Learning to rank related entities in web search

Changsung Kang Dawei Yin[1] Ruiqiang Zhang
Nicolas Torzec Jianzhang He Yi Chang
Yahoo! Labs

Abstract

Entity ranking is a recent paradigm that refers to retrieving and ranking related objects and entities from different structured sources in various scenarios. Entities typically have associated categories and relationships with other entities. In this work, we present an extensive analysis of Web-scale entity ranking, based on machine learned ranking models using an ensemble of pair-wise preference models. Our proposed system for entity ranking uses structured knowledge bases, entity relationship graphs and user data to derive useful features to facilitate semantic search with entities directly within the learning to rank framework. We also describe a suite of novel features in the context of entity ranking and present a detailed feature space analysis. The experimental results are validated on a large-scale graph containing millions of entities and hundreds of millions of entity relationships. We show that our proposed ranking solution clearly improves simple user behavior based ranking and several baselines.

Keywords: entity ranking, structured data, semantic search, object ranking

1. Introduction

The focus of current Web search engines is to retrieve relevant documents on the Web, and more precisely documents that match with the query intent of the user. Some users are looking for specific information, while other just want to access rich media content (images, videos, etc.) or explore a topic. In the latter scenario, users do not have a fixed or pre-determined information need, but are using the search engine to discover information related to a

[1]Corresponding author

particular object of interest. In this scenario one can say that the user is in a "browse" mode.

To support users in their exploratory search, we propose a machine learned framework for ranking related entities. This framework ranks related entities according to two dimensions: a lateral dimension and a faceted dimension. In the lateral dimension, related entities are of the same nature as the entity queried (e.g. Hyderabad and Bangalore, or Angelina Jolie and Brad Pitt). In the faceted dimension, related entities are usually not of the same type as the queried entity, and refer to a specific aspect of the queried entity (e.g. Hyderabad and India, or Brad Pitt and Fight Club). Entity ranking, is a recent paradigm (1; 2; 3) that focuses on retrieving and ranking related entities from different (structured) sources. Entity ranking can occur in various forms and scenarios as proposed in (4; 5). Entities typically have a canonical name, a main type, alternate names and several subtypes. They are related to each other through labelled relationships (e.g. Bengalore IsLocatedIn India, or Brad_Pitt CastsIn Fight_Club). This kind of information can be represented as an Entity-Relationship graph, which shows many similarities to the graphs underlying social networks (6).

The amount of structured data sources such as DBPedia[2] and Freebase[3] on the Web are increasing (7). The availability of such large collections of structured data enables a realm of possibilities beyond the basic textual Web search. Popular Web search engines are already providing a rich experience, mining structured data, query logs and web documents to provide rich information in the search results (e.g. movie plot, genre, cast, review and show times at the user location) or direct answers to the users (e.g. "date of birth Brad Pitt"), while displaying related news articles, images, videos and tweets for queries about popular persons, organizations, media works and locations whenever possible.

We propose to enhance this experience by providing well-qualified related entities. A snapshot of the overall experience we propose can be seen in Figure 1. Related entities, which are points of interest for the query 'India' in this example, are shown as suggestions on the left-hand rail of the search results page. The types of the related entities to show depend upon the category of the query entity. For example, for movie queries, the goal is to show

[2]http://dbpedia.org

[3]http://www.freebase.com

both lateral information in terms of related movies and faceted information in terms of cast information. The challenge that we propose to address in this work is to select the appropriate set of related entities depending upon the queried entity and its type, and to rank them in the order of relevance.

We cast the problem of entity ranking as a supervised machine learning problem (8; 9) with the goal of predicting the relevance of the related entity to the query entity. While the previous work in this area (10; 11) focuses on optimizing the Click Through Rate (CTR) of the related entities alone, we present an approach to jointly learn the relevance among the entities using both the user click data and the editorially assigned relevance grades. In contrast to web search, the entity search results are grouped by categories of related entities, which complicates the ranking problem. We address how to incorporate the categories of related entities into the loss function and show how to leverage relationships between related entities with different categories ("inter-category" relationships) to improve relevance.

In this work, we present an extensive analysis of Web-scale object ranking, based on machine learned ranking models using ensemble of pairwise preference models. Our proposed system for entity ranking uses structured knowledge bases, entity-relationship graphs and user data to derive useful features to facilitate semantic search with entities directly within the learning to rank framework. We also describe a suite of novel features in the context of entity ranking and present a detailed feature space analysis. We further discuss how entity ranking is different from regular Web search in terms of presentation bias and the interaction of categories of query entities and result facets. The experimental results validated on a large-scale graph containing millions of entities and hundreds of millions of relationships show that our proposed ranking solution clearly improves a simple user behavior based ranking model.

Our main contributions of this work can be summarized as follows:

- extensive feature analysis of Web-scale object ranking, based on several structured and semi-structured data sources.
- learning pairwise preferences from multiple click models to obtain more robust pairwise preferences than a single click model and enriching the training data with high-confidence pairwise preferences.
- incorporate the categories of the related entities into the loss function of the ranking model to improve relevance.

- learn the category preference of the related entities based on the category of the query entity from user behavior data.

The rest of the paper is organized as follow. In section 2, we provide background details about the terminology that we use, describe the knowledge base and the Web search experience where entity ranking is used. Section 3 describes the extensive features that we utilize based on both various data sources and the entity-relationship graph itself. Section 4 provides details about the pairwise comparison model that produces highly robust pairwise preferences and describes how to incorporate category information in the loss function. Section 5 presents the experimental results obtained on a large collection of structured knowledge sources and Section 6 concludes the paper with some directions about the future work.

2. Background

In this section, we describe the terminology that we use throughout the rest of the paper, give an overview of the system used for building the knowledge base that supports the whole experience, and describe the Web search experience where this entity ranking is utilized.

2.1. Terminology

This section introduces the terminology used in this paper. The application of entity ranking, as presented in this paper, is to support users in Web search by providing related entity information, which allows them to explore a topic of their interest. Given a **user query (q)** entered in the search box and a large knowledge base of known entities with binary directed relationships between them, we detect entities present in the query. We refer to such an entity as the **query entity (qe)**. A **facet (f)** is defined as the directed relationship between two entities, starting from the query entity to a **facet entity (fe)**. For popular entities we typically have in the order of several hundreds of facets in our knowledge base, and a few dozen facets for the infamous entities. Figure 2 shows the distribution of the facets per entity in our knowledge base.

