When Recommendation Goes Wrong - Anomalous Link Discovery in Recommendation Networks

Bryan Perozzi*
Stony Brook University
bperozzi@cs.stonybrook.edu

Michael Schueppert
Google New York
mschueppert@google.com

Jack Saalweachter
Google New York
saalweachter@google.com

Mayur Thakur*
maythakur@gmail.com

ABSTRACT

We present a secondary ranking system to find and remove erroneous suggestions from a geospatial recommendation system. We discover such anomalous links by "double checking" the recommendation system's output to ensure that it is both structurally cohesive, and semantically consistent.

Our approach is designed for the Google Related Places Graph, a geographic recommendation system which provides results for hundreds of millions of queries a day. We model the quality of a recommendation between two geographic entities as a function of their structure in the Related Places Graph, and their semantic relationship in the Google Knowledge Graph.

To evaluate our approach, we perform a large scale *human evaluation* of such an anomalous link detection system. For the long tail of unpopular entities, our models can predict the recommendations users will consider poor with up to 42% higher mean precision (29 raw points) than the live system.

Results from our study reveal that structural and semantic features capture different facets of relatedness to human judges. We characterize our performance with a qualitative analysis detailing the categories of real-world anomalies our system is able to detect, and provide a discussion of additional applications of our method.

Categories and Subject Descriptors

D.2.8 [**Database Management**]: Database applications—*Data mining*

Keywords

anomaly detection; knowledge graph; link prediction; recommendation systems

*Work performed while at Google, Inc.

KDD '16 August 13-17, 2016, San Francisco, CA, USA

ACM ISBN 978-1-4503-4232-2/16/08.

DOI: http://dx.doi.org/10.1145/2939672.2939734

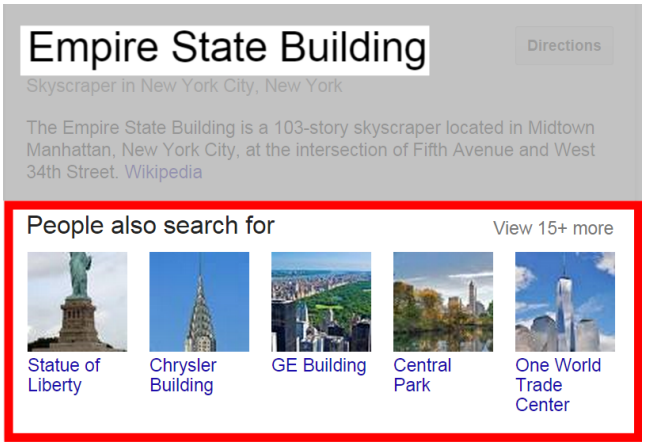


Figure 1: The *People Also Search For* feature, showing five good recommendations from the Related Places Graph for the `Empire State Building`. Prominent use of recommendations raises the risk of bad suggestions negatively affecting a user's product experience.

1. INTRODUCTION

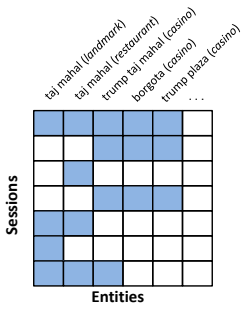
Recommendation systems have become an integral part of modern information retrieval systems. They are used to suggest almost anything to users, including places, products, publications, and on social networks - even friends.

Despite their prevalence, recommendation systems are still capable of making recommendations that users might find irrelevant or unhelpful. These bad recommendations can occur from noise in the real world processes that generate the data that they are trained on, or can be the result of a subtle dependency that the recommendation system doesn't properly model. As the size of a dataset grows, so does its long tail of less popular items, which worsens both problems. Not only do spurious correlations occur more often, but the effects of improperly modeled dependencies become more apparent. These sources of error directly affects the utility of these recommendation systems for information retrieval and content recommendation tasks.

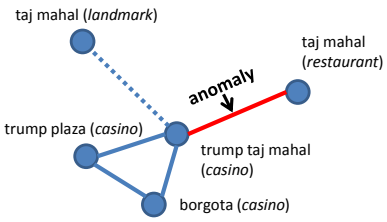
In the literature this problem is usually addressed by changing the original model to include additional features and dependencies. Unfortunately, the cost of properly engineering and validating such an enhanced model for web-scale

t	Query	Entity
1	taj mahal	`taj mahal` (*landmark*) `trump taj mahal` (*casino*) `taj mahal` (*restaurant*)
2	atlantic city casinos	`borgota` (*casino*) `trump taj mahal` (*casino*) `trump plaza` (*casino*)
3	trump plaza	`trump plaza` (*casino*) `trump taj mahal` (*casino*) `borgata` (*casino*)

(a) Hypothetical session queries



(b) Session-Entity Matrix



(c) Related Places Graph

Figure 2: Overview: The Related Places Graph is formed by observing entity co-occurrence in web sessions (2a), constructing a Session-Entity matrix (2b), and estimating an entity similarity score (2c). This graph can be filtered by distance (dashed lines), but more subtle anomalies (red) may remain.

recommendation is frequently not justified by speculative performance gains.[1] However, simply ignoring the problem is often equally undesirable as incorrect recommendations have been shown to lower a user's opinion of the system [6].

In this work, we present a secondary ranking system for the detection of such erroneous recommendations in the Google Related Places Graph, a large geographic recommendation system which provides recommendations for hundreds of millions of queries a day. Our approach detects anomalous entity recommendations by fusing semantic information from the Google Knowledge Graph with network features from the Related Places Graph. While we focus on a specific problem instance, the approach we present is general, and can be used with any similarly constructed recommendation network and knowledge graph (e.g. Freebase [5]).

To evaluate our approach, we perform what is (to our knowledge) the first comprehensive human evaluation of such an anomalous link detection system, and show that we are able to achieve relative increases of 9% to 42% in mean precision (7 to 29 raw points, respectfully) in an anomaly detection task against a very challenging baseline - the live system itself. We also perform a qualitative evaluation which illustrates the categories of anomalies our system is able to detect.

