns, verbs, adjectives, and adverbs are grouped into sets of synonyms called synsets, each of which expresses a distinct concept. In our study, we focus on the synsets of nouns in WordNet. The synsets are ordered by their estimated frequency of use. Each synset has an associated synset offset which is the byte offset of the synset in the WordNet database file called data.noun. The synset offset of a synset is an 8 digit, zero-filled decimal integer which can be used to uniquely identify the synset. Table 1 shows the different synsets in WordNet containing the noun *table.*

Table 1. Synsets containing the noun *table* with the corresponding concepts and synset.

Synset	Concept	Synset offset
{table, tabular array}	A set of data arranged in rows and columns	08266235
{table}	A piece of furniture having a smooth flat top that is usually supported by one or more vertical legs	04379243
{table}	A piece of furniture with tableware for a meal laid out on it	04379964
{mesa, table}	flat tableland with steep edges	09351905
{table}	A company of people assembled at a table for a meal or game	08480135

3.3 Image Management Component

Database. We first built the database with image IDs, URLs to download the images, and object bounding box annotations for all the images available on the ImageNet. An image ID is a concatenation of POS (Part of Speech) tag and synset offset of WordNet ID followed by the underscore character (_) and a number which is unique for each image.

For example, the image ID of the image of *table* (see Fig. 2) is n04379243_17932, where 'n' is the POS tag for nouns, 04379243 is the WordNet ID (wnid), and 17932 is the unique number for the image. ImageNet provides the URLs to download the images as a text file and the annotations of object bounding boxes of images as XML files.

Fig. 2. Sample real-world image of *table* from ImageNet

The XML file contains the image ID, image dimensions (width, height, and depth) and bouning box coordinates of each foreground object in the image (see Fig. 3). We deserialized the XML files to extract the image ID, image dimensions (width and height) and coordinates of the top left corner and bottom right corner of the object bounding box. Then, we stored all the information we gathered from the ImageNet into a SQL database to improve the performance of the system and keep the system highly available even if the ImageNet website doesn't respond.

```
<?xml version="1.0"?>
- <annotation>
    <folder>n04379243</folder>
    <filename>n04379243_11010</filename>
  - <source>
      <database>ImageNet database</database>
    </source>
  - <size>
      <width>448</width>
      <height>265</height>
      <depth>3</depth>
    </size>
    <segmented>0</segmented>
  - <object>
      <name>n04379243</name>
      <pose>Unspecified</pose>
      <truncated>0</truncated>
      <difficult>0</difficult>
    - <bndbox>
        <xmin>0</xmin>
        <ymin>26</ymin>
        <xmax>447</xmax>
        <ymax>255</ymax>
      </bndbox>
    </object>
</annotation>
```

Fig. 3. Sample XML file containing object bounding box annotations

Pre-download Images. We download 10 random real-world images for each unique image ID from the ImageNet in order to minimize the waiting time for the end user as image downloading time can vary from 2 to 8 s. Additionally, we also extract the object image from the original image using the bounding box coordinates and the GrabCut algorithm and store the processed images in the local file storage of the machine that runs the web service.

Word Tokenization. During the word tokenization process, the system tokenizes the input sentence that contains two common nouns and a preposition of location. In this step, the system will extract main noun, dependent noun, and preposition in the sentence using the tokenization, word singularization, and POS tag detection features from the Natural Language Toolkit [9]. The POS tag determines which object becomes the main noun or the dependent noun and which word is the preposition.

Image Segmentation. Our system extractsthe objects of interest from the images. To achieve this, the system uses the GrabCut algorithm proposed by Rother, Kolmogorov, and Blake in [10]. GrabCut is an interactive foreground extraction tool where the user drags a rectangle around the foreground region. The foreground region must be completely inside the rectangle since everything outside the rectangle will be taken as sure background. The algorithm then segments the image iteratively till the foreground/background classification converges. Our system automates the algorithm without the user having to select the region of interest by using the bounding box information. Figure 4 shows the image of a cat from ImageNet with a rectangle drawn using the object bounding box coordinates and the output from the GrabCut algorithm with a transparent background.

