

Hyperlink Classification via Structured Graph Embedding

*Geon Lee¹, Seonggoo Kang², and Joyce Jiyoung Whang^{*1}*

¹*Sungkyunkwan University (SKKU)*, ²*Naver Corporation*, *corresponding author

The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019

Contents

- 1. Outline**
- 2. Datasets and Challenges**
- 3. Knowledge Graph Embedding**
- 4. Hyperlink Classification Model**
- 5. Result**
- 6. Conclusion and Future Work**

Outline

Three types of hyperlinks:

Navigation Link

Links related to navigation within a site, such as site path or site recommendation.

Suggestion Link

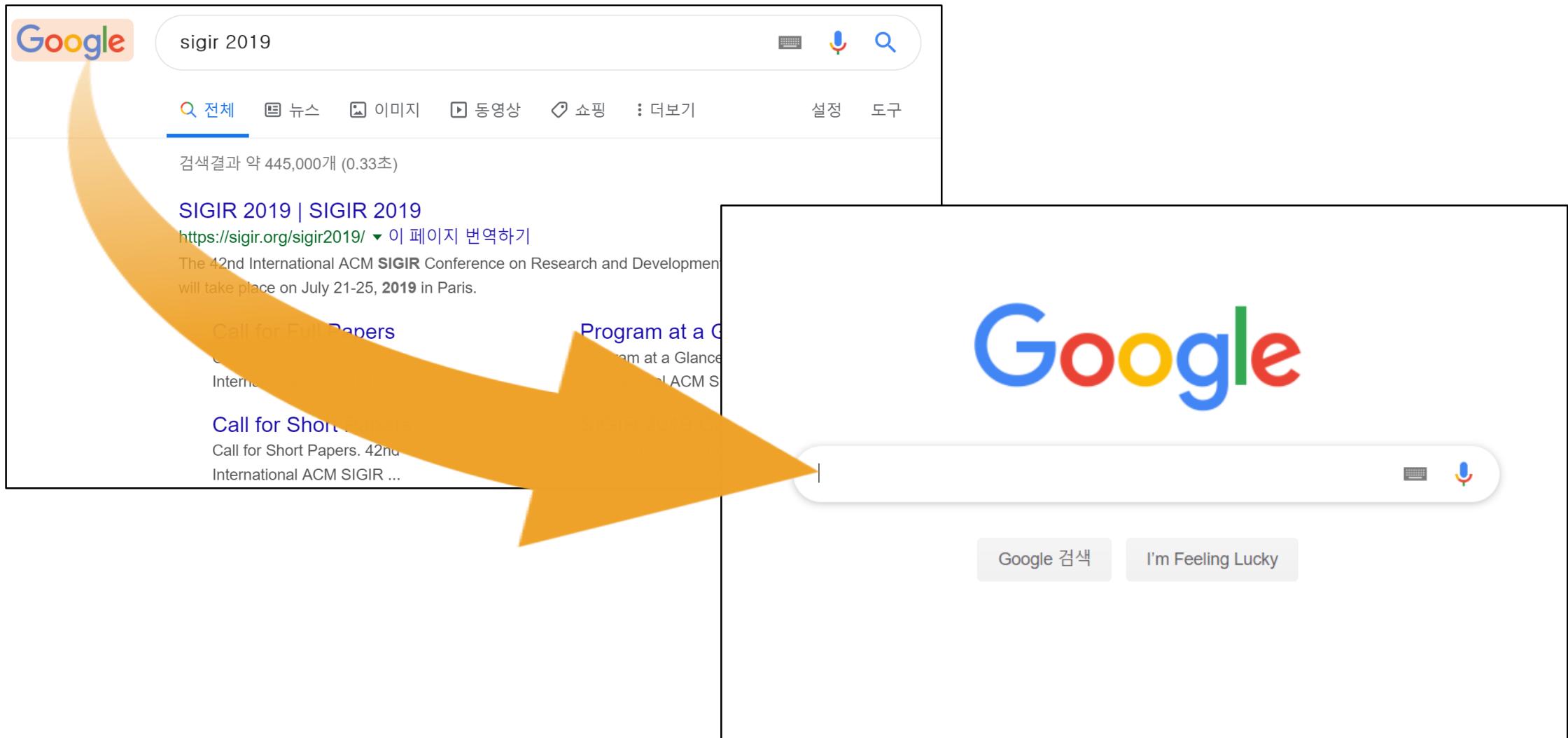
Links recommended directly by the user using the site.