For each entity the following information is maintained. The *reference* is used internally to identify and manage the entity. The *canonical name* is shown to the user when the entity ranking experience is triggered in the search engine results page. The *type* indicates the semantic class of the

Table 1: Example entity

reference	ID::286186
canonical name	India
variant	India; Bharat
type	location
subtype	country

Table 2: Example facet

query entity	ID::286186 (i.e. India)
facet entity	ID::2295414 (i.e. city of Hyderabad, India)
type	has_point_of_interest

entity, that is whether the entity is a person, a location, a media work, etc. The *subtypes* provide a more fine grained typology for the entity. An entity can have several subtypes. For example a person can be both an actor and a producer. In addition, an entity can have one or more *variants* (e.g. alternate names, birth names, aliases) which capture colloquial references to this entity. We assume that problems related to de-duplication of known entities with identical canonical names and types are resolved within the knowledge base, as well as the handling of other disambiguation problems. For ease of reference when computing a ranking, we assume that an entity can be uniquely identified through its *normalized* canonical name, and type.

For each facet the following information is maintained: the reference to the query entity, the reference to the facet entity, as well as the relationship type and how frequently we observe that relationship in the sources feeding our knowledge base. Typically multiple facets can be defined between an **entity-facet pair** $(f(qe, fe))$, reflecting the different roles that can occur between any two entities.

To illustrate the above with an example, consider the location *India*, table 1 shows the typical data that we would have on file for this particular location. When detecting this entity in any of our ranking sources both the canonical name, and its variants are used as references for this entity. India is a *location*, of subtype *country*.

Table 2 shows the information stored for a facet, which simply contains a reference to both entities and the type of the relationship.

2.2. Knowledge Base

As already mentioned, the type of experience described in this paper relies on a knowledge base of entities and the relationships between them. The construction of the first version of that knowledge base is described in (11). The present section describes the new knowledge base, and its building process.

The system we have designed and implemented for building this knowledge base is called Yalinda. Yalinda extracts various form of knowledge - including entities, their attributes (i.e. reference, canonical name, variants, type, subtypes, other attributes) and the relationships between them (i.e. labeled directed binary relationships). Yalinda extracts this knowledge from structured sources. Selected sources include both internal specialized sources, such as Yahoo! properties (e.g. Y! Movies, Y! Sport, Y! TV, Y! Music, Y! GeoPlanet, etc.), as well as broad-coverage reference sources such as online collaborative encyclopedias (e.g. Wikipedia, Freebase). This extraction is done automatically and frequently, using data scraping and data feed processing techniques. The extracted knowledge is normalized and serialized into semantic graphs, one per input source and domain, providing unified views convenient for consumption. Yalinda is implemented as a framework consisting of general modules providing the common features and pluggable modules providing special features such as wrappers for specific data. It follows a three-step process - the data acquisition step, the knowledge extraction step and the knowledge serialization step - which is described below.

Regarding input source selection, a typology of potential sources has been defined, and potential sources have been reviewed and analyzed regarding practical knowledge extraction. In order to maximize the tradeoff between precision, coverage and cost, the focus has been set on extracting knowledge from large high-quality structured sources. Depending on the source, the knowledge base is updated daily, weekly or quarterly.

In the data acquisition step, new data is retrieved from remote locations and made available locally in a standard processable way, to ease extraction. Main challenges include dealing with various protocols, APIs, encodings and formats. Sometime input data must also be retrieved and combined from several sources to form a convenient input data set. Resulting data and metadata are stored locally, as structured data feeds. In the knowledge extraction step, entities, attributes and relationships are extracted from the data feeds, and normalized into the corresponding canonical names. Entities and associated facts are extracted using wrappers specific to a schema or

format. Entities and their attributes are normalized according to their semantics using rules, focusing on the main attributes and the attributes that can be used as pivot for building relationships. Challenges depend on the source processed. In the knowledge serialization step, extracted knowledge is refined and serialized into Entity-Relationship graphs. The main challenge is to identify and model the meaningful relationships, and to materialize them using specific attribute values as pivots.

Overall, the resulting knowledge base include millions of entity instances (100+ fine-grain types) and hundreds of millions of relationship instances (300 fine-grain relationship types, including both 1st order and second order relations). The domain currently covered include Automotive, Book, Finance, Movie, TV, Music, Notability, Periodical, Product, Sport, etc. For popular entities we typically have in the order of several hundreds of relationships in our knowledge base, and a few relationships for the infamous entities.

2.3. Web search experience

The research presented in this paper is powering the faceted search experience in Web Search. Figure 1 depicts a screen shot of the experience, where the user has searched for *India*. The Web search results page is organized into three columns, where the left column is used to present entity ranking results whenever the query contains entity terms, the middle column contains the traditional web search results along with their snippets that show a summary of the match between the Web page and the query, and the right column is used to display relevant ads if the query contains commercial intention. In addition to the three column layout, rich results are embedded in the middle column above the traditional Web search results whenever the corresponding structured data is available, depending upon the type of the query.

Though our ranking strategy blends the facets entities of different types, when shown to the user, the facets are grouped by their type to enhance the user comprehension. In addition, we show a mini-thumbnail for each facet to aid the user, and capture their attention. Both aspects are variable across different queries, and will affect the user engagement. When training and evaluating the ranking strategies presented here, we deploy click-through behavior as well as editorial assessments. The latter allows us to eliminate any bias in the evaluation, with respect to these two variables.

3. Feature Space Analysis

In our previous work, we have introduced a probabilistic feature framework, that allows us to quickly derive a pool of features from various rankings sources, such as web search query logs, Flickr, and Twitter. In Section 3.1, we'll give a brief overview of these features. In addition to that, we experiment with a new set of features, based on the analysis of the entity graph that forms our knowledge base in Section 3.2.

3.1. Probabilistic Feature Framework

As introduced in van Zwol et al. (12), we have setup a framework to uniformly compute a set of features from various ranking sources. In this paper, we use the framework to compute the features over Web search query logs, tags used to annotate photos in Flickr, and entity pairs detected in tweets from Twitter users. For each source we can compute both term based, and session based statistics.