Specifically, our contributions are the following:

- **Unified Approach**: We fuse *semantic information* from a knowledge graph, with *structural features* from the recommendation network to detect bad recommendations with much higher precision (9% to 42% relative improvement in mean precision) over the base recommendation system.
- **Human Evaluation**: To capture the nuanced semantics assigned with geographic similarity, we evaluate our approach using trained human raters. Results from our study reveal that structural and semantic features capture different facets of relatedness to human judges.
- **Qualitative Analysis**: We provide a comprehensive qualitative analysis of the capabilities and limitations of our system, along with example applications for our system's output. We believe this will motivate further research in this area.

[1]For example, the winning algorithm for the Netflix Challenge was never launched into production [3].

2. RELATED PLACES GRAPH

Our dataset for this paper is the Google Related Places Graph, a very large geographic recommendation system with hundreds of millions of entities and tens of billions of similarity relations. It is used to provide pairwise geographic entity similarity results for hundreds of millions of search queries a day. In this section, we briefly describe its construction and challenges associated with web-scale geographic recommendation.

2.1 Overview

The Google Related Places Graph $G = (V, E)$ is a similarity network composed of V entities which are geolocated businesses or organizations, and E edges which encode a similarity score between them (i.e. edge $E_{ij} = s(i, j)$, the similarity between entities v_i, and v_j). It is an instance of an item-based collaborative filtering system [24], which intuitively captures how related two places are for the purposes of web search. Exact details of the similarity function are proprietary, but a number of memory and model based approaches to similar problems have been discussed in the literature [28]. The top k highest weighted outgoing edges of an entity can be used to return ranked lists of similar places for geographic recommendation. A typical result of such use is shown in Figure 1.

An outline of the Related Places Graph is shown in Figure 2. The process starts by collecting entities associated with user search sessions (2a). This is used to populate a Sessions-Entity matrix M from the set of sessions S and establishments V (2b). Finally a similarity function $s : V \times V \to \mathbb{R}$ is used to construct the recommendation network (2c).

2.2 Challenges

Unfortunately, the relations captured by the Related Places Graph are not perfect. In particular, there are a number of adverse effects which make it difficult to correctly model establishment similarity, including:

- **Distance is relative**: The distance which an individual considers two things to be related varies greatly with location and intent. A pizza restaurant two blocks away may be too far in New York City, but perfectly reasonable in rural Montana. This variance poses challenges for simple thresholding methods.
- **Frequency Imbalance**: Some entities (such as the `Empire State Building`) are much more popular than

many of their surrounding establishments. This imbalance can result in low quality similarity links for less popular establishments.

- **Geographic Sparsity**: Geographic regions with smaller populations have correspondingly lower query volume. Statistical estimates for establishments in these regions are therefore more prone to noise.
- **Conglomerate Entities**: An entity may have multiple semantic senses associated with it. For example, a grocery store might have an automated teller machine (ATM), a pharmacy, or a coffee shop in addition to food products. Composite relationships like this can greatly broaden the scope of entities considered to be related.
- **Ambiguous Queries**: Similarly, the ambiguity of natural language used for search itself poses issues. Consider the `Taj Mahal`, which is both the name of a famous mausoleum and a common name for restaurants which serve Indian cuisine. This polysemy creates superficial similarity.

Dealing with such challenges is a non-trivial task, and geographic recommendation systems are is the subject of active research [15, 33]. The most straightforward strategy is to discard all relationships with low similarity ($E_{ij} < \epsilon_S$) or high distance ($\text{dist}(i,j) > \epsilon_D$). While this thresholding strategy can mitigate some forms of errors (e.g. from frequency imbalance), it adversely effects recall (especially for establishments with low volume). Additionally, it does nothing to address other, more semantic, sources of error. These remaining errors can have an extremely negative influence on how users perceive some queries, typically when multiple semantic meanings or polysemous queries link two seemingly dissimilar places. Examples of anomalous relationships detected by our system are shown in Section 7.2.

2.3 Definition of 'Relatedness'

We note that there are many possible definitions of what might be considered a 'related place'. For the purposes of this work, we consider a place to be 'related' when it is:

- **Relevant**: A relevant recommendation captures a user's internal sense of the similarity between two places.
- **Useful**: A useful recommendation is one which is helpful if shown to users in addition to place they are searching for.

These attributes are subjective, and can vary with the type of entity. For example, whether a business is 'useful' will vary based on location (2 blocks may be too far in NYC, while ten miles may be reasonable in rural Kentucky), or its type (two amusement parks may be far away and still be related). The concept of 'relevance' is also deeply integrated with the nature of the search task which a user is performing at the time of query. For example, when searching for a hotel to stay in, a user might wish to see alternative hotels in the area. However, a user who has already chosen a hotel (and may be already staying there) might prefer to see nearby restaurants or tourist attractions.

In order to capture this relation in all of its nuance, the evaluation of our system utilizes human raters. More details are discussed in Section 5.

3. ANOMALOUS LINK DISCOVERY

Given a graph $G = (V, E)$, the Anomalous Link Discovery (ALD) problem is to find a set of anomalous edges $E_a \subset E$ which represent a deviation from the dominant patterns in G [21]. ALD is related to the link prediction problem [13], which models the evolution of the network by predicting unseen edges.

3.1 Class Imbalance

Link prediction can be viewed as classifying the set of possible edges of G to those that existent and should nonexistent. Many real world graphs are sparse, and so the existent edges (positive label) are a small fraction ($m \approx O(n)$) out of all possible edges ($O(n^2)$) This asymptotically skewed class distribution is a core challenge of link prediction, increasing the variance of link models, and making complete model evaluation computationally expensive.

Unlike link prediction, ALD focuses on the m edges which actually exist - a distinction which becomes increasingly important in sparse real world graphs. We note that this does not eliminate class skew entirely, as most graphs of interest will have more good edges than noisy ones. As such, ALD models must also account for variance arising from both the structure and higher-order semantic properties of a network.

3.2 ALD Modeling

Just as with link prediction, early approaches to ALD consisted of using individual connectedness metrics to provide a ranking of edges by their anomaly value (i.e. edges with $\text{score}(i,j) < \epsilon$ are returned as the anomalies). Modern link prediction uses supervised learning [14], which provides much more flexibility for link modeling. We apply a similar supervised approach in our ALD model, extracting multiple topological properties as features, and seamlessly combining them with additional semantic information.

