Probabilistic Prototype Model for Serendipitous Property Mining

Taesung Lee[*] **Seung-won Hwang** **Zhongyuan Wang**[†]
IBM T. J. Watson Research Center Yonsei University Microsoft Research
taesung.lee@ibm.com seungwonh@yonsei.ac.kr wzhy@outlook.com

Abstract

Besides providing the relevant information, amusing users has been an important role of the web. Many web sites provide serendipitous (unexpected but relevant) information to draw user traffic. In this paper, we study the representative scenario of mining an amusing quiz. An existing approach leverages a knowledge base to mine an unexpected property then find quiz questions on such property, based on prototype theory in cognitive science. However, existing *deterministic model* is vulnerable to noise in the knowledge base. Therefore, we instead propose to leverage probabilistic approach to build a prototype that can overcome noise. Our extensive empirical study shows that our approach not only significantly outperforms baselines by 0.06 in accuracy, and 0.11 in serendipity but also shows higher relevance than the traditional relevance-pursuing baseline using TF-IDF.

1 Introduction

Unlike the traditional purpose of the web providing relevant information or answers to user questions, conversely, recent web services ask users unexpected trivia questions to amuse them. Bing provides a set of interesting quizzes with an image of the day on the front page. Figure 1 describes an amusing quiz question generation on a long 'neck' of `giraffe`, which we use as a motivating scenario throughout this paper.

Inspired by Figure 1, we study the problem of finding a "serendipitous" property a such as 'neck' for any given entity e. Table 1 categorizes existing automatic quiz generation efforts pursuing relevance and unexpectedness, respectively. Inspired by Jeopardy!, Seyler et al. (2015) focus on relevance to the domain and a certain difficulty level. Inspired by Bing questions, Lee et al. (2016) seek unexpected entity-property pair (e, a).

In clear contrast, we complement the solution space by pursuing the intersection of finding unexpected but still relevant properties, which is often named as *serendipitous* (or Case B). As the existing *deterministic* model (DM, Lee et al. (2016)) fails to distinguish Case B and C, we propose a new *probabilistic* model (PM).

- DM: Unexpectedness of 'neck' can be found by building a deterministic prototype using the average of (normalized) property frequencies of all MAMMALS. As the frequency of 'neck' for a `giraffe` is far higher than the average, this can be found.

- PM: DM is effective when the underlying data (*i.e.*, knowledge base) to derive probabilities are not noisy, but an automatically harvested knowledge base inevitably contains noise. For example, Probase, mining textual patterns such as "<property> of <category>," may contain 'some part' as a property of a `giraffe`, which can be recognized as a desirable unexpected property by DM. We propose a probabilistic model that overcomes this problem.

[*] This work was done at Yonsei University and Microsoft Research.

[†] Zhongyuan Wang is now with Facebook Inc.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 663–673, Osaka, Japan, December 11-17 2016.

Our extensive evaluation results using real-life Flickr data and the Probase knowledge base validate that our approach not only significantly outperforms baselines by 0.06 in accuracy and 0.11 in serendipity, but also shows higher relevance than the traditional relevance-pursuing baseline using TF-IDF.

Figure 1: Bing quiz scenario used by DM (Lee et al., 2016).

	Expected	Unexpected (Lee et al., 2016)
Relevant (Seyler et al., 2015)	Case A	Case B
Irrelevant	-	Case C

Table 1: Dimensions of a mining problem.

2 Related work

This paper studies the serendipitous property mining problem of finding relevant yet unexpected properties for a given entity based on a knowledge base. Therefore, our work is closely related to knowledge acquisition. Also, mining unexpected part of knowledge can be considered as serendipitous mining or outlier detection.

2.1 Knowledge acquisition

Our serendipitous property mining system leverages automatically harvested knowledge on the web. Particularly, measuring unexpectedness requires knowing the expectation, which requires worldly knowledge. The basis of such knowledge acquisition works is taxonomies that contain categories and their entities (Carlson et al., 2010; Suchanek et al., 2007; Wu et al., 2012). Among them, Probase (Wu et al., 2012) provides a conditional probability of an entity for a category, and also that of a property given an entity, from which we probabilistically model the expectation (*i.e.*, prototype in our approach).

Besides the automatically harvested knowledge bases, other types of knowledge such as a traditional DB (Merzbacher, 2002) or a linked open data (Marie et al., 2013) are manually generated and often do not cover new entities, like new idols possibly attracting click-through. Therefore, we rely on an automatically harvested knowledge base.