The features can be classified in four groups:

- Atomic features that work on one of the entities in the facet (qe, fe), for example the entity probability, or its entropy.
- Symmetric features such as the point-wise mutual information and joint probability.
- A-symmetric features like the conditional probability and KL-divergence.
- Combinations of features like $P_u(f|e) * P(f)$ that combine the conditional (user) probability of a facet f given entity e and the probability of the facet. According to Skomrow and Araki (13), this allows the learning algorithm to make a more informed decision, if the combined feature is more descriptive.

The corpus of Web documents is a valuable source to compute the similarity among related entities. All of the probabilistic features described above can be computed within the context of Web pages. In fact a simple approximation of these corpus based features can be computed by retrieving the number of documents which contain the entity alone, facet alone, and both the entity and facet together. These co-citation features for the entity and the facet are computed from the web search results as total hits and deep hits.

3.2. Graph-based Entity Popularity Feature

An entity-facet pair $f(qe, fe)$ illustrates the relation between a query entity and a facet entity. We can deduce a whole entity network over all the entities if we connect all the pairs. The network can be built by simply connecting the facet of one pair to the entity of another pair if the two are of the same surface form. Figure 4 shows a subnet of the network, centered around "Angelina Jolie". Labeled nodes represent entities, "Angelina Jolie", "Brad Pitt", "Troy". A direct connection between "Angelina Jolie" and "Brad Pitt" denotes an entity-facet pair $f(qe, fe)$. But the entity "Angelina Jolie" and "Troy" is related through "Brad Pitt".

The entity network is very similar to a social network. Each node in the social network refers to a user, while this equals an entity in the entity network. We can extract many features from the entity network which are useful in the context of entity ranking. Some obvious features include the shortest distance between two entities, number of paths between two nodes, and the number of shared connections. The concept of shared connections is inspired from the idea of mutual friends in social networks (6). The intuition is that if two entities have many shared nodes or connections in the entity graph, they are more related to each other. We utilize normalized shared connections are various depths as features in our framework.

In addition to these graph based features, we incorporated another feature based on the entity popularity on the entity network. The intuition is that more popular entities are more likely to be eye sparking and more often clicked by users. Mathematically, we represent this graph as a $m \times m$ adjacency matrix, $\mathbf{W}$, where $W_{ij} = 1$ if entity i connects to entity j. In practice, we normalize $\mathbf{W}$ so that $\sum_j W_{ij} = 1$. Given this matrix and an eigen system, $\mathbf{W}\pi = \lambda\pi$, the eigenvector, π, associated with the largest eigenvalue, λ, provides a natural measure of the centrality of the user (14). The analog in web search is the PageRank of a document (15). This eigenvector, π, can be computed using power iteration,

$$\pi_{t+1} = (\lambda\mathbf{W} + (1 - \lambda)\mathbf{U})\pi_t \tag{1}$$

where $\mathbf{U}$ is a matrix whose entries are all $\frac{1}{m}$. The interpolation of $\mathbf{W}$ with $\mathbf{U}$ ensures that the stationary solution, π, exists. The interpolation parameter, λ, is set to 0.85. We use perform fifteen iterations (i.e. $\tilde{\pi} = \pi_{15}$).

We computed $\tilde{\pi}$ for four million entities. In Table 3 the top 6 entities associated with the highest values of $\tilde{\pi}_i$ are listed.

Table 3: Entity popularity feature values of top entities

entity	$\tilde{\pi}_i$
law and order	-8.24
er	-8.43
strong medicine	-9.27
bill kurtis	-9.32
las vegas	-9.81
michael mckean	-9.81

4. Machine-learned Ranking for Entities

Machine learning has been extensively used for many ranking tasks (8). We follow the gradient boosted decision trees framework applied to pairwise preferences (9), which has been successfully used for some ranking problems. In our entity ranking problem, there are three main challenges:

- There is small amount of editorial data, which is a common situation when developing a ranking function for a new domain.
- User clicks are sparse and very noisy. Since entity search results are shown along with web search results, clicks are considerably fewer compared to web search. Also, the thumbnails for entities add a strong bias, which leads to very noisy clicks.
- Entity search results are grouped by categories of facets, which complicates the ranking problem. This also requires a new problem definition for learning a ranking function.

To overcome the problem of sparse editorial data, we propose to augment the training data using the click-through data (in Section 4.3). However, without a proper mechanism to deal with sparse and noisy clicks, the augmented training data would not produce a robust ranking function. Hence, we propose a method of combining multiple click models to tackle the problem of sparse and noisy clicks (in Section 4.2). To deal with categories of facets, we propose a new loss function that utilizes the category information (in Section 4.3).

4.1. Problem Definition

In contrast to web search, the entity search results are grouped by categories of facets. For example, for a movie actor entity, a group of person

facets are first shown in a group and a group of movie facets follow. The order of these groups is pre-determined by user behavior (Section 5.2) based on the category of the query entity.

In the web search ranking problem, a set of n documents $\mathcal{D}$ is given as input and a permutation τ of $\{1, \ldots, n\}$ is returned as output. In our entity ranking problem, $\mathcal{D}$ is split into a set of groups by categories of results: $\mathcal{D} = \mathcal{D}_1 \cup \ldots \cup \mathcal{D}_m$ where $\mathcal{D}_i$ contains results with category i. Our goal is to generate a permutation or ranking τ_i for each category i.

A straightforward approach would be to train a ranking function for each category separately using only the subset of training data for the category. However, there are two problems: i) The training data for each category may be too small to train a robust ranking function (even the entire training data available is small). ii) We may loose some useful relationships between facets with different categories. For example, suppose that $f \in \mathcal{D}_1$ and $f', f'' \in \mathcal{D}_2$. If we have "inter-category" relationships in our training data that $f' > f$ and $f > f''$, these may be leveraged to provide an "intra-category" constraint $f' > f''$. In this way, "inter-category" relationships may help "intra-category" ranking. Thus, we propose to generate a single ranking τ for the whole $\mathcal{D}$ to leverage these "inter-category" relationships. The ranking τ_i for each category i is then derived from τ by simply ordering $\mathcal{D}_i$ according to τ.

