Multi-Modal Graph Inductive Learning with CLIP Embeddings

David Roth, Adi Srikanth, Tanya Naheta, Andre Chen for Zillow Group

Problem Definition

- Generate a Knowledge Graph that can store and relate multiple modes of data (text, numeric, image, etc)
- Learn embeddings that utilize graph structure by using proximal data to more accurately represent data
- Use link prediction to evaluate graph-based modal embeddings
- Future Application Areas: multi-modal search, hero image ranking, recommender systems

Contributions

- Construction of domain-specific Multi-Modal Knowledge Graph (MMKG) using Zillow listing images, keyword tags, and scenes initialized using pre-trained CLIP embeddings
- Training and evaluation of GraphSAGE, an inductive graph convolutional network (GCN) learning approach, over MMKG
- Development and evaluation of multiple frameworks for link prediction on previously unseen nodes

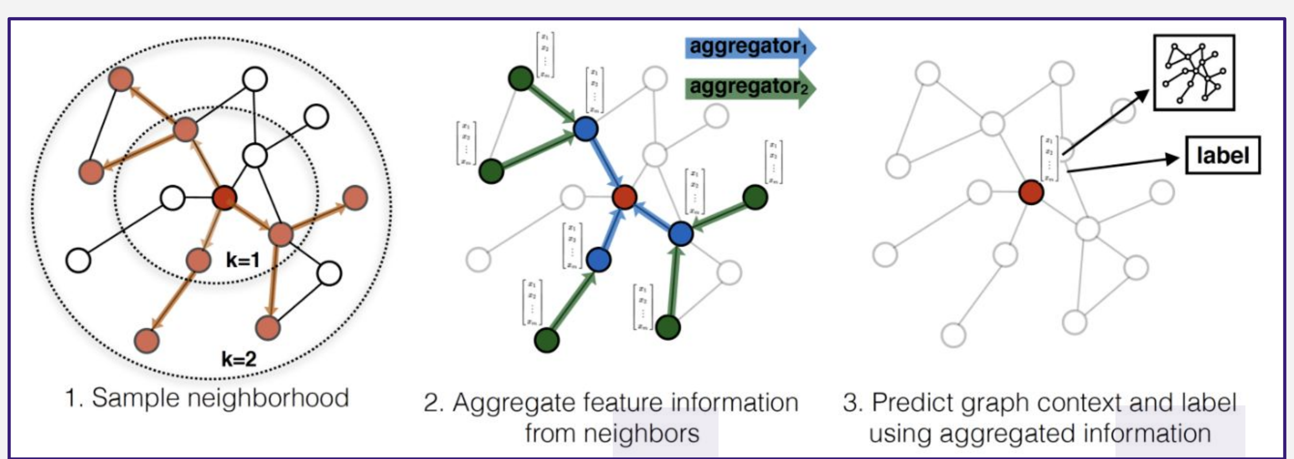
Approach and Implementation

Data

- COCO - Open-source, labeled images used for development
- Zillow Data - weakly labeled & human verified datasets (embedded images & keywords via CLIP)
- CLIP - Open AI method for generating pre-trained image and text embeddings

Model: GraphSAGE

- GraphSAGE is an inductive GCN approach that uses a contrastive objective to learn functions for updating node embeddings



GraphSAGE Contrastive Objective

$$J_{\mathcal{G}}(\mathbf{z}_u) = -\log\left(\sigma(\mathbf{z}_u^\top \mathbf{z}_v)\right) - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log\left(\sigma(-\mathbf{z}_u^\top \mathbf{z}_{v_n})\right)$$

Approach

Build MMKG using CLIP image, keyword, and scene embeddings → Partition MMKG into train and holdout subgraphs → Define graph sampling strategy and model parameters → Train GraphSAGE Model on training subgraph → Evaluate GraphSAGE model via keyword link prediction on test subgraph

Graph Sampling Strategy:

1:1 negative sampling ratio with uniform negative sampling, limit message passing to three neighbors per GCN layer during training

