Записки научных
семинаров ПОМИ
Том 529, 2023 г.

V. Firsanova

WHAT DO TEXT-TO-IMAGE MODELS KNOW ABOUT THE LANGUAGES OF THE WORLD?

Abstract. Text-to-image models use user-generated prompts to produce images. Such text-to-image models as DALL-E 2, Imagen, Stable Diffusion, and Midjourney can generate photorealistic or similar to human-drawn images. Apart from imitating human art, large text-to-image models have learned to produce combinations of pixels reminiscent of captions in natural languages. For example, a generated image might contain a figure of an animal and a symbol combination reminding us of human-readable words in a natural language describing the biological name of this species. Although the words occasionally appearing on generated images can be human-readable, they are not rooted in natural language vocabularies and make no sense to non-linguists. At the same time, we find that semiotic and linguistic analysis of the so-called hidden vocabulary of text-to-image models will contribute to the field of explainable AI and prompt engineering. We can use the results of this analysis to reduce the risks of applying such models in real life problem solving and to detect deepfakes. The proposed study is one of the first attempts at analyzing text-to-image models from the point of view of semiotics and linguistics. Our approach implies prompt engineering, image generation, and comparative analysis. The source code, generated images, and prompts have been made available at `https://github.com/vifirsanova/text-to-image-explainable`.

§1. Introduction

Text-to-image generation is a machine learning task that refers to an image synthesis conditioned by a prompt, i.e., a natural language description. While being asked to generate some pictures containing text (for example, with a prompt "an advertising board with a word on it"), these models can produce images containing human-readable letters. However, the letters put together do not form natural language words. Figure 1 shows two images generated with the prompt "an advertising board with a word on it" by a multilingual text-to-image model Kandinsky 2.0 [14]. It is possible to

Key words and phrases: Explainable Artificial Intelligence and Text-to-Image Synthesis and Diffusion Models.

make out the words “tia” and “g’ioniy” (ending with a dot) in the images. Although the words themselves hardly make any sense, what catches the eye is that each of their letters uses one and the same colour and font, the letter spacing is uniform, and punctuation marks (period, apostrophe etc.) are used in accordance with regular grammar. This observation shows that although the pictured text generated by text-to-image models seems random, it is evident that the model has derived some laws of the underlying natural language.

In this work, we aim to describe and analyze the hidden language of text-to-image models from a linguistic perspective. The novelty of our study is that it is the first attempt to give a linguistic description of the text-to-image model’s mechanism. We also see a wide research gap in the analysis of multilingual language models, which is why the present work focuses on multilingual analysis. We hypothesize that outputs of text-to-image models containing text in different languages are culturally biased and conditioned by linguistic typological generalizations similar to those that autoregressive models for text generation make. The study contributes to explainable AI, computational linguistics, and semiotics, as the results of this study can help us find a solution to the AI black box problem, use text-to-image models more consciously by prompt engineering, reduce risks of applying them in solving real-life tasks, and make further steps in semiotic analysis of fakery.

§2. Related Work

In [10], the authors posit that such nonsensical words as “tia” and “g’ioniy” produced by a text-to-image model might be a part of the so-called hidden vocabulary. The hidden vocabulary is a list of words or phrases that text-to-image models develop internally, while being fed with images containing some captions. While nonsensical to humans, those words have a certain meaning in the hidden text-to-image model knowledge base. To prove this hypothesis, Daras and Dimakis use some of the generated nonsensical words as prompts for a text-to-image model DALL-E 2. They found that using nonsensical captions for images of birds generated by DALL-E 2 as prompts for the same text-to-image model often results in producing pictures of similar birds. That means that a nonsensical word occasionally appearing on generated images might have a certain meaning, like in a human dictionary every word has its own meaning. The authors of the paper also discover some properties of the hidden vocabulary. For

Fig. 1. Images generated by Kandinsky 2.0 with the prompt “an advertising board with a word on it”.

example, compositionality means that different words of the hidden text-to-image model vocabulary can be combined to form sentences, like words of a natural language.