4.2. Pairwise Comparison Model

When generating pairwise preferences as additional training data using click models (16; 17), the high accuracy of the pairwise preferences is necessary to learn a robust ranking function. However, in our entity ranking problem, a common way of generating the pairwise preferences (16; 17) may not work due to sparseness and noisiness of user clicks.

In this section, we introduce an approach, called the *pairwise comparison model (PCM)* that learns robust pairwise preferences for entity pairs based on *pairwise* click features.

Some click models (16; 17) have been used to enrich the training data for the boosting algorithm (9) in two steps:

- First, they compute a relevance score for each (query, url) pair.
- Second, the pairwise preference between two urls is decided by the relevance scores of the two urls: a facet f_i is preferred to a facet f_j if $r(\mathbf{x}_i) > r(\mathbf{x}_j)$ where r is a click model and $\mathbf{x}_i$ is a feature vector for f_i.

The second step raises some questions. First, this method relies on a single click model r to generate pairwise preferences. If the accuracy of the click model is not sufficiently high (due to noisiness of clicks in our data), the generated preference data may not improve a ranking function when it is added to the editorial data. Thus, it arises the question whether we may leverage multiple click models, which can possibly complement each other to get more reliable preferences. Second, in this method, the pairwise preferences are indirectly derived from "pointwise" scores ($r(\mathbf{x}_i)$ and $r(\mathbf{x}_j)$). This motivates us to design a model that directly predicts a preference between two facets.

We propose the *pairwise comparison model (PCM)*, which takes a "pairwise" feature vector as input and predicts a preference. Given two facets f_i and f_j, we extract a pairwise feature vector $\mathbf{w}_{ij}$. Then, the pairwise comparison model h is applied to $\mathbf{w}_{ij}$ to obtain the preference between f_i and f_j: f_i is preferred to f_j if $h(\mathbf{w}_{ij}) > 0$. The key insight is that we use the responses of multiple click models as features ($\mathbf{w}_{ij}$) and train a model (h) using them. We first describe how we extract a pairwise feature vector $\mathbf{w}_{ij}$ for two facets. Then, we show how we train the pairwise comparison model.

Some pairwise features can be derived from two facets. For each (entity, facet i, facet j) tuple, we have the following pairwise features.

- SkipAbove$_{ij}$: ncc_{ij}/cnc_{ij} for the click sessions in which the facet i is ranked higher than the facet j where ncc_{ij} is the number of sessions in which facet i was not clicked but facet j was clicked and cnc_{ij} is the number of sessions in which facet i was clicked but facet j was not clicked.

- SkipNext$_{ij}$: cnc_{ij}/ncc_{ij} for the click sessions in which the facet i is ranked one position higher than the facet j.

Also, we have some features derived from each facet. For each (entity, facet) pair, we have the following pointwise features.

- CTR

- skipCTR : $\#clicks/(\#clicks+\#skips)$ where $\#clicks$ denotes the number of sessions where a facet f was clicked and $\#skips$ denotes the number of sessions where f is not clicked but some other facets ranked below f are clicked. skipCTR is a reasonable approximation of the DBN model score (16).

- Cumulated relevance (Cumrel) (17) : a state-of-the-art click model that estimates the relevance of a document based on user behavior.

Although these features are pointwise ones, the concatenation or ratio of two pointwise features can be considered as a pairwise feature. We define the feature vector for each (entity, facet i, facet j) as follows.

$$\begin{aligned}\mathbf{w}_{ij} = (&SkipAbove_{ij}, SkipAbove_{ji}, SkipNext_{ij}, SkipNext_{ji},\\ &CTR_i, CTR_j, skipCTR_i, skipCTR_j, Cumrel_i, Cumrel_j,\\ &CTR_i/CTR_j, skipCTR_i/skipCTR_j, Cumrel_i/Cumrel_j)\end{aligned}$$

Given all these pairwise features, we have the following training data for each training entity e:

$$\mathcal{T}_e = \{(\mathbf{w}_{ij}, l_i - l_j) \mid i, j \in \{1, \ldots, N\}, i \neq j\}$$

where l_i is a numerical label given by human editors to facet i out of a finite set of labels L (e.g., $L = \{4, 3, 2, 1, 0\}$) and N is the number of facets to be ranked for e. We choose $N = 10$ to get enough click information among the facets and restrict the size of the training data.

We apply the gradient boosting algorithm (18) on our training data $\{\mathcal{T}_e \mid e \text{ is a training entity}\}$ to obtain a function $h(\mathbf{w}_{ij})$ which predicts the relative relevance of two facets f_i and f_j.

4.3. Training Ranking Function

In this section, we propose how to incorporate facet categories in the loss function to learn a ranking function. We start with a simple loss function that ignores facets categories and then show a new loss function incorporating facet categories.

The boosting algorithm (9) uses pairwise preferences as input to learn a ranking function. We have two sets of pairwise preferences:

- $\mathcal{P}_E = \{(f_i, f_j) \mid l_i > l_j\}$ where l_i is a numerical label given by human editors to a facet f_i (the larger, the more relevant).

- $\mathcal{P}_C = \{(f_i, f_j) \mid h(\mathbf{w}_{ij}) > \lambda\}$ where h is the pairwise comparison model described in Section 4.2 and λ is a threshold to obtain reliable preferences.

For each (entity, facet) pair, we extract a feature vector $\mathbf{x}$ containing all the features described in Section 3. The boosting algorithm optimizes the following loss function:

$$\frac{1-\delta}{|\mathcal{P}_E|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{P}_E} \max(0, 1-(f(\mathbf{x}_i)-f(\mathbf{x}_j)))^2 + \\ \frac{\delta}{|\mathcal{P}_C|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{P}_C} \max(0, 1-(f(\mathbf{x}_i)-f(\mathbf{x}_j)))^2 \tag{2}$$

where δ is a parameter that controls the balance between the two sets. $f(\mathbf{x})$ here is the predicting function of gradient boosting decision trees.

