CWI Centrum Wiskunde & Informatica

Escuela Politécnica Superior

Better Contextual Suggestions from ClueWeb12

Using Domain Knowledge Inferred from The Open Web

Thaer Samar

Alejandro Bellogin and Arjen P. de Vries

Our Submission

- **Contextual Suggestion model:**
  - Find attractions in ClueWeb12
  - Generating user profiles
  - Similarity between candidate attractions and users
  - Rank suggestion per (user, context) pair
- **RQ:**

  can we improve the performance of the contextual suggestions by applying domain knowledge?
- **Approach:**
  - Filter collection using domain knowledge to create sub-collections
  - Apply same retrieval model to different sub-collections
  - Compare differences in effectiveness

Creating Sub-collections

- GeoFiltered sub-collection
  - Applying geographical filter
    - Exact mention of the given contexts

      format: {City, ST} e.g., Miami, FL
    - Exclude documents that mention multiple contexts

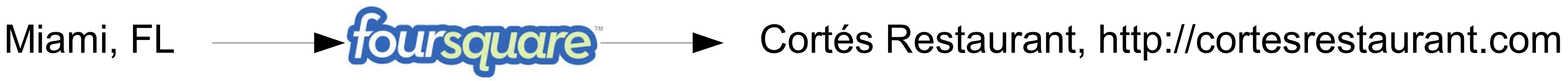
      e.g., a Wikipedia page about cities in Florida state

TouristFiltered sub-collection

- Applying domain knowledge extracted from the structure of the Open Web:
  - Domain Oriented
    - Manual list of tourist websites
      {yelp, tripadvisor, wikitravel, zagat, xpedia, orbitz, and travel.yahoo}
    - From ClueWeb12
      - extract any document whose host in the list **(TouristListFiltered)**
        e.g., http://www.zagat.com/miami
    - Expand TouristListFiltered
      - Extract outlinks
      - Search for outlinks in ClueWeb12 **(TouristOutlinksFiltered)**

TouristFiltered sub-collection

- Attraction Oriented
    - Use Foursquare API to get attractions for given contexts