Action Link

Links that require other user actions.

Outline

Navigation Link



Outline

Suggestion Link

The screenshot displays two side-by-side web pages from Stack Overflow and Computational Science. A large orange arrow points from the left page to the right page, highlighting a specific link.

Stack Overflow (Left):

- Header: stackoverflow
- Search bar: Search...
- Navigation: Home, PUBLIC, Stack Overflow, Tags, Users, Jobs, Teams Q&A for work, Learn More
- Section: 3 Answers
- List:
 - If you need the inverse, you should use `inv`.
 - The inverse is calculated via LU decomposition, whereas the backslash operator `\` calculates the solution to your linear system using different methods depending on the properties of your matrix `A` (see <https://scicomp.stackexchange.com/a/1004>), which can yield less accurate results for the inverse.
 - It should be noted that if you want to solve a linear system, the calculation is likely going to be much faster and more accurate using `\`. The MATLAB documentation of `inv` is basically one big warning not to use `inv` to solve linear systems.
- Post 7: `eps(n)` is a command that returns the distance to the next larger double precision number from `n` in MATLAB. So, `eps(1) = 2.2204e-16` means that the first number after 1 is $1 + 2.2204e-16$. Similarly, `eps(3000) = 4.5475e-13`. Now, let's look at the precision of your calculations:

```
n = 100;
A = rand(n);
inv_A_1 = inv(A);
inv_A_2 = A \ eye(n);

max(max(abs(inv_A_1-inv_A_2)))
ans =
1.6431e-14
```

Computational Science (Right):

- Header: Stack Exchange
- Search bar: Search on Computational Science...
- Navigation: Home, Questions, Tags, Users, Unanswered
- Section: Hot Network Questions
- List:
 - In Matlab, the '`\`' command invokes an algorithm which depends upon the structure of the matrix `A` and includes checks (small overhead) on properties of `A`.
 - 1. If `A` is sparse and banded, employ a banded solver.
 - 2. If `A` is an upper or lower triangular matrix, employ a backward substitution algorithm.
 - 3. If `A` is symmetric and has real positive diagonal elements, attempt a Cholesky factorization. If `A` is sparse, employ reordering first to minimize fill-in.
 - 4. If none of criteria above is fulfilled, do a general triangular factorization using Gaussian elimination with partial pivoting.
 - 5. If `A` is sparse, then employ the UMFPACK library.
 - 6. If `A` is not square, employ algorithms based on QR factorization for undetermined systems.
- Post 1: Assuming I am dealing with a 10000×10000 unstructured dense matrix with all elements nonzero (High level of density), what would be my best bet? I want to isolate that 1 algorithm which works for dense matrices. Is it LU, QR or Gaussian Elimination? – [Inquest](#) Jan 25 '12 at 20:14
- Post 1: Sounds like a Step 4 where Gaussian Elimination is invoked which corresponds to the most general case where no structure of `A` can be exploited to boost performance. So, basically this is a LU factorization and subsequent one forward followed by a backward substitution step. – [Allan P. Engsig-Karup](#) Jan 25 '12 at 20:18

Outline

Action Link

The screenshot illustrates the 'Action Link' feature in Wikipedia. It shows two views of the same article: the 'Read' view and the 'Edit' view.

Read View: The top part shows the article 'Association for Computing Machinery'. The title is bolded. Below it is a summary: "From Wikipedia, the free encyclopedia". The main content describes the Association for Computing Machinery (ACM) as an international learned society for computing, founded in 1947, and the world's largest scientific and educational computing society. It also mentions its status as a profit professional membership group and its headquarters in New York City. A large orange arrow points from the 'Edit' button in the top navigation bar to the 'Edit' view below.

