

DYNAMIC RECOMMENDATIONS FOR FINDING LOCAL EXPERT USING LAZY RANDOM WALK WITH APPLICATION TO SHOPPING PLACES RECOMMENDER SYSTEMS
Diyah Puspitaningrum, Julio Fernando, Edo Afriando, Ferzha Putra Utama
University of Bengkulu, Indonesia
IDA 2017, London, Oct 26-28, 2017



DYNAMIC RECOMMENDATIONS FOR FINDING LOCAL EXPERT USING LAZY RANDOM WALK WITH APPLICATION TO SHOPPING PLACES RECOMMENDER SYSTEMS

Diyah Puspitaningrum et al - DYNAMIC RECOMMENDATIONS FOR FINDING LOCAL EXPERT USING LAZY RANDOM WALK WITH APPLICATION TO SHOPPING PLACES RECOMMENDER SYSTEMS
IDA 2017, London, Oct 26-28, 2017

Dynamic Recommendations For Finding Local Expert Using Lazy Random Walk With Application to Shopping Places Recommender Systems



University of Bengkulu

Web Mining & Information Retrieval Research Group
Dept. of Computer Science, University of Bengkulu, Sumatera, Indonesia
Diyah Puspitaningrum, Julio Fernando, Edo Afriando, Ferzha Putra Utama

Diyah Puspitaningrum et al ~ DYNAMIC RECOMMENDATIONS FOR FINDING LOCAL EXPERT USING LAZY RANDOM WALK WITH APPLICATION TO SHOPPING PLACES RECOMMENDER SYSTEMS
IDA 2017, London, Oct 26-28, 2017

Dynamic Recommendations For Finding Local Expert Using Lazy Random Walk With Application to Shopping Places Recommender Systems

Agenda\Topics:

- Project Description
- Project Methodology
- Dataset & Evaluation Method
- Experiments Results
- Conclusion

```

graph TD
    Experiments --> Results
    Experiments --> Baselines
    Results --> HowToImprove[How to improve?]
    HowToImprove --> SupervisedLRW[Supervised LRW]
    Baselines --> RW[Random Walk RW]
    Baselines --> NetworkProperties[Network Properties]
    NetworkProperties --> PageRank
    NetworkProperties --> closeness
    NetworkProperties --> betweenness
    NetworkProperties --> degree
    
```

Diyah Puspitaningrum et al ~ DYNAMIC RECOMMENDATIONS FOR FINDING LOCAL EXPERT USING LAZY RANDOM WALK WITH APPLICATION TO SHOPPING PLACES RECOMMENDER SYSTEMS
IDA 2017, London, Oct 26-28, 2017

Dynamic Recommendations For Finding Local Expert Using Lazy Random Walk With Application to Shopping Places Recommender Systems

Agenda\Topics:

- Project Description
- Project Methodology
- Dataset & Evaluation Method
- Experiments Results
- Conclusion

```

graph TD
    Experiments --> Results
    Experiments --> Baselines
    Results --> HowToImprove[How to improve?]
    HowToImprove --> SupervisedLRW[Supervised LRW]
    Baselines --> RW[Random Walk RW]
    Baselines --> NetworkProperties[Network Properties]
    NetworkProperties --> PageRank
    NetworkProperties --> closeness
    NetworkProperties --> betweenness
    NetworkProperties --> degree
    
```

DYNAMIC RECOMMENDATIONS FOR FINDING LOCAL EXPERT USING LAZY RANDOM WALK WITH APPLICATION TO SHOPPING PLACES RECOMMENDER SYSTEMS

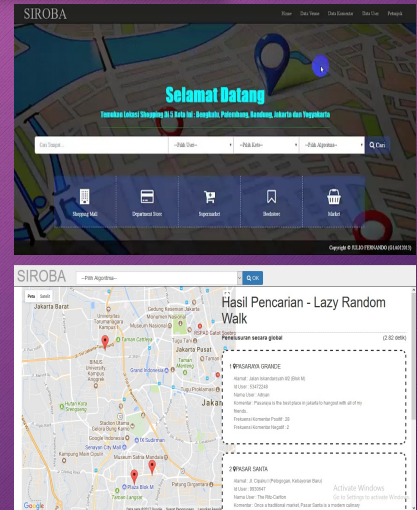
Project Description

Creating a good recommender system that:

- 1) alleviate static recommendation,
- 2) guarantee appropriate recommendation by listen to local experts' opinions using Lazy Random Walk.
- 3) scalable

Benefits:

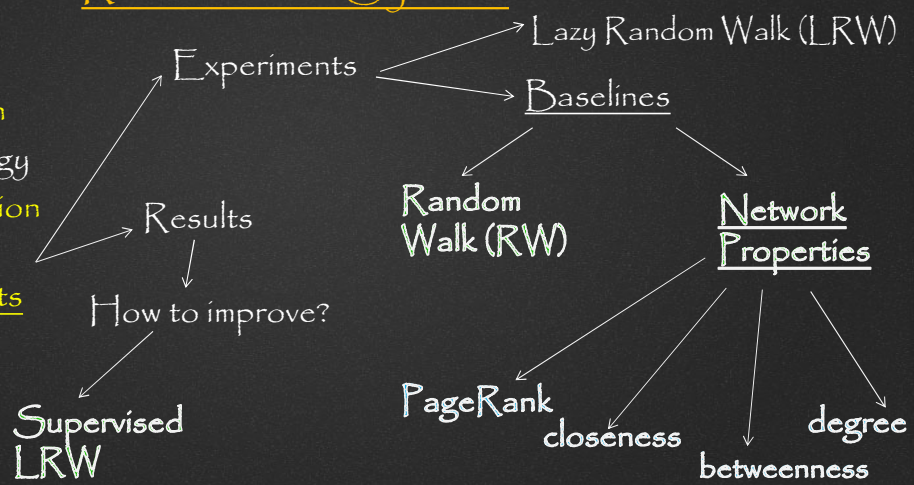
- High score precisions
 $p@1 = 0.5$, $p@3 = 0.37$, $p@5 = 0.26$.
- fast execution time
 3.222 seconds in average
- For the developer: *"the database used to powered recommender system no need to be very frequently updated in order to produce good recommendation"*