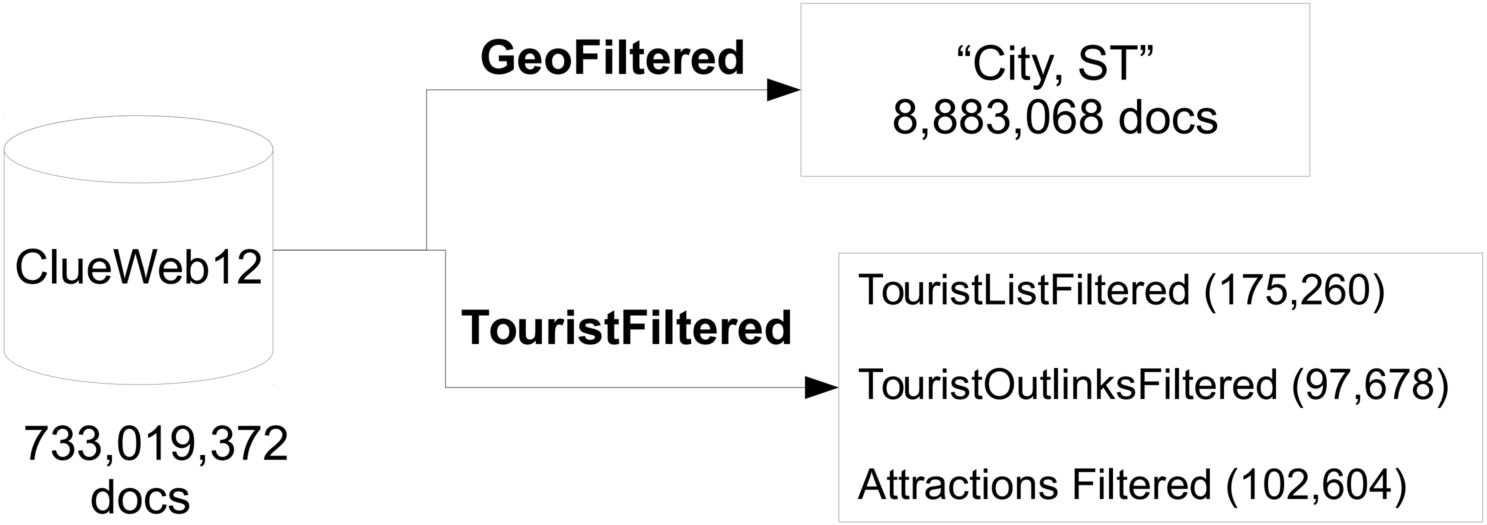


Miami, FL → Cortés Restaurant, http://cortesrestaurant.com

  - If URL is missing for the attraction, then use Google API query: "Cortés Restaurant Miami, FL"

  - For found attractions
    - Get host names of their URLs
    - From ClueWeb12 get any document whose host from the above **(AttractionFiltered)**

Sub-collections Summary



ClueWeb12

733,019,372 docs

GeoFiltered

“City, ST”
8,883,068 docs

TouristFiltered

TouristListFiltered (175,260)

TouristOutlinksFiltered (97,678)

Attractions Filtered (102,604)

Generating Users Profiles

- Aggregation of attractions descriptions
- Take into account ratings given by users
  - Build positive and negative profiles

Similarity

- Represent attractions and users in weighted VSM
  - Vector element <term, frequency>

- Cosine similarity

$$sim(u^+, d) = \cos(u^+, d) = \frac{\sum_i {u_i}^+ \cdot d_i}{\sqrt{\|u^+\|_2}\sqrt{\|d\|_2}} \tag{1}$$

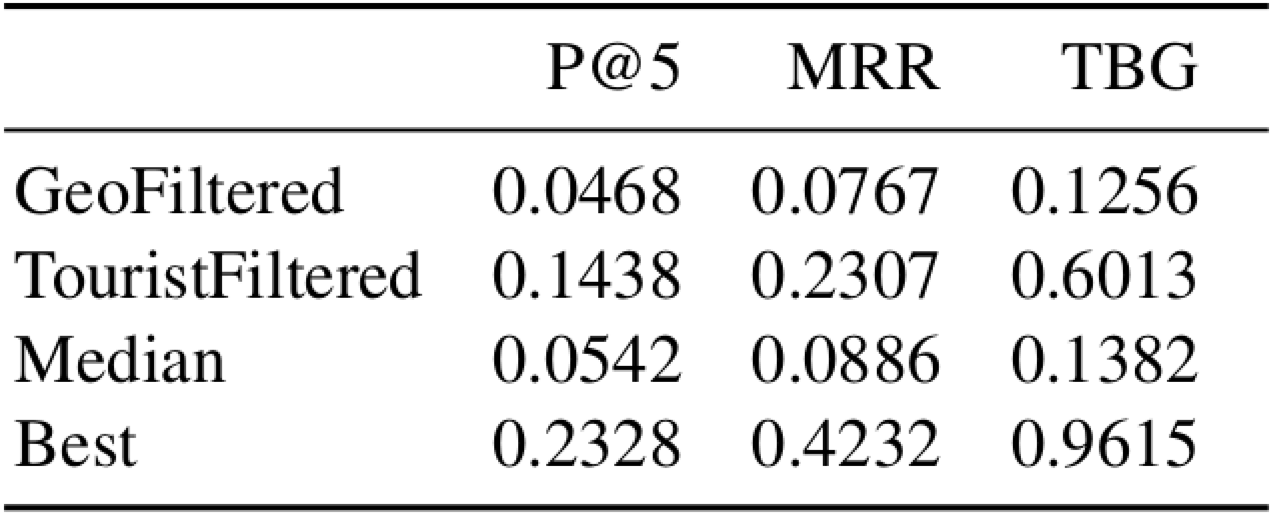
$$sim(u^-, d) = \cos(u^-, d) = \frac{\sum_i {u_i}^- \cdot d_i}{\sqrt{\|u^-\|_2}\sqrt{\|d\|_2}} \tag{2}$$

$$score = a \cdot sim(u^+, d) + b \cdot sim(u^-, d) \tag{3}$$

Ranked suggestions

- For each (user, context) pair
  - Rank suggestions based on similarity score
  - Generate titles to represent attraction:
    - Extract from <title> or <header> tags
  - Generate descriptions tailored to the user
    - Extract content of <description> tag
    - Break documents into sentences
      - rank sentences based on their similarity with the user
    - Concatenate until 512 bytes reached

