More Knowledge, Less Bias: Unbiasing Scene Graph Generation with Explicit Ontological Adjustment (Supplementary Material)

Zhanwen Chen[1] Saed Rezayi[2] Sheng Li[1]
[1]School of Data Science, University of Virginia
{pct4et, shengli}@virginia.edu
[2]School of Computing, University of Georgia
saedr@uga.edu

1. Algorithm for Knowledge Graph Extension

In this section, we describe the detailed procedures for extending the commonsense knowledge graph (KG). We start with the set of 150 object labels from Visual Genome [1] as our initial concept entities. After training a BLINK model, we feed the object labels to the BLINK model to obtain a new set of entities linked to the Wikidata entities with their edges. Because each entity on Wikidata has multiple neighbors, we randomly select one neighbor in our construction to prevent the graph from becoming unwieldy. The output from our Wikidata knowledge graph consists of triplets whose format is similar to that of ConceptNet (*e.g.*, ⟨“flower”, “/r/PartOf”, “plant_part”⟩). In our final Wikidata edges, the head entities are strictly on-scene, but the tail entities, such as “flower_part” can be off-scene. The above entity linking process is shown between lines 1 and 4 of Algorithm 1 below.

With our combined ConceptNet [3] and Wikidata [4] edges, we now construct the commonsense knowledge graph. Because ConceptNet [3] uses WordNet [2] synset prefixes as IDs, we first manually link each off-scene tail entity to WordNet [2] and use its WordNet prefix to query its associated ConceptNet entities [3]. We then reconstruct the ConceptNet and Wikidata knowledge bases by initializing and populating the combined edge matrix with the union of the two edge sets. We use their combined and sorted ConceptNet IDs as the index. There are four types of directed edges in the graph: entities to entities (e2e), predicates to predicates (p2p), entities to predicates (e2p), and predicate to entities (p2e). Furthermore, as is the case in GB-Net [6], we consider some ConceptNet edge types such as “/r/PartOf” as single-directional edges and others like “/r/RelatedTo” as two-directional edges. For the single-directional edges, each direction counts as an edge type in the commonsense knowledge graph. However, a two-directional edge only counts as one. For example, the two directions of “/r/PartOf” are edge types 2 and 3, respectively. In contrast, the two directions of “/r/RelatedTo” share the edge type 4. The product is the combined commonsense knowledge graph. The complete algorithm for this extension is shown in Algorithm 1.

With our commonsense knowledge graph, we add the statistical components to the graph. Because the original statistical priors by Xu *et al*. [5] are computed on the 150 on-scene entities, we must remap them to include the additional off-scene entities with trivial values. These statistics include conditional probabilities between objects, subjects, and predicates as well as covariance of entities and predicates. The key to remapping the statistics is an index remapping from the original entity order to the new extended one. We set the off-scene entity statistics to 0 because they do not appear in the ground truth at all. Lastly, we concatenate the matrix components into four subgraphs [6]. This full remapping process is shown in Algorithm 2.

Algorithm 1 Our algorithm to extend GB-Net's Knowledge Graph

function EXTEND-EBNET(SE, SP, edges)
  $CE_{on} = SE$;
  $CP = SP$;
  $E' = BLINK(CE_{on})$;
  $CE_{wiki} = \{e_{tail} | (e_{head}, t, e_{tail}) \in E'\}$;
  $CE_{ext} = CE \cup CE_{wiki}$;
  $CE_{off} = CE_{ext} - SE$;
  $E = \emptyset$;
  for $c \in CE_{ext}$ **do**
    wordnet_id = synset[c];
    $id_{cn}[c]$ = query_conceptnet(wordnet_id);
  Manually edit edges;
  $N_{CE} = |CE_{ext}| + 1$; ▷ With the background class
  $N_{CP} = |CP| + 1$; ▷ With the background class
  $E_e 2e$ = **0** ($|T_{ent2ent}| \times N_{CE} \times N_{CE}$);
  $E_p 2p$ = **0** ($|T_{pred2pred}| \times N_{CP} \times N_{CP}$);
  $E_e 2p$ = **0** ($|T_{ent2pred}| \times N_{CE} \times N_{CP}$);
  $E_p 2e$ = **0** ($|T_{pred2ent}| \times N_{CP} \times N_{CE}$);
  for $(e_{head}, t, e_{tail}) \in E \cup E'$ **do**
    if $\{e_{head}, e_{tail}\} \subseteq CE_{ext}$ **then**
      $E^{ext}_{e2e}[t, id_{CE}[id_{cn}[e_{head}]], id_{CE}[id_{cn}[e_{tail}]]] = 1$;
      $E^{ext}_{e2e}[t, id_{CE}[id_{cn}[e_{tail}]], id_{CE}[id_{cn}[e_{head}]]] = 1$;
    else if $\{e_{head}, e_{tail}\} \subseteq CP$ **then**
      $E^{ext}_{p2p}[t, id_{CP}[id_{cn}[e_{head}]], id_{CP}[id_{cn}[e_{tail}]]] = 1$;
      $E^{ext}_{p2p}[t, id_{CP}[id_{cn}[e_{tail}]], id_{CP}[id_{cn}[e_{head}]]] = 1$;
    else
      $E^{ext}_{e2p}[t, id_{CE}[id_{cn}[e_{head}]], id_{CP}[id_{cn}[e_{tail}]]] = 1$;
      $E^{ext}_{p2e}[t, id_{CP}[id_{cn}[e_{tail}]], id_{CE}[id_{cn}[e_{head}]]] = 1$;
  return

$$\begin{Bmatrix} E^{ext}_{e2e} & E^{ext}_{p2p} \\ E^{ext}_{e2p} & E^{ext}_{p2e} \end{Bmatrix}$$

Algorithm 2 Our algorithm to remap the statistical subgraphs and add them to EB-Net

function ADD-STATS(CE, CE_{ext}, KG, stats)
  E^{ext}_{e2e}, E^{ext}_{p2p}, E^{ext}_{e2p}, $E^{ext}_{p2e} = KG$;
  Build index mapping im from CE_{on} to CE_{ext}; ▷ Both are alphabetically sorted
  $index_{off} = index(CE_{ext}) - im$;
  stats =

$$\left\{ \begin{matrix} \{p(obj|subj), p(subj|obj), \\ p(subj|pred), p(pred|subj), \\ p(obj|pred), p(pred|obj), \\ cov(SE), cov(SP), word\_sim\} \end{matrix} \right\}$$

  ;
  for $s \in stats$ **do**
    $s^{ext}[im] = s$;
    $s^{ext}[index_{off}] = 0$;
  return

$$\left\{ \begin{matrix} E^{ext}_{e2e} \oplus p(obj|subj)^{ext} \oplus \\ p(subj|obj)^{ext} \oplus cov(SE)^{ext} \oplus word\_sim^{ext} \\ E^{ext}_{p2p} \oplus cov(SP)^{ext} \\ E^{ext}_{e2p} \oplus p(subj|pred)^{ext} \oplus p(obj|pred)^{ext} \\ E^{ext}_{p2e} \oplus p(pred|subj)^{ext}, p(pred|obj)^{ext} \end{matrix} \right\}$$

References

[1] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[2] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, nov 1995.

[3] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge, 2017.

[4] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014.

[5] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *Proceedings of the European conference on computer vision (ECCV)*, August 2020.