Edit View: The bottom part shows the 'Editing Association for Computing Machinery' page. It features a warning message: "You are not logged in. Your IP address will be publicly visible if you make any edits. If you log in or create an account, your edits will be attributed to a user name, among other benefits." Below this is a note about copyright violations and verifiability. The edit interface includes a toolbar with bold, italic, and image insertion tools, and a text area containing the wikitext for the infobox organization.

Left Sidebar (Visible in both views):

- WIKIPEDIA The Free Encyclopedia
- Main page
- Contents
- Featured content
- Current events
- Random article
- Donate to Wikipedia
- Wikipedia store
- Interaction
- Help
- About Wikipedia
- Community portal
- Recent changes
- Contact page

Top Navigation Bar (Visible in both views):

- Not logged in Talk Contributions Create account Log in
- Article Talk Read Edit View history Search Wikipedia

Edit View Toolbar:

- B I ↲ 🔍 📸 🖼 🎥 🖊 Advanced Special characters Help Cite

Edit View Text Area (Wikitext):

```
 {{Use mdy dates|date=June 2012}}
{{Infobox organization
| name      = Association for Computing Machinery
| image     = Association for Computing Machinery (ACM) logo.svg
| image_border =
| size      = 100px
| alt       = }}  
The ACM is an umbrella organization for academic and professional members in computer science. Its motto is "Advancing Computing Profession".
```

Datasets and Challenges

Conduct a biased random walk from the seed page. (By NAVER)

The screenshot shows a Stack Overflow question page. The question title is "Why is it faster to process a sorted array than an unsorted array?". The post has 22628 upvotes and 10362 downvotes. The code snippet provided is:

```
#include <algorithm>
#include <ctime>
#include <iostream>

int main()
{
    // Generate data
    const unsigned arraySize = 32768;
    int data[arraySize];

    for (unsigned c = 0; c < arraySize; ++c)
        data[c] = std::rand() % 256;

    // !!! With this, the next loop runs faster
    std::sort(data, data + arraySize);

    // Test
    clock_t start = clock();
    long long sum = 0;

    for (unsigned i = 0; i < 100000; ++i)
    {
        // Primary loop
        for (unsigned c = 0; c < arraySize; ++c)
        {
            if (data[c] >= 128)
                sum += data[c];
        }
    }

    double elapsedTime = static_cast<double>(clock() - start) / CLOCKS_PER_SEC;
    std::cout << elapsedTime << std::endl;
    std::cout << "sum = " << sum << std::endl;
}
```

Below the code, there are two bullet points:

- Without `std::sort(data, data + arraySize);`, the code runs in 11.54 seconds.
- With the sorted data, the code runs in 1.93 seconds.

A note at the bottom states: "Initially, I thought this might be just a language or compiler anomaly. So I tried it in Java."

To the right of the question, there is a sidebar titled "Looking for a job?" listing several job openings:

- Cloud-Python Developer
- Back-end(Python) Developer
- Front End Developer
- Senior iOS Engineer

At the bottom of the sidebar, there are three "Linked" questions:

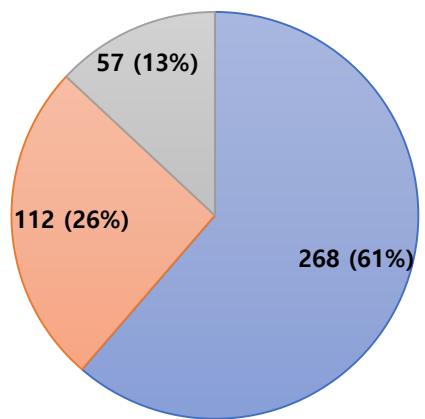
- Is "==" in sorted array not faster than unsorted array?
- Complexity of comparison operators
- SQL Server: dynamic columns based on row values (Date)

Seed Page

Datasets and Challenges

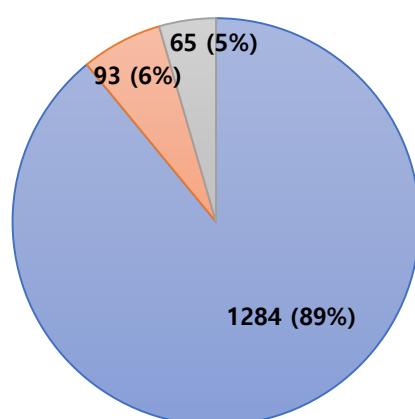
Datasets

	graph_437	graph_1442	Graph_10000
No. of Hyperlinks	437	1442	10000
No. of Pages	404	332	2202
No. of Domain	18	100	120
No. of Host	-	22	25