2.2 Serendipitous mining

The primary metric for recommender system is prediction accuracy. However, focusing solely on this metric is reported to limit user satisfaction by always recommending predictable items, such as a new comedy for a comedy fan who can discover it without recommendation. To amuse users, serendipitous mining is studied in recommendation and search. Several approaches (Onuma et al., 2009; Nakatsuji et al., 2010) focus on finding serendipitous items such as funny zombie movie, which is both relevant and unexpected, from user-item matrices. Another direction is pursuing serendipity in search: The existing approaches propose to consider emotional expressions (Hauff and Houben, 2012; Bordino et al., 2013) or presentations such as bold font (O'Brien, 2011), to detect surprises and apply that in search results. These efforts cannot be applied to our problem as a user-item matrix or the text format is unavailable, but our knowledge-based signals are orthogonal and thus can be straightforwardly applied to improve both lines of the work.

2.3 Outlier detection

The unexpected property mining can be considered as an outlier detection problem. Outlier detection has been extensively studied with different approaches. Deviation-based approaches find observations whose removal greatly reduces the sample variance (Arning et al., 1996). Distance-based approaches define a distance measure and consider observations whose distance to others is above a threshold as outliers (Knorr and Ng, 1997; Ramaswamy et al., 2000; Fan et al., 2006). Density-based approaches

consider observations which have little or no neighbors as outliers (Jin et al., 2006). Most of these approaches are designed, trained, or tuned for a target domain using labeled data or using user specified parameters, which is not suitable for our scenario targeting diverse categories. Moreover, these solutions categorize observations into two: normal observations and outliers. Lee et al. (2016), mining quiz questions given images, also leverage the similar notion of classifying the properties into the two, which we use as a baseline and show their ineffectiveness in our experiments, due to their inability to distinguish noise and serendipitous properties.

3 Knowledge base

We leverage an automatically harvested knowledge base for human-like unexpected and relevant property mining. In particular, we leverage Probase *'is-a'* knowledge and its property data to mine probabilities, which is essentially equivalent to deriving probabilities from a huge corpus. Probase is a large knowledge base containing millions of categories and their information including *entities*, *properties*, and their *typicalities* which we describe below.

Is-A knowledge

The backbone of Probase is probabilistic knowledge of huge amount of *'is-a'* relations between entities and categories (Wu et al., 2012)[1]. For example, Probase contains a relation: "`giraffe` *is-a* MAMMAL" where `giraffe` is an entity and MAMMAL is a category. A category may contain many entities (*e.g.*, MAMMAL contains `giraffe` and `platypus`), and an entity may also belong to several categories (*e.g.*, `giraffe` belongs to both MAMMAL and ANIMAL). Such relationships are mined from a huge amount of web documents using patterns (Wu et al., 2012).

Property knowledge

Properties are words representing a certain aspect of an entity. For example, 'neck' is a property of the entities in category MAMMAL. Properties of an entity are obtained by several approaches including pattern-based extraction methods (Lee et al., 2013).

Typicality

Based on the number of pattern occurrences, we can compute the conditional probability of a certain element given a condition, which we also call *typicality*. For example, given category MAMMAL, people would usually think of typical mammals such as `dog`. In particular, Probase has diverse types of typicalities including the *entity typicality for a category*, and *property typicality for an entity*.

$P(E|C)$ is a conditional probability of entity E given category C. For example, we can obtain the probability $P(E = \text{giraffe}|C = \text{mammal})$, representing how typical entity `giraffe` is for category MAMMAL. Such probability can be obtained using all occurrences of MAMMAL, and the co-occurrences of MAMMAL and `giraffe`:

$$P(E = \text{giraffe}|C = \text{mammal}) = \frac{\text{Freq}(\text{mammal}, \text{giraffe})}{\text{Freq}(\text{mammal})} \quad (1)$$

where $\text{Freq}(x, y)$ represents the co-occurrence of x and y, and $\text{Freq}(x)$ indicates the occurrence of x in Probase. We can similarly compute other probabilities including $P(C)$, $P(E)$, $P(C|E)$. Also, we can obtain $P(A|E)$ where A is a property and E is an entity.

We can consider a typicality as the amount of people's interest in the topic since it is derived from how frequently we discuss the topic. We usually have general topics of interest for entities in a certain category: 'lifespan', 'diet', 'size' and so on for mammals. But, if the heart of a giraffe is often discussed in comparison to other mammals, it can be an unexpected topic about a giraffe.

[1]Probase knowledge base is publicly available online at https://concept.research.microsoft.com/.