Specifically, we seek to model $P(E_{ij})$, the probability of an edge found between v_i and v_j in G. We assume that edges are a function of topological properties from G, and the intrinsic semantic attributes of the nodes in our network as represented in the Google Knowledge Graph (denoted X_{KG}), and therefore concern ourselves with $P(E_{ij}|G, X_{KG})$. We assume that the input to our process is a similarity graph where edges are weighted with $E_{ij} = s(i,j)$ and that G has already been appropriately thresholded to only contain high similarity edges (i.e. $E_{ij} > \epsilon$).

4. FEATURES

In this section we discuss the features used for link modeling in our anomalous link detection system for recommendation networks. Our choice of features is based on the assumption that recommendation networks should contain strong homophily - that is they should exhibit both structural transitivity, and semantic consistency. We model structural transitivity through features derived from the network itself and model semantic consistency through features captured by the Google Knowledge Graph. For each feature we briefly discuss its motivation and outline the its construction function $f(v_i, v_j) \mapsto \mathbb{R}$.

4.1 Structural Transitivity

Our first assumption is that a recommendation network should exhibit homophily through transitivity. That is, if

entity A is related to entity B and C, then items B and C should have a much higher chance of being related. We quantify this relationship by introducing topological features from the recommendation network representing the structural connectedness of two entities. These network features are very valuable, and can implicitly capture information which has not been explicitly annotated.

4.1.1 Neighborhood

The simplest measures of structural transitivity can be derived from the neighborhoods of two vertices. Features of this variety have been widely used in link prediction[13]. We denote the neighbors of a node v by $\mathcal{N}(v)$, (i.e. $\mathcal{N}(v) = \{i; (i,v) \in E \ \vee (v,i) \in E\}$).

- **Common Neighbors**: $f(x,y) = |\mathcal{N}(x) \cap \mathcal{N}(y)|$
  The simplest link prediction feature is the size of the intersection between two neighborhoods.
- **Jaccard Index**: $f(x,y) = \frac{|\mathcal{N}(x) \cap \mathcal{N}(y)|}{|\mathcal{N}(x) \cup \mathcal{N}(y)|}$
  To avoid bias towards neighborhoods with high size, this measure normalizes the size of the intersection by the total size of the combined neighborhood.
- **Preferential Attachment**: $f(x,y) = |\mathcal{N}(x)| * |\mathcal{N}(y)|$
  In this measure, the size of each neighborhood determines the creation of a link.

4.1.2 Graph Distance

More advanced measures of structural equivalence consider information beyond the node neighborhoods.

- **Personalized PageRank**: $f(x,y) = PPR_x(y)$
  The Personalized PageRank vector[10] PPR_x captures information about probability that node y will be reached in a random walk started at node x.

We note that many additional features have been proposed for link modeling, but limit our discussion to the techniques listed here.

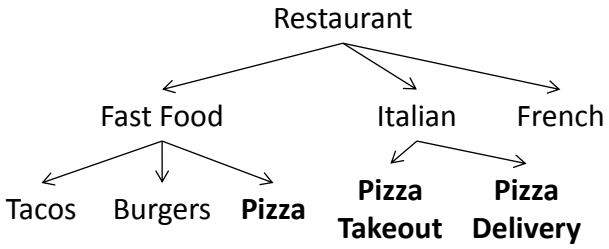
4.2 Semantic Consistency

Our second assumption is that good recommendations are those which are semantically consistent with the entity of interest. This semantic consistency varies with both the type of entity being considered and the location of the entity. For example, consider two medical practitioners: one who specializes in cardiology, and the other in pediatrics. Although they are both doctors, they might be bad recommendations for one another in an area with many doctors. However, they might well be reasonable recommendations if they were the only two doctors in town. Fine grained semantic consistency can be quite nuanced in other domains as well, such as food service. To capture these relationships, we annotated entities in our recommendation network with information from the Google Knowledge Graph.

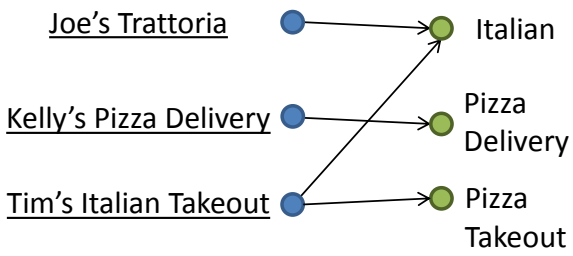
4.2.1 Knowledge Graph

The Google Knowledge Graph is a comprehensive knowledge store that contains semantic information about entities in the real world [26]. Storing information on people, places, and things, it aids in query disambiguation and summarization.

In this work, we are concerned with the subset of the knowledge graph which refers to places. For our purposes, a place is a location entity which people may wish to visit such as a business, monument, or school. Each place in the knowledge graph has an associated set of semantic categories associated with it. These categories are related to each other through a hierarchy ranging from the specific (e.g. pizza-delivery, French restaurant) to general (store). Figure 3 illustrates an example of an idealized hierarchy (3a) and the semantic labels for several similar hypothetical businesses (3b). Notice how related entities do not necessarily share the same semantic categories.



(a) Idealized Knowledge Graph Category Hierarchy



(b) Example: Three businesses which all offer pizza (left), connected to their corresponding semantic labels (right).

Figure 3: An idealized hierarchy of entity categories in the Knowledge Graph (a), showing how even a seemingly straightforward concept like *pizza restaurant* can be encoded in a number of different ways. In (b), an example of 3 hypothetical businesses which all offer pizza, but due to noise in the labeling process, have different sets of semantic categories.

The size and scope of the Knowledge Graph afford a variety of ways to create features for classification tasks. We briefly discuss some of them below:

- **Distance**: $f(x,y) = dist(x,y)$
  The distance between two locations is an important part of geographic recommendation. By itself weak (not all close businesses are related), it can be powerful when combined with other features.
- **Common Categories**: $f(x,y) = |\mathrm{Cat}(x) \cap \mathrm{Cat}(y)|$
  The categories which two entities have in common is also very relevant to recommendations. Similar to the common neighbors, this metric can also be normalized using Jaccard similarity $\left(f(x,y) = \frac{|\mathrm{Cat}(x) \cap \mathrm{Cat}(y)|}{|\mathrm{Cat}(x) \cup \mathrm{Cat}(y)|}\right)$ or related measures.
- **Categorical Hierarchy**
  In some cases, there may be no exact overlap between two entities' categories, although they are related (Figure 3b). To capture broader senses of relation, we can expand the set of categories each entity has by traversing upwards through the categorical hierarchy (Figure

3a), searching for common ancestors. The overlap between these expanded sets can be normalized based on their size, or based on the distance up the hierarchy the terms are added at.

