IEEE International Conference on Artificial Intelligence & Knowledge Engineering (AIKE 2023)

Prompt Recommendation for AI Art

Kapil Wanaskar
Software (Machine Learning) Co-op Intern at Product Security Team,
Intuitive, Sunnyvale

An armchair in the shape of an avocado

Hyelim Yang,
Kapil Wanaskar,
Harshika Shrivastava,
Shahbaz Mansahia,
Shakshi Richhariya,
Prof. Magdalini Eirinaki

Introduction

- Dall-E and Midjourney have reshaped digital art creation.
- Picture this: "**a white cat sitting on a Disney-themed window**" and witnessing art spring to life. But what if we could make it even easier to choose the right "idea" for art ?
- Optimizing art concept recommendations is our focus.
- Our data: user IDs, ideas, and images, without ratings.
- We use graph analysis and user clustering for refined suggestions.
- Goal: Advance AI art prompt recommendations.

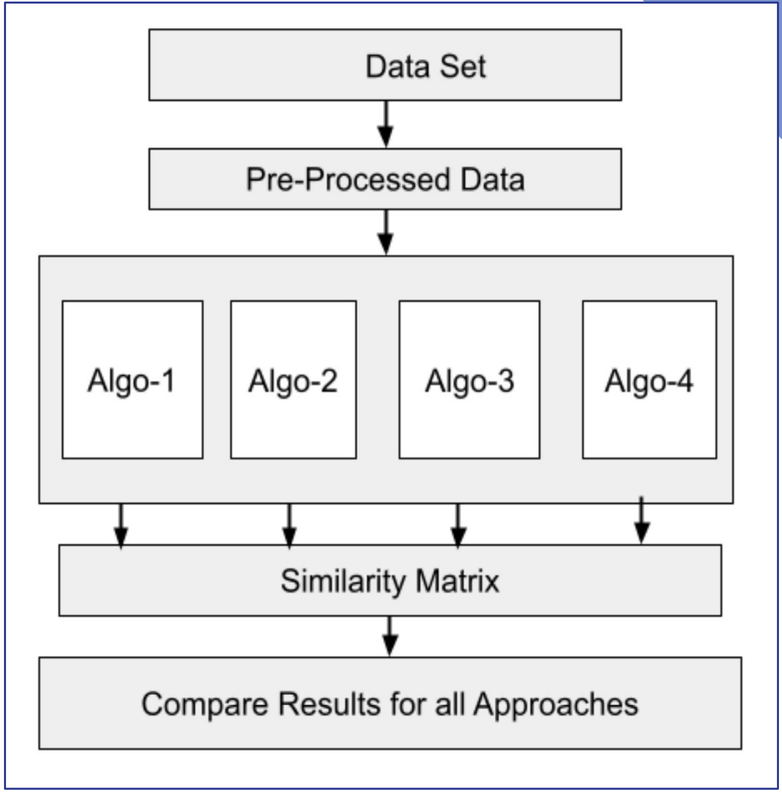
Challenges and Objective

- Absence of explicit ratings and traditional metrics.
- Limited data availability: user ID, prompt, image.
- Addressing the gap in AI art creation research.
- Focus on prompt recommendation.
- System creation for prompt improvement.

SJSU

System Design & Methodologies

- Architecture designed for flexibility.
- Four main algorithms: Text embeddings, Ensemble models, Hybrid approach, Object detection.
- Role of text embeddings and image feature extraction.
- Object detection using YOLOv4.
- Versatility and adaptability of the system.



SJSU

Dataset & Preprocessing

- Dataset: Subset of Diffusion DB, generated in August 2022.
- Features: Image, Prompt, User name.
- Preprocessing: Stopwords removal, Emoji removal, Stemming.
- Example after preprocessing: ***"the pope's hard rock band, with instruments"*** to ***"pope hard rock band instrument."***
- Importance of preprocessing in refining data.

- Prompt: Figure 2
  After Preprocessing: 'pick fox face wolf face rocket full moon cityscap'

Fig. 2. Image prompt example

SJSU

Algorithm-1

Goal: Given a prompt, recommend similar prompts in the dataset without any state-of-art embedding methods

TfidVectorizer

- 6553 features
ex) zuckerberg, abel, etc
- No min-max normalization

Calculate Cosine similarity $= \mathrm{Sim}_{\mathrm{TfidVec}}$

CounterVectorizer

- 6553 features
- Same features with TfidVectorizer
- min-max normalization

Calculate Cosine similarity $= \mathrm{Sim}_{\mathrm{CounterVec}}$

$$0.5(\mathrm{Sim}_{\mathrm{TfidVec}}) + 0.5(\mathrm{Sim}_{\mathrm{CounterVec}})$$

Pros:

- Recognizable features → better interpretability

Cons:

- Limitation on capturing hidden meaning in a prompt
- Ignore the context of a prompt

Algorithm-2 (Ensemble Models)

- Deep dive into embeddings: Word2Vec, GloVe, BERT, CLIP.
- Representation of words and ideas.
- Differences in suggesting art prompts.
- Importance of the right tool for art suggestions.
- Insights from experimental evaluation.