The authors, however, did not analyze the linguistic component of the hidden vocabulary. For example, according to Daras and Dimakis, the phrase “apoploe vesrreaitais” means a bird in the DALL-E 2 hidden vocabulary, but it is not clear why the model developed internally such a combination of symbols. Our linguistic intuition tells us that the name “apoploe vesrreaitais” has much in common with binomial nomenclature, a formal system of naming species that uses Latin. Scientific names of species consist of two words, a generic name and a specific epithet. We can say that “apoploe” is a generic name, and “vesrreaitais” is a specific epithet. One of the enthusiasts on social media analyzed the byte pair encoding used in DALL-E 2 and found that the model parsed “apoploe vesrreaitais” as “apo, plo, e” and “ve, sr, re, ait, ais” finding that consonant words “apodidae”, “ploceidae" and “apodiformes" that include syllables “apo” and “plo” recognized by DALL-E 2 denote large families and orders of birds [1]. We must add that suffixes “-e” and “-is” occure in Latin third declension of nouns and adjectives. Nouns and adjectives play roles of generic names and specific epithets in our example.

This shallow and toy analysis of “apoploe vesrreaitais” shows that joint application of prompt engineering (experiments on discovering “apoploe vesrreaitais” conducted by Daras and Dimakis), the analysis of encoding and algorithm (the analysis of byte pair encoding used in DALL-E 2 by a social media enthusiast), and the linguistic or semiotic exploration presented by some of our intuitive guesses, can contribute to the solution of

the AI black box problem. In what follows, we will try to conduct a similar analysis in expanded form, using in-depth methods.

Analysis of the internal AI linguistic structure has been conducted with BERT [11]. In [4], the authors provide observational evidence of the fact that BERT's attention (which is the fundamental mechanism of Transformer-based models including BERT) corresponds to linguistic dependency syntax structure and captures coreference. In [17], the authors provide experimental evidence that BERT approximately encodes dependency trees in its contextual representations by using a structural probe and reconstructing the geometrical tree-like structures of the vector space used in BERT. Since many text-to-image models use pretrained Transformer-based encoders such as BERT, for example XMC-GAN [34] or Imagen [30], we find this evidence essential for the analysis proposed in this study.

§3. Method

The study proposes a complex approach to analyze the hidden linguistic structure of text-to-image models that continues and expands the method provided in [10]. First, we use methods of prompt engineering to discover prompts that would allow us to generate representative samples of images containing captions produced by text-to-image models. Then, we get image samples with different pipelines and use computational methods to tokenize our prompts and extract features from the generated images. We then use this data to analyze samples from the point of view of linguistics and semiotics. We aim to capture cultural biases and linguistic generalizations in text-to-image models through the generated images and compare our findings with other state-of-the-art language models.

3.1. Prompt Engineering. With the development of large language models such as the GPT family including GPT-2 [27], GPT-3 [3], InstructGPT [24], and ChatGPT [22], a new concept called prompt engineering has emerged in AI. Prompting implies providing instructions, descriptions, or samples for completion in a natural language as input for a deep learning model, i.e., it uses open-ended text. On the one hand, that opens infinite possibilities for image creation, and on the other hand, that might lead to numerous trials before getting the needed result. Prompt engineering is the search for prompts that will result in desirable outputs [31].

Our task is to generate images that contain human-readable text. The subtask is to capture cultural bias by analyzing multilingual data. That means we want to generate images containing captions in different languages. Because there are over 7,000 languages across the world, we need to choose criteria to limit the choice of languages for our study. Our sample should be representative, and one way to achieve this is to use linguistic classifications. For example, we can use either the genealogical or typological classification. Genealogical classification divides languages into language families according to their history of evolution, while typological classification groups languages by their structural and functional traits [25].

Since the processing of captions on images still remains an Achilles' heel of text-to-image models [10], we do not expect text-to-image models to recognize language structure information on the same level as BERT does [17]. But we can assume that computer vision engines in text-to-image models can be capable of distinguishing writing systems because diffusion models and generative adversarial networks (GANs) that are widespread in image processing have proven their efficiency in font generation [16] and calligraphy synthesis [21]. This fact demonstrates their capacity towards recognizing styles of lettering. As a result, we have chosen the classification of languages by writing systems as a baseline criterion to select languages for our study.

There are various writing system typologies, among which the classification proposed by Gelb [15] is the seminal one. We have chosen the approach proposed by Daniels [9] as one of the most influential works on language typology of the last three decades [2]. In his book "The World's Writing Systems", Daniels also observed the history of analogue and digital computer-mediated writing, which indicated the relevance of this study to our work [9]. Daniels suggests six writing systems types presented in Table 1.