4 Methods

In this section, we describe our approach to model prototype and mine serendipitous properties. We follow the framework presented in (Lee et al., 2016) that leverages the category of a given entity, and propose a method for unexpected property mining. Therefore, we assume that we have the given topic entity e, and its category c.

4.1 Modeling a prototype

Identifying an unexpected property of an entity requires comparison to what we consider expected. If we expect most mammals have a heart with the size in a specific range, the extra large heart of a `giraffe` can be unexpected. Therefore, defining the prototype of a category—which represents the human expectation for entities in the category— is the key step to find serendipitous properties. We may consider selecting a typical entity among the existing entities in the category (such as `dog` for MAMMAL). However, typical properties such as 'lifespan' can be missing with `dog` due to data sparsity. In this case, 'lifespan' of any entity can be rather considered unexpected.

DM (Lee et al., 2016) models a deterministic prototype model $\mathcal{R}_c^{\text{DM}}$ as a *hypothetical* entity using the average typicalities in the category. Note that, instead of the values of properties, their typicalities are leveraged to directly capture the human interest on the properties for entities and the category. That is, high $P(\text{height}|\text{giraffe})$ means that 'height' draws human interest so that it is discussed frequently with `giraffe` on the web. If some property is frequently mentioned with most entities in a category, it can be considered the representative property of the category. Then, we can consider some property of an entity is unexpected if the property is particularly more frequently mentioned with the entity than with other entities in the category.

Thus, using the average of typicalities allows us to compute the representativeness of the property for the category (*e.g.*, MAMMAL). Formally, $\mathcal{R}_c^{\text{DM}}$ is defined as an ordered set of the average of property typicalities in the category c as follows:

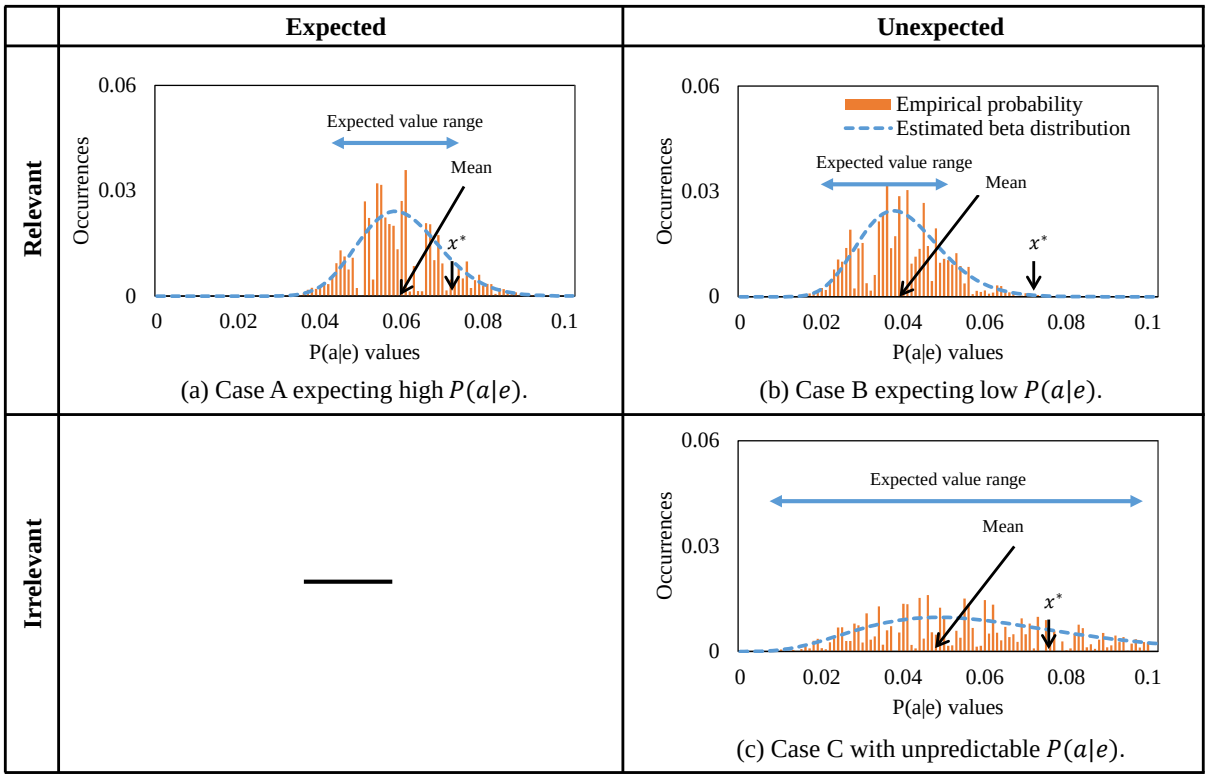
$$\mathcal{R}_c^{\text{DM}} = \{\text{avg}_{e \in c}\, P(a|e)P(e|c) \mid a \text{ is a property}\} \tag{2}$$