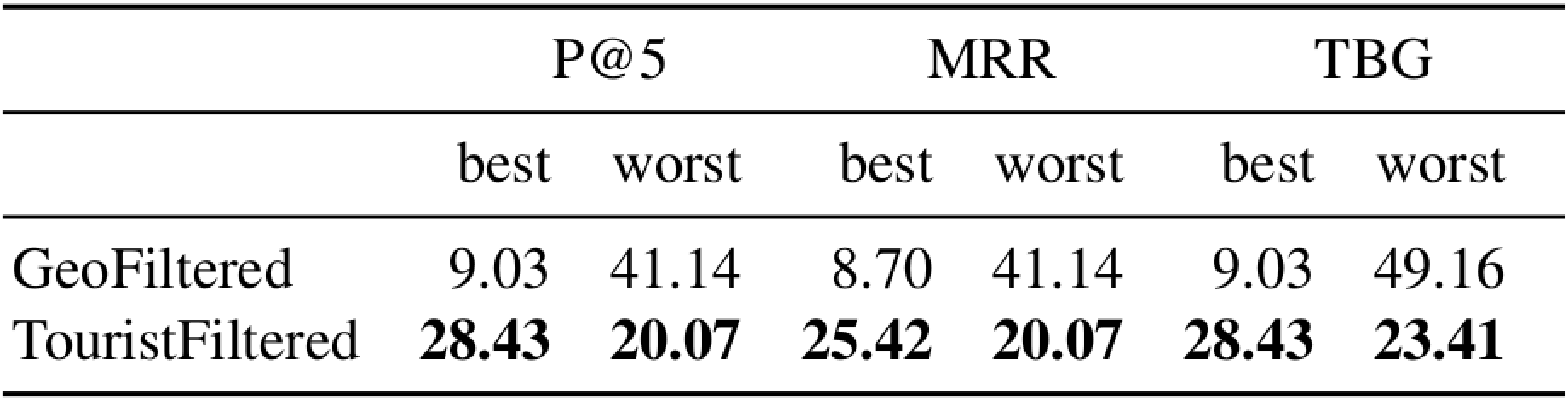
Results (General Performance)



	P@5	MRR	TBG
GeoFiltered	0.0468	0.0767	0.1256
TouristFiltered	0.1438	0.2307	0.6013
Median	0.0542	0.0886	0.1382
Best	0.2328	0.4232	0.9615

Analysis (General)