Note that this loss function ignores facet categories. We now introduce a new loss function that considers facet categories. First, we define some notation. $\mathcal{P}_E$ and $\mathcal{P}_C$ can be split into two sets:

$$\mathcal{P}_E = \mathcal{P}_E^{inter} \cup \mathcal{P}_E^{intra}$$
$$\mathcal{P}_C = \mathcal{P}_C^{inter} \cup \mathcal{P}_C^{intra}$$

where

$$\mathcal{P}_E^{inter} = \{(f_i, f_j) \mid (f_i, f_j) \in \mathcal{P}_E, \text{category of } f_i \neq \text{category of } f_j\}$$
$$\mathcal{P}_E^{intra} = \{(f_i, f_j) \mid (f_i, f_j) \in \mathcal{P}_E, \text{category of } f_i = \text{category of } f_j\}$$
$$\mathcal{P}_C^{inter} = \{(f_i, f_j) \mid (f_i, f_j) \in \mathcal{P}_C, \text{category of } f_i \neq \text{category of } f_j\}$$
$$\mathcal{P}_C^{intra} = \{(f_i, f_j) \mid (f_i, f_j) \in \mathcal{P}_C, \text{category of } f_i = \text{category of } f_j\}.$$

The new loss function is

$$\frac{\alpha(1-\delta)}{|\mathcal{P}_E|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{P}_E^{inter}} \max(0, 1-(f(\mathbf{x}_i)-f(\mathbf{x}_j)))^2 + \\ \frac{1-\delta}{|\mathcal{P}_E|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{P}_E^{intra}} \max(0, 1-(f(\mathbf{x}_i)-f(\mathbf{x}_j)))^2 + \\ \frac{\alpha\delta}{|\mathcal{P}_C|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{P}_C^{inter}} \max(0, 1-(f(\mathbf{x}_i)-f(\mathbf{x}_j)))^2 + \\ \frac{\delta}{|\mathcal{P}_C|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{P}_C^{intra}} \max(0, 1-(f(\mathbf{x}_i)-f(\mathbf{x}_j)))^2.$$

In this new loss function, we introduce a parameter α that controls the weight for "inter"-category pairs of facets. If $\alpha = 1$, the new loss function is equivalent to (2). If $\alpha = 0$, we are considering different groups of facets as if they are from different queries. α between 0 and 1 may help inter-category ranking, which is empirically shown in Section 5.

5. Experimental Results

In this section we present experimental results to validate our approach. Two types of experiments are conducted to validate our algorithms. The first set of experiments are to evaluate the performance of our approach on editorially judged entity pairs and the second set of experiments use the user behavior data on the search results page to compute the efficacy of our approach.

5.1. Experimental Setup

We use the query log from Yahoo! Search Engine to sample entity queries that match from our dictionary of entity names. For each of these entities, we extract the related entities from its connections in the entity graph. For all of these related entities, a five-point relevance grade is obtained that indicates the match between the query entity and the facet entity. A feature vector is computed for each of these relationships. The data set for our experiments consists of the pair of entities that are related, the relevance grade that indicates the match between the entities and the feature vector.

Our data set consists of 6000 query entities and overall 33000 entity-facet pairs including both training and test data (50% training and 50% test). These entity-facet pairs were given a five-point editorial grade that indicates the relevance of the facet entity to the query entity. The inter-editor agreement among the editors for this study were close to 78%. Since the relevance of the facet to the query is sometimes subjective, this is understandable and thus we propose methodologies to combine the editorial data with the user preference data obtained from click-through logs. In our training data, we combine two sets $\mathcal{P}_E$ and $\mathcal{P}_C$: $\mathcal{P}_E$, the set of pairwise preferences generated by the above editorial data contains 93000 (entity, facet i, facet j) tuples. $\mathcal{P}_C$, the set of pairwise preferences generated by the pairwise comparison model consists of 189000 (entity, facet i, facet j) tuples.

Our baseline is a simple linear combination of the conditional probabilities, as explained in Section 3.1, across different feature sources such as Web

Search Query Terms, Web Search Query Logs, Flickr and Twitter. We chose this baseline because it is a good representation of a simplistic user preference model that is derived from user behavior data. The conditional probability is also normalized for the number of users to make sure that each user is counted only once.

The evaluation is based on the Discounted Cumulative Gain (DCG) (19) and the pairwise accuracy. The DCG is computed as follows:

$$\text{DCG-K} = \sum_{i=1}^{K} \frac{g(i)}{log(1+i)}$$

where $g(i)$ is the gain associated with the rating of result at rank i and K is the maximum depth result to consider. In this paper, we use gains of $10, 7, 3, 0.5$, and 0, respectively, corresponding to the five ratings or relevance grades.

The pairwise accuracy is the ratio of correct pairs

$$\frac{\{(f_i, f_j) \mid \tau(f_i) < \tau(f_j), h(\mathbf{w}_{ij}) > \lambda\}}{\{(f_i, f_j) \mid \tau(f_i) < \tau(f_j)\}}$$

where $\tau(f_i)$ is the position of f_i in the search results and h is the pairwise comparison model.

A metric is computed for each query and the average values over all the queries in our test data are reported.

5.2. User Data based Evaluation

5.2.1. CTR Analysis of Web Search and Entity Ranking

Before we present the results on our proposed methodologies, we show the differences in presentation and user behavior between traditional Web search results and the entity ranking results to help understand the difference in the presentation bias between the two presentations. As shown in Figure 1, the Web search results page is organized into three columns, where the left column is used to present entity ranking results whenever the query contains entity terms, the middle column contains the traditional web search results along with their snippets that show a summary of the match between the Web page and the query, and the right column is used to display relevant ads if the query contains commercial intention. In addition to the three column layout, rich results are embedded in the middle column above the traditional

Web search results whenever the corresponding structured data is available, depending upon the type of the query. Figure 5 shows the position-wise CTR for both traditional Web search results and the entity ranking results. The CTR information for both the results presentation are normalized so that CTR at position #1 is 1. It can be seen that Web search has a sharp decline at position #2 and slowly decays monotonically for the lower positions. The entity ranking results also experience a similar monotonic decay, but the decline is not that steep. In fact, there is a short plateau region between position #3 and position #7. This indicates the users are more in exploratory mode when interacting with the entity ranking results and browse through various facet entities. The clicks on the entity ranking results also depend on the quality of the thumbnails shown for the entities. Choosing the right thumbnails for the candidate entities is important to mitigate the presentation bias, but solving this problem is beyond the scope of this work.