Next, we should select several languages from each class, and our sample should correlate to the machine learning training data. One way to do this is to choose languages from the list of languages most widely used online. Another way is to use statistics of a large multilingual dataset such as Common Crawl [6]. We could try using statistics of the most spoken languages in the world [13]. But it would not be representative for our purposes because widespread spoken languages are not necessarily common on the Internet, meaning that some languages from our sample will not

Table 1. The classification of writing systems proposed by Daniels [9], as described in [2], and our selection of examples.

Type	Features	Examples
Logosyllabary (morphosyllabary)	Characters encode words (morphemes)	Chinese, Japanese (Kanji)
Syllabary	Characters encode syllables	Japanese (Kana)
Abjad (Semitic-type script)	Characters encode consonants	Hebrew, Yiddish, Persian, Arabic
Alphabet (Greek-type script)	Each character encodes a consonant or a vowel	English, Russian, Greek
Abugida (Sanskrit-type script)	Uses units of consonant-vowel sequences	Hindi, Thai
Featural	Encodes phonetic features of designated segments	Korean (Hangul)

likely be present in machine learning datasets. So we do not expect them to be processed well by deep learning models.

After the analysis of accessible language statistics, we decided to use a combination of W3Techs statistics of the top 10 million websites that provides a list of languages used in more than 0.1% of Web sources as of March 1, 2023 [33] and statistics of the distribution of languages mined by Common Crawl [6]. The "Examples" column in Table 1 shows our selection.

The next stage of our work is prompt engineering. Studies show that phrasing and selection of connecting words have no significant influence on the outputs of text-to-image models. However, we should focus on the subject we want to generate and use specific keywords. We should avoid prompts that may be misinterpreted. Due to the stochastic nature of generative algorithms, we should be ready to generate several samples for each prompt. Because generation relies on random initialization parameters, the model outputs may vary, and using different random seeds to calibrate the model might be beneficial. Using different seeds and initialization parameters might give us a broader understanding of the model's capabilities [20].

For our study, we propose to use prompts with the structure "SUBJECT in LANGUAGE", where SUBJECT is an item containing some text in natural language, and LANGUAGE is a name of a language from our selection (see the column "Examples" in Table 1). Since we aim to get images containing any free-form text, the styling or decoration of generated text is not significant for our research purposes. The phrasing is also not important [20]. So, we ended up using a baseline prompt "text in LANGUAGE" (e.g., "text in Russian"). We also prepared two backup prompts that can be used if our baseline does not work: "word in LANGUAGE" and "caption in LANGUAGE". We abandoned prompts with a broader meaning, such as "content in LANGUAGE", because they are ambiguous (for example, the word "content" refers to different modalities, including images, audio etc.). We also abandoned narrower prompts, for example, "a book with a title in LANGUAGE", because they are detailed and specific, which might reduce the quality of generated images and increase the risk of producing blurry outputs. The latter is a common issue in diffusion probabilistic models, so we would like to avoid it by simplifying our prompting.

We used our prompts to generate images with the following models: DALL-E 2 [28], Stable Diffusion [29], and VQGAN+CLIP [8]. DALL-E 2 is a two-stage model that applies a CLIP [26] image embedding to the prompt and uses a diffusion-based decoder to generate an image conditioned on the prompt. The original DALL-E 2 model is not an open-source project; we can use a public demo of the model [23] or try working with open-source implementations such as Craiyon [5]. Since we need only a limited number of images for our experiments, we decided to use the original DALL-E 2 through the public demo. That would allow us to access the full potential of the model.

Stable Diffusion uses the latent diffusion model. Diffusion probabilistic models use autoencoders to learn distributions by gradually denoising training data, i.e. images. Stable Diffusion compresses images to a latent space that encodes the semantics behind images. The model encodes text prompts with CLIP. Then the model gradually diffuses the information by applying Gaussian noise, recovers the latent representation, and generates the resulting image. Stable Diffusion is an open-source project; we can fine-tune such hyperparameters as a random seed and the number of inference steps (number of iterations for denoising).