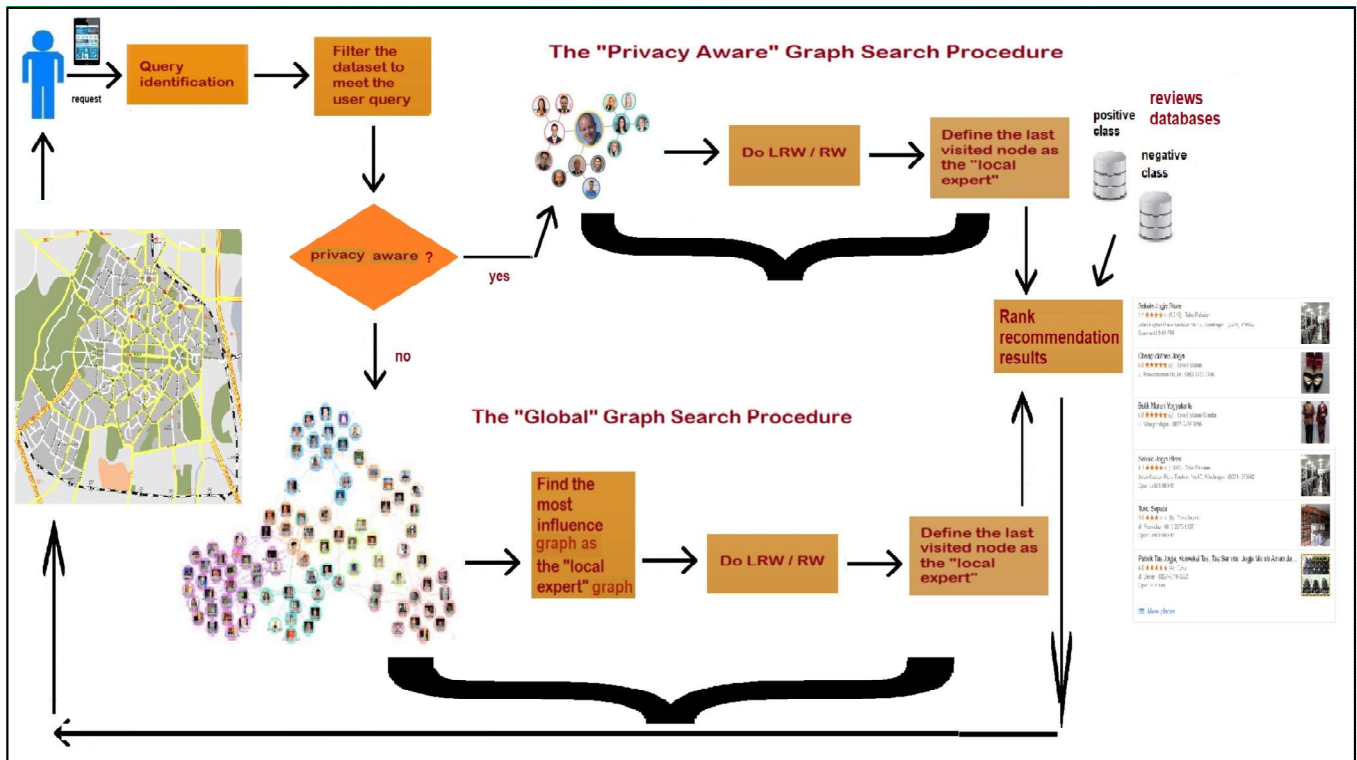


Dynamic Recommendations For Finding Local Expert Using Lazy Random Walk With Application to Shopping Places Recommender Systems

Agenda\Topics:

- Project Description
- Project Methodology
- Dataset & Evaluation Method
- Experiments Results
- Conclusion





Project Methodology

- Procedure: generate recommendation by listen to local experts' opinions using Random Walk technique with $k\text{-steps} = \{5, 7, 9, 11, 13, 15, 17, 19, 21, 23\}$
- Key assumptions: local authority and topical authority
- A sentiment analysis is required before generate the recommendations.
- *Gold standard*: choose recommendations from reviews dataset **from a user input city (local authority)**, and find shopping places that **meet user query (topical authority)** as well as also the member of positive reviews class.

Our propose recommendation system as follows:

1. Query identification: query q (or category such as: "shopping mall", "department store", "supermarket", "bookstore", "market"), userid, city, and algorithm.
2. Filter the visitor review data only to a graph that meet the query criteria. Select only reviews that originated from userid that located in the city stated in point 1.
3. If a user is a person without social contacts then use a global graph search local expert, otherwise use a graph derived from his social contacts to do local expert finding.
4. As we use network properties and random walk variants to test our system, then

If method="PageRank" (or other network centrality measures e.g. "betweenness", "closeness", and "degree"), for the highest local expert score, find his recommendation of places that meet the query.

Else:

If method="Lazy Random Walk" or "Random Walk", for source node of all users in graph (see point 3) with walking-steps = f5, 7, 9g use the last step of the walk as local expert candidate and then find his recommendation of places that meet q .

5. Sort recommendations by number of positive reviews in its descending order.

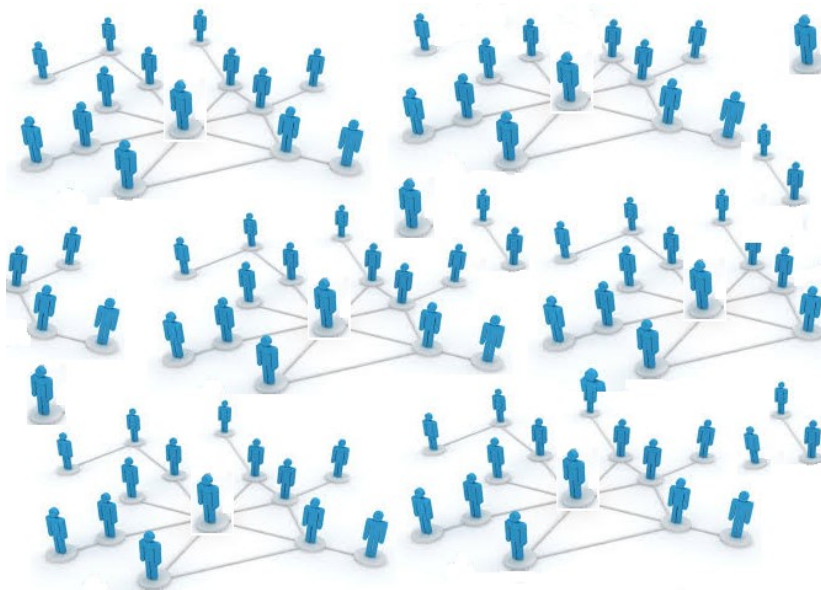
Method:

Lazy Random Walk

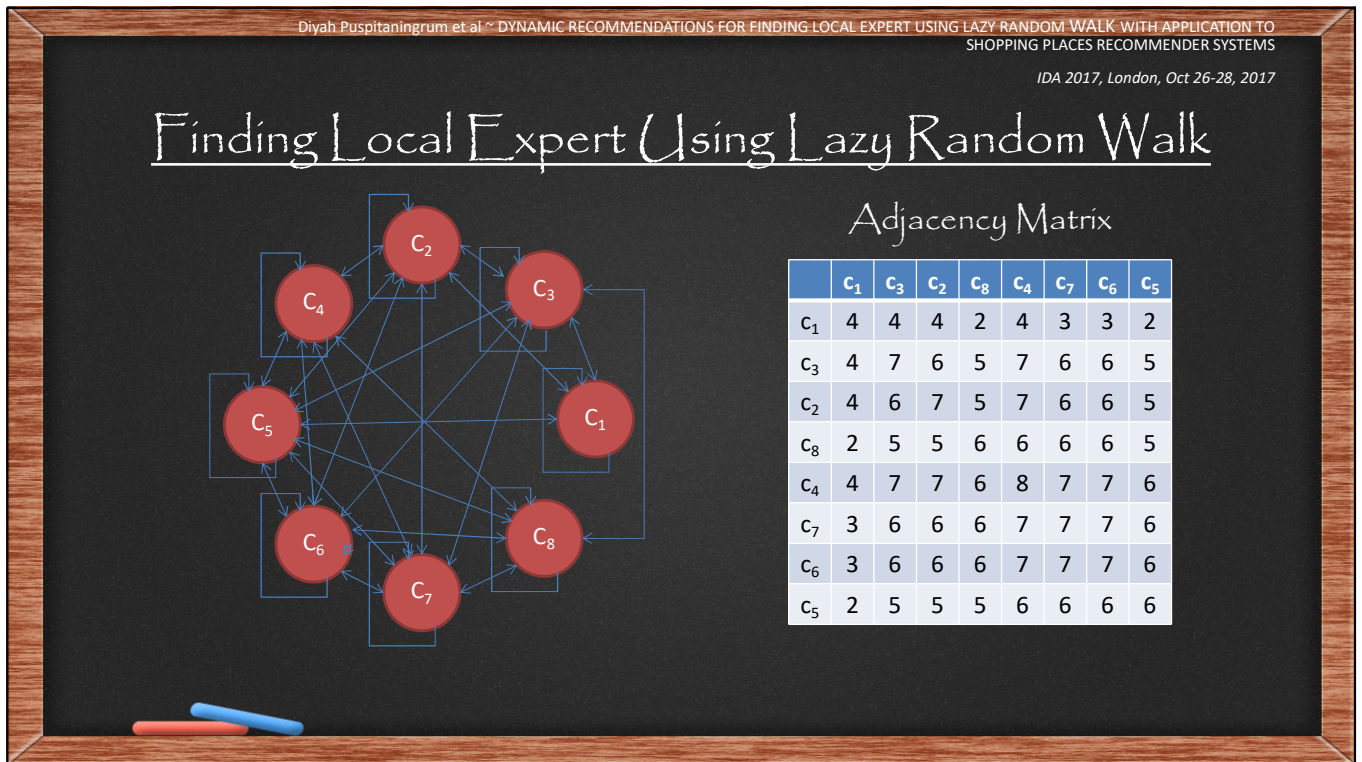
Baselines:

1. Random Walk
2. PageRank
3. Betweenness
4. Closeness
5. Degree

Global View Of User Data



- connected graphs
- isolated graphs
- single nodes



Procedure: generate recommendation by listen to local experts' opinions using Random Walk technique with $k\text{-steps} = \{5, 7, 9, 11, 13, 15, 17, 19, 21, 23\}$

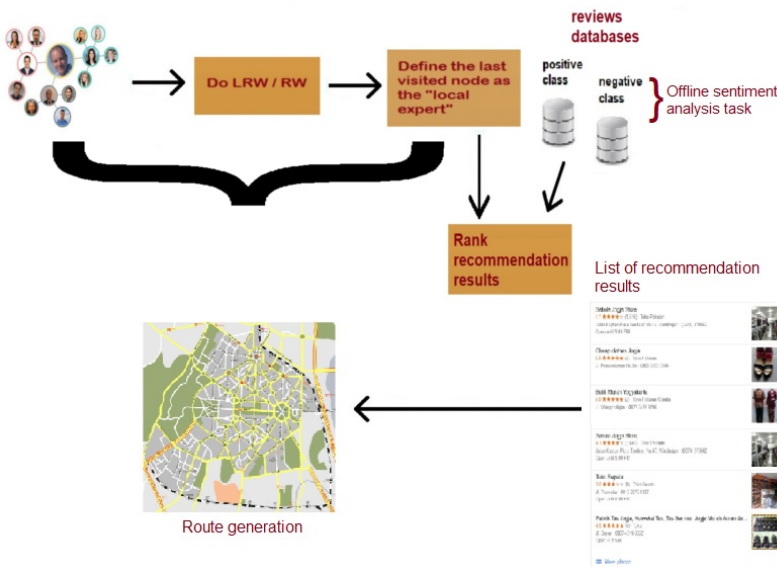
The "Global" Graph Search Procedure:

1. Build global graph(s) of reviewers (users who have comments on a shopping place that meet the user query q) by finding each reviewers' contacts until two intermediate nodes. A reviewer in this graph can be connected or isolated so it is possible to have more than one global graph.
2. Since a local expert should have influence on his local network (in his city), find the highest degree graph as the origin of candidate of local experts.
3. Do lazy random walk or pagerank on the graph with number of walking steps $k\text{-steps} = \{5, 7, 9, 11, 13, 15, 17, 19, 21, 23\}$.
4. The last visited node was chosen as the local expert.

The "Global" Graph Search Procedure

Procedure: generate recommendation by listen to local experts' opinions using Random Walk technique with $k\text{-steps} = \{5, 7, 9, 11, 13, 15, 17, 19, 21, 23\}$

The "Privacy Aware" Graph Search Procedure



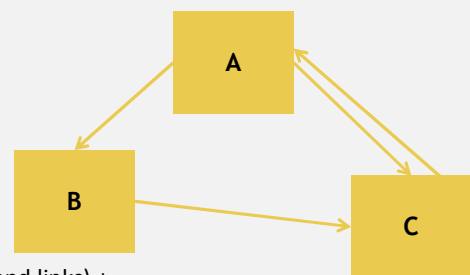
The "Privacy Aware" Graph Search Procedure:

1. Build a global graph until 2 intermediate nodes ahead from a user who request the query.
2. Do lazy random walk on the graph.
3. The last visited node was chosen as the local expert.

PAGERANK

$$PR(A) = (1-d) / N + d (PR(Ti) / C(Ti) + \dots + PR(Tn) / C(Tn))$$

PR(A) : Pagerank of page A
 PR(Ti) : Pagerank of pages Ti which link to page A
 C(Ti) : number of *outbond links* on page Ti
 d : *damping factor* which can be set between 0 and 1.
 N : number of nodes
 or,



PageRank of node A = Initial pagerank + (total ranking power/number of outbond links) + ...

Example: $N = 3 ; d = 0.7$

$$PR(B) = (1 - d) \times (1 / N) + d \times (PR(A) / 2) = 0.1 + 0.35 PR(A)$$

$$PR(C) = (1 - d) \times (1 / N) + d \times ((PR(A) / 2) + (PR(B) / 1)) = 0.17 + 56/200 PR(A)$$

$$PR(A) = (1 - d) \times (1 / N) + d \times (PR(C) / 1) = 0.1 + 0.7 \times (0.17 + 56/200 PR(A)) = 0.219 + 0.196 PR(A)$$

Linear equations

0.804 PR(A) = 0.219, so we have $PR(A) = 0.2723880597, PR(B) = 0.1953358209, PR(C) = 0.2462686567$

Key assumptions: *local authority* and *topical authority*

Local Authority

- a user is said a candidate of local expert if the user's location, and the *query location*, is in the same city.

Topical Authority

- a user is said a candidate of local expert if the user has considerable knowledge about the query category, or
- a user is said a local expert candidate if *he has reviews about user query category* ($c(q)$).

Places

- Recommendations are generated from shopping places that categorize as positive class (positive reviews) and suggested by a local expert that meet both the *local authority* and the *topical authority* criteria.