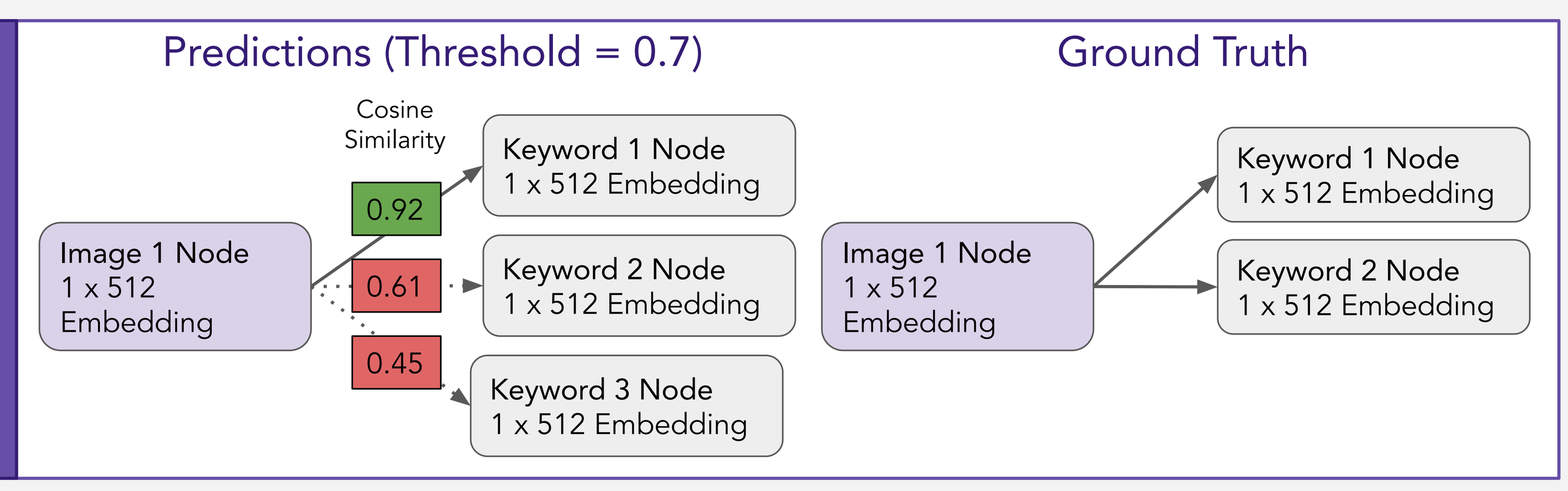
Link Prediction Experiments for New Nodes

To simulate link prediction on previously unseen nodes, we tested three different approaches of reconnecting test subgraph nodes to the train subgraph prior to node updates and link prediction:

Experiment 1: Cosine Similarity	Experiment 2: Scene Connection	Experiment 3: Self-Loops
Reconnect test graph nodes by making connections based on node similarity (cosine similarity)	Reconnect test graph nodes by making connections between nodes that share the same scene attribute	Induce "self-edges" that connect each test graph node back to itself