where $P(a|e)$ is the property typicality given entity, and $P(e|c)$ is the entity typicality given category (Section 3). This hypothetical prototype does not suffer from missing properties caused by data sparsity. However, this approach is vulnerable to noise of another cause: 'some part', wrongfully identified as a property, would be considered unexpected due to its infrequency.

Instead of this deterministic approach, we leverage a probabilistic method similar to (Eskin, 2000), which is originally designed for intrusion detection, to model the prototype. Unlike its category-agnostic approach using a Markov chain, we leverage the category information to model the prototype with beta distributions. In particular, we define a prototype of category c hypothetically as an ordered set of random variables $\mathcal{R}_c = \{X_{a,c}|a \text{ is a property}\}$ (an example distribution of $X_{a,c}$ is shown as the blue dashed line in Figure 2(b)). That is, unlike DM using averages to produce an ordered set of expected typicalities, we build probability distributions.

Given this model, we can consider that the properties of an entity in the category are realizations of the random variables. To illustrate, suppose we have $\mathcal{R}_c = \{X_{lifespan,mammal}, X_{tail,mammal}, X_{heart,mammal}\}$, and typicalities of `dog` $P(\text{lifespan}|\text{dog})$, $P(\text{tail}|\text{dog})$, and $P(\text{heart}|\text{dog})$ are 0.3, 0.6, and 0.1 respectively. Then, we consider 0.3, 0.6 and 0.1 are realizations of $X_{\text{lifespan,mammal}}$, $X_{\text{tail,mammal}}$ and $X_{\text{heart,mammal}}$ with the corresponding probabilities $P(X_{\text{lifespan,mammal}} = 0.3)$, $P(X_{\text{lifespan,mammal}} = 0.6)$, and $P(X_{\text{lifespan,mammal}} = 0.1)$. By measuring the likelihood of this event (having 0.3, 0.6, and 0.1) using these probabilities, we can measure how unexpected this entity or its properties are. Later in Section 4.2, we will argue how this model distinguishes Case B and C in Figure 2 and explain how we measure the unexpectedness.

Formally, we consider an entity of the category as an ordered set of properties represented as $\{P(a|e)\}$ (*e.g.*, $\{0.3, 0.6, 0.1\}$) each of which is drawn from the corresponding random variable in $\mathcal{R}_c$. The co-occurrences of properties and entities can be modeled to be drawn from a multinomial distribution. Then,



(a) Case A expecting high $P(a|e)$.

(b) Case B expecting low $P(a|e)$.

(c) Case C with unpredictable $P(a|e)$.

Figure 2: Distribution of $X_{a,c}$.

the typicality $P(a|e)$ for each a constitutes parameter probabilities $p_1, \ldots, p_k$ of the multinomial distribution $Mult(n|p_1, \ldots, p_k)$. We exploit a Dirichlet distribution of a single dimension, which is a beta distribution, to model $P(a|e)$ since a Dirichlet distribution is a conjugate prior of a multinomial distribution. That is, each random variable $X_{a,c}$ in prototype $\mathcal{R}_c$ is modeled as a beta distribution corresponding to a property of entities in category c: $X_{a,c} \sim Beta(x; \alpha_{a,c}, \beta_{a,c})$.

Now we have to learn the parameters specifying the beta distributions $Beta(x; \alpha_{a,c}, \beta_{a,c})$ of the prototype $\mathcal{R}_c$. In particular, we use a set $O_{a,c}$ of property typicality $P(a|e)$ for each entity e in the identified category c as samples together with its occurrence probability $P(e|c)$, that we obtain from Probase. That is, we give a larger weight to a more typical entity in the model. Let $X_{a,c}$ be a random variable modeled by a beta distribution $Beta(x; \alpha_{a,c}, \beta_{a,c})$. We find parameters $\alpha_{a,c}$ and $\beta_{a,c}$ so that $X_{a,c}$ generates the observation $O_{a,c} = \{x^e_{a,c} = P(a|e) | e \in c\}$ with their associated occurrences $P(e|c)$. Then, we have $P(x_1 \leq X_{a,c} < x_2) = \sum_{e \text{ s.t. } x_1 \leq P(a|e) < x_2} P(e|c)$. Thus, the mean μ of the beta distribution $Beta(x; \alpha_{a,c}, \beta_{a,c})$ is $\sum_{e \in c} P(a|e)P(e|c)$, and the variance σ^2 is $\sum_{e \in c} (P(a|e) - \mu)^2 P(e|c)$. The parameters $\alpha_{a,c}$ and $\beta_{a,c}$ can be fitted by the widely adopted method of moments using the mean and the variance, of exploiting the first and second moments of $Beta(x; \alpha_{a,c}, \beta_{a,c})$.