We note that the inclusion of categorical features from the Knowledge Graph has the potential to greatly increase a model's complexity. Our approach relies on using a small number of training examples, and so we have discussed aggregating categorical information. However, as available training data grows, model complexity can be easily increased by treating each particular category (or pairwise combination of categories) as separate features.

5. EXPERIMENTAL DESIGN

Our approach to Anomalous Link Discovery uses user input to train discriminative link models. This user input is collected during the course of routine product quality evaluations. The anomalies detected by these models are then evaluated by human raters to determine model performance. In this section, we briefly discuss the process used to construct datasets and link models, the design of the experiments, and our human evaluations.

5.1 Training Dataset Construction

We use human evaluations of the similarity between two place entities as our training data. This data was collected in the course of routine product quality evaluations.

For each pair of entities, at least three different human raters were asked to judge the recommendation (on a scale of {*Not*, *Somewhat*, *Very*}) based on its *relevance* (their internal view of how analogous the two places were), and its *usefulness* (how helpful it would be to a user searching for a the original place). The distinction between relevance and usefulness is elaborated on in Section 2.3.

To convert their ratings to binary labels, a value of {1,2,3} was assigned to each judgment of {*Not*, *Somewhat*, *Very*}. An edge was labeled as relevant (or useful) when its respective average score for that category was above 2. As we assume that a good geographic recommendation is both relevant and useful, we labeled an edge as related only when judged as having both properties.

Using this process, we distilled over 28,800 human ratings into 9,600 labeled examples pairs for training. Of these 7,637 pairs were labeled as related and 1,963 labeled as not-related (a positive class label imbalance of 5:1).

5.2 Classification

Having discussed our method for generating training data, we turn to our choice of classifier for link modeling.

5.2.1 Models Considered

We compared the performance of Logistic Regression (regularized with the L_2 loss), and Random Forest Classifiers to model the probability of a recommendation link being related. We used stratified 5-fold cross validation for model selection, where Random Forests significantly out performed Logistic Regression (shown in Table 1). For the remainder of our paper, we present results using models trained with Random Forests, which we abbreviate RFC.

The baseline models we consider in decreasing order of difficulty:

- **$\text{RFC}_{\text{Related}}$**: This model uses the raw similarity score (edge weight) of the existing relatedness estimate from the Related Places Graph. This estimate is computed from a very large sample of web traffic, and is a very competitive baseline.
- **$\text{RFC}_{\text{Common}}$**: A model which considers only the common neighbors between entities.
- **$\text{RFC}_{\text{Distance}}$**: A model which considers only the distance between two entities.
- **Random**: This model simply returns the links under consideration in an random order, which illustrates the difficulty of the task.

The method under consideration are:

- **$\text{RFC}_{\text{Network}}$**: This model considers a number of network features designed to capture structural transitivity (Section 4.1).
- **$\text{RFC}_{\text{Knowledge}}$**: This model considers a number of features derived from the Knowledge Graph, in order to capture semantic similarity (Section 4.2).
- **RFC_{All}**: This model is trained on both semantic and structural features.

Classifier	Feature Set	ROC AUC
Logistic Regression	Structural	0.621
Random Forest	Structural	**0.656**
Logistic Regression	Semantic	0.696
Random Forest	Semantic	**0.723**
Logistic Regression	All	0.713
Random Forest	All	**0.778**

Table 1: Average model performance with different feature sets in stratified 5-fold cross validation.

5.3 Human Evaluation

The decision as to whether two geographic entities are related is nuanced, varying with their location, the entities, and ultimately the individual. In order to satisfactorily capture this relation, we evaluate our model's performance using human evaluation.

The human raters we use in our experiments are paid evaluators used for product quality assessment. These raters have been professionally trained using a rigorous process and guidelines. They all reside in the region where our study takes place (United States), and have a full proficiency in English. These vetted raters allow us to avoid some of the quality issues which may be present in other large scale human evaluations (e.g. Mechanical Turk [12]).

We generate test datasets for our evaluation by sampling entities from a subgraph of the Related Places Graph containing only entities located in the United States. This subgraph is large enough to be interesting (millions of nodes and billions of edges), but it eliminates some cultural variance which might complicate evaluation. Our test datasets are generated in one of two ways:

- **Uniform**: Our first test dataset is created by uniform random sampling of entities without replacement. It consists of 12,593 nodes and their top 5 best recommendations (highest weighted edges).
- **Traffic Weighted** - Our second test dataset is created by random sampling of entities in proportional to their average web traffic volume (without replace-

ment). It consists of 1,187 nodes and their top 5 best recommendations (highest weighted edges).

To evaluate the performance of a model, we score the entire test datasets, and send each of the 200 most anomalous edges (highest $P(E_{ij} = 0|G, X_{KG})$) out to three human raters. An individual rater was allowed to answer a maximum of 20 out of the 600 judgments per model, ensuring that at least 30 unique raters contributed to each assessment. The raters were instructed to perform research (e.g. visit an entity homepage, read reviews, etc.) about the two entities and then to deliver a judgment. The raters results are converted to labels (as discussed in Section 5.1), which we use to calculate the area under the precision curve.[2]

6. EXPERIMENTS

After evaluating our models through cross validation, we conducted two large scale human evaluations of feature quality from links present in the Google Related Places Graph. As described in Section 5.2, one evaluation consists of location entities which were sampled uniformly at random and the other of entities sampled in proportion to their web traffic volume.

6.1 Uniform Sampling

The results of our uniform sampling evaluation are presented in Figure 2. Here, we briefly discuss some observations in terms of *relevance* and *usefulness*.