VQGAN [12] is a generative adversarial network that encodes and vector quantizes input images. The vector quantized data form a codebook

representing the data used for Transformer processing. The Transformer learns the images' composition for further image synthesis. CLIP is a multimodal neural network that connects images with texts. The model learns visual perception from natural language supervision. VQGAN+CLIP is a combination of the two: VQGAN generates images, while CLIP evaluates the correspondence of the generated image to the prompt.

We generated 48 images (4 images for each of the 12 languages from our sample) using three models (DALL-E 2, Stable Diffusion and VQGAN+CLIP), 144 images in total. The images are stored in the project repository on GitHub for research purposes. To generate images with DALL-E 2, we used the public demo with our baseline prompt "text in LANGUAGE". The size of each image is 1024×1024 pixels. No hyperparameter tuning was applied, as the model is not open source.

To generate images with Stable Diffusion and VQGAN+CLIP, we used an NVIDIA T4 graphics processor unit provided by the Google Colab environment. We used Stable Diffusion version 2.1 base, which is smaller and more accessible than version 2.1 for the provided hardware. We accessed this version of the model through the HuggingFace Diffusers library [19]. The code for image generation is provided in our repository. We generated images of size 512×512 pixels with the baseline prompt "text in LANGUAGE". We set random seed values and the number of inference steps manually and chose the seed values randomly. We selected the number of inference steps according to the HuggingFace tutorial guidelines, and checked these values by trial and error. We ended up with the following values of inference steps: 15, 25, 30, and 50. On average, the time for image generation was between 8 and 30 seconds, depending on the number of inference steps.

To access VQGAN+CLIP, we used a Colab tutorial by Katherine Crowson [7]. The size of each generated image is 400×400 pixels. We used different prompts ("text in LANGUAGE", "word in LANGUAGE", "caption in LANGUAGE") as the model did not show much detailing in generating images with our baseline prompt "text in LANGUAGE". The backup prompt "word in LANGUAGE" gave us more detailed images. With that prompt, the model tried to paint one word in a large font, and in that way, it captured more details of the language script. While being fed with prompts "text in LANGUAGE" or "caption in LANGUAGE", VQGAN+CLIP tried to paint a paper sheet with small blurry and thus unreadable letters, which is not appropriate for an in-depth semiotic analysis. We used different

manual random seed values, as in Stable Diffusion experiments. We tried various numbers of inference steps from 50 to 400. It took us about 4 minutes to generate one image with the maximum number of inference steps. During the image generation, we output an intermediate image every 50th step and picked the most detailed for further research. We found that 200 is the optimum number of inference steps, although sometimes a larger or smaller number gives better results, which agrees with the results in [20].

To reproduce our experiments, we provided the images with the information on used prompts and chosen hyperparameters. Images generated with DALL-E 2 are stored in sets of 4 items named according to the prompt used for their production. Each image created with Stable Diffusion or VQGAN+CLIP is named according to the prompt, the number of inference steps, and seed value. The names of the images have the following structure: "{prompt}_steps_{number of inference steps}_gen_{seed}.png". Sets of images, code and supplementary information on image generation are given on each model separately and provided in folders "dalle2", "stable_diffusion" and "vqgan+clip" at `https://github.com/vifirsanova/text-to-image-explainable`.

3.2. Feature Extraction. Feature extraction is an idea that lies behind machine learning algorithms; a neural network learns to capture features corresponding to certain classes that the model should recognize. In computer vision (CV), features encode specific parts of images that a model associates with certain classes (consider a child learning to recognize cats by long whiskers). The features extracted by CV models are not human-readable, which makes it difficult to decode and interpret the model outputs. One of the solutions for latent diffusion models, such as Stable Diffusion, is to construct Diffusion Attentive Attribution Maps (DAAM) [32].

DAAM heatmaps allow to evaluate the degree of influence of words from the prompt on different pieces of an image. Consider the following sentence: "When Tucker found Alice, she was alone". An attention-based natural language processing (NLP) model learns to refer "Alice" to the pronoun "she", thus revealing the ability to encode semantics [4]. In the same way, cross-attention mechanisms in diffusion models refer text embeddings to latent image representations [29]. In [32], the authors propose to aggregate two-dimensional cross-attention mapping in diffusion models to synthesize heatmaps that indicate the connection strength between the pixels of an image and individual words of the prompt.