4.2 Mining unexpected properties

With the obtained probabilistic prototype $\mathcal{R}_c = \{X_{a,c} | a \text{ is a property}\}$ for category c, we discover serendipitous properties of the given entity, and show why noisy properties are not mined. We can consider that entities in the category are created by drawing each property typicality $P(a|e)$ from the beta distribution of the prototype. If a property typicality of the given entity is unlikely based on the prototype, we can consider the property is serendipitous.

We show how a property of the topic entity can be compared with that of the prototype. We divide the comparison into three cases that we have described in the introduction (Table 1): relevant and expected, unexpected but relevant (serendipitous), and noisy. Figure 2 shows examples of the component random variables $X_{a,c}$ of the model for Case A, B, and C in Table 1. Then, we consider how likely the given entity to be generated.

Case A: Property is relevant and expected

First, Figure 2(a) depicts a distribution skewed to high values. For example, 'lifespan' is a common property for the most of entities in MAMMAL. Therefore, high $P(\text{lifespan}|\text{giraffe})$ does not mean `giraffe` has an unexpected property.

Case B: Property is unexpected, but relevant (serendipitous)

In contrast, Figure 2(b) shows a distribution skewed to low $P(a|e)$. In this case, we expect a relatively low value, and thus high $P(a|e)$ of the topic entity implies a serendipitous event. For example, although the most entities in MAMMAL have a property 'neck,' it is not frequently mentioned, and hence we expect to have low $P(\text{neck}|e)$ value. Therefore, the distribution of $X_{\text{neck,mammal}}$ is skewed to low values. Then, the high value of $P(\text{neck}|\text{giraffe})$ indicates that `giraffe` has an serendipitous property 'neck.'

Case C: Property is noisy

Unlike Case A and B modeling relevant properties, the prototype may include a random variable representing a noisy property. Specifically, when a noisy property is modeled, it looks like Figure 2(c). A noisy property does not show a general tendency, and it is rather randomly distributed as often modeled by a uniform distribution. That is, since a noisy property is modeled like a uniform-esque distribution as shown in Figure 2(c), the most value of $P(a|e)$ is expected to happen, and would not be considered serendipitous.

Computing unexpected relevance

Based on these observations, we compute our measure of serendipitous properties. Formally, as exploited in (Eskin, 2000), we compute the log-likelihood of an entity-property probability $P(X_{a,c} = x^e_{a,c})$ of the given topic entity using the obtained distribution for $X_{a,c}$ of $\mathcal{R}_c$. In addition, we use a minus sign to obtain a measure indicating a more serendipitous property with a greater value.

$$\begin{aligned} I(x^e_{a,c}) &= -\log(P(X_{a,c} = x^e_{a,c})) \\ &= -\log Beta(x; \alpha_{a,c}, \beta_{a,c}) \end{aligned} \tag{3}$$

Upon this value, we also consider that a serendipitous property of a well-known or more typical entity can more easily draw user attention. Thus, we quantify the degree of being serendipitous for property a of entity e that belongs to category c as follows.

$$H(c, e, a) = I(x^e_{a,c})P(e|c) \tag{4}$$

We measure this value for each property of the given topic entity, and consider one with the highest score the most interesting, so that we present it to users.

5 Evaluation

In this section, we evaluate our systems using various measures evaluating several facets of the proposed work. Throughout the evaluation, we compare our method PM with DM (Lee et al., 2016) in the same setting of using Flickr and Yahoo! Answers.

5.1 Serendipitous property discovery

In this section, we analyze and evaluate serendipitous property mining. Table 2 first shows examples of the top serendipitous entity-property pairs for diverse categories returned by PM and DM. We can observe that DM is prone to pick long phrasal nouns or noisy properties, which are less commonly mentioned. DM considers (possum, plural) and (intel, fourth piece) unexpected because the conditional probabilities $P(\text{plural}|\text{mammal})$ and $P(\text{fourth piece}|\text{company})$ are very low while $P(\text{plural}|\text{possum})$ and $P(\text{fourth piece}|\text{intel})$ are not. On the other hand, our approach does not propose noisy properties, by modeling such noises together. Instead, our approach suggests entity-property pairs such as (marsupial, embryo), (microsoft, founder), or (china, population) that would lead to interesting information. For

Table 2: Mined unexpected entity-property pairs for diverse categories.