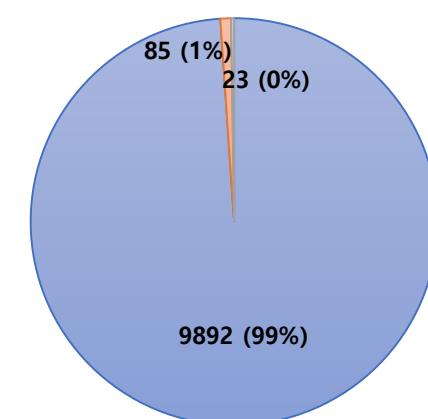
■ Navigation ■ Suggestion ■ Action

graph_437



■ Navigation ■ Suggestion ■ Action

graph_1442

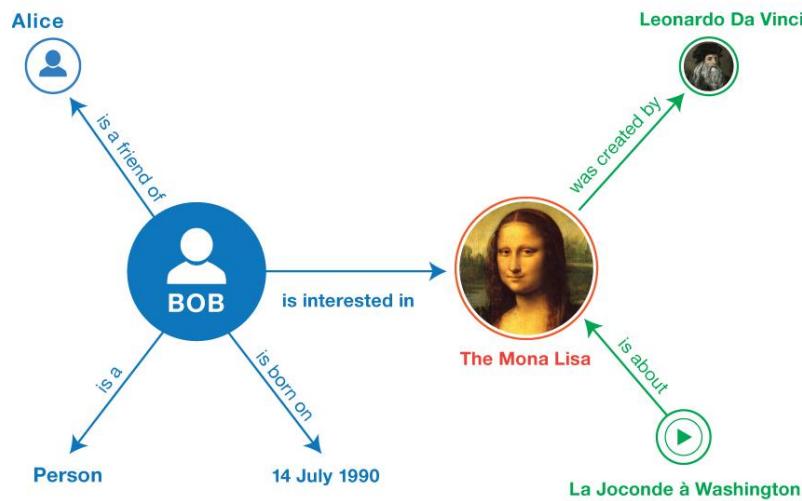


■ Navigation ■ Suggestion ■ Action

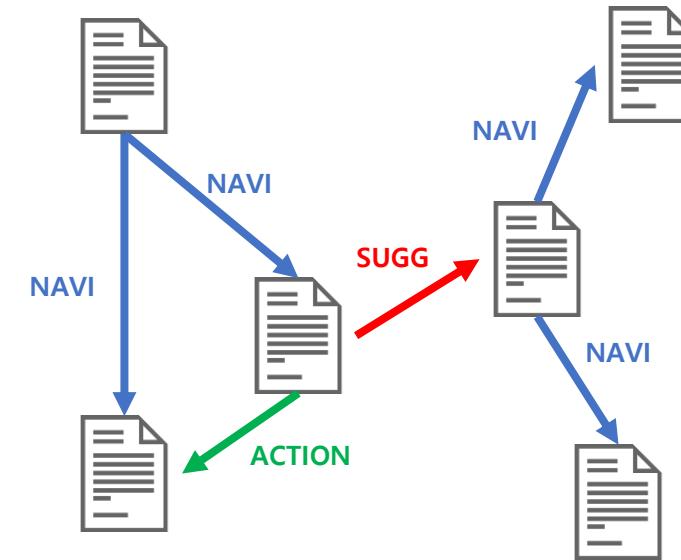
graph_10000

Knowledge Graph Embedding

Can the **knowledge graph embedding** method be applied to **hyperlink embedding**?



Knowledge Graph



Web Graph

Knowledge Graph Embedding

Can the **knowledge graph embedding** method be applied to **hyperlink embedding**?