Of all the features considered, we find that the those from the Knowledge Graph perform best for detection of links which are not relevant, initially outperforming the strong $\text{RFC}_{\text{Related}}$ baseline by 14 precision points (k=25), and narrowing to 5 at k=200. Structural features were much less competitive on this task, failing to outperform the baseline at all. A final observation is that the distance model is not able to determine relevance (as judged by humans) significantly better than random.

In contrast, we find that structural features are much better predictors of recommendations which are not useful. We attribute the strong performance of structural features for k<=50 to those features (e.g. a low number of common neighbors) which can provide strong evidence for non-relation. As k grows larger, the network structure's signal is less valuable, and performance degrades.

Both models perform well on detecting recommendations which are not related, with $\text{RFC}_{\text{Network}}$ initially beating the baseline by 42% at k=25, but again degrading as more results are returned. The joint model (RFC_{All}) performs generally well on this task, leveraging the strengths of both the structural and semantic features to beat the baseline by 37% at k=25, to 9% at k=200. However, we note that in some cases the model performs worse than its constituent parts. This occurs when the structural features provide poor discriminating power, and is a result of the highly heterogeneous phenomenon we are modeling. The performance of the joint model will improve as our system collects more human judgments.

We remark that the use of uniform sampling emphasizes entities which lie in the *long tail* of popularity. These relatively unpopularity entities have similarity estimates based on weaker statistical evidence, and are more likely to have data quality issues. Lower popularity then, impacts both structural and semantic feature quality. Our results show that even under such constraints, it is indeed possible to identify anomalous recommendations with a high degree of precision.

[2]As our task is anomaly detection, we consider the positive class label to be 0 when calculating the mean precision.

6.2 Traffic Weighted Sampling

In order to better understand the user impact of our system, we designed an evaluation where the entities were sampled in proportion to their percentage of total search volume. This experiment reflects the real world situation in which our product is used. Instead of dealing with unpopular entities, this experiment allows us to understand how our approach works when the entities have both good similarity estimates and detailed Knowledge Graph entries. This allow the study of nuanced anomalies (and not those that are simply a result of noise). The results of this evaluation are presented in Table 3.

With respect to detecting relations categorized as not relevant, we find that models using features from the Knowledge Graph ($\text{RFC}_{\text{Knowledge}}$, RFC_{All}) again perform much better than $\text{RFC}_{\text{Network}}$. We also see that the relative performance of the baseline is much stronger on this task, and that only the combination of topological and semantic features (RFC_{All}) is able to exceed it (by up to 12% at k=200)

For not useful recommendations, we again see that network features are very strong indicators of whether users consider a recommendation to be useful. When entities have good similarity estimates, the network features much more closely model user behavior. Conversely, the relatively poor performance of the Knowledge Graph features on this task can be explained by the more robust semantic similarities captured between entities. For example, when given an airport, we may recommend nearby hotels. Such a recommendation is quite useful, but may be flagged as an anomaly because it is between two very distinct semantic categories (and there was a lack of training data indicating that these types of entities are appropriate to recommend for each other). We discuss this further in Section 7.

Finally we see that the performance of structural features again carries over to detecting entities which are not related, beating the baseline by 19% at k = 25 and still by 5% at k = 200. We see that RFC_{All} initially suffers from its reliance on semantic features, but recovers and is superior to both methods by k=200. The worst performing model is $\text{RFC}_{\text{Knowledge}}$, as it can not take advantage of the enhanced similarity estimates available for popular entities.

7. DISCUSSION

Here we discuss the conclusions we have derived from our experiments, provide a qualitative analysis of the types of anomalies we detect, and discuss applications of our work.

7.1 Results

As seen in Section 6, our experiments show that our proposed approach is able to detect erroneous recommendations on a very large recommendation system with much higher precision than any of the baselines we consider. In addition to the raw performance benefits, we have draw several higher-level conclusions from our human evaluation, which we highlight here:

Structural and semantic features are not equal. Our results on the long tail of entities (Uniform Random Sampling) show

	Model	Mean Precision @k				% vs $RFC_{Related}$			
		k=25	k=50	k=100	k=200	k=25	k=50	k=100	k=200
Not Relevant	Random	0.147	0.146	0.200	0.235	-75.7	-76.3	-69.4	-63.9
	$RFC_{Distance}$	0.154	0.165	0.179	0.191	-74.5	-73.2	-72.6	-70.7
	RFC_{Common}	0.297	0.294	0.265	0.250	-50.8	-52.4	-59.4	-61.6
	$RFC_{Related}$	0.604	0.618	0.654	0.653	-	-	-	-
	$RFC_{Network}$	0.603	0.516	0.461	0.437	-0.18	-16.6	-29.5	-33.1
	$RFC_{Knowledge}$	**0.748**	**0.715**	**0.695**	**0.703**	**23.7**	**15.6**	**6.26**	**7.67**
	RFC_{All}	0.766	0.714	0.679	0.654	26.7	15.5	3.89	0.21
Not Useful	Random	0.468	0.417	0.433	0.444	-29.6	-38.1	-39.0	-37.6
	$RFC_{Distance}$	0.285	0.351	0.399	0.443	-57.2	-47.9	-43.8	-37.7
	RFC_{Common}	0.456	0.498	0.478	0.446	-31.5	-26.2	-32.6	-37.2
	$RFC_{Related}$	0.666	0.675	0.710	0.711	-	-	-	-
	$RFC_{Network}$	**0.858**	**0.807**	0.731	0.683	**28.8**	**19.5**	2.89	-3.90
	$RFC_{Knowledge}$	0.779	0.771	**0.762**	**0.785**	17.1	14.2	**7.36**	**10.4**
	RFC_{All}	0.893	0.820	0.773	0.745	34.2	21.46	8.87	4.85
Not Related	Random	0.468	0.423	0.443	0.452	-29.6	-37.3	-37.7	-36.8
	$RFC_{Distance}$	0.285	0.358	0.409	0.451	-57.2	-46.9	-42.5	-36.9
	RFC_{Common}	0.456	0.502	0.489	0.462	-31.5	-25.6	-31.3	-35.4
	$RFC_{Related}$	0.666	0.675	0.712	0.715	-	-	-	-
	$RFC_{Network}$	**0.951**	**0.867**	**0.767**	0.708	**42.8**	**28.4**	**7.85**	-0.95
	$RFC_{Knowledge}$	0.779	0.771	0.762	**0.785**	17.1	14.2	7.14	**9.74**
	RFC_{All}	0.912	0.856	0.812	0.779	37.0	26.9	14.1	8.98

Table 2: Anomaly detection results for our Uniform Random Sampling experiment. Bold indicates whether structure or semantics ($RFC_{Network}$ or $RFC_{Knowledge}$) performed best on the task. For the long tail of unpopular entities, our proposed models can predict the recommendations users will consider not related up to 42% better than our strongest baseline.