Category	PM	DM
mammal	(marsupial, embryo), (giraffe, heart), (chimpanzee, brain), ...	(wolf, strength), (muskrat, best part), (otter, presence), ...
company	(microsoft, founder), (facebook, founder), (amazon.com, success), ...	(intel, fourth piece), (dell, model number), (coca-cola, original color), ...
country	(china, great wall), (china, population), (india, population), ...	(china, great wall), (china, choices), (india, reserve bank), ...
metal	(copper, resistivity), (gold, price), (gold, purity), (lead, density), ...	(copper, discovery), (lead, presence), (iron, presence), (aluminum, presence), ...
drug	(cocaine, price), (marijuana, legalization), (marijuana, odor), ...	(marijuana, 80 pounds), (cocaine, freebase form), (cocaine, last use), ...

Table 3: Unexpectedness of discovered properties

Method	SS@5
PM	0.31
DM	0.25

example, unlike other mammals, a marsupial has a pouch on her stomach to carry her babies. China is known to have the largest population in the world.

We also quantitatively evaluate the accuracy, or how well the discovered serendipitous properties are aligned with human judges. The serendipitous property (Case B in Table 1) should be relevant to the given entity and category pair, but peculiar in the category. In addition, the property is the most useful in our scenario drawing users if the property leads to a novel information (less known). We build a labeled dataset by annotating the mined property with the serendipitous score on a scale of 0, 1/3, 2/3 and 3/3. In particular, 3/3 point is given if the property is relevant, peculiar, and novel; 2/3 point is given if the property is relevant, and peculiar; 1/3 point is given if the property is relevant; and 0 point is given if the property is irrelevant. For example, the population of China is extraordinarily large which is serendipitous, but this property is well known so that we give 2/3. We have built a gold standard of size 386 independently labeled by three assessors from which we observe sufficient agreement (0.64 pairwise cosine similarity), and thus use the average of the scores assigned by the assessors. We leverage *SS@5*, the average serendipitous scores of the top-5 properties for each entity.

Table 3 shows the evaluation result of serendipitous property discovery methods. We see that ours show the highest score. While our approach might find a typical property for a typical entity (*e.g.*, 'size' for `dog`), the discovered properties are mostly relevant, and often peculiar. On the other hand, DM frequently present irrelevant ones such as noisy properties (*e.g.*, 2 cups, oldddddddd guns), and thus show the lower scores.

5.2 Trivia question mining evaluation

We plug in our method into the framework of (Lee et al., 2016) that seeks trivia quizzes from image tags. We compare the trivia quiz mining results using our serendipitous property mining module and that of (Lee et al., 2016). Our evaluation procedure is based on an serendipitous document mining work (Bordino et al., 2013). Bordino et al. (2013) use both rank-agnostic accuracy of top-N results (Section 5.2.1) and rank correlation of treating its rank differently (Section 5.2.2).

5.2.1 Relevance and serendipity

We measure both relevance, and *serendipity* (*i.e.*, pleasant surprise) using the evaluation procedure introduced by (Ge et al., 2010). to evaluate a recommender system or a serendipitous web search method. It shares the same spirit as our goal that we amuse users with serendipitous properties.

Both for relevance and serendipity, we label question relevance into *relevant* (1) and *irrelevant* (0). We randomly show a query image and one of the top N result questions from any methods so that an assessor is not biased to a certain method. Note that no other information is shown to the assessor.

Then, we first measure the average relevance of the results for all images in J for each method. Formally, we compute the relevance as follows.

$$Average\ relevance@N = \frac{\sum_j \sum_k^N rel_{j,k}}{N|J|} \tag{5}$$

where $rel_{j,k}$ is the relevance (1 or 0) of the rank k question for $j \in J$.

Measurements of serendipity have been established in the context of evaluating recommendation systems (Ge et al., 2010; Shani and Gunawardana, 2011). Specifically, (Ge et al., 2010) evaluates serendipity as the ratio of unexpected but relevant results based on a *benchmark model* that generates expected recommendations, which are relevance-pursuing results. That is, from the result questions of each method, we remove the expected ones that are retrieved by the benchmark model. Then the remainder is considered unexpected. By checking the relevance of the remainder, we measure the relevance of the unexpected results.