Word2Vec

- Text embedding
- 300 features
- Capture the semantic relationships between words.

Glove

- Text embedding
- 300 features
- Combines the strengths of both global matrix factorization and local context window methods

Bert

- Text embedding
- 768 features
- Context-specific embeddings
- Known to be better capture the nuances in the text data

CLIP

- Image embedding
- Trained on a variety of (image, text) pairs
- Known to be efficiently learns visual concepts from natural language supervision

Algorithm-3

Goal: Leveraging both text and image embeddings

Prompt-to-prompt recommendation

- Input: Text embeddings from TFIDF + Image embeddings from Inception V3.
- Cosine similarity to find similar prompts based on query prompt ID.

+

Recommendation based on image

- Input: Image embeddings from ResNet and Inception v3
- Cosine similarity to find similar prompts based on query image.

+

User-based recommendation

- Forming Clusters using K-means algorithm of users with similar artistic interests (based on prompt tried by user)
-Recommending prompt based on similar user in cluster.

Pros:

- Able to recommend relevant prompts and images. AI generation model can give different images for same prompt, good to use (Prompt+images)

Cons:

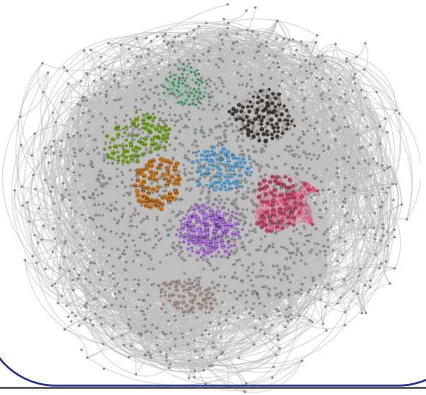
- High computational resources required.

SJSU

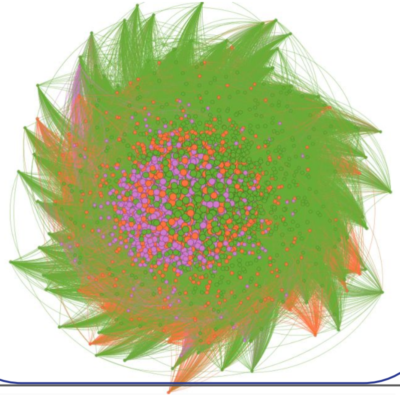
Evaluation: Comparison between algorithms