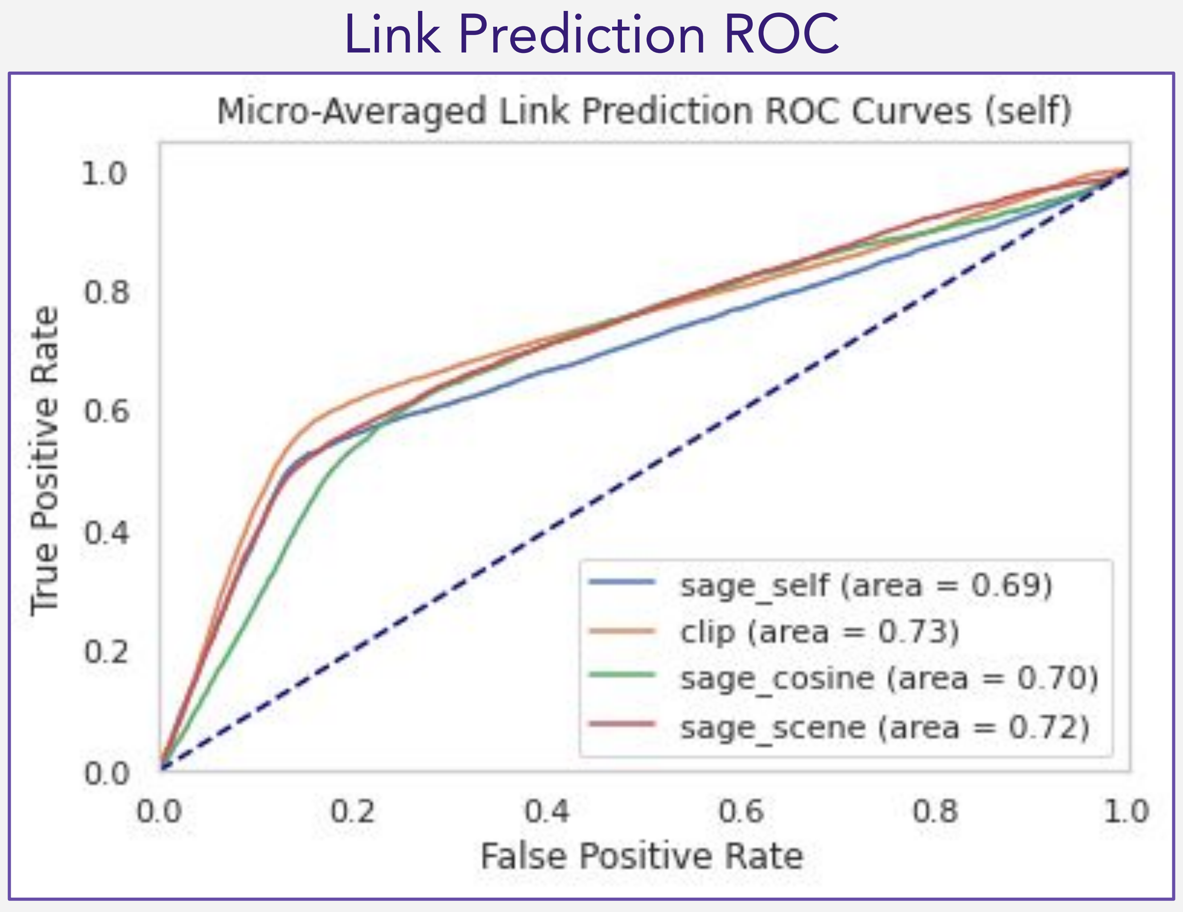
Link Prediction

We conduct cosine similarity based link prediction using our original CLIP embeddings and our updated embeddings to measure the quality of our updated embeddings