5.2.2. Category Interaction Analysis of Entities

For some categories of query entities, such as movie actors and sports athletes, the facet entities could be of different categories. For example, movie entities will have movie actors and related movies as results, both of which belong to two different categories. There are several ways of presenting the category information of the facet entities in the entity ranking results section. We chose to group the related entities of the same category and decide the order of the categories depending upon the type of the query entity. However it is not trivial to determine the order of categories given a query entity. To understand how the categories of the query entities and facet entities interact together, we experimented with a small portion of the Web search traffic where we randomize the order of various categories of the facet entities. Table 4 shows the CTR for various categories of query entities on the categories of the facet entities. The CTR values have been normalized to the maximum values in each row. For example the first row shows normalized CTR for entities of the type 'Person' across various categories such as 'Person', 'Movie', 'TV Show', etc, i.e., the normalized CTR on movies for 'Person' queries is 0.43%. Similarly the normalized CTR for various combinations of categories of the entities is shown in the table. From this experiment, it is evident that the person entities typically have higher CTR compared to other entities such as movies and TV shows and sports teams. To understand this interaction of the categories among various facet entities, we conducted other experiments to evaluate the effect of including the category information of the facets into

Table 4: CTR on category of the query entity vs. facet entity. Each row represents the category of the query entity and the column represents the category of the facet entity and each cell represents aggregate CTR at the intersection. The CTR values are normalized for each row such that the category with the highest CTR in each row is given 1.0. The missing entries indicate that the data for the intersection of those particular categories is not available.

Query/Facet	Person	Movie	TV Show	Actor	Music Artist	Sports Team	Athlete
Person	0.98%	0.43%	0.34%	**1.0%**	0.63%	-	0.72%
Movie	**1.0%**	0.92%	-	0.85%	0.37%	-	0.53%
Movie Actor	**1.0%**	0.57%	0.52%	0.96%	0.66%	-	0.61%
Music Artist	0.63%	0.52%	0.59%	0.56%	**1.0%**	-	0.56%
TV Show	**1.0%**	-	-	0.27%	-	-	-
Music Album	0.93%	-	-	-	**1.0%**	-	0.81%
Sports Team	0.66%	-	-	-	0.95%	0.74%	**1.0%**
Sports Athlete	0.91%	-	-	0.73%	0.31%	0.73%	**1.0%**

the loss function which are described in following sections.

5.2.3. Evaluation of Pairwise Comparison Model

The pairwise comparison model is trained on 4.1M preference pairs. We evaluate the model based on the test data consisting of 400K preference pairs. Each click model/feature used in the pairwise comparison model can be used to predict the preference between two facets. Given a click model c, we predict that f_i is preferred to f_j if $c(f_i) - c(f_j) > \tau$. The accuracy of the prediction can be measured by editorial labels given to f_i and f_j. Hence, this is a binary classification problem. With a different τ, we can plot a precision-recall graph. Figure 6 shows the precision-recall for each click model and the pairwise comparison model (PCM). It is clear that a single model is not robust: for small recall, precison is not high. Even a state-of-the-art click model such as the cumulated relevance model (cumrel) shows the same trend. This implies that the user clicks in entity ranking are highly noisy, which we suspect is affected by the presentation bias due to thumbnail images. However, the pairwise comparison model combining these models outperforms all the single models and seem to be much more robust to noisy clicks.

Table 5: Relevance improvements with various inter-category weights over the baseline. The smaller α, the more the intra-category relationships between facets are emphasized. DCG is computed for each group of facets with the same category

Inter-category Weight	DCG-1 Gain	DCG-5 Gain	DCG-10 Gain	Pairwise Accuracy Gain
$\alpha = 0.0$	2.50%	1.97%	0.88%	20.68%
$\alpha = 0.2$	2.58%	1.95%	0.88%	20.87%
$\alpha = 0.4$	2.52%	1.98%	0.86%	20.90%
$\alpha = 0.6$	2.41%	1.97%	0.86%	20.90%
$\alpha = 0.8$	2.45%	1.94%	0.87%	20.82%
$\alpha = 1.0$	2.14%	1.96%	0.84%	20.82%

5.3. Editorial Evaluation

5.3.1. Evaluation of Category Based Loss Function

Table 5 shows the DCG gains with various inter-category weights α over the baseline, which is a linear combination of the conditional probabilities across various feature data sources. If $\alpha = 1$, facet categories are ignored in the training of a ranking function. If $\alpha = 0$, pairs of facets with different categories are not used. The result shows that the relevance is improved by using $\alpha < 1$. Also, α between 0 and 1 provides the best relevance, which implies that the inter-category relationships between facets help the intra-category ranking (ranking within each group of facets with the same category).

5.3.2. Evaluation of Various Types of Features

Table 6 shows the DCG gains with various types of features over the baseline, which is a linear combination of the conditional probabilities across various feature data sources. The table shows that the query log features by themselves are not better than the baseline, but when they are combined with the other feature sources such as Flickr, the overall gain is more than the gain from individual sources. Of all the individual data sources, Flickr seems to be most valuable as it is natural that if two entities, mostly celebrities, appear together in a picture it is likely that these two celebrities are related. While the user data features provide a DCG-10 gain of 1.53%, all features including the graph features such as popularity features and corpus based features provide an overall DCG-10 gain of 2.25%.

Table 6: DCG Gain of various sets of features over the baseline

Feature Sources	DCG-1 Gain	DCG-5 Gain	DCG-10 Gain
Query Terms Only	-1.18%	0.31%	-0.15%
Query Session Only	-3.83%	-1.12%	-0.86%
Flickr Only	2.30%	1.09%	0.43%
All User Data Features	5.38%	3.35%	1.53%
All Features	8.01%	4.98%	2.25%

6. Conclusions

In this paper, we presented a system for ranking related entities in the context of the Web search. We presented an extensive analysis of features for Web-scale entity ranking. We also proposed novel techniques for entity ranking based on machine learned ranking models using an ensemble of pair-wise preference models. We showed how to work with categories in the context of entity ranking by introducing inter-category and intra-category weighting. We showed the results on one of the large knowledge base containing millions of entities and hundreds of millions of relationships. The experiments reveal that our proposed ranking solution clearly improves simple user behavior based ranking and several baselines. The future directions for our work include investigating the effect of time-sensitive recency based features on related entity ranking for the buzzy entities. Another line of future work is to extend this framework to rank and recommend related entities for a given Web page given the content and the context around the page. One of the limitations of our framework is that it is not flexible enough to handle the dynamic nature of entity categories. In some cases, the category of an entity can be only probabilistically described and it can also change over time. We will investigate a way to extend our framework to exploit probabilistic nature of entity categories.