(Bob, is_interested_in, The_Mona_Lisa)

(Bob, _is_a_friend_of, Alice)

(Barack Obama, _place_of_birth, Hawaii)

(Albert Einstein, _follows_diet, Veganism)

(San Francisco, _contains, Telegraph Hill)



(www.naver.com, NAVI, www.news.naver.com)

(www.google.com/larry_page, SUGG, www.wikipedia.org/Larry_Page)

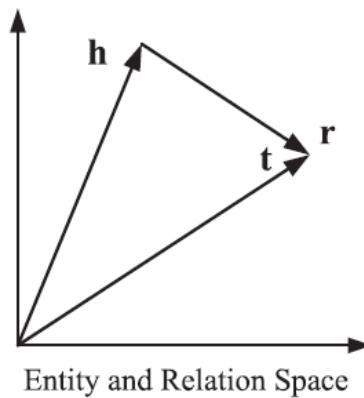
(www.wikipedia.org/Larry_Page, ACTION, www.wikipedia.org/Larry_Page/edit)

Knowledge Graph Embedding

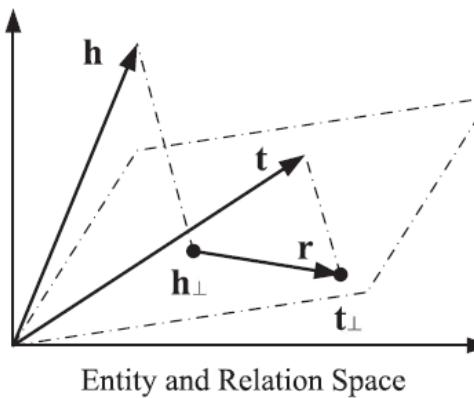
TransE : Translating Embeddings for Modeling Multi-relational Data (NIPS 2013)

TransH : Knowledge Graph Embedding by Translating on Hyperplanes (AAAI 2014)

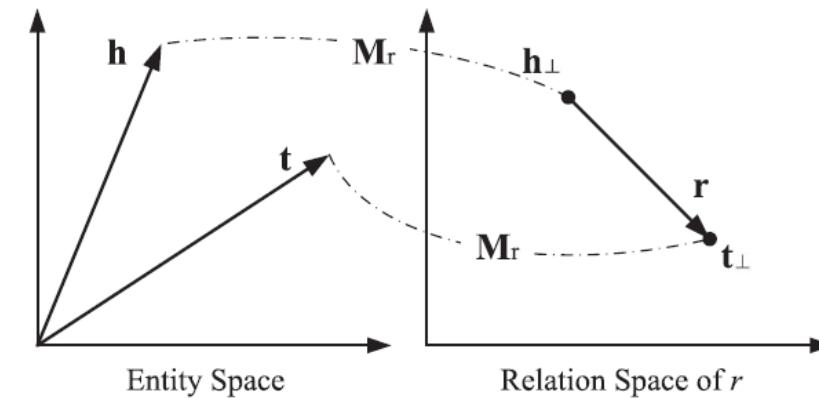
TransR : Learning Entity and Relation Embeddings for Knowledge Graph Completion (AAAI 2015)



(a) TransE.



(b) TransH.



(c) TransR.

Knowledge Graph Embedding

TransE, **TransH**, and **TransR** methods minimize the following loss function:

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'} [f(h, r, t) + \gamma - f(h', r, t')]_+$$

where $[x]_+ \equiv \max(0, x)$ and γ is the margin.

S is a set of golden triplets and S' is a set of corrupted triplets.

The method for calculating $f(h, r, t)$ depends on the model.

$$\text{TransE} : f(h, r, t) = \|h + r - t\|_2^2$$

$$\text{TransH} : f(h, r, t) = \|h_\perp + r - t_\perp\|_2^2$$

$$\text{TransR} : f(h, r, t) = \|h_r + r - t_r\|_2^2$$

e_\perp and e_r represent projected entity e on relation-specific hyperplane and space, respectively.

Hyperlink Classification Model

Given two web pages P_u and P_v , and set of hyperlink types R, estimate the type of hyperlink between P_u and P_v as follow.

$$r^* = \underset{r \in R}{\operatorname{argmin}} f(P_u, r, P_v)$$

where $f(h, r, t)$ is the loss of triplet (h, r, t) .

r^* is the predicted relation.