Graph after the community detection

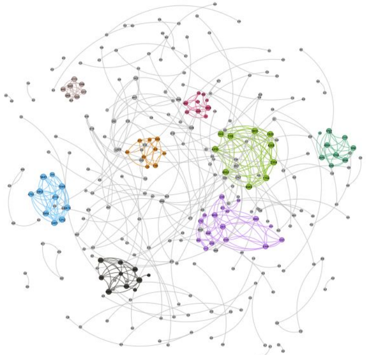
Algorithm 1



Algorithm 2



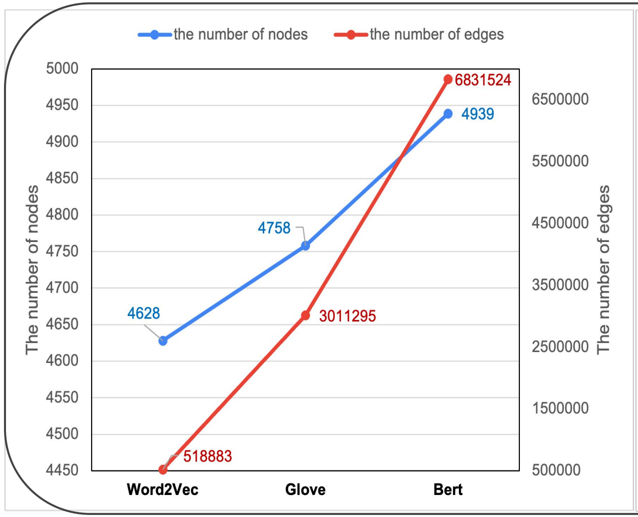
Algorithm 3



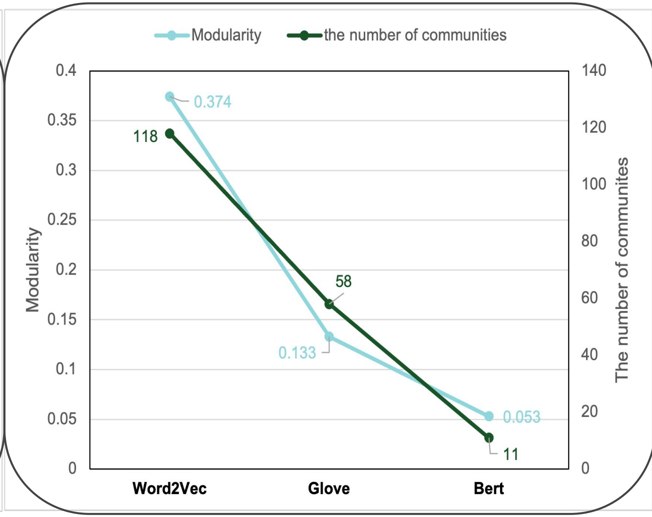
	Algorithm 1	Algorithm 2	Algorithm 3
# of nodes	3587	4940	282
# of edges	29020	7,255,901	391
Avg weighted degree	13.53	2,204.88	2.24
Graph density	0.005	0.595	0.01
Modularity	0.943	0.07	0.941
The number of communities	607	4	88

Evaluation: Comparison between algorithms

Compare graph properties



Compare community detection results

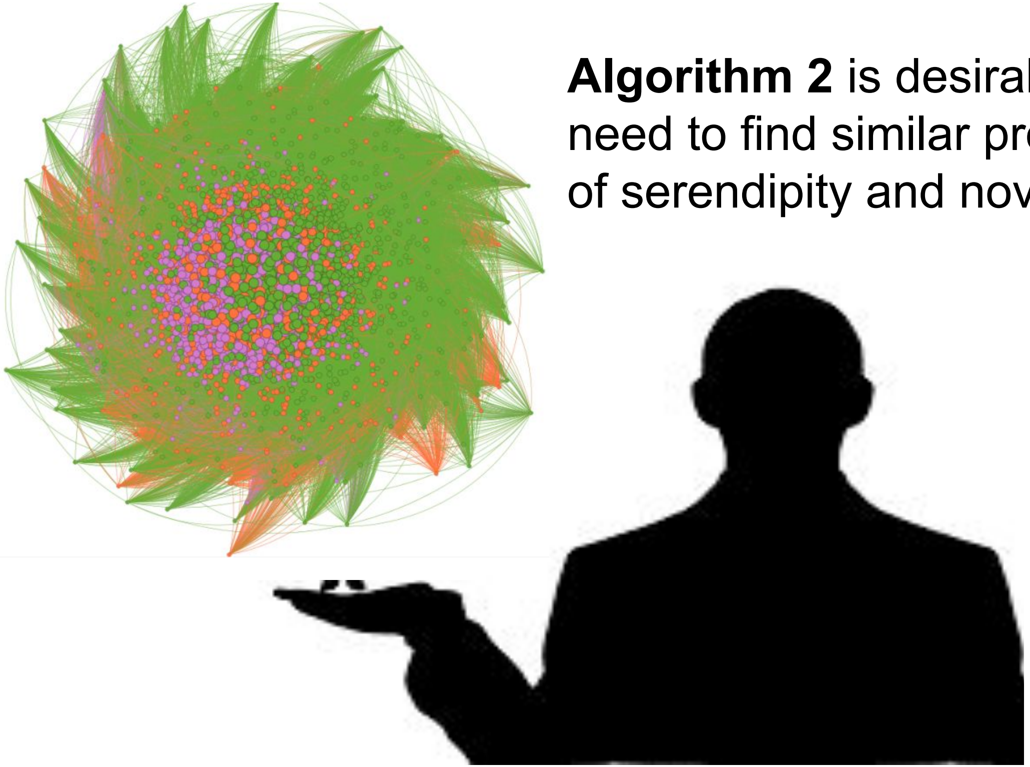


- Increasing in embedding complexity leads:
  - More nodes and edges
  - Higher average weighted degree
  - Increased graph density
  - Modularity decreases
  - Number of detected communities decreases

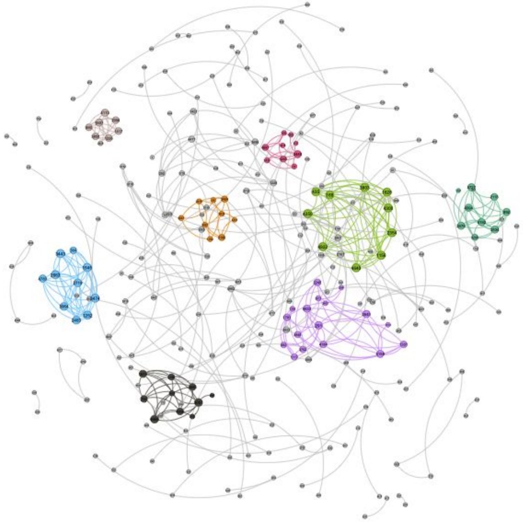
- BERT embedding can recognizes subtle nuances in prompts → finds similar prompts for 99% of prompts with similarity threshold of 0.7

