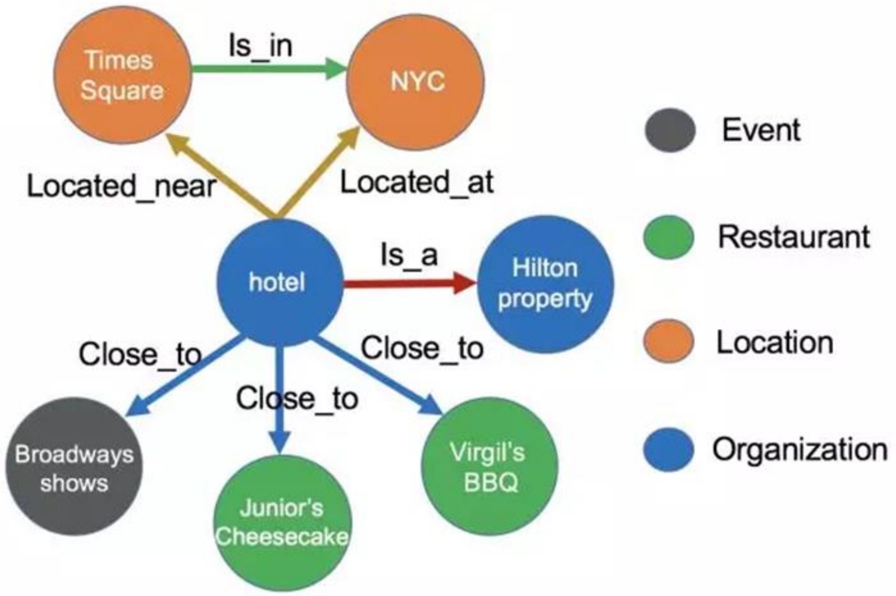
Query Knowledge Graph via Ranking Given Examples

gsc

This hotel is my favorite Hilton property in NYC! It is located right on 42nd street near Times Square in New York, it is close to all subways, Broadways shows and next to great restaurants like Junior's Cheesecake, Virgil's BBQ



MOTIVATION:

Inside knowledge graph hides the information that we need for further purpose. Though there is a query language SPARQL served as a query approach, sometimes people just can't organize the inner logic which covers the demand in a clear way but giving examples.
So a query by example approach is needed.

EXAMPLE: potential crime or criminal

所有周星驰参演过的电影

```
PREFIX : <http://www.kgdemo.com#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX vocab: <http://localhost:2020/resource/vocab/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX map: <http://localhost:2020/resource/#>
PREFIX db: <http://localhost:2020/resource/>

SELECT ?n WHERE {
  ?s rdf:type :Person.
  ?s :personName '周星驰'.
  ?s :hasActedIn ?o.
  ?o :movieTitle ?n
}
```

```
n

"功夫"
"琉璃樽"
"英雄本色"
"少林足球"
"西游记第壹佰零壹回之月光宝盒"
"长江七号"
"西游记大结局之仙履奇缘"
"建国大业"
"审死官"
"龙在天涯"
"大内密探零零发"
```

Problem description:

In our model, we provide positive examples from the set of ground truth and negative examples outside.

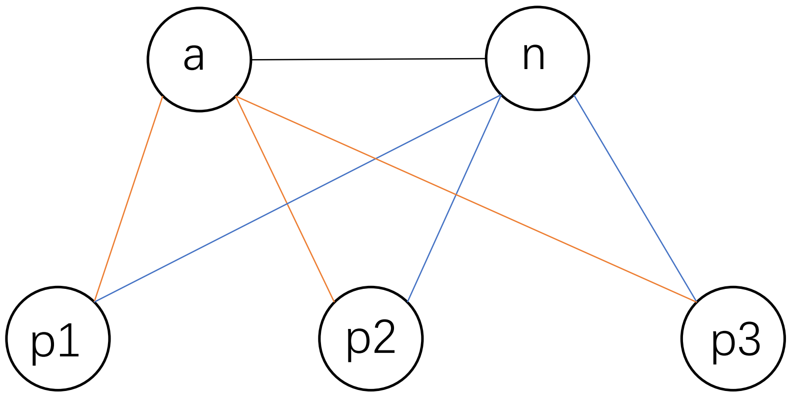
If we want to know the demand from the given examples, the attributes that the examples associated to should get more attention. We should first rank the attributes to get the ones that people really care about. Then query the knowledge base using these attributes.

Problem definition:

We organize Graph G which is under Marcov property using these examples and the attribute for ranking. For each attribute, we could build a Graph to calculate the score of it using an approach of MRF(marcov random field).