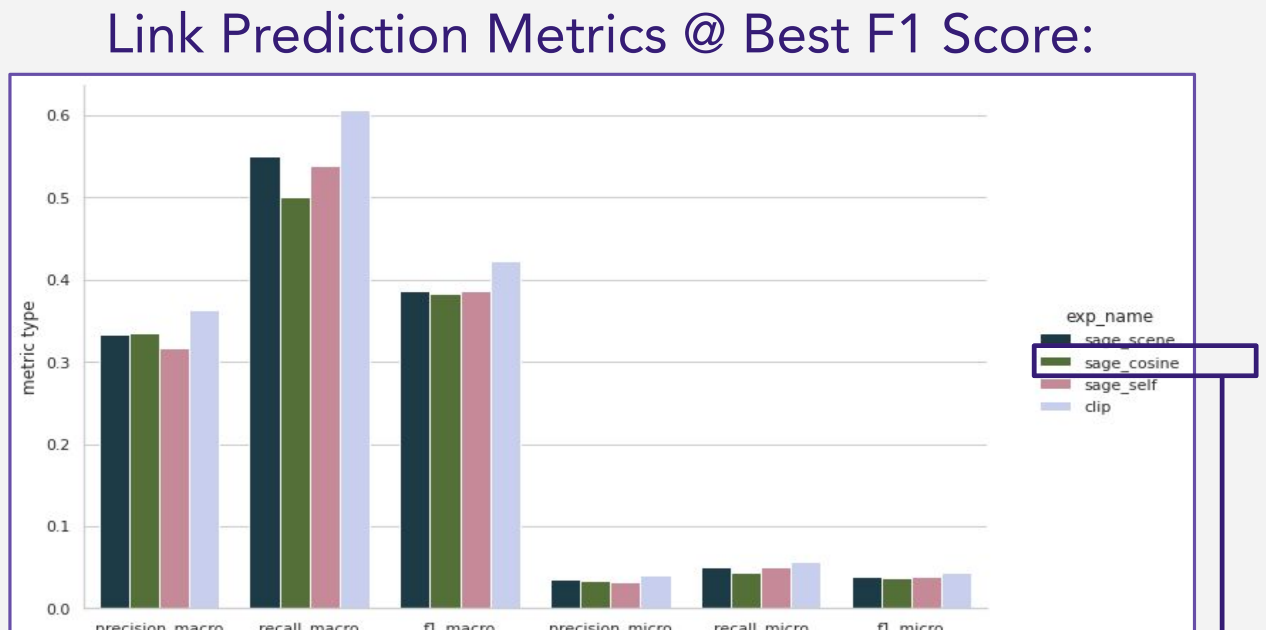
Results and Discussion

Link Prediction ROC



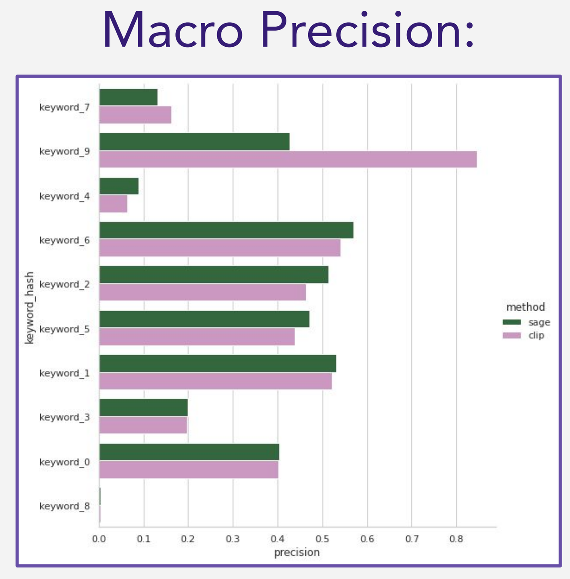
Experiment Name	Micro-Averaged ROC AUC	Macro-Averaged ROC AUC
clip	0.726	0.749
sage_self	0.718	0.701
sage_cosine	0.697	0.726
sage_scene	0.688	0.724

Link Prediction Metrics @ Best F1 Score:

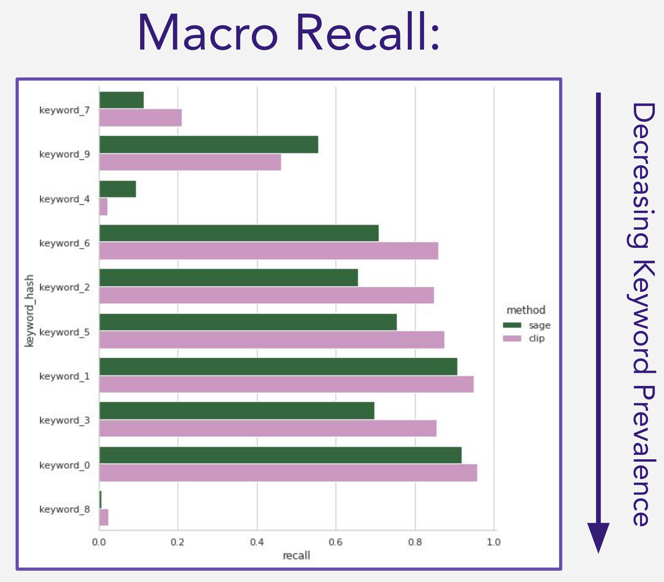


Metrics Breakdown by Keyword (Cosine Method)

Macro Precision:



Macro Recall:



Key Findings

- Inducing self-loops on new nodes shows comparable performance to scene and cosine-based graph connection methods
- Under current settings, CLIP embeddings slightly outperform GraphSAGE embeddings on link prediction
- Prediction performance appears worse for keywords that occur more frequently across images
- GraphSAGE outputs highly polarized, sparse embeddings compared to CLIP

Future Work

- Regularization - Introduce additional objectives to enforce smoothness and reduce information loss during training.
- Edge Weights - Use Graph Attention or heuristic methods to vary contributions of neighboring nodes to the final node representation.

Acknowledgements

- Zillow Team - Shourabh Rawat, Jyoti Prakash Maheswari, Raghav Jajodia, Supriya Anand
- NYU Faculty Advisor - Najoung Kim