Positive dictionaries

No	Words
1	Sangat bagus
2	Rekomendasi
3	Nilai
4	Historik
5	sejarah
6	Edukasi
7	Bagus
8	Menyenangkan

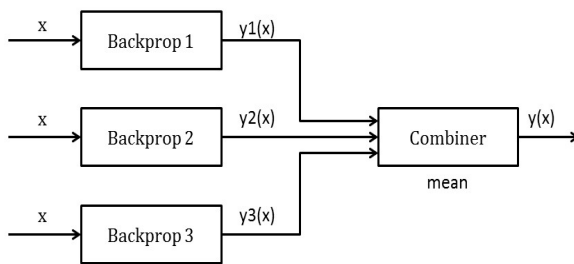
Negative dictionaries

No	Words
1	Terlalu sempit
2	Kurang terawat
3	Jauh

No	Kalimat	Jumlah Positif	Jumlah Negatif	Kesimpulan
1	Museum AAA <i>sangat bagus.</i>	2	0	Positif
2	Museum AAA kami <i>rekomendasikan</i> untuk sekolah-sekolah.	1	0	Positif
3	Museum Asia Afrika membawa kembali <i>nilai-nilai historik</i> dalam <i>sejarah</i> perjuangan bangsa.	3	0	Positif
4	Museum Anekit memiliki misi <i>edukasi.</i>	1	0	Positif
5	Tempat parkir Museum AAA <i>terlalu sempit.</i>	0	1	Negatif
6	Museum Asia Afrika saat ini agak <i>kurang terawat.</i>	0	1	Negatif
7	Untuk ke Museum AAA jalannya <i>jauh.</i>	0	1	Negatif
8	Museum AAA <i>bagus</i> untuk berfoto.	1	0	Positif
9	Bagong museum science mengajarkan fungsi-fungsi panca indera secara <i>menyenangkan.</i>	1	0	Positif

A **sentiment analysis** is required before generate the recommendations.

- **Lazy Random Walk:** sentiment analysis using single SVM
- **Random Walk:** sentiment analysis using single SVM
- **Network properties** (Pagerank, betweenness, closeness, degree): sentiment analysis using ensemble backpropagation.
- BP1 = Learning rate 0.00005, max iteration 20, number of hidden layer 10.
- BP2 = Learning rate 0.00010, max iteration 30, number of hidden layer 20.
- BP3 = Learning rate 0.00015, max iteration 40, number of hidden layer 30.
- SVM kernel nonlinear RBF degree 3, gamma = 3, polynomial degree = 3 .



Ensemble Backpropagation

Klasifier	Validasi Sistem (%)	Pengujian Sistem (%)
Backprop 1	95.04	94.08
Backprop 2	97.99	96.18
Backprop 3	99.77	97.10
Ensemble	98.06	96.18

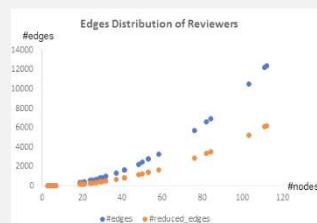
- **Gold standard:** choose recommendations from reviews dataset **from a user input city (local authority)**, and find shopping places that **meet user query (topical authority)** as well as also the member of **positive reviews** class.





Statistics of Data Set

- 1) 176 shopping places data from 5 cities in Indonesia: Jakarta, Bandung, Yogyakarta, Palembang, and Bengkulu.
- 2) 3844 visitors' reviews trained using SVM or Ensemble Backpropagation Neural Networks to classify positive and negative reviews of shopping places dataset (sentiment analysis tasks).
- 3) 14309 users data.



- 4) We have 220 queries q_i , combinations of: query + city + mode + k -walking steps
 - query = {"mall", "inexpensive market", "discount books", "dress shop", "comfortable shopping", "complete and inexpensive shopping"}.
 - city = {Jakarta, Bandung, Yogyakarta, Palembang, Bengkulu}.
 - mode = {"global", "privacy aware"}.
 - k -walking_steps = {5,7,9,11,13,15,17,19,21,23}.

Evaluation

A. Performance measurements:

- *R-score* link prediction task
- *p@x* (precision at x)

B. Scalability measurements:

- *t_graph*: computational cost of time for building graph
- *t_algo*: time for running random walker
- *t_other*: time for others required for producing output from recommender systems
- total execution time: $t_{total} = t_{graph} + t_{algo} + t_{other}$

R-score

- *R-score* and *average over links* is used to handle extreme case of very rare local experts and very popular local experts.
- User *u*'s likelihood:

$$p(u, t) = 2^{-\frac{j_t - 1}{\lambda - 1}}$$

- λ is the number of walking steps divide by 2
- j_t is the position of a shopping place *t* on a ranked list of statistics-based shopping places which has most positive reviews in decrease order.

Link Prediction Functions

A utility function to predict whether a local expert candidate has a non popular shopping place:

$$U(u_i, g_i) = -\log\left(\frac{1}{\text{len}(\text{nodes})} * \text{walking_steps}\right)$$

Expected utility or **R-score**:

$$R_u = \sum_{i=1}^n p(u, g_i) * U(u_i, g_i)$$

n = number of places.

R_u^{\max} is a maximum $p(u, g_i) * U(u_i, g_i)$.

Maximum possible R-score:
recommendations, $m=10$

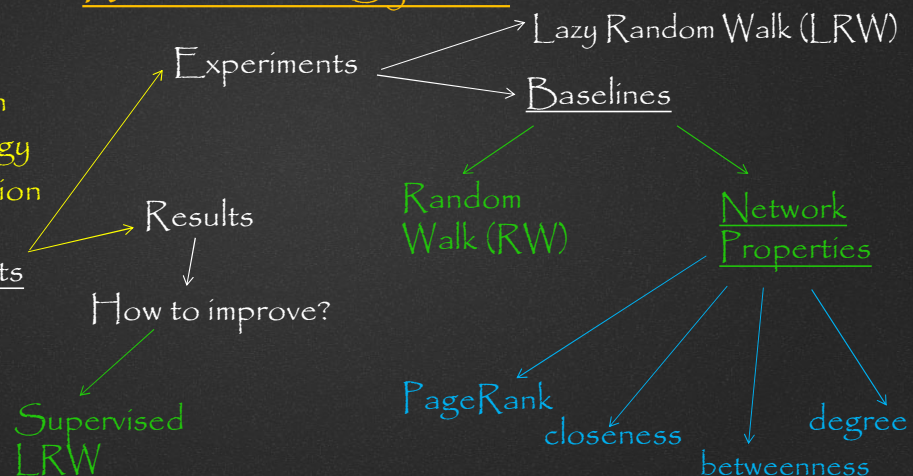
$$E[R_u] = \frac{1}{m} \sum \frac{R_u}{R_u^{\max}}$$

