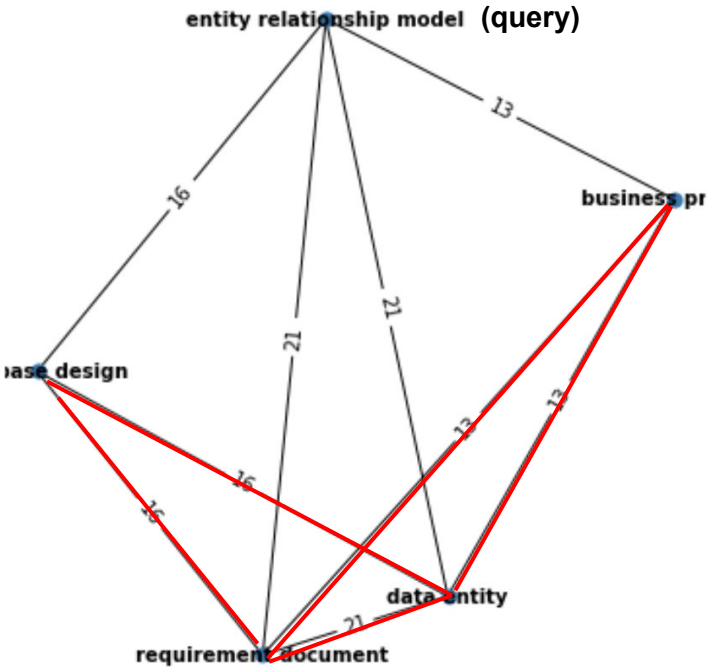
Community Ranking

University of Illinois at Urbana-Champaign
Bruno Seo
sbseo2@illinois.edu

Motivation 1

We observed that successful related keywords form a strong community

=> Reason to leverage ***community*** search



Motivation 2

Naive sorting method is effective, but it fails to break a tie

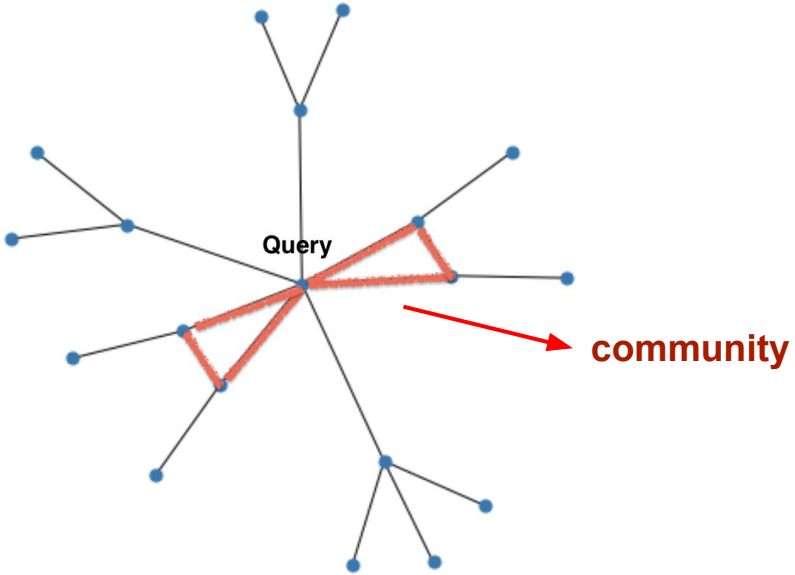
=> Needs a better ***ranking*** function

```
query = 'computer science'
naive_sorting(query, top=20)
```

```
[('knowledge based software engineering', {'weight': 10.812489107402115}),
 ('neural immune', {'weight': 10.812489107402115}),
 ('grammar school', {'weight': 10.812489107402115}),
 ('knowledge transmission', {'weight': 10.812489107402115}),
 ('dna testing', {'weight': 10.812489107402115}),
 ('sleepwalking', {'weight': 10.812489107402115}),
 ('goal structure', {'weight': 10.812489107402115}),
 ('mead conway revolution', {'weight': 10.812489107402115}),
 ('scansion', {'weight': 10.812489107402115}),
 ('computer science curriculum', {'weight': 10.812489107402115}),
 ('organizational capacity', {'weight': 10.812489107402115}),
 ('sound visualization', {'weight': 10.812489107402115}),
 ('web intelligence', {'weight': 10.812489107402115}),
 ('geneva conventions', {'weight': 10.812489107402115}),
 ('morphophonology', {'weight': 10.812489107402115}),
 ('technology based interventions', {'weight': 10.812489107402115}),
 ('empty sum', {'weight': 10.812489107402115}),
 ('string to string correction problem', {'weight': 10.812489107402115}),
 ('sentience', {'weight': 10.812489107402115}),
 ('cultural artifact', {'weight': 10.812489107402115})]
```