Fig. 4. (left) Original image of a *cat* from ImageNet (middle) The bounding box containing the *cat* (right) Output of GrabCut algorithm with transparent background

Create the Image Illustrating the Sentence. We create the output image visualizing the sentence by placing the images of the objects obtained from the previous step on a canvas according to the preposition of location. The sizes of the images obtained from ImageNet can vary greatly from 75 × 75 pixels to 1024 × 768 pixels. Therefore, the system resizes the images of the objects to a suitable and consistent size keeping the aspect ratio. After the images are resized, alpha blending is used on both main and dependent noun object images as foreground image to combine it with a background image to create the appearance of transparency and smooth out the boundaries.

The alpha blending uses the alpha channel of the image encoding and the following equation is used to overlay the image of the object on the canvas:

$$I_A = \alpha I_F + (1 - \alpha) I_B$$

In the equation, I_A is the alpha blended image, I_F is the foreground image of the object, and I_B is the background canvas image. The created image would not look as natural without the alpha blending (see Fig. 5).

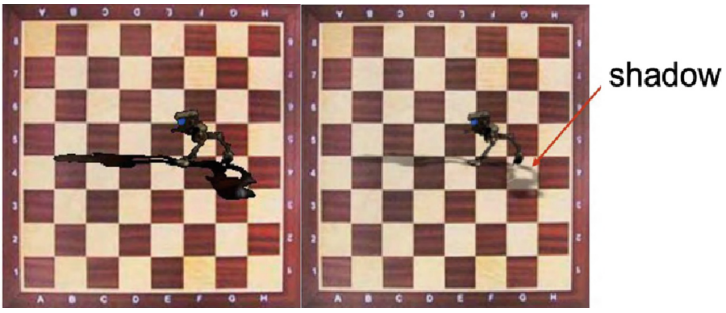

Fig. 5. (left) Image created without alpha blending (right) Image created with alpha blending [11]

3.4 Web Service

Our Web Service component is built on the Node.js platform with Express.js web framework. This component starts an HTTP server and listens on the specific requests. When the web service receives the request, it creates a child process to start a job on the Image Management component to perform the image generation or image retrieval. When the job is completed, the child process will callback with either success or error. If the Image Management component has successfully completed, the web service will send the response back to the client with HTTP status code 200 (OK) and the image. Alternatively, the web service will send HTTP status code 500 (Internal Server Error) and error message when the child process fails.

3.5 Mobile Application

For the mobile application, we focused on the usability and performance as it is the end user facing application. Our mobile application is designed to target the English language learners and we tried to use as much visualization as possible to assist the users in understanding how to use the mobile application. We also added speech recognition and a drag and drop interaction that would help in making language education more engaging.

Speech Recognition. Apple supports speech recognition framework for the iOS 10 and above [11] and our mobile application uses this framework to recognize spoken words in live audio. The user will be seeing his/her voice translated into text on the

mobile screen in live and uses the spoken words as input parameter to generate an image visualizing the sentence. Network connectivity is required for this feature as the speech recognition framework relies on Apple's server.

URL Sessions. Our mobile application uses Apple's URLSession APIs to download the image data from the web service [12]. This functionality has been written to support asynchronous calls to allow the main UI thread to continue to run smoothly while the data is being fetched in the background.

User Interaction. We have added drag and drop gesture support on our mobile application for Image-to-Text conversion feature. This feature is only available in the mobile application as the user has to use drag and drop gesture to move around the two images within the mobile screen. Once the user has finished moving the images, the 'Generate' button will become enabled and the user will see the text describing the spatial relationship between the two objects in the images.