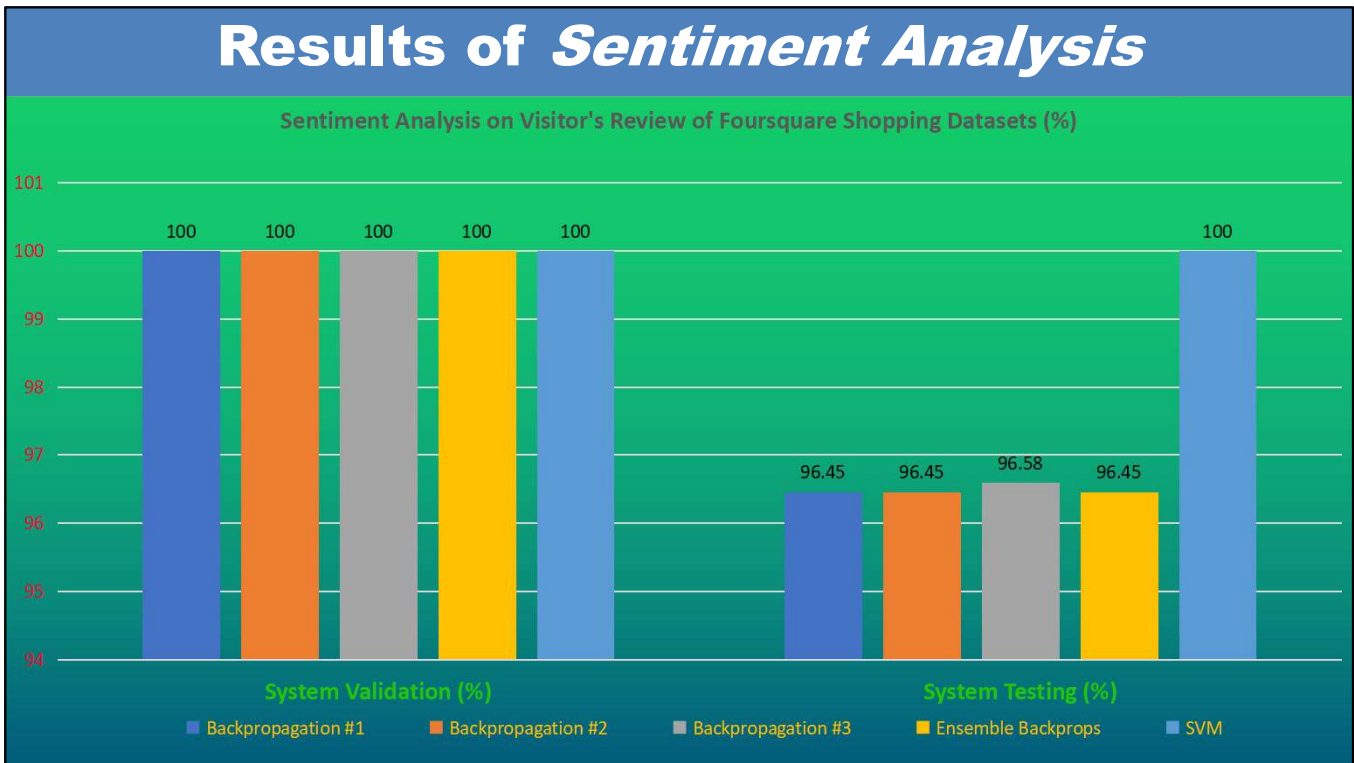
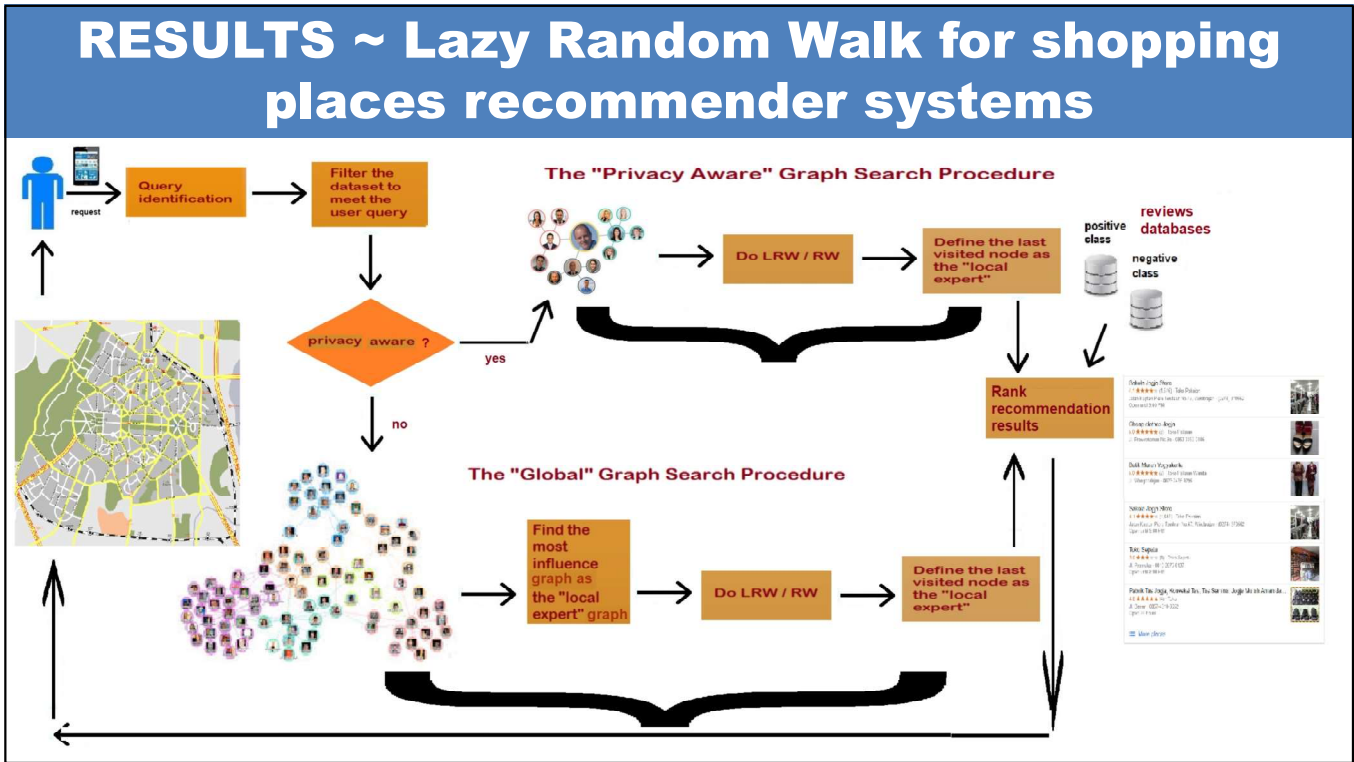
m times generate

Dynamic Recommendations For Finding Local Expert Using Lazy Random Walk With Application to Shopping Places Recommender Systems

Agenda\Topics:

- Project Description
- Project Methodology
- Dataset & Evaluation Method
- Experiments Results
- Conclusion