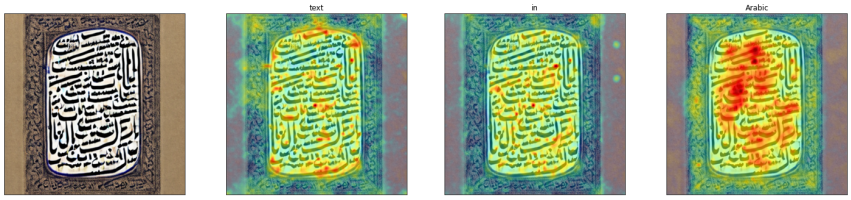

Fig. 2. DAAM heatmap for the prompt “text in Arabic” produced with Stable Diffusion v. 2.1 base. From left to right: original image generated with Stable Diffusion, heatmap for the word “text”, heatmap for the word “in”, heatmap for the word “Arabic”.

We applied DAAM to image generation with Stable Diffusion v. 2.1 base with our baseline prompt “text in LANGUAGE”. We used the same hyperparameters described in Section 3.1 to reproduce similar images. We generated images and computed DAAM heatmaps simultaneously. Then we applied maps for each of the three words of the prompt: “text”, “in”, and “LANGUAGE”. Figure 2 shows the result.

The generated image in Figure 2 shows a piece of text put in a brown frame. The letters are not human-readable, although they strongly resemble the Arabic script by their shape. The page layout presented on the generated image reminds us of the Quran, the religious text of Islam written in Arabic. The heatmap shows a strong correlation between the word “Arabic” and letters (see reddish spots in Figure 2). The imaged frame has a notable but not strong correlation with the word “Arabic” and shows some relation with the word “text”. What catches the eye is that both words “text” and “in” do not strongly affect the final result of image generation.

We can see from the DAAM heatmaps that latent diffusion models extract features associated with language scripts and their cultural backgrounds. The strong correlation between the name of the language (“Arabic”) and imagined shapes of letters shows that these models use attention to associate languages with their scripts by learning the letters’ shapes, writing direction, and even the cultural context of language use, e.g., Arabic is associated with the Quran. Results on image generation with DALL-E 2 also support this conclusion.

For example, in Chinese, horizontal and vertical writing directions are appropriate, and DALL-E 2 generated both options with the prompt “text

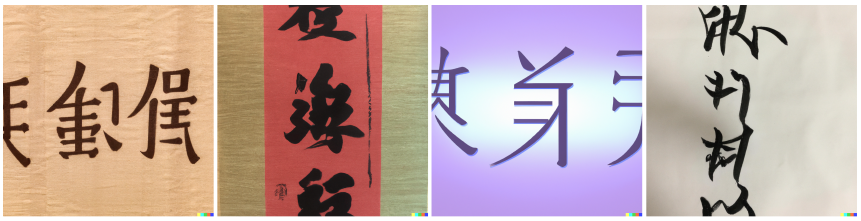

Fig. 3. DALL-E 2 image generation results with the prompt "text in Chinese".

Fig. 4. VQGAN+CLIP image generation with different prompts. From left to right: with the prompt "word in Chinese", with the prompt "word in Hebrew", with the prompt "word in Hindi".

in Chinese" (see Figure 3). Furthermore, some images imitate ink lettering on craft paper, which reminds us of Chinese calligraphy, a form of writing and art. That indicates a cultural bias similar to the example with Stable Diffusion and the Arabic language.

GANs do not show such cultural conditioning, while they tend to imitate specific shapes of different scripts (for example, separated Chinese characters with rounded angles, squared Hebrew lettering, and a few words in Hindi with an upper horizontal line known as shirorekhā); they also capture the imaging of the writing direction in different languages, although it is not explicit in their outputs (see Figure 4).