Spatial Relationship Description Based on Images Location. Our mobile application calculates the distance between the location of the two object images to generate the text representing the scene on the mobile screen. It is currently supporting six different prepositions to show the spatial relationship between the two objects. The prepositions we support are 'On', 'Under', 'Beside', 'Above', 'Below', and 'Near'.

Text-to-Image Use Case Sequence Diagram. Figure 6 shows the use case of the Text-to-Image API from the mobile application. The mobile application first receives the voice command from the user and convert the audio input to text. Then, the mobile application sends GET request to the web service which starts a job in Image Management component. Upon successful call back to the web service with image location, the web service responds back to the mobile application with image file. Finally, the mobile application displays the image on the screen. The three components of the system work together as a single service for creating visual representations of natural language sentences.

4 Results

We have developed an educational application to demonstrate the effectiveness of the proposed approach to visualize natural language sentences. Our mobile application has been designed to help the English language learners with usability in our mind. The GUI of the mobile application for visualizing the spoken words on the mobile screen is shown in Fig. 7. As we can see, the main mobile application UI displays the tab bar buttons to switch between the Text-to-Image conversion view and Image-to-Text view on the bottom. The 'START'/'STOP' button for voice recognition is also visually appealing to the user for the interaction. The live text feedback on the center of the mobile screen helps the user to know if the voice recognition API has successfully converted the spoken words to text or not. The user can use this button to re-try if needed as the tapping on the 'STOP' button will clear the text. We used bright and high contrast colors and large font sizes for the usability and design purpose.