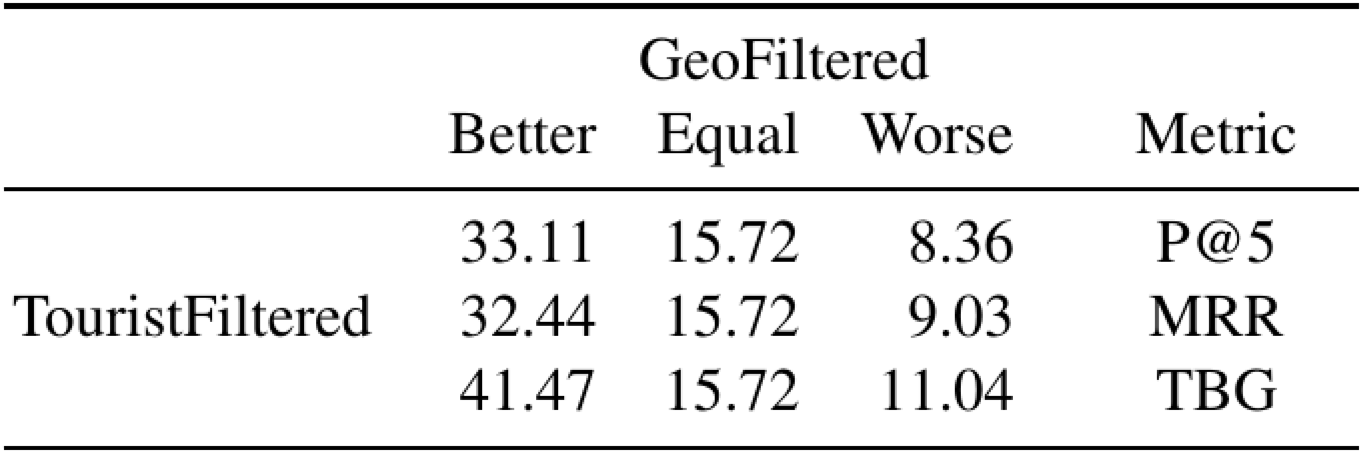
- Percentage of best and worst topics given by each run
- Exclude topics where best score=worst=0
- Compared with all runs based on ClueWeb12



	P@5		MRR		TBG	
	best	worst	best	worst	best	worst
GeoFiltered	9.03	41.14	8.70	41.14	9.03	49.16
TouristFiltered	**28.43**	**20.07**	**25.42**	**20.07**	**28.43**	**23.41**

Analysis (TouristFiltered vs. GeoFiltered)

- Compare our runs against each other
- Percentage of topics where **TouristFiltered** is better than equal to and worse than **GeoFiltered**
- In case of equality, ignore topics when best score is zero

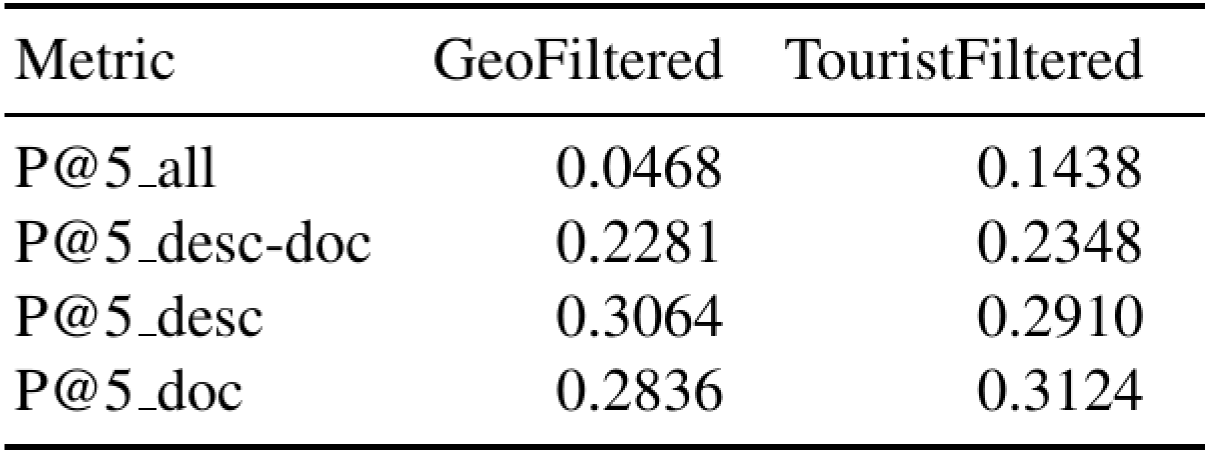


	GeoFiltered			
	Better	Equal	Worse	Metric
TouristFiltered	33.11	15.72	8.36	P@5
	32.44	15.72	9.03	MRR
	41.47	15.72	11.04	TBG

Analysis (decompose metrics dimensions)

- P@5 and MRR consider three dimensions of relevance
  - Geographical (geo), description (desc) and document (doc) relevance