Figure 5 illustrates the text encoding used in the models. We tokenized our prompts with CLIP. CLIP tokenization is based on the byte pair encoding. We can see that the words "text" and "word" were not separated

```
SAMPLE:  text in Arabic
BPE:     text   in   A ra bic</w>
ENDODED: [4160, 530, 320, 1735, 10675, 34308, 342, 285]
DECODED: text in a ra bic </ w >

SAMPLE:  word in Russian
BPE:     word   in   R u ssian</w>
ENDODED: [2653, 530, 337, 340, 31218, 34308, 342, 285]
DECODED: word in r u ssian </ w >

SAMPLE:  caption in English
BPE:     cap tion   in   E ng lish</w>
ENDODED: [3938, 740, 530, 324, 5215, 2354, 34308, 342, 285]
DECODED: cap tion in e ng lish </ w >
```

Fig. 5. Examples of prompts tokenization with CLIP.

while fed to the models. However, in the names of the languages the model separated suffixes that form adjectives. Knowing that text-to-image models distinguish adjectives that encode names of the languages, we assume that this could increase the chances of recognizing such features as lettering shape and language cultural context and thus leading to a precise language distinction shown in our study.

The code and illustration for this subsection are provided in the folder "feature_extraction" at
`https://github.com/vifirsanova/text-to-image-explainable`.

§4. Results

We used the results of conducted experiments to identify specific features of the generated images that indicate how text-to-image models perceive natural language text. First, we found that while being fed with the prompt "text in LANGUAGE", diffusion models show a repetitiveness in their outputs similar to generative language models such as LSTM [18] or GPT [27]. Because recursive patterns form natural languages and behind generative models lies the probability distribution over big data, repetitiveness is a common issue in language modeling. Figure 6 shows some examples of repetitive generation occurring in text-to-image synthesis. We

Fig. 6. Examples of repetitive generation. From left to right DALL-E 2 outputs for the following prompts: “text in Korean”, “text in Hebrew”, “text in Yiddish”, “text in Japanese”.

can see that a different architecture, i.e. a diffusion model, while being prompted to generate natural language text, shows the same behaviour.

Figure 6 illustrates different scenarios of generative repetitiveness in diffusion models: the model generates one pattern, i.e. a sequence of different symbols, several times; the model repeats one character several times in a row; the model produces the same or confusingly similar symbol in different positions. All three scenarios are not typical for human-created text in natural language, although similar behaviour is common for generative language models.

Next, we found that diffusion models tend to generate text in English, even if the prompt did not contain the task of producing Latin characters. Figure 7 shows that such prompts as “text in Japanese” or “text in Greek” might be recognized by the model as “text containing the word “Japanese” (“Greek”) in English” because the prompt itself is in English. Figure 7 shows some attempts of DALL-E 2 to generate the words “Japan” or “Japanese” and “Greek” in English. It seems that the model has learned such basics of the English language as definite and indefinite articles and even captured some common syllables (e.g., -in-, -sis-, -ad-), which might be caused by the byte pair encoding provided by CLIP embeddings.

Figures 8 and 9 show cultural bias in text-to-image models. We found that diffusion models associate different languages with specific written sources or public places. Figure 8 presents different types of written sources generated by Stable Diffusion and DALL-E 2 in association with various languages. For example, the models often produced images of 20th-century newspapers for the Yiddish language. We assume that because

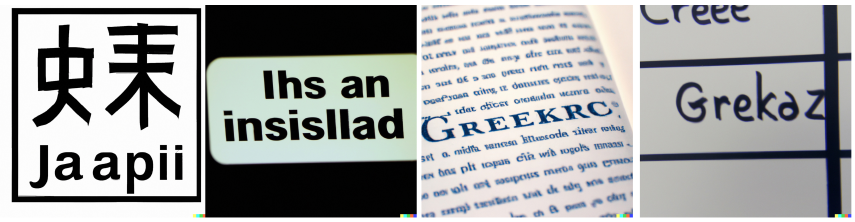

Fig. 7. Examples of human-readable samples in English. From left to right DALL-E 2 outputs for the following prompts: “text in Japanese”, “text in English”, “text in Greek”, “text in Greek”.

the model generated page layouts, specific shades of yellow and fonts typical for printed newspapers, and the pages contained photographs of people in old-fashioned clothes. The model produced pictures reminding us of the Quran for Arabic, calligraphy for Chinese, and the scrabble game for English. The model created a notebook and pencil for the Russian language. The setup in the picture resembles a type of school workbook popular in the Russian schooling system. Each example is strongly associated with stereotypes of different cultures and indicates an explicit bias in text-to-image models.