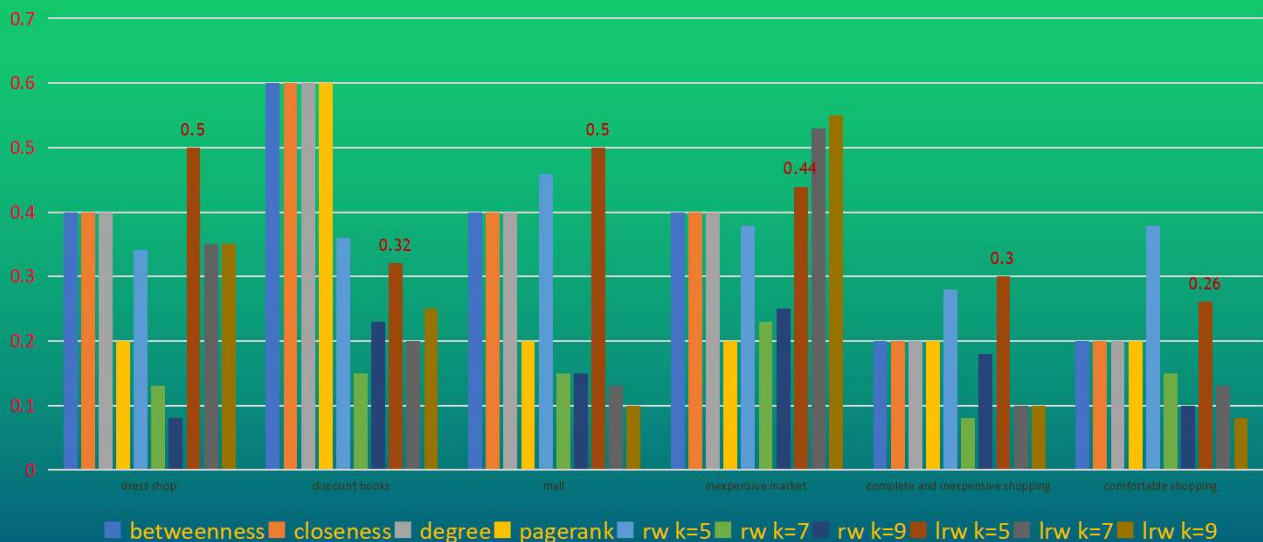
FourSquare log queries	
Query	mall_city = "Bandung", global
lrw + svm	[(('Istana Plaza (IP)', 24L, 5L), ('LotteMart', 23L, 3L), ('Gramedia', 20L, 3L)), ('Festival Citylink', 19L, 5L),
rw + svm	[(('Istana Plaza (IP)', 24L, 5L), ('LotteMart', 23L, 3L), ('METRO Department Store', 21L, 6L)]
svm gold standard	[(('Istana Bandung Electronic Center (BEC)', 26L, 1L), ('Trans Studio Mall (TSM)', 24L, 4L), ('Istana Plaza (IP)', 24L, 5L), ('Cihampelas Walk (GWalk)', 23L, 1L), ('LotteMart', 23L, 3L), ('Braga CityWalk', 23L, 4L), ('Bandung Trade Centre - BTC Fashion Mall', 23L, 3L), ('Bandung Indah Plaza (BIP)', 21L, 2L), ('riaunction', 21L, 5L), ('METRO Department Store', 21L, 6L), ('Gramedia', 20L, 3L), ('Festival Citylink', 19L, 5L), ('Paris Van Java (PVJ)', 18L, 3L), ('Liana swalayan', 5L, 0L), ('Living Plaza Dago', 3L, 0L)]
btw + ens. bp	[(('Trans Studio Mall (TSM)', 27L, 1L), ('Istana Plaza (IP)', 25L, 4L), ('Festival Citylink', 21L, 3L)]
cls + ens. bp	[(('Trans Studio Mall (TSM)', 27L, 1L), ('Istana Plaza (IP)', 25L, 4L), ('Festival Citylink', 21L, 3L)]
deg + ens. bp	[(('Trans Studio Mall (TSM)', 27L, 1L), ('Istana Plaza (IP)', 25L, 4L), ('Festival Citylink', 21L, 3L)]
pr + ens. bp	[(('Trans Studio Mall (TSM)', 27L, 1L), ('Istana Plaza (IP)', 25L, 4L), ('Festival Citylink', 21L, 3L)]
ens. bp gold standard	[(('Setiabudhi Supermarket', 29L, 1L), ('Trans Studio Mall (TSM)', 27L, 1L), ('Istana Bandung Electronic Center (BEC)', 26L, 1L), ('Istana Plaza (IP)', 25L, 4L), ('Bandung Trade Centre - BTC Fashion Mall', 25L, 1L), ('Toko Buku Togamas', 24L, 4L), ('riaunction', 24L, 2L), ('METRO Department Store', 24L, 3L), ('Gramedia', 22L, 1L), ('Braga CityWalk', 22L, 5L), ('Cihampelas Walk (GWalk)', 21L, 3L), ('Festival Citylink', 21L, 3L), ('Bandung Indah Plaza (BIP)', 21L, 2L), ('Paris Van Java (PVJ)', 20L, 1L), ('LotteMart', 19L, 7L), ('Liana swalayan', 5L, 0L)]

*lrw = lazy random walk
 *rw = random walk
 *svm = support vector machine
 *ens. bp = ensemble backprops

* btw = betweenness
 *cls = closeness
 *deg = degree
 *pr = PageRank

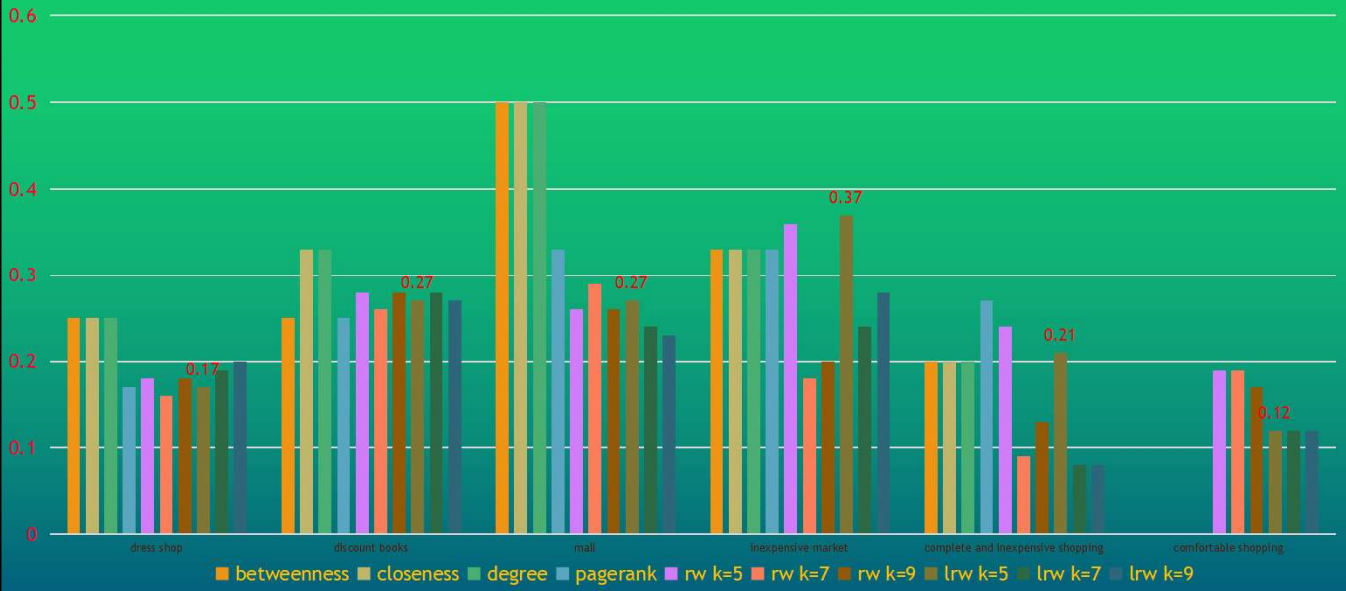
Results $p@1$ of Local Expert Finding (1)