SJSU

Discussion

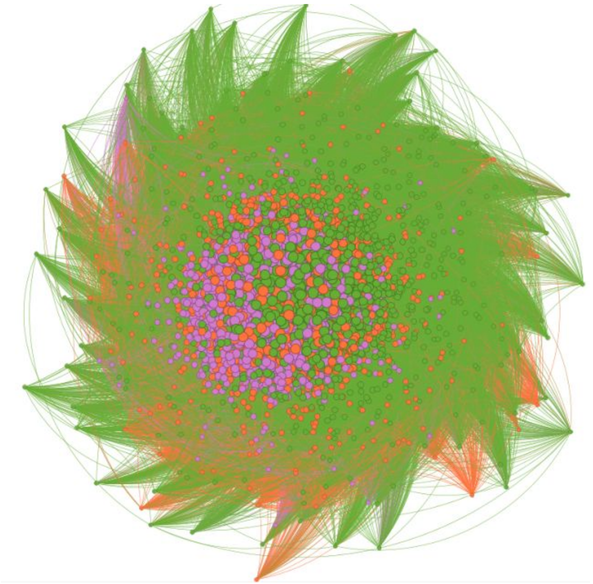


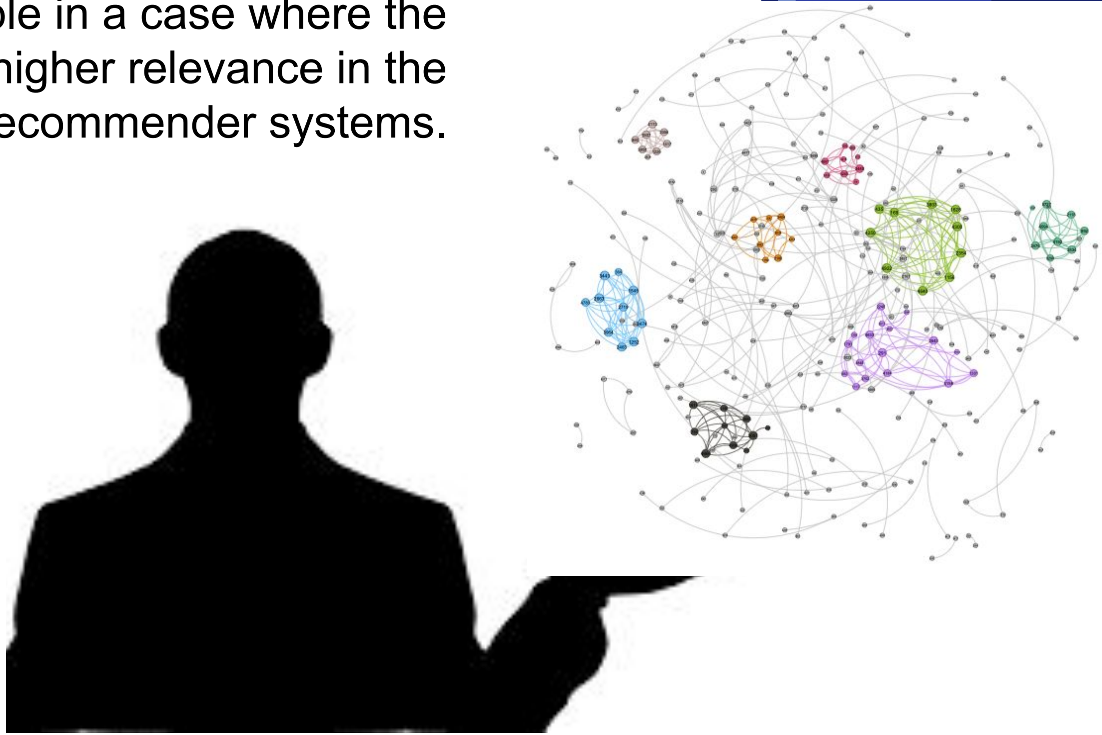
Algorithm 2 is desirable in a case where we need to find similar prompts efficiently in terms of serendipity and novelty



Discussion

Algorithm 3 is desirable in a case where the desired outcome is higher relevance in the recommender systems.





SJSU

Kapil Wanaskar
Software (Machine Learning) Co-op Intern,
at Product Security Team,
Intuitive, Sunnyvale

Thank you

Any
Questions?

SJSU