Figure 9 shows another type of bias that appeared only in Stable Diffusion, which might be specific to the dataset LAION-5B used to train that model. Along with generating texts for our prompts, the model output images of different locations, such as temples or city streets. It seems that the name of the language is encoded similarly to the name of the country associated with that language due to the byte pair encoding. The images may show stereotypical scenery for the prompted country, such as ancient temples for Greece, or some textual items immersed in the city landscape, such as a signboard in a Russian city (probably from the times of the USSR).

§5. Discussion

In this section, we formulate the main takeaways from our study. So how do text-to-image models perceive the natural languages of the world?

Fig. 8. Examples of imitation of written sources. From left to right, Stable Diffusion outputs for the following prompts: “text in Yiddish”, “text in Arabic”. Next, from left to right, DALL-E 2 outputs for the following prompts: “text in Chinese”, “text in English”, “text in Russian”, “text in Greek”.

Fig. 9. Examples of landscape generation. From left to right, Stable Diffusion outputs for the following prompts: “text in Greek”, “text in Korean”, “text in Russian”, “text in Thai”.

We found that both diffusion-based models and GANs can reproduce different scripts. Apart from encoding the information about the visual representation of languages, those models scan the cultural context. They are undoubtedly biased by cultural stereotypes related to different languages, which manifests itself in the generation of written sources or even public places that are often associated with countries of origin for these languages.

Text-to-image models have the potential to produce text in natural languages. We can see that texts in English produced by DALL-E 2 are already human-readable; the model begins to fix common combinations of letters and even such basic linguistic units as articles and can repeat the prompt (for example, the model explicitly generates the word "Greek" for the prompt "text in Greek"). We expect to observe improved natural language processing capabilities from the next generations of text-to-image models. That increases the risks of applying such models in producing fakes, fraud, and other types of cybercrime.

According to the results of our study, we can detect a false image containing generated text by excessive repetitiveness, explicit cultural bias, and artefacts in generating text in small fonts. The quality of the text generation by text-to-image models allows us to easily detect fake text at this point, of course. However, considering the rapid introduction of new architectures, it is essential that we can learn to detect such signals.

§6. Conclusion

In this study, we have presented an in-depth analysis of the ability to generate text in natural language in different text-to-image models. We used DALL-E 2, Stable Diffusion, and VQGAN+CLIP architectures to generate 144 images with the following prompts: "text in LANGUAGE", "word in LANGUAGE", and "caption in LANGUAGE". We produced these prompts using methods of prompt engineering. The prompt "text in LANGUAGE" was our baseline, while the other two were our backups in case the models could not deal with the task.

We explored the extracted features in the generated images using DAAM mapping and CLIP tokenization. We used this knowledge to conduct semiotic and linguistic analysis of the study material (144 generated images). We registered cultural bias in the picture generation that manifested in imaging written sources and public places. We discovered that text-to-image models are acute to typological generalizations when it comes to reproducing certain scripts, however, they struggle with such limitations

as repetitiveness that occurs usually in generative language models (LSTM, GPT family, etc.). Our hypothesis that text-to-image models are culturally biased and conditioned by linguistic generalizations is thus proved.

We assume that our conclusions can be used in fake detection and prompt engineering. The study contributes to explainable AI development. The supplementary materials for the study are open and are available at `https://github.com/vifirsanova/text-to-image-explainable`.