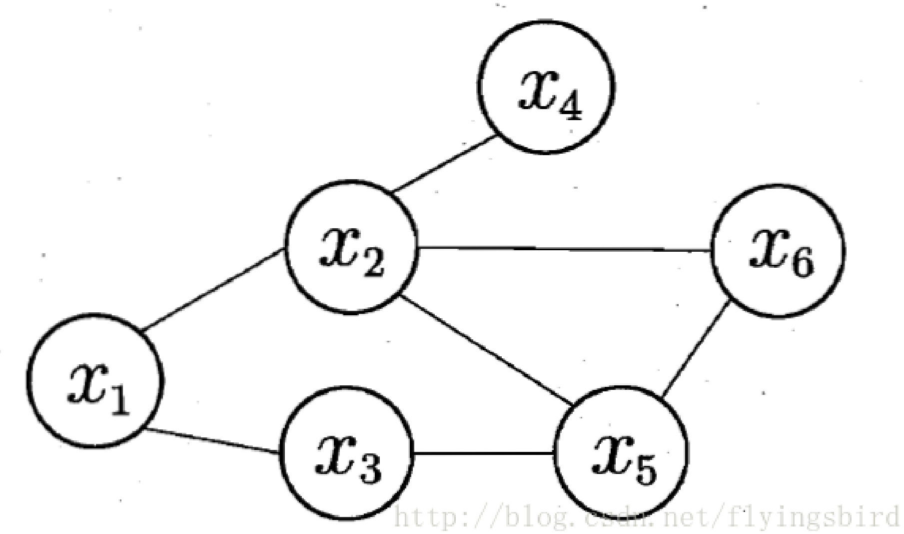
Different relation edge in Graph G:

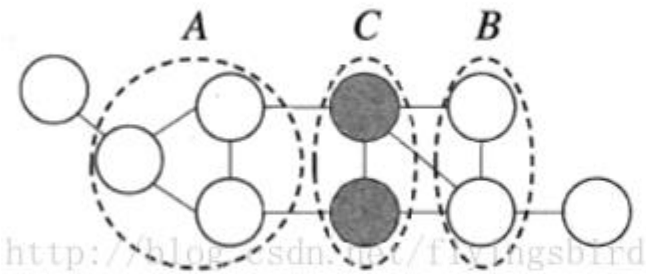
- attribute-positive example
- attribute-negative example
- negative example-positive example



Graph G

About MRF:



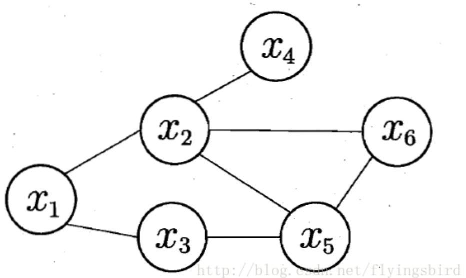


1）局部马尔可夫性（Local Markov Property）：给定某个变量的邻接变量，则该变量条件独立于其他变量。

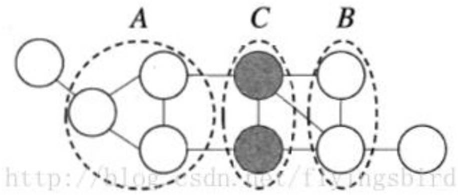
2）成对马尔可夫性（Pairwise Markov Property）：给定所有其它变量，两个非邻接变量条件独立。

About MRF:

$$P(x) = \frac{1}{Z} \prod_{Q \in \mathcal{C}} \psi_Q(\mathbf{x}_Q)$$

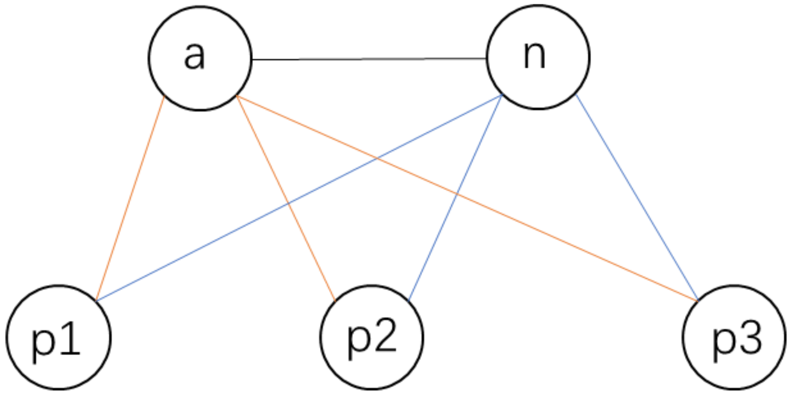


Q：图上的一个团
Z：规范化因子



With Graph G, how to design the potential function?