Precisions $p@1$ of Shopping Places Dataset with k -steps = {5, 7, 9}



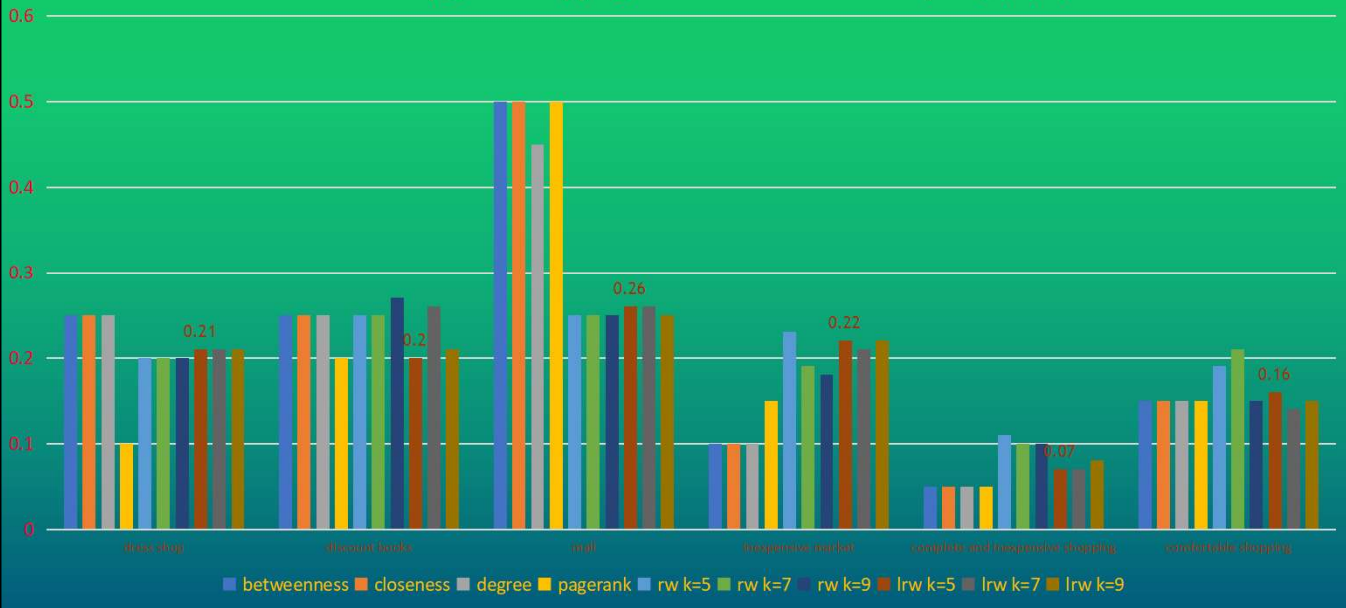
Results $p@3$ of Local Expert Finding (2)

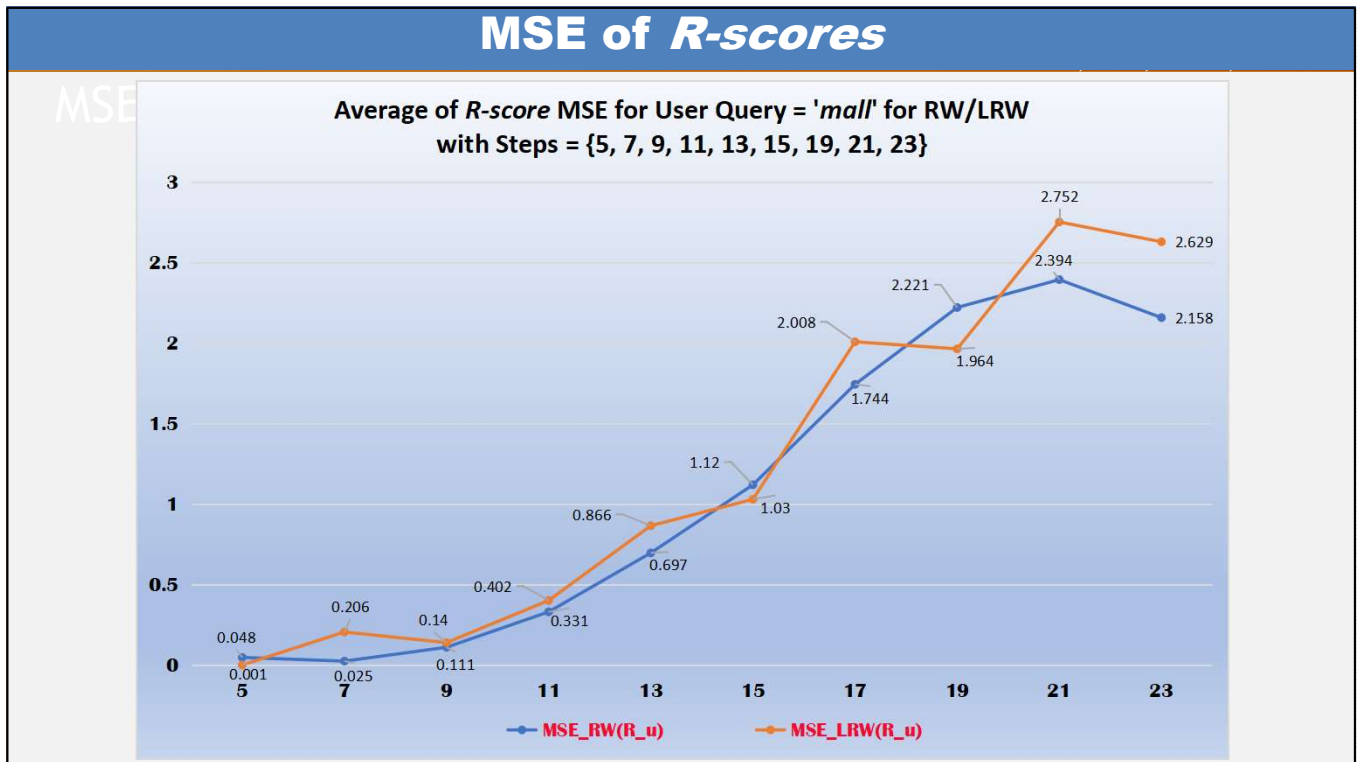
Precisions $p@3$ of Shopping Places Dataset with k -steps = {5, 7, 9}



Results $p@5$ of Local Expert Finding (3)