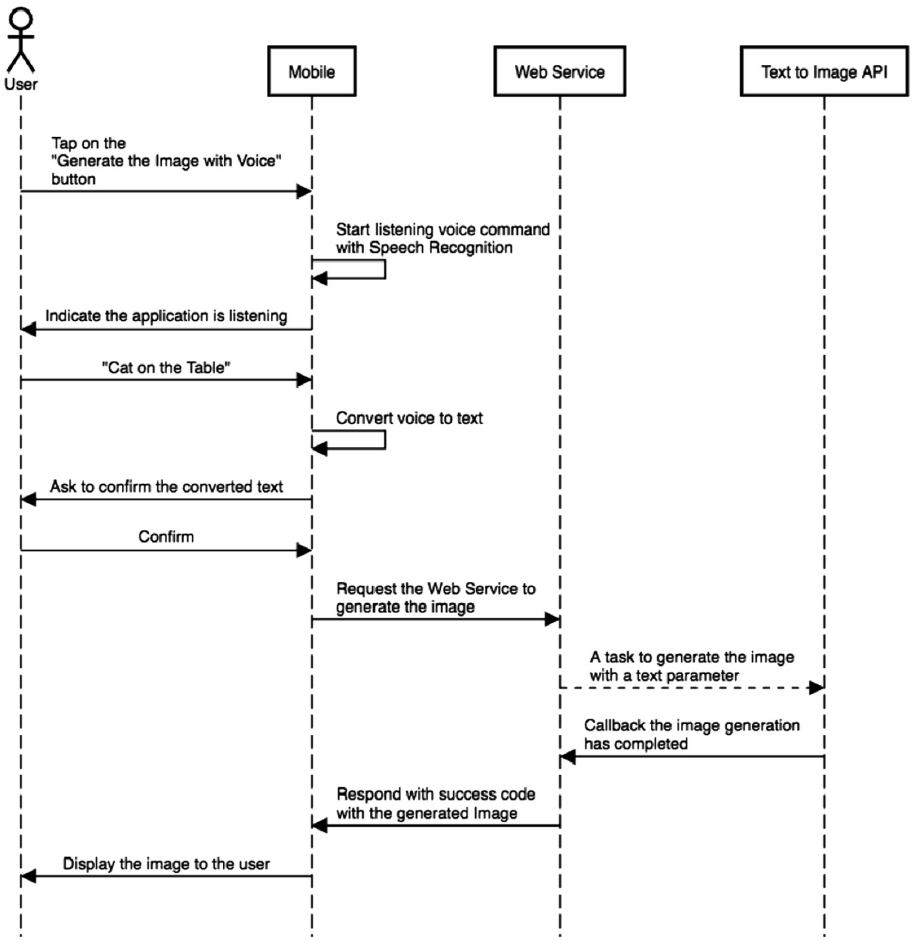

Fig. 6. Sequence diagram

The user interface of the converting the two images on the screen to a text describing the spatial relationship between two objects (see Fig. 8) is similar to Text-to-Image view. This view follows the same theme from the Text-to-Image conversion view and re-use the button with the font, font size, and color for consistency and usability. Although the drag and drop gesture to move around the images are not obvious, enabling the 'GENERATE' button only when the user interact with the mobile application suggest interaction is required.

The following Table 2 gives a summary of the performance measureswe collected. It took an average of 1.0594 s to extract the foreground object from the image using the GrabCut algorithm with bounding box annotations. An average number of the bytes the mobile application downloaded from the web service was 558502.875 bytes and the elapsed time to fetch the image was 3.78247 s.

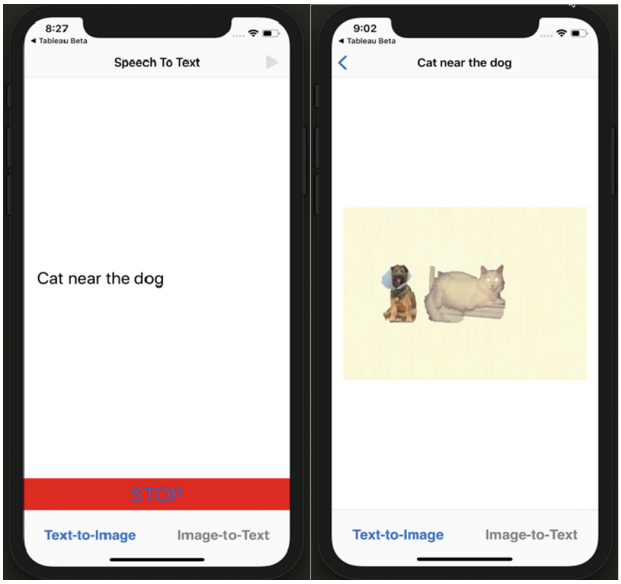

Fig. 7. Text-to-Image conversion user interface

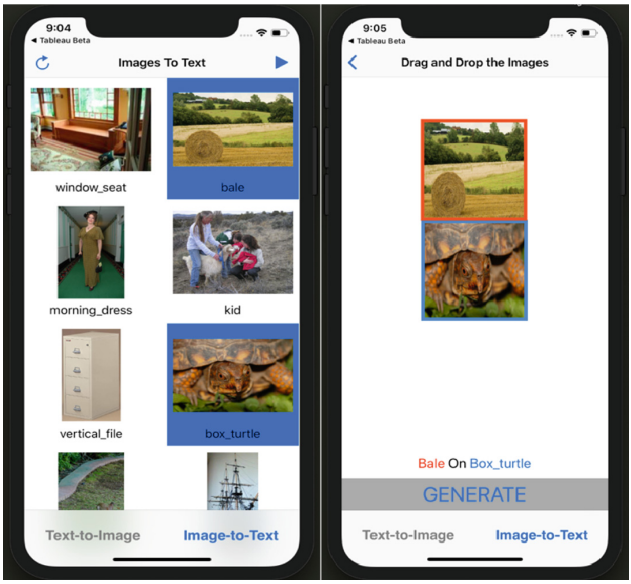

Fig. 8. Image-to-Text conversion user interface

Table 2. Elapsed time taken to fetch image created via Text-to-Image API.