- The clique: should we go with the maximal clique or just a plain one; (maximal clique)
- The pattern: for edge with different vertex type, how to describe the need in a potential function; (two patterns)

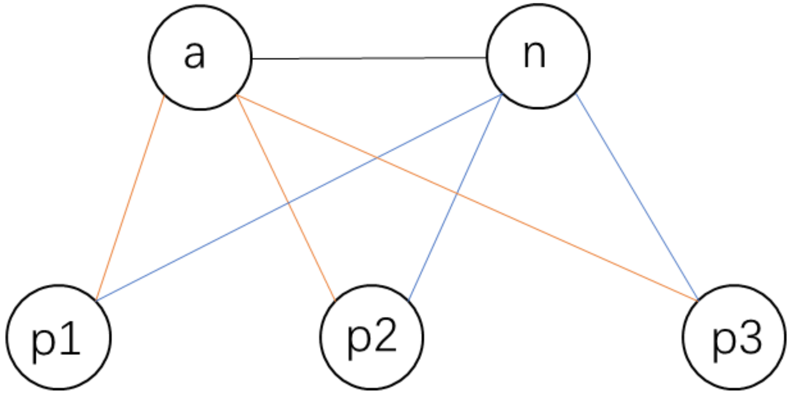


Graph G

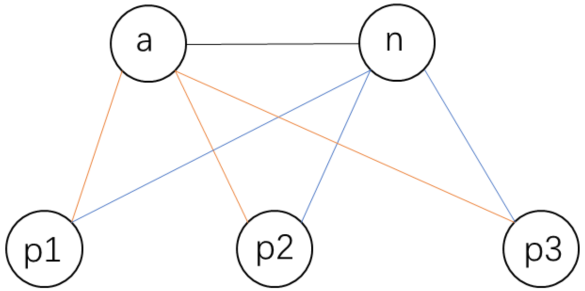
$$S_p = |\{< a, t >\in positive\_example\}|$$

$$S_n = |\{< a, t >\in negative\_example\}|$$

$$a : attribute, t : value$$



Graph G



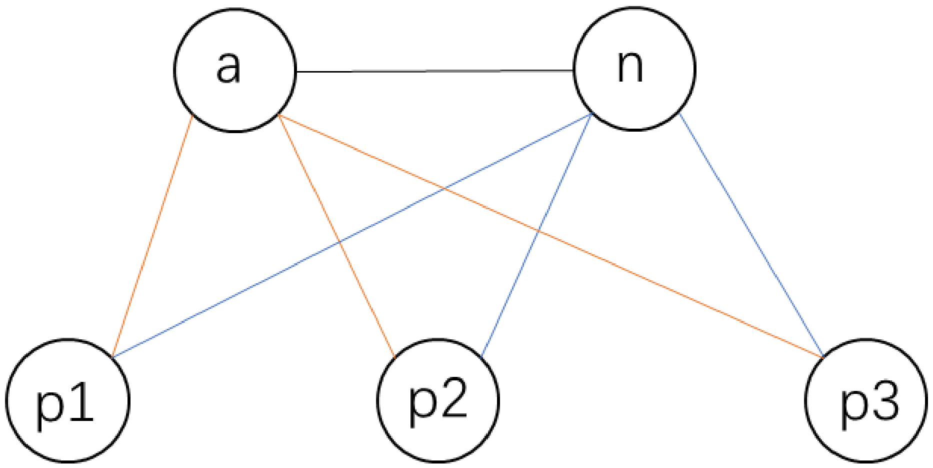
Graph G

$$f_1 = \frac{|S_p - S_n| \times |S_n - S_p|}{|S_p| \times |S_n|}$$

$$f_2 = -\frac{|\{< a, v >\in positive\_example\}|}{|S_p - S_n|} \times \frac{|E_{all}|}{|\{e \in E_{all}| < a, v >\in e\}|} \ldots\ldots (E_{all} = all\_the\_entities)$$

$$\Psi(a,p,n) = exp(\lambda_1 f_1 + \lambda_2 f_2)$$

$$p(a|E_p,E_n) = \frac{1}{Z} \prod_{Q_i \in C(G); p_i, n_i \in Q_i} \Psi(a,p_i,n_i)$$