Problem Statement

We define finding related keyword task as a ***Community Ranking Problem.***



Naive Ranking Function

Naive ranking function sorts by PMI

$$f(q,t) = W_{qt}$$

q : query

t : target word (connected node)

w : weight of edge between query and target

Method: Community Search

We assign extra weight if the target word belongs to a community

$$f(q,t) = W_{qt} + \alpha$$

α : constant

Method: Query Expansion

We assign another extra weight if the target word includes an unigram from query.
ex) Query: Computer Science, Target word: Science => m = 1

$$f(q,t) = W_{qt} + \alpha + m \cdot \beta_{qexp}$$

$$\beta_{qexp} : constant$$

$$m = n(\forall \text{unigrams in } q \cap t)$$

Methods: Vector Similarity

Finally, we add a semantic similarity between query and the target word

$$f(q,t) = W_{qt} + \alpha + m \cdot \beta_{qexp} + sim(q,t)$$

$$sim(q,t) : \text{cosine similarity}$$

Result

Note that there are no more ties

```
1  naive_sorting('data mining')
```

```
[('temporal data mining', {'weight': 10.953221088512144}),
 ('career decision', {'weight': 10.953221088512144}),
 ('data mining algorithm', {'weight': 10.953221088512144}),
 ('big data mining', {'weight': 10.953221088512144}),
 ('semantic query optimization', {'weight': 10.953221088512144})]
```

```
1  community_ranking('data mining')
```

```
[('big data mining', {'weight': 23.87301034142753}),
 ('relational data mining', {'weight': 23.862132587823364}),
 ('data mining algorithm', {'weight': 23.858869131229305}),
 ('geospatial intelligence', {'weight': 21.461073157053107}),
 ('operational intelligence', {'weight': 21.449719617681655})]
```

Result

```
1  naive_sorting('computer science')
```

```
[('knowledge based software engineering', {'weight': 10.812489107402115}),
 ('neural immune', {'weight': 10.812489107402115}),
 ('grammar school', {'weight': 10.812489107402115}),
 ('knowledge transmission', {'weight': 10.812489107402115}),
 ('dna testing', {'weight': 10.812489107402115})]
```

```
1  community_ranking('computer science')
```

```
[('computer science curriculum', {'weight': 13.747134303977846}),
 ('knowledge based software engineering', {'weight': 11.54591597654004}),
 ('technology based interventions', {'weight': 11.398283186586255}),
 ('technology forecasting', {'weight': 11.379199966269923}),
 ('web intelligence', {'weight': 11.378285783747277})]
```

Result

```
1  naive_sorting('programming language')
```

```
[('distributed logic', {'weight': 11.629423006271736}),
 ('universal protein resource', {'weight': 11.629423006271736}),
 ('watchdog timer', {'weight': 11.629423006271736}),
 ('new executable', {'weight': 11.629423006271736}),
 ('java programming language', {'weight': 11.629423006271736})]
```

```
1  community_ranking('programming language')
```

```
[('general purpose programming language', {'weight': 24.53182924512161}),
 ('binary xml', {'weight': 22.043037522097485}),
 ('system programming language', {'weight': 14.559581441785244}),
 ('java programming language', {'weight': 14.547467590515701}),
 ('synchronous programming language', {'weight': 14.531592996511337})]
```

Result

```
1  naive_sorting('education')
```

```
[('technology education', {'weight': 10.715671654233072}),
 ('higher education', {'weight': 10.715671654233072}),
 ('primary education', {'weight': 10.715671654233072}),
 ('cross site request forgery', {'weight': 10.715671654233072}),
 ('formal education', {'weight': 10.715671654233072})]
```

```
1  community_ranking('education')
```

```
/home/ec2-user/.local/lib/python3.7/site-packages/ipykernel
```

```
[('nonformal education', {'weight': 22.715671657092052}),
 ('education curriculum', {'weight': 22.65666897195555}),
 ('vocational education', {'weight': 22.634977428700996}),
 ('continuing education', {'weight': 22.627558600642512}),
 ('teacher education', {'weight': 22.617686715173576})]
```

Reference

M. Gupta, J. Gao, X. Yan, H. Cam and J. Han, "Top-K interesting subgraph discovery in information networks," 2014 IEEE 30th International Conference on Data Engineering, Chicago, IL, 2014, pp. 820-831, doi: 10.1109/ICDE.2014.6816703.

Guisado-Gámez, J., Dominguez-Sal, D., & Larriba-Pey, J. L. (2013). Massive query expansion by exploiting graph knowledge bases. arXiv preprint arXiv:1310.5698.