As a benchmark model, we might consider Yahoo! Answers search results or TF-IDF-based results using the host page keywords. Unfortunately, Yahoo! Answers returns very few or no results when there are more than two keywords, which is often much less than the number of keywords an image has (*e.g.*, "giraffe zoo savanna" gives no result on Yahoo! Answers). To make the matters worse, obtaining top 100 results for subsets of keywords on Yahoo! Answers and joining the results also gives few or no intersection. Therefore, we use TF-IDF to obtain the benchmark results.

Formally, suppose that $BM@N$ is the top N questions retrieved by the benchmark model, and $RS@N$ is the top N questions retrieved by a method we want to test. Then, we calculate the unexpected recommendation set as follows.

$$UNEXP@N = RS@N - BM@N \tag{6}$$

These unexpected recommendations may or may not be relevant to the query, but we want unexpected but still relevant ones. Therefore, we measure the serendipity based on the relevance of each item in $UNEXP$. Based on relevance labels $rel_{j,k}$ of k-th result for image j, serendipity is defined as follows.

$$SRDP(RS)@N = \frac{\sum_{(j,k)\in UNEXP@N} rel_{j,k}}{N} \tag{7}$$

Table 4: Average question relevance and serendipity at N.

Method	*Rel.*@5	*Rel.*@10	*SRDP@5*	*SRDP@10*
PM	**0.6689**	**0.6607**	**0.6662**	**0.6275**
DM	0.5672	0.5820	0.5562	0.5819
TF-IDF	0.6190	0.6048	-	-

Table 4 shows the average relevance and the serendipity of each method. As a reference, we also include TF-IDF, which does not consider unexpectedness, to show the results of a typical relevance-pursuing model. Note that TF-IDF is the benchmark model and hence it does not provide unexpected questions for serendipity.

We may anticipate that the methods pursuing serendipity may have lower relevance, since such methods would avoid highly relevant results, which are often expected. Thus, DM shows lower relevance

Table 5: Average Kendall coefficient for the questions.

Methods	Q1.	Q2.	Q3.	Q4.
PM	**0.4960**	**0.3980**	**0.2486**	**0.3538**
DM	-0.0558	-0.0764	-0.2567	-0.2026

than PM because it gives lower score to those close to the average, but it may instead acquire and leverage noisy properties as unexpected. For example, DM may consider 'fourth piece' of `Intel`, which is extracted due to ambiguity and hence rare in the category, as unexpected. Therefore, it leverages such a property and in turn mines irrelevant trivia quizzes. We can also see that our approach shows higher serendipity than the baseline regardless of N. As in relevance, the baseline shows low serendipity because they highlight noisy properties as we have already seen in Section 5.1. Note that we can see PM show comparable or better relevances with TF-IDF as our approach pursuing serendipitous properties distinguishes noises.

5.2.2 Kendall's tau-b rank correlation coefficient

We evaluate the results considering their ranks using *Kendall's tau-b rank correlation coefficient* (Kendall, 1938). Kendall's coefficient ranges from -1 (perfect ranking disagreement) to +1 (perfect ranking agreement).

We build a reference ranking according to the several evaluation dimensions since "pleasant" in serendipity (pleasant surprise) can be interpreted in different ways. Therefore, we use several criteria such as 'more relevant questions rank higher' based on (Bordino et al., 2013) and evaluate individual dimensions. Specifically, we generate a task with a photograph and two randomly chosen result trivia quizzes out of those returned by all tested methods. A result trivia quiz that is more suitable for each of the following criteria is labeled as 'better.'

- Q1. The trivia quiz is relevant to the image.
  (*i.e.*, 'pleasant' means 'relevant')
- Q2. If someone is interested in the image, they would also be interested in the trivia quiz.
  (*i.e.*, 'pleasant' means 'interesting given interest to the image')
- Q3. Even if you are not interested in the image, the trivia quiz is interesting to you personally.
  (*i.e.*, 'pleasant' means 'interesting regardlessly to the image')
- Q4. You learn something new about the image from the trivia quiz.
  (*i.e.*, 'pleasant' means 'novel')

Note that Q2 and Q3 try to evaluate both the image or topic-centric view of interestingness (Q2), and the intrinsic interestingness of the trivia quiz (Q3) as in (Bordino et al., 2013). For example, if a method shows high performance on Q3 but not on Q2, its trivia quiz tends to deviate much from the image/topic. However, the method is anyway presenting interesting trivia quiz.