	Model	Mean Precision @k				Relative % to $RFC_{Related}$			
		k=25	k=50	k=100	k=200	k=25	k=50	k=100	k=200
Not Relevant	Random	0.145	0.152	0.171	0.179	-75.9	-74.9	-71.9	-70.8
	$RFC_{Distance}$	0.195	0.207	0.207	0.210	-67.5	-66.0	-66.1	-65.7
	RFC_{Common}	0.399	0.354	0.329	0.311	-33.5	-41.7	-46.1	-49.2
	$RFC_{Related}$	0.601	0.608	0.612	0.613	**-**	-	-	-
	$RFC_{Network}$	0.454	0.411	0.394	0.374	-24.4	-32.3	-35.5	-39.0
	$RFC_{Knowledge}$	0.580	0.593	0.592	**0.620**	-3.55	-2.54	-3.18	**1.13**
	RFC_{All}	0.560	0.618	0.653	0.692	-6.88	1.68	6.86	12.8
Not Useful	Random	0.273	0.260	0.258	0.260	-60.8	-62.6	-62.8	-62.4
	$RFC_{Distance}$	0.543	0.517	0.509	0.510	-22.2	-25.6	-26.6	-26.4
	RFC_{Common}	0.675	0.637	0.574	0.527	-3.27	-8.29	-17.1	-23.9
	$RFC_{Related}$	0.698	0.695	0.693	0.693	-	-	-	-
	$RFC_{Network}$	**0.864**	**0.819**	**0.767**	**0.722**	**23.8**	**17.9**	**10.7**	**4.25**
	$RFC_{Knowledge}$	0.649	0.672	0.677	0.697	-6.98	-3.26	-2.31	0.66
	RFC_{All}	0.682	0.725	0.751	0.769	-2.30	4.39	8.31	11.0
Not Related	Random	0.359	0.299	0.301	0.325	-50.5	-57.8	-57.0	-53.3
	$RFC_{Distance}$	0.543	0.523	0.519	0.518	-25.2	-26.1	-25.9	-25.5
	RFC_{Common}	0.675	0.637	0.579	0.536	-6.96	-10.1	-17.3	-23.0
	$RFC_{Related}$	0.725	0.709	0.700	0.696	-	-	-	-
	$RFC_{Network}$	**0.864**	**0.819**	**0.769**	**0.730**	**19.0**	**15.6**	**9.81**	**4.93**
	$RFC_{Knowledge}$	0.649	0.689	0.699	0.716	-10.53	-2.84	-0.14	2.89
	RFC_{All}	0.682	0.725	0.751	0.774	-6.03	2.36	7.24	11.2

Table 3: Anomaly detection results for our Traffic Weighted Sampling experiment. Bold indicates whether structure or semantics ($RFC_{Network}$ or $RFC_{Knowledge}$) performed best on the task. For popular entities, the combination of structural and semantic features is necessary to discover relations which are not relevant, while structural features provide a strong signal for entity pairs which are not useful.

that semantic features from the Knowledge Graph are good at identifying recommendations which are judged as not relevant, while structural features from the recommendation network are good at determining relationships which judged to be are not useful.

Network features help, regardless of location popularity. As locations receive more web traffic, we are able to build a recommendation network that more accurately models the underlying relationships. This decreases the noise of features directly derived from the network, allowing them to perform well even as the number of anomalies decreases (as in our traffic weighted experiment).

Distance alone is not enough. Many geographic recommendation systems simply include distance-based constraints or regularization to model geospatial dependencies. Our results show that these models can be improved by using structural features of the network, and semantic features of locations.

7.2 Qualitative Analysis

We have performed a qualitative analysis of the types of anomalies our approach is able to capture, which we briefly

Entity Location (*Category*)	**Anomalous Recommendation(s) in Top-5** (*Category*)	**Reason**
`IBM` New York, NY (*Software Company*)	`Victoria's Secret` (*Intimate Apparel Store*)	Nearby
`Boys and Girls Club` Orlando, FL (*Youth Organization*)	`3 Strip Clubs` (*Adult Entertainment*)	Auto-Completion & Polysemy
`Mom's Bar` Los Angeles, CA (*Bar*)	`Los Angeles County Bar Association` (*Professional Association*)	Ambiguous Phrases
`Tony's Small Engine Services` Ashland, KY (*Auto Repair Shop*)	`Academy Animal Hospital` (*Veterinarian*)	Data Sparsity
`IKEA` Canton, MI (*Home Furnishings Store*)	`Comfort Suites` (*Hotel*) `Hampton Inn` (*Hotel*) `La Quinta Inn` (*Hotel*) `Fairfield Inn` (*Hotel*) `Extend Stay America` (*Hotel*)	Unmodeled Phenomenon
`Florida Department of Agriculture` Tampa, FL (*State Government Agency*)	`Tampa Private Investigators` (*Private Detectives*) `Equip 2 Conceal Firearms` (*Gun shop*) `Shoot Straight` (*Gun shop*) `Florida Firearms Academy` (*Shooting Range*)	Conglomerate (issues firearm permits)

Table 4: Representative examples of real anomalies detected by our system, and our categorization of their cause.

discuss here. Table 4 shows a summary of representative examples of actual anomalies we have detected using our approach. Specifically, the variety of anomalies which we were able to discover included those due to:

Unmodeled Phenomena: Unmodeled interactions between entities can lead to very interesting anomalies, such as the recommendation between IKEA and hotels. We discuss how this class of anomalies can be used for targeted ontology improvements in Section 7.3.2.