Precisions $p@5$ of Shopping Places Dataset with k -steps = {5, 7, 9}





FourSquare log queries	
Query	mall_city = "Bandung", global
lrw + svm	[('Istana Plaza (IP)', 24L, 5L), ('LotteMart', 23L, 3L), ('GrameDia', 20L, 3L)], ('Festival Citylink', 19L, 5L),
rw + svm	[('Istana Plaza (IP)', 24L, 5L), ('LotteMart', 23L, 3L), ('METRO Department Store', 21L, 6L)]
svm gold standard	[('Istana Bandung Electronic Center (BEC)', 26L, 1L), ('Trans Studio Mall (TSM)', 24L, 4L), ('Istana Plaza (IP)', 24L, 5L), ('Cihampelas Walk (GWalk)', 23L, 1L), ('LotteMart', 23L, 3L), ('Braga CityWalk', 23L, 4L), ('Bandung Trade Centre - BTC Fashion Mall', 23L, 3L), ('Bandung Indah Plaza (BIP)', 21L, 2L), ('riaunction', 21L, 5L), ('METRO Department Store', 21L, 6L), ('GrameDia', 20L, 3L), ('Festival Citylink', 19L, 5L), ('Paris Van Java (PVJ)', 18L, 3L), ('Liana swalayan', 5L, 0L), ('Living Plaza Dago', 3L, 0L)]
btw + ens. bp	[('Trans Studio Mall (TSM)', 27L, 1L), ('Istana Plaza (IP)', 25L, 4L), ('Festival Citylink', 21L, 3L)]

cls + ens. bp	[('Trans Studio Mall (TSM)', 27L, 1L), ('Istana Plaza (IP)', 25L, 4L), ('Festival Citylink', 21L, 3L)]
deg + ens. bp	[('Trans Studio Mall (TSM)', 27L, 1L), ('Istana Plaza (IP)', 25L, 4L), ('Festival Citylink', 21L, 3L)]
pr + ens. bp	[('Trans Studio Mall (TSM)', 27L, 1L), ('Istana Plaza (IP)', 25L, 4L), ('Festival Citylink', 21L, 3L)]
ens. bp gold standard	[('Setiabudhi Supermarket', 29L, 1L), ('Trans Studio Mall (TSM)', 27L, 1L), ('Istana Bandung Electronic Center (BEC)', 26L, 1L), ('Istana Plaza (IP)', 25L, 4L), ('Bandung Trade Centre - BTC Fashion Mall', 25L, 1L), ('Toko Buku Togamas', 24L, 4L), ('riaunction', 24L, 2L), ('METRO Department Store', 24L, 3L), ('GrameDia', 22L, 1L), ('Braga CityWalk', 22L, 5L), ('Cihampelas Walk (GWalk)', 21L, 3L), ('Festival Citylink', 21L, 3L), ('Bandung Indah Plaza (BIP)', 21L, 2L), ('Paris Van Java (PVJ)', 20L, 1L), ('LotteMart', 19L, 7L), ('Liana swalayan', 5L, 0L)]
*lrw = lazy random walk *rw = random walk *svm = support vector machine *ens. bp = ensemble backprops	*btw = betweenness *cls = closeness *deg = degree *pr = PageRank

System Scalability (Time) of LRW vs RW

$$t_{total} = t_{graph} + t_{algo} + t_{other}$$

Internal time per stages for finding local expert using recommender system with LRW:

- $t_{graph_{LRW}} : t_{algo_{LRW}} : t_{other_{LRW}} = 93\% : 3\% : 4\%$

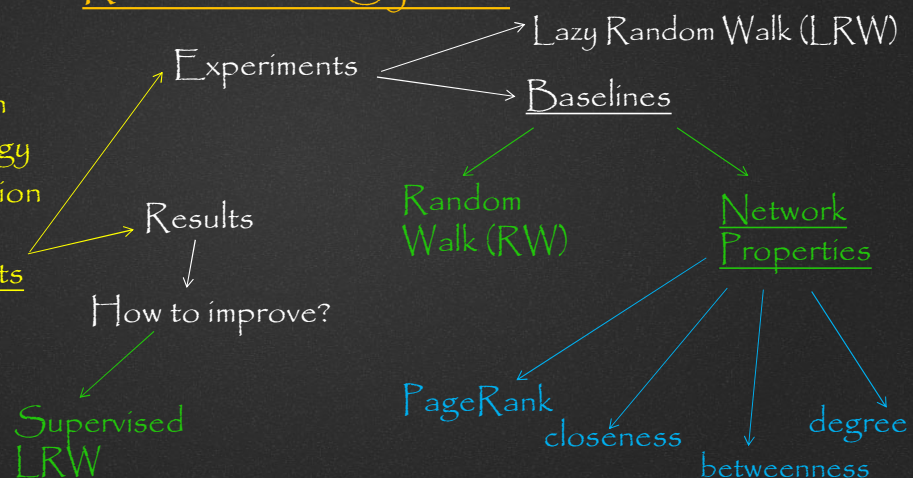
Total time for finding local expert using recommender system with LRW vs RW:

- $t_{total_{LRW}} : t_{total_{RW}} = 3.222 : 3.215$ seconds

Dynamic Recommendations For Finding Local Expert Using Lazy Random Walk With Application to Shopping Places Recommender Systems

Agenda\Topics:

- Project Description
- Project Methodology
- Dataset & Evaluation Method
- Experiments Results
- Conclusion



Conclusion

- We have shown the use of Lazy Random Walk for generating dynamic recommendations (for finding local expert) with implementation in shopping places recommender systems.
- Our proposed Lazy Random Walk recommender system shows both good in accuracy as well as time efficient: MSE of link prediction in local experts finding ≤ 0.206 , $p@1=0.5$, $p@3=0.37$, $p@5=0.26$ and is very scalable (execution time was only 3.222 seconds).
- A recommender system need preprocessing task: *sentiment analysis*, where the quality of *sentiment analysis* is depend on the classifier method. A *single SVM* classifier is best. It is better than *single backprop* or even *ensemble backprop*.
- Using dynamic recsys, the database of user reviews is no need to be very frequently updated to produce good recommendations.

Future Works

- The Lazy Random Walk algorithm for recommendation systems can be improved using the Supervised Lazy Random Walk (LRW) version.
- The supervised version will classify a node in a selected graph-of-candidates-of-local-expert automatically into category of “*expert*” or “*non expert*” using local node metrics, global graph metrics, and parameters of LRW.
- We expect more robust than the LRW method used in this current prototype of recommendation system that use only the parameters of LRW as its inputs.

Possible Improvement Method

IDA 2017, London, Oct 26-28, 2017

Supervised Lazy Random Walk for Recommender Systems

IDA 2017, London, Oct 26-28, 2017

Supervised Lazy Random Walk

- Features of input classifier:
 - i. Common neighbor $(x,y) \approx CN(x,y)$
 - ii. Sorensen index $(x,y) \approx Sorensen(x,y)$
 - iii. Hub promoted index $(x,y) \approx HPI(x,y)$
 - iv. Leicht-Holme-Newman $(x,y) \approx LHN(x,y)$
 - v. Total distance Dijkstra $(x,y) \approx Dijkstra(x,y)$
 - vi. Hitting time $(x,y) \approx hit(x,y)$
 - vii. k_stepwalks $(x,y) \approx step(x,y)$