[1] G. Demartini, C. Firan, T. Iofciu, R. Krestel, W. Nejdl, A model for ranking entities and its application to wikipedia, in: Web Conference, 2008. LA-WEB '08., Latin American, 2008, pp. 29–38.

[2] T. Cheng, X. Yan, K. C.-C. Chang, Entityrank: Searching entities directly and holistically, in: Proceedings of the 33rd International Confer-

ence on Very Large Data Bases, VLDB '07, VLDB Endowment, 2007, pp. 387–398.

[3] A.-M. Vercoustre, J. A. Thom, J. Pehcevski, Entity ranking in wikipedia, in: Proceedings of the 2008 ACM Symposium on Applied Computing, SAC '08, ACM, New York, NY, USA, 2008, pp. 1101–1106.

[4] Z. Nie, Y. Zhang, J.-R. Wen, W.-Y. Ma, Object-level ranking: Bringing order to web objects, in: Proceedings of the 14th International Conference on World Wide Web, WWW '05, ACM, New York, NY, USA, 2005, pp. 567–574.

[5] X. He, M. Baker, xhrank: Ranking entities on the semantic web, in: 9th International Semantic Web Conference (ISWC2010), 2010.

[6] Y. Jin, Y. Matsuo, M. Ishizuka, Ranking entities on the web using social network mining and ranking learning, in: World Wide Web Conference (WWW), 2008.

[7] J. Pound, P. Mika, H. Zaragoza, Ad-hoc object retrieval in the web of data, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10, ACM, New York, NY, USA, 2010, pp. 771–780.

[8] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender, Learning to rank using gradient descent, in: Proceedings of the 22Nd International Conference on Machine Learning, ICML '05, ACM, New York, NY, USA, 2005, pp. 89–96.

[9] Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, G. Sun, A general boosting method and its application to learning ranking functions for web search, in: Advances in Neural Information Processing Systems 20, MIT Press, 2008, pp. 1697–1704.

[10] R. van Zwol, L. Garcia Pueyo, M. Muralidharan, B. Sigurbjörnsson, Machine learned ranking of entity facets, in: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, ACM, New York, NY, USA, 2010, pp. 879–880.

[11] R. van Zwol, B. Sigurbjornsson, R. Adapala, L. Garcia Pueyo, A. Katiyar, K. Kurapati, M. Muralidharan, S. Muthu, V. Murdock, P. Ng,

A. Ramani, A. Sahai, S. T. Sathish, H. Vasudev, U. Vuyyuru, Faceted exploration of image search results, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10, ACM, New York, NY, USA, 2010, pp. 961–970.

[12] R. van Zwol, L. G. Pueyo, M. Muralidharan, B. Sigurbjörnsson, Ranking entity facets based on user click feedback, in: Fourth IEEE International Conference on Semantic Computing (IEEE ICSC2010), Pittsburgh, PA, USA, 2010.

[13] M. Skowron, K. Araki, Effectiveness of combined features for machine learning based question classification, Information and Media Technologies 1 (1) (2006) 461–481.

[14] P. Bonacich, Factoring and weighting approaches to clique identification., Journal of Mathematical Sociology 2 (1972) 113–120.

[15] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, in: Proceedings of the Seventh International Conference on World Wide Web 7, WWW7, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1998, pp. 107–117.

[16] O. Chapelle, Y. Zhang, A dynamic bayesian network click model for web search ranking, in: Proceedings of the 18th International Conference on World Wide Web, WWW '09, ACM, New York, NY, USA, 2009, pp. 1–10.

[17] G. Dupret, C. Liao, A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10, ACM, New York, NY, USA, 2010, pp. 181–190.

[18] J. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Statist. 29 (2001) 1189–1232.

[19] K. Järvelin, J. Kekäläinen, Ir evaluation methods for retrieving highly relevant documents, in: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00, ACM, New York, NY, USA, 2000, pp. 41–48.

Figure 1: Screenshot of entity ranking results embedded into the left-hand rail of the search results page.

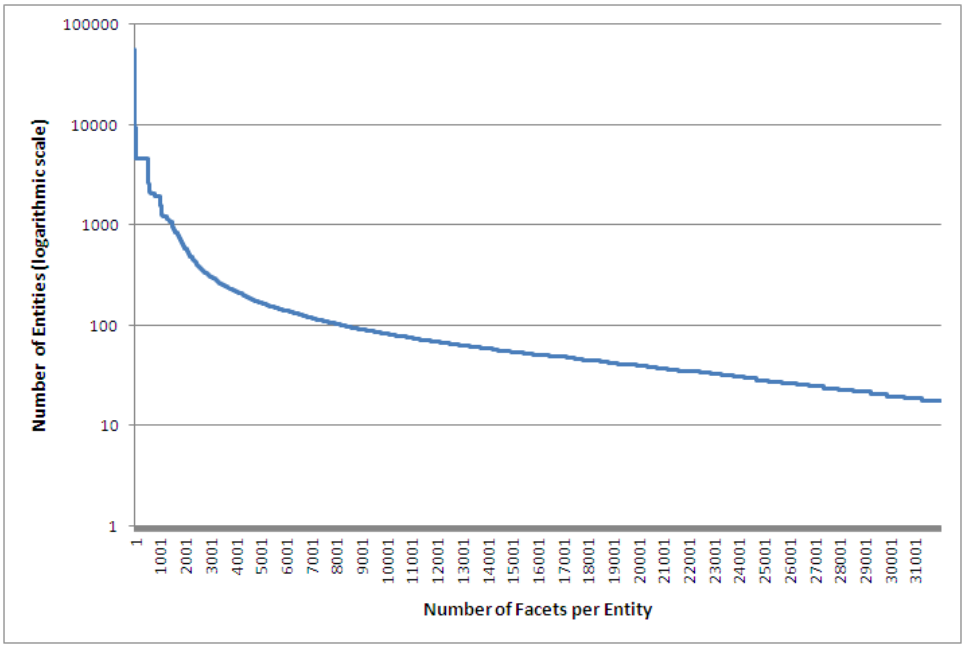

Figure 2: Distribution of the number of facets per entity. The number of entities is in logarithmic scale. While some entities have large number of facets (typically these are location entities which are connected to many other points of interest), some entities have fewer related facets.