Graph G

Data set: DBpedia collection v2

- `queries-v2.txt` : The set of 467 queries, where each line contains a query ID and query text pair.
- `queries-v2_stopped.txt` : The same queries, with stop patterns and punctuation marks removed.
- `qrels-v2.txt` : Relevance judgments in standard TREC format.
- `folds/` : Partitioning of queries for 5-fold cross validation. This is provided to make results directly comparable by using the same partitioning for supervised approaches. A separate file is provided for each query subset; if training is done over the set of all queries, use the `all_queries.json` file.
- `annotator_agreements.tsv` : Inter-annotator agreements between crowd workers (and expert annotators, if applicable) for each query-entity pair. The agreement scores are computed according to the Fleiss' kappa index (i.e., Eq (3) of its Wikipedia article). This information may be used as a proxy for query difficulty.

```
INEX_LD-20120431	bicycle benefits health
INEX_LD-20120432	bicycle benefits environment
INEX_LD-20120511	female rock singers
INEX_LD-20120512	south korean girl groups
INEX_LD-20120521	electronic music genres
INEX_LD-20120522	digital music notation formats
INEX_LD-20120531	music conferences
INEX_LD-20120532	intellectual property rights lobby
INEX_LD-2009022	Szechwan dish food cuisine
INEX_LD-2009039	roman architecture
INEX_LD-2009053	finland car industry manufacturer saab sisu
INEX_LD-2009061	france second world war normandy
INEX_LD-2009062	social network group selection
INEX_LD-2009063	D-Day normandy invasion
INEX_LD-2009074	web ranking scoring algorithm
INEX_LD-2009096	Eiffel
INEX_LD-2009111	europe solar power facility
INEX_LD-2009115	virtual museums
```

```
INEX_LD-2009022 Q0	<dbpedia:Afghan_cuisine>	0
INEX_LD-2009022 Q0	<dbpedia:Akan_cuisine>	0
INEX_LD-2009022 Q0	<dbpedia:Ambuyat>	0
INEX_LD-2009022 Q0	<dbpedia:American_Chinese_cuisine>	1
INEX_LD-2009022 Q0	<dbpedia:Ants_climbing_a_tree>	2
INEX_LD-2009022 Q0	<dbpedia:Baingan_bharta>	1
INEX_LD-2009022 Q0	<dbpedia:Bamischijf>	0
INEX_LD-2009022 Q0	<dbpedia:Black_cardamom>	0
INEX_LD-2009022 Q0	<dbpedia:Brazilian_cuisine>	0
INEX_LD-2009022 Q0	<dbpedia:British_cuisine>	0
INEX_LD-2009022 Q0	<dbpedia:Caribbean_cuisine>	0
INEX_LD-2009022 Q0	<dbpedia:Cayenne_pepper>	0
INEX_LD-2009022 Q0	<dbpedia:Cellophane_noodles>	1
INEX_LD-2009022 Q0	<dbpedia:Ceviche>	0
INEX_LD-2009022 Q0	<dbpedia:Chana_masala>	0
INEX_LD-2009022 Q0	<dbpedia:Chen_Kenichi>	1
INEX_LD-2009022 Q0	<dbpedia:Chen_Kenmin>	1
INEX_LD-2009022 Q0	<dbpedia:Chicago-style_pizza>	0
INEX_LD-2009022 Q0	<dbpedia:Chicken_(food)>	0
INEX_LD-2009022 Q0	<dbpedia:Chifle>	0
INEX_LD-2009022 Q0	<dbpedia:Chili_oil>	2
INEX_LD-2009022 Q0	<dbpedia:Chinatown,_Los_Angeles>	0
INEX_LD-2009022 Q0	<dbpedia:Chinatown>	1
INEX_LD-2009022 Q0	<dbpedia:Chinese_cuisine>	2
INEX_LD-2009022 Q0	<dbpedia:Churumuri_(food)>	0
INEX_LD-2009022 Q0	<dbpedia:Cookbook>	0
INEX_LD-2009022 Q0	<dbpedia:Cooking>	0
INEX_LD-2009022 Q0	<dbpedia:Couscous>	0
INEX_LD-2009022 Q0	<dbpedia:Cuban_cuisine>	0
INEX_LD-2009022 Q0	<dbpedia:Cuisine>	0
INEX_LD-2009022 Q0	<dbpedia:Cuisine_of_Jharkhand>	0
INEX_LD-2009022 Q0	<dbpedia:Cuisine_of_the_Southern_United_States>	0
INEX_LD-2009022 Q0	<dbpedia:Cuisine_of_the_United_States>	0
INEX_LD-2009022 Q0	<dbpedia:Culture_of_the_Song_dynasty>	0
INEX_LD-2009022 Q0	<dbpedia:Curry>	0
```

Experiment:
To be continued