Hyperlink Classification Model

Negative Sampling

TransE / TransH / TransR → **only changes entities** when making negative samples

$$(h, r, t) \rightarrow (\textcolor{red}{h}', r, t) \text{ or } (h, r, \textcolor{red}{t}')$$

Introduce parameter α ($0 \leq \alpha \leq 1$).

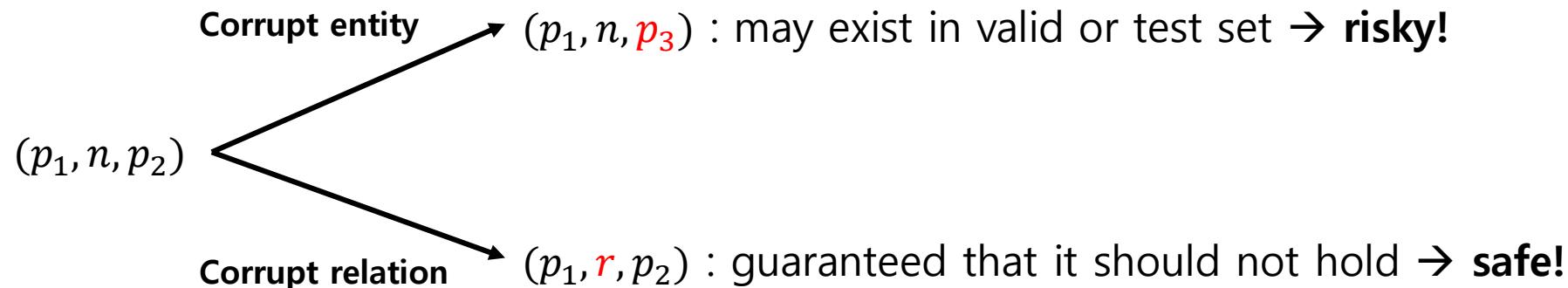
- Replace the **entity** with the probability α . $(h, r, t) \rightarrow (\textcolor{red}{h}', r, t) \text{ or } (h, r, \textcolor{red}{t}')$
- Replace the **relation** with the probability $1 - \alpha$. $(h, r, t) \rightarrow (h, \textcolor{red}{r}', t)$

Hyperlink Classification Model

Negative Sampling

False Negative

When we corrupt entities, there is a chance that it is not a *corrupted* one but just an *unobserved* one in the training set.



Result

The average F1 scores(%) of our model with different α and the original TransE, TransH, and TransR.

		TransE	TransH	TransR
web_437	Our model, $\alpha = 0.3$	34.29	60.25	57.99
	Our model, $\alpha = 0.5$	34.39	58.87	57.32
	Our model, $\alpha = 0.7$	33.88	58.91	59.83
	The original model	36.22	54.04	53.22
web_1442	Our model, $\alpha = 0.3$	23.39	53.42	50.04
	Our model, $\alpha = 0.5$	24.86	55.16	46.18
	Our model, $\alpha = 0.7$	21.18	52.70	45.12
	The original model	20.05	29.94	10.35
web_10000	Our model, $\alpha = 0.3$	20.68	76.00	53.86
	Our model, $\alpha = 0.5$	17.98	74.64	46.99
	Our model, $\alpha = 0.7$	19.50	72.94	44.11
	The original model	15.31	25.35	2.08