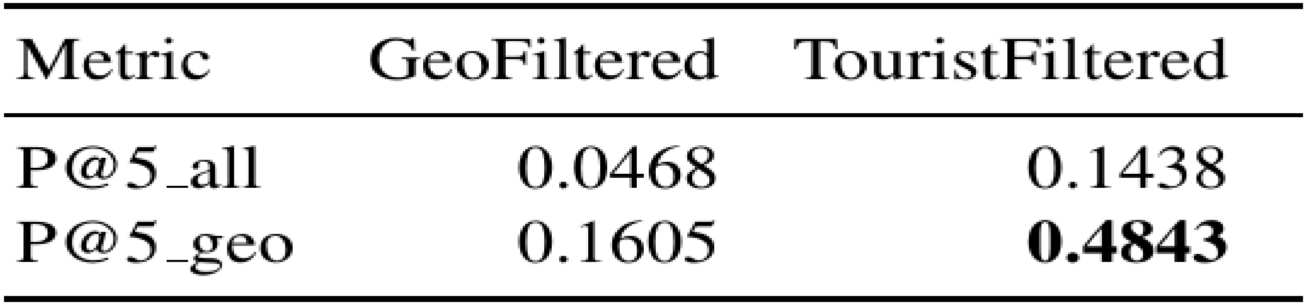
- Considering the desc and doc relevance
  - Two runs have similar effectiveness



Metric	GeoFiltered	TouristFiltered
P@5_all	0.0468	0.1438
P@5_desc-doc	0.2281	0.2348
P@5_desc	0.3064	0.2910
P@5_doc	0.2836	0.3124

Analysis (decompose metrics evaluation)

- Considering the geo aspect only
  - TouristFiltered is geographically appropriate

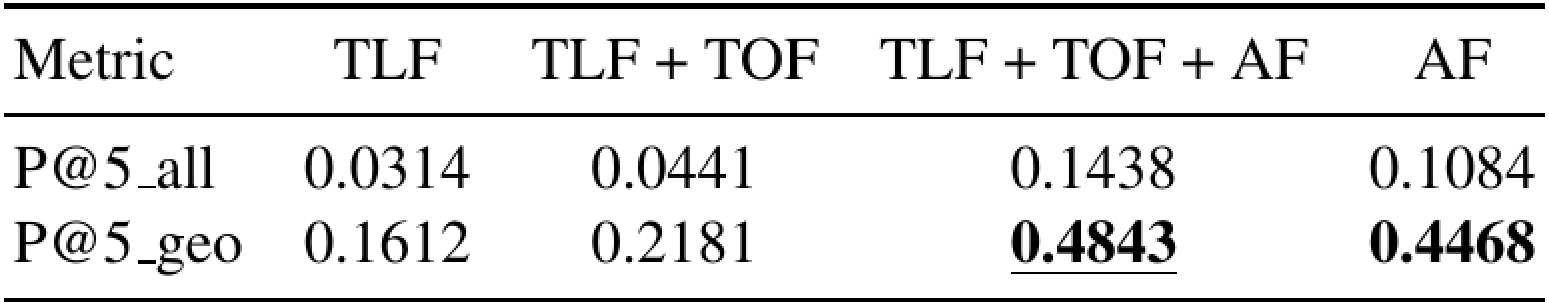


Metric	GeoFiltered	TouristFiltered
P@5_all	0.0468	0.1438
P@5_geo	0.1605	**0.4843**

Analysis (Effect of sub-collection parts)

- TouristFiltered sub-collection consists of three parts
  - TouristListFiltered (TLF)
  - TouristOutlinksFiltered (TOF)
  - AttractionFiltered (AF)

- Measure how each part contributes to the performance



Metric	TLF	TLF + TOF	TLF + TOF + AF	AF
P@5_all	0.0314	0.0441	0.1438	0.1084
P@5_geo	0.1612	0.2181	**0.4843**	**0.4468**

Conclusions and Future work

- Applying Open Web domain knowledge leads to have better suggestions
- We can think of each part in **TouristFiltered** collection as a binary filter
- For future work:
  - We can combine different weighted filters
  - Each filter can represent a different source of knowledge