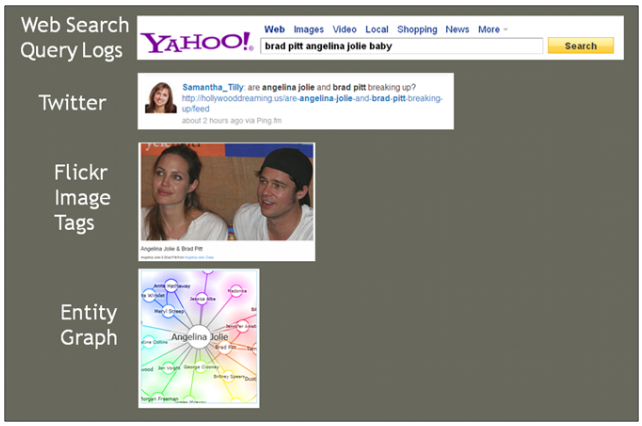

Figure 3: Feature Sources for Entity Ranking.

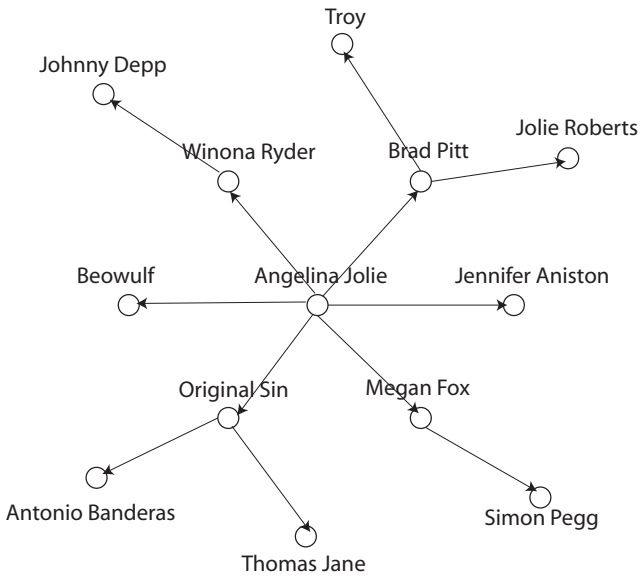

Figure 4: Entity graph: an example

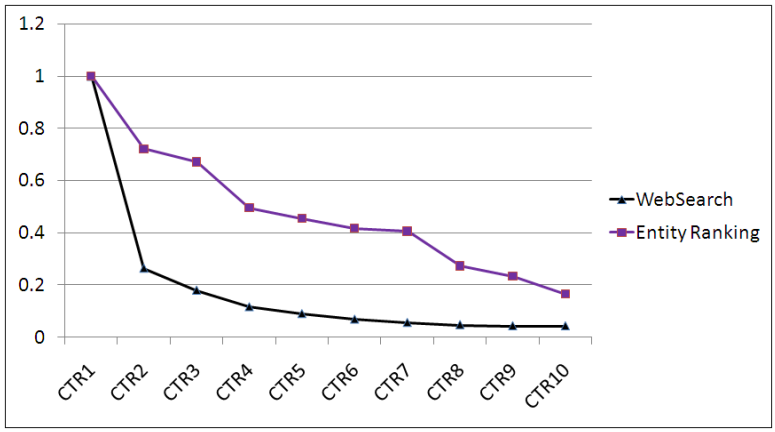

Figure 5: Comparison of position-wise CTR between traditional Web Search results and Entity Ranking. The CTR information for both the results presentation are normalized so that CTR at position #1 is 1.

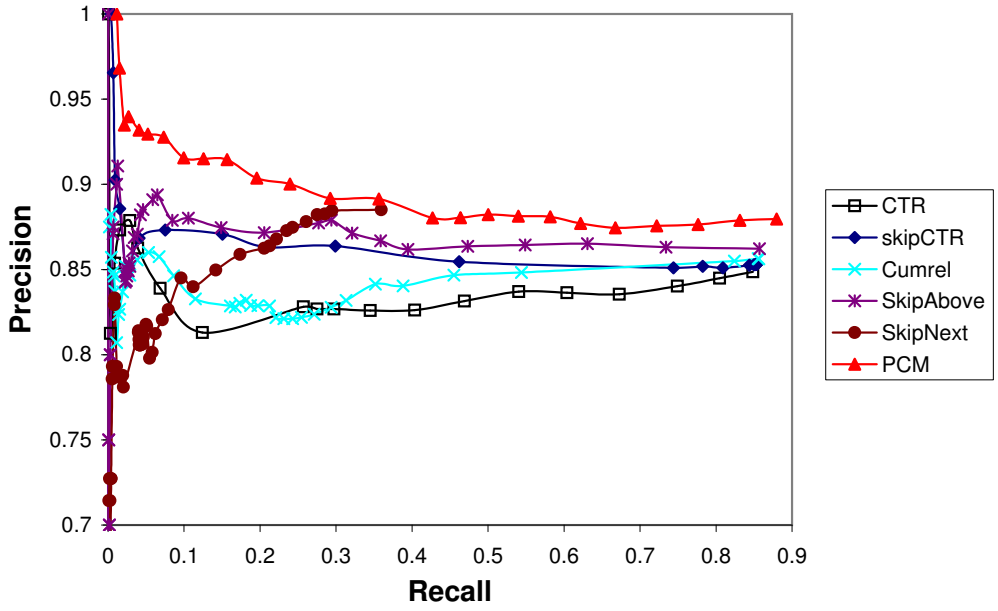

Figure 6: Precison vs. recall of several click models and the pairwise comparison model (PCM). The trade-off between precison and recall is obtained by different thresholds used for a pairwise preference prediction.