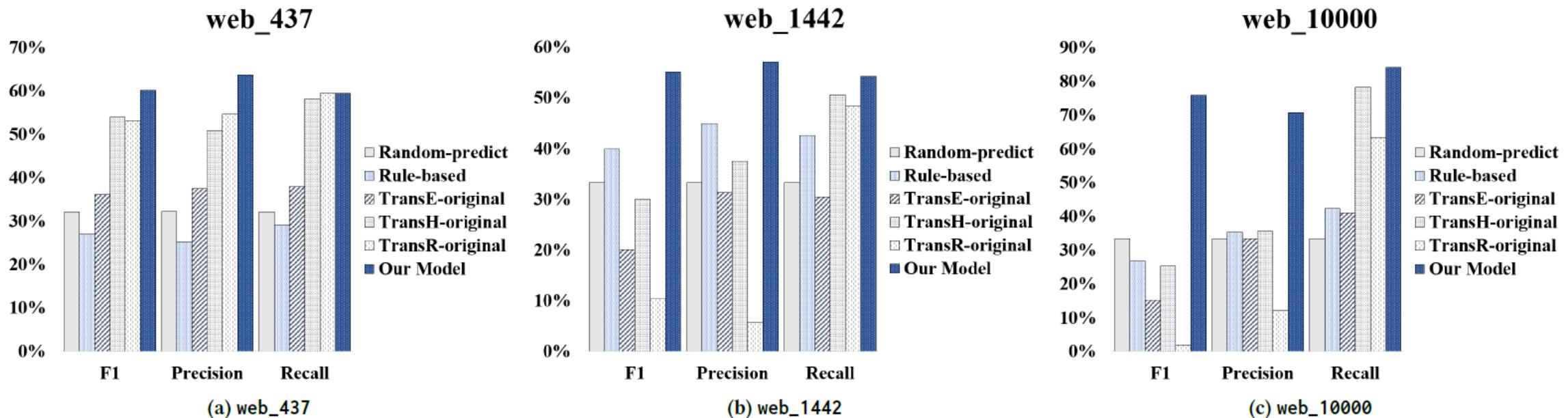
Result

F1 Scores(%) of each relation and the average F1 scores.

		<i>navigation</i>	<i>suggestion</i>	<i>action</i>	Average
web_437	Random-predict	59.75	25.81	11.07	32.21
	Rule-based	60.20	20.96	0.00	27.05
	TransE-original	55.78	31.96	20.93	36.22
	TransH-original	70.80	52.75	38.56	54.04
	TransR-original	67.87	52.86	38.94	53.22
	Our Model	77.04	57.05	46.64	60.25
web_1442	Random-predict	89.13	5.18	5.65	33.32
	Rule-based	72.98	10.20	36.67	39.95
	TransE-original	42.54	8.57	9.05	20.05
	TransH-original	54.80	13.57	21.45	29.94
	TransR-original	0.00	12.97	18.09	10.35
	Our Model	93.48	22.88	49.12	55.16
web_10000	Random-predict	98.91	1.60	0.00	33.50
	Rule-based	68.81	1.74	9.92	26.82
	TransE-original	43.25	2.06	0.61	15.31
	TransH-original	63.01	12.02	1.03	25.35
	TransR-original	0.00	5.61	0.61	2.08
	Our Model	99.66	83.22	45.12	76.00

Result

The average F1, precision, recall on the three web graphs.



- **Random-predict** indicates the performance of random prediction while preserving the number of hyperlinks in each class.
- We use the result of TransH with $\alpha = 0.3$, $\alpha = 0.5$, and $\alpha = 0.3$ for web_437, web_1442, and web_10000, respectively.

Result

Performance on the original web graphs and randomly shuffled graphs where the relation labels are randomly assigned while preserving the number of each relation type.

		navigation			suggestion			action		
		F1 (%)	Pre. (%)	Rec. (%)	F1 (%)	Pre. (%)	Rec. (%)	F1 (%)	Pre. (%)	Rec. (%)
web_437	Original Graph	77.04	78.82	75.37	57.05	50.43	65.77	46.64	62.00	37.43
	Randomly Shuffled Graph	58.60	60.51	56.88	25.36	24.39	26.59	13.79	13.26	14.42
web_1442	Original Graph	93.48	92.22	94.78	22.88	30.66	18.28	49.12	48.52	49.74
	Randomly Shuffled Graph	86.08	88.94	83.41	6.19	5.28	7.53	5.68	4.58	7.52
web_10000	Original Graph	99.66	99.82	99.50	83.22	77.84	89.41	45.12	34.91	63.77
	Randomly Shuffled Graph	98.43	98.94	97.92	1.28	0.99	1.83	0.61	0.38	1.45

The real-world web graphs have characterized structures, which enables the knowledge graph embedding techniques to reasonably work well on predicting the relation labels.

Conclusion and Future Work

Conclusion

- The knowledge graph embedding techniques can be efficiently used for hyperlink embedding.
- Performance improvement can be observed by modifying the negative sampling method.

Future Work

- Utilize information from web pages or hyperlinks (such as anchor text) in hyperlink embeddings.

Thank You