Ambiguous Phrases: We are able to detect anomalies created both polysemous words and phrases (e.g. `Mom's Bar` / `Bar Association`).

Conglomerate entities Multi-sense entities can result in surprising recommendations. For example, normally one would not expect recommendations between a government entity and firearm clubs. However, as the `Florida Department of Agriculture` also issues gun permits, several such recommendation links appear. We note that such recommendations for conglomerate entities may in fact be useful to users. In such cases, (as with unmodeled phenomena) the anomalies we find can be used to improve the quality of data stored in the Knowledge Graph itself.

Data-sparsity: In some rural locations, there is not enough data for confident estimates of similarity (e.g. `Tony's Small Engine Services` / `Academy Animal Hospital`). When this happens, features from the Knowledge Graph allow detection of entities which are not related.

Mobile users & Search Completion: Finally, two sources of anomalies seemed tied to mobile users and search completion technologies. First are *auto-completion errors*. This category of errors seemed due to substring similarity between entity names. Examples include people's names to locations (e.g. `Tuscano` / `Tuscany`), and between very unrelated entity types (e.g. `Boy's and Girls Club` / `Girls Club`). Second, are *nearby entities*. This category of anomalies contained dissimilar businesses entities which are very close to each other. We suspect this behavior is due to individuals searching for information about that location (e.g. in order to plan a shopping trip).

Our qualitative analysis shows that our approach is able to address the challenges we outlined earlier in Section 2.2, and that we are able to discover anomalous links across a variety of categories, causes, and geographic locations.

7.3 Applications and Future Work

Although our work has thus far focused on a specific problem instance, we believe that the approach we present is general, and has applications to any recommendation network which has structured information available. In this section we highlight additional applications of our method which go beyond the simple removal of low-quality recommendations.

7.3.1 Entity Outlier Score

In our qualitative analysis, we have seen that anomalous links (specifically due to unmodeled phenomena, multi-sense entities, and data sparsity) tend to come in 'clumps' for an entity. A natural extension of our approach, then, is to consider the detection of anomalous entities in addition to anomalous links. We note that an outlier score for an entity may be constructed as a function of it's link probabilities, for example:

$$score(i) = \frac{\sum_{j \in \mathcal{N}(i)} Pr(E_{ij} = 0 | G, X_{KG})}{|\mathcal{N}(i)|} \quad (1)$$

Maintaining an accurate web-scale knowledge base is a very challenging endeavor, and this score could be useful for highlighting entities which have incorrect information about them (perhaps from recent changes). Entities flagged as anomalous by this measure could then be prioritized to be investigated by the relevant data quality team.

7.3.2 Targeted Ontology Improvements

When an anomalous entity is not the result of incorrect information, it can indicate that our underlying semantic

Entity	Location	Anomalous Recommendation(s) in Top-5 (*Category*)
IKEA	Brooklyn, NY	IKEA Dock (*Transportation*)
		Wall Street-Pier 11 (*Transportation*)
		New York Water Taxi (*Transportation*)
		St. George Terminal (*Transportation*)
		Hoboken Terminal (*Transportation*)
IKEA	Centennial, CO	Embassy Suites Denver - Tech Center (*Hotel*)
		Comfort Suites Denver Tech Center (*Hotel*)
IKEA	Charlotte, NC	Hilton Charlotte University Place (*Hotel*)
		Comfort Suites University Area (*Hotel*)
IKEA	West Chester, OH	Kings Island (*Amusement Park*)
		Newport Aquarium (*Aquarium*)

Table 5: Additional examples of the IKEA store anomaly, found during our qualitative evaluation. Our investigation reveals that in many areas, users treat IKEA more like a *Tourist Attraction* or *Travel Destination* than a normal furniture store.

model is not expressive enough. These data-driven improvements can result in groups of entities acquiring new semantic categories, or even changes to the hierarchical relationships between categories.

As a case study, we return to the furniture store IKEA. Table 5 illustrates some highlights from our investigation into the IKEA store anomaly found during our qualitative analysis. Interestingly, we see two things. First, in areas of high density (such as Brooklyn, NY), users focus on transportation methods to/from the store. This is understandable, as many residents of New York City do not have cars capable of transporting large furniture. Second, in areas adjacent to large rural regions, IKEA is treated as a travel destination. Top recommendations include *Hotels* and tourist attractions like *Amusement Parks* or *Aquariums*. These anomalies suggest that IKEA should be modeled differently than a traditional furniture store, perhaps with additional semantic senses (such as a *Tourist Attraction* or *Travel Destination*).

We believe that the investigative analysis of anomalies exposed by our approach will allow us to not only improve the quality of our recommendation system, but also extend the expressiveness of the Knowledge Graph itself. Such improvements can allow a better understanding of user intent and improve user experience across a variety of products.

8. RELATED WORK

The problem of detecting anomalous links touches on the domains of Link Prediction, and Recommender Systems, which we briefly discuss here.

Link Prediction models the strength of relations, usually in order to find high probability links which are missing from a network. Early work on link prediction was unsupervised, using topological features [13]. More recently, the problem has been addressed as in a supervised fashion [2, 14]. Supervision allows the blending of topological features with semantic ones, and a number of methods doing such have been proposed. Semantic features used in the literature include textual features (e.g. paper titles or keywords[2], sometimes with TF-IDF weighting [30]), location features, [7, 18], or social interactions [32]. Other recent work examines using community information [27] or transferability of models across networks [29]. More information is available from several surveys [9, 16, 22].

Although there has been considerable work on link prediction, the vast majority of the literature deals with the discovery of non-existent links and not the detection and removal of anomalous ones. The anomalous link discovery problem was introduced by Rattigan and Jensen [21], who noted that topological measures of similarity performed well for anomalous co-authorship detection. Relatively little work followed. In [11] Huang and Zeng examined anomalous links in an email network, and Ding et al. [8] examined clustering and betweenness centrality for detecting links that represented network intrusion. Our work helps address this gap in the literature by analyzing the performance of an anomalous link discovery system in a very large industrial setting. Furthermore, our comprehensive human evaluation is the first such human evaluation (to our knowledge) of any anomalous link detection approach.

More recent work in graph anomaly detection has focused on discovering anomalous communities [19, 20]. A comprehensive survey is available from Akoglu et al. [1].