References

1. BarneyFlames, *Twitter*, `https://twitter.com/BarneyFlames/status/1531736708903051265`, 2023, Last accessed 12 Mar 2023.
2. S. R. Borgwaldt, T. Joyce, *Typology of writing systems*, vol. 51, John Benjamins Publishing, 2013.
3. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., *Language models are few-shot learners.* — Advances in Neural Information Processing Systems **33** (2020), 1877–1901.
4. K. Clark, U. Khandelwal, O. Levy, C. D. Manning, *What does bert look at? an analysis of bert's attention*, arXiv preprint arXiv:1906.04341 (2019).
5. Craiyon, *Craiyon*, `https://www.craiyon.com/`, 2023, Last accessed 15 Mar 2023.
6. Common Crawl, *Statistics of common crawl monthly archives*, `https://commoncrawl.github.io/cc-crawl-statistics/plots/languages`, 2023, Last accessed 14 Mar 2023.
7. K. Crowson, *Vqgan+clip tutorial*, `https://colab.research.google.com/github/justinjohn0306/VQGAN-CLIP/blob/main/VQGAN%2BCLIP(Updated).ipynb`, 2023, Last accessed 17 Mar 2023.
8. K. Crowson, S. Biderman, D. Kornis, D. Stander, E. Hallahan, L. Castricato, E. Raff, *Vqgan-clip: Open domain image generation and editing with natural language guidance*, European Conference on Computer Vision, Springer, 2022, pp. 88–105.
9. P. T. Daniels, W. Bright, *The world's writing systems*, Oxford University Press, 1996.
10. G. Daras, A. G. Dimakis, *Discovering the hidden vocabulary of dalle-2*, arXiv preprint arXiv:2206.00169 (2022).
11. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805 (2018).
12. P. Esser, R. Rombach, B. Ommer, *Taming transformers for high-resolution image synthesis*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 12873–12883.
13. Ethnologue, *Languages of the world*, `https://www.ethnologue.com/`, 2023, Last accessed 14 Mar 2023.
14. A. I. Forever, *Kandinsky 2.0*, `https://github.com/ai-forever/Kandinsky-2.0`, 2023, Last accessed 12 Mar 2023.
15. I. Gelb, *A study of writing*, University of Chicago Press, Chicago, 1952.

16. H. He, X. Chen, C. Wang, J. Liu, B. Du, D. Tao, Yu Qiao, *Diff-font: Diffusion model for robust one-shot font generation*, arXiv preprint arXiv:2212.05895 (2022).
17. J. Hewitt, C. D. Manning, *A structural probe for finding syntax in word representations*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4129–4138.
18. S. Hochreiter, J. Schmidhuber, *Long short-term memory.* — Neural Computation **9**, No. 8 (1997), 1735–1780.
19. HuggingFace, *Diffusers*, `https://github.com/huggingface/diffusers`, 2023, Last accessed 17 Mar 2023.
20. V, Liu, L. B. Chilton, *Design guidelines for prompt engineering text-to-image generative models*, Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022, pp. 1–23.
21. P. Lyu, X. Bai, C. Yao, Z. Zhu, T. Huang, W. Liu, *Auto-encoder guided gan for chinese calligraphy synthesis*, 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, IEEE, 2017, pp. 1095–1100.
22. OpenAI, *Chatgpt*, `https://openai.com/blog/chatgpt`, 2023, Last accessed 12 Mar 2023.
23. OpenAI, *Dall-e*, `https://labs.openai.com/`, 2023, Last accessed 15 Mar 2023.
24. L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., *Training language models to follow instructions with human feedback.* — Advances in Neural Information Processing Systems **35** (2022), 27730–27744.
25. V. A. Plungyan, *Modern linguistic typology*, Herald of the Russian Academy of Sciences **81** (2011), 101–113.
26. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., *Learning transferable visual models from natural language supervision*, International conference on machine learning, PMLR, 2021, pp. 8748–8763.
27. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., *Language models are unsupervised multitask learners.* — OpenAI Blog **1**, no. 8, 9 (2019).
28. A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen, *Hierarchical text-conditional image generation with clip latents*, arXiv preprint arXiv:2204.06125 **1** (2022), no. 2, 3.
29. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, *High-resolution image synthesis with latent diffusion models*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 10684–10695.
30. C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. G. Lopes, B. K. Ayan, T. Salimans, et al., *Photorealistic text-to-image diffusion models with deep language understanding.* — Advances in Neural Information Processing Systems **35** (2022), 36479–36494.
31. V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. Le Scao, A. Raja, et al., *Multitask prompted training enables zero-shot task generalization*, arXiv preprint arXiv:2110.08207 (2021).

32. R. Tang, A. Pandey, Z. Jiang, G. Yang, K. Kumar, J. Lin, F. Ture, *What the daam: Interpreting stable diffusion using cross attention*, arXiv preprint arXiv:2210.04885 (2022).
33. W3Techs, *Usage statistics of content languages for websites*, `https://w3techs.com/technologies/overview/content\_language`, 2023, Last accessed 14 Mar 2023.
34. H. Zhang, J. Y. Koh, J. Baldridge, H. Lee, Y. Yang, *Cross-modal contrastive learning for text-to-image generation*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 833–842.

St. Petersburg State University,
St. Petersburg, Russia
E-mail: `st085687@student.spbu.ru`

Поступило 6 сентября 2023 г.