Experiment count	Elapsed time to execute the GrabCut algorithm	Downloaded data bytes	Elapsed time to fetch the created image
1	2.0129 s	567736 bytes	3.89415 s
2	1.7314 s	585643 bytes	4.52450 s
3	0.8248 s	524609 bytes	3.39177 s
4	1.2437 s	563855 bytes	3.67760 s
5	0.5785 s	586443 bytes	4.52157 s
6	1.8266 s	561345 bytes	3.29475 s
7	0.8926 s	517657 bytes	3.37896 s
8	0.3305 s	560735 bytes	3.57648 s
Average	**1.0594 s**	**558502.875 bytes**	**3.78247 s**

5 Conclusion and Future Works

Visualization of natural language sentences has the potential to spark innovative applications. In this paper, we have proposed a novel approach to visualize natural language sentences using a large online image dataset called ImageNet. We have also developed an educational application to demonstrate the effectiveness of the proposed approach. The developed mobile application shows that the proposed approach can be effectively used to create illustrations of sentences containing common nouns and prepositions of location to assist in building new vocabulary and structures during language education.

One of the unique features of our approach is the use of a large online hierarchical image database. This helps in obtaining real-world images of a rich set of nouns in different contexts without the overhead of creating and maintaining an image database. By using the mapping between WordNet and ImageNet to automatically label images, the proposed approach eliminates the need to manually tag images.

The method proposed in this paper to illustrate natural language sentences can be used in many other applications. It can be used in language learning to assist learners enriching their vocabulary and developing a detailed understanding of the various grammatical structures of the language such as using the prepositions. The method can be also used for automatic illustration of language worksheets.

Currently, ImageNet provides annotations of object attributes like color (black, blue, brown, gray, green, orange, pink, red, violet, white, yellow), pattern (spotted, striped), shape (long, round, rectangular, square), and texture (furry, smooth, rough, shiny, metallic, vegetation, wooden, wet) for about 400 synsets. Examples of attributes provided by ImageNet for various objects is shown in Fig. 9. This information can be integrated into the system to use sentences that also contain adjectives for the nouns. For example, 'The *black cat* is *on* the *round table*.' The sentences can also include the number of objects like 'There are *three black cats on* the *table.*'

Fig. 9. Annotations of object attributes provided by ImageNet (Color figure online)

References

1. Omaggio, A.C.: Pictures and second language comprehension: do they help? Foreign Lang. Ann. **12**(2), 107–116 (1979)
2. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
3. Miller, G.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
4. Delgado, D., Magalhaes, J., Correia, N.: Assisted news reading with automated illustration. In: Proceedings of the International Conference on Multimedia – MM 2010 (2010)
5. Inaba, S. Kanezaki, A., Harada, T.: Automatic image synthesis from keywords using scene context. Ibn: Proceedings of the ACM International Conference on Multimedia – MM 2014 (2014)
6. Zitnick, C., Parikh, D., Vanderwende, L.: Learning the visual interpretation of sentences. In: 2013 IEEE International Conference on Computer Vision (2013)
7. Coyne, B., Sproat, R.: WordsEye. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques – SIGGRAPH 2001 (2001)
8. Mano, T., Yamane, H. Harada, T.: Scene image synthesis from natural sentences using hierarchical syntactic analysis. In: Proceedings of the 2016 ACM on Multimedia Conference – MM 2016 (2016)
9. Bird, S., Loper, E., Klein, E.: Natural Language Processing with Python. O'Reilly Media Inc., Sebastopol (2009)
10. Rother, C., Kolmogorov, V., Blake, A.: GrabCut. ACM Trans. Graph. **23**(3), 309 (2004)
11. Efros, A.: Image Compositing and Blending, Carnegie Mellon University (2007). http://graphics.cs.cmu.edu/courses/15-463/2007_fall/Lectures/blending.pdf. Accessed 4 Feb 2019
12. Apple Developer Documentation Web Page. https://developer.apple.com/documentation/speech. Accessed 4 Feb 2019
13. Apple Developer Documentation Web Page. https://developer.apple.com/documentation/foundation/urlsession. Accessed 4 Feb 2019