Recommender Systems model user/item interactions to suggest additional content to users. Similar to link prediction, the focus is on finding highly confident recommendations (i.e. those which future users will have the highest likelihood of interacting with). Item based filtering was introduced in [24]. Since then, entity recommendation has been proposed for use in search in a variety of different settings [4, 34, 35]. Removal of anomalous recommendations is typically performed implicitly during model improvement, usually through the inclusion of additional features to the model , and several recent works target geographic recommendation [15, 33].

We view our work as complementary to that of the existing recommender system development process for two reasons. First, by explicitly modeling the detection of anomalous links, our approach provides an insightful view of what is being effectively captured by a recommendation system. This analysis is quite useful for understanding the limits of an existing system, and where likely areas of the best improvements are (e.g. Qualitative Analysis in Section 7.2). Second, it is sometimes impossible to replace an entire recommender system used in production as substantial model changes require validation (which can be both expensive and time consuming). Our lightweight modeling of anomalous link discovery approach allows for high precision *focused changes* changes to existing models, which can allow for easier validation.

Finally, we note that the disambiguation of user intent from keyword search is a mainstay of research in information retrieval [17, 23, 25, 31]. We believe that our work helps further the discussion on this important problem.

9. CONCLUSIONS

In this work, we have proposed and evaluated a unified approach for anomalous link discovery in recommendation systems. Given a recommendation system, we treat its out-

put as a network and extract graph features which quantify the structural transitivity present in the recommendations. We fuse this information with features constructed from an appropriate knowledge graph, which capture the semantic consistency of the entities.

Experiments on the Google Related Places Graph using human raters show the effectiveness of our approach on a very large geographic recommendation system. Interestingly, our experiments also show that structural features from the recommendation network capture a sense of a recommendation's *usefulness* to users, while semantic features better capture a sense of the *relevance* of a recommendation.

In addition to strong quantitative results, our qualitative analysis illustrates how the anomalies exposed by our system provide a valuable lens to study a recommendation system's behavior. Investigating such anomalies can enhance understanding of user intent and has the potential to improve user experience across all information retrieval systems leveraging the same knowledge graph.

Acknowledgments

We thank the anonymous reviewers for their comments.

References

[1] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3), 2015.

[2] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *SDM'06: Workshop on Link Analysis, Counter-terrorism and Security*, 2006.

[3] X. Amatriain and J. Basilico. Netflix recommendations: Beyond the 5 stars (part 1). http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html, May 2012.

[4] R. Blanco, B. B. Cambazoglu, P. Mika, and N. Torzec. Entity recommendations in web search. In *The Semantic Web–ISWC 2013*, pages 33–48. Springer, 2013.

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. SIGMOD'08, pages 1247–1250, 2008.

[6] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl. Is seeing believing?: How recommender system interfaces affect users' opinions. CHI '03, pages 585–592, 2003.

[7] D. J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences*, 107(52):22436–22441, 2010.

[8] Q. Ding, N. Katenka, P. Barford, E. Kolaczyk, and M. Crovella. Intrusion as (anti)social communication: Characterization and detection. KDD '12, pages 886–894, 2012.

[9] M. Hasan and M. Zaki. A survey of link prediction in social networks. In C. C. Aggarwal, editor, *Social Network Data Analytics*, pages 243–275. Springer US, 2011.

[10] T. Haveliwala, S. Kamvar, and G. Jeh. An analytical comparison of approaches to personalizing pagerank. 2003.

[11] Z. Huang and D. Zeng. A link prediction approach to anomalous email detection. volume 2 of *SMC '06*, pages 1131–1136, Oct 2006.

[12] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. HCOMP '10, pages 64–67, 2010.

[13] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[14] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. KDD '10, pages 243–252, 2010.

[15] B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning geographical preferences for point-of-interest recommendation. KDD '13, pages 1043–1051, 2013.

[16] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

[17] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. SIGIR '98, 1998.

[18] J. O'Madadhain, J. Hutchins, and P. Smyth. Prediction and ranking algorithms for event-based network data. *SIGKDD Explor. Newsl.*, 7(2):23–30, Dec. 2005.

[19] B. Perozzi and L. Akoglu. Scalable anomaly ranking of attributed neighborhoods. SIAM SDM, 2016.

[20] B. Perozzi, L. Akoglu, P. Iglesias Sánchez, and E. Müller. Focused clustering and outlier detection in large attributed graphs. KDD, 2014.

[21] M. J. Rattigan and D. Jensen. The case for anomalous link discovery. *SIGKDD Explor. Newsl.*, 7(2):41–47, Dec. 2005.

[22] R. A. Rossi, L. K. McDowell, D. W. Aha, and J. Neville. Transforming graph data for statistical relational learning. *J. Artif. Int. Res.*, 45(1):363–441, Sept. 2012.

[23] M. Sahami and T. D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. WWW '06, 2006.

[24] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. WWW '01, pages 285–295, 2001.

[25] M. Shokouhi, M. Sloan, P. N. Bennett, K. Collins-Thompson, and S. Sarkizova. Query suggestion and data fusion in contextual disambiguation. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, 2015.

[26] A. Singhal. Introducing the knowledge graph: things, not strings. http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html, May 2012.

[27] S. Soundarajan and J. Hopcroft. Using community information to improve the precision of link prediction methods. WWW '12 Companion, pages 607–608, 2012.

[28] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, Jan. 2009.

[29] J. Tang, T. Lou, and J. Kleinberg. Inferring social ties across heterogenous networks. WSDM, 2012.

[30] C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. ICDM '07, pages 322–331, Washington, DC, USA, 2007. IEEE Computer Society.

[31] X. Wang and C. Zhai. Mining term association patterns from search logs for effective query reformulation. CIKM '08, 2008.

[32] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. WWW '10, pages 981–990, 2010.

[33] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. SIGIR '11, pages 325–334, 2011.

[34] X. Yu, H. Ma, B.-J. P. Hsu, and J. Han. On building entity recommender systems using user click log and freebase knowledge. WSDM '14, pages 263–272, 2014.

[35] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. Personalized entity recommendation: A heterogeneous information network approach. WSDM '14, pages 283–292, 2014.