A human assessor has to label each of criteria for a given task. As used in (Bordino et al., 2013), the reference ranking for each criterion is built by a simple voting-based approach, by ranking items with the greater number of 'better' votes higher. Since this evaluation may heavily depend on human assessors, we validate the gold standard by measuring the agreement. Thirty tasks are randomly sampled and evaluated by four human assessors. We measure Fleiss' kappa (Fleiss, 1971) to obtain $\kappa = 0.57$, which shows moderate agreement (Landis and Koch, 1977). In addition, considering only confident answers by removing those tasks with any 'not sure' vote, we obtain higher agreement $\kappa = 0.71$, which shows substantial agreement.

The evaluation result using Kendall coefficient is shown in Table 5. We can see that our method outperforms the baseline. In particular, our approach retains a high positive correlation with the reference rankings based on user perception. On the other hand, the baseline method shows low or negative correlation with all reference rankings.

6 Conclusions and future work

This paper studies the serendipitous property mining problem of finding relevant yet unexpected properties for a given entity. Although the serendipitous information mining is important for industry to increase website traffic, it has not been studied actively on general knowledge due to lack of knowledge or having noisy knowledge. Such noisy knowledge is challenging as noisy properties can be evaluated as unexpected. We empirically show that probabilistic modeling of a prototype for each category alleviates the noise problem, while the existing approach is prone to pick up the noisy properties that may significantly detract the user experience of the applications like trivia questions. Our evaluation results suggest that our approach shows not only higher serendipity than the baselines, but also higher relevance than a traditional baseline using TF-IDF optimized for relevance.

We expect many research topics can be stemmed from our work. One possibility of modeling a prototype would be neural embedding. Comparing its performance with our probabilistic model will be a good research direction. Meanwhile, our approach is limited to properties, and the proposed framework evaluates one property of an entity at each time. But taking relations into account, or considering more than one properties/relations may give even more interesting questions and facts. Also, we expect a similar approach can be exploited to mine outstanding issues from social network data which have a considerable amount of noise.

7 Acknowledgments

This work was supported by the Yonsei University New Faculty Research Seed Funding Grant of 2015, the Yonsei University Research Fund (Post Doc. Researcher Supporting Program) of 2015 (project no.: 2015-12-0211), and Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.B0101-16-0307, Basic Software Research in Human-level Lifelong Machine Learning (Machine Learning Center)).

References

Andreas Arning, Rakesh Agrawal, and Prabhakar Raghavan. 1996. A linear method for deviation detection in large databases. In *Proc. 2nd International Conference on Computational Linguistics (KDD 1996)*, pages 164–169. ACM.

Ilaria Bordino, Yelena Mejova, and Mounia Lalmas. 2013. Penguins in sweaters, or serendipitous entity search on user-generated content. In *Proc. 20th ACM International Conference on Information and Knowledge Management (CIKM 2013)*. ACM.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proc. 24th Conference on Artificial Intelligence (AAAI 2010)*. AAAI Press.

Eleazar Eskin. 2000. Anomaly detection over noisy data using learned probability distributions. In *Proc. 17th International Conference on Machine Learning (ICML 2000)*, pages 255–262. Morgan Kaufmann.

Hongqin Fan, Osmar R Zaïane, Andrew Foss, and Junfeng Wu. 2006. A nonparametric outlier detection for effectively discovering top-n outliers from engineering data. In *Proc. Pacific-Asia conference on knowledge discovery and data mining (PAKDD 2006)*, pages 557–566. Springer.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proc. 4th Conference on Recommender Systems (RECSYS 2010)*, pages 257–260. ACM.

Claudia Hauff and Geert-Jan Houben. 2012. Serendipitous browsing: Stumbling through wikipedia. In *Proc. 34th European Conference on IR Research (ECIR 2012)*.

Wen Jin, Anthony KH Tung, Jiawei Han, and Wei Wang. 2006. Ranking outliers using symmetric neighborhood relationship. In *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2006)*, pages 577–593. Springer.

M. G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93.

Edwin M. Knorr and Raymond T. Ng. 1997. A unified approach for mining outliers. In *Proc. 2rd International Conference on Computational Linguistics (KDD 1997)*, pages 219–222. ACM.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

Taesung Lee, Zhongyuan Wang, Haixun Wang, and Seung-won Hwang. 2013. Attribute extraction and scoring: A probabilistic approach. In *Proc. 29th International Conference on Data Engineering (ICDE 2013)*, pages 194–205. IEEE.

Taesung Lee, Seung-won Hwang, and Zhongyuan Wang. 2016. Trivia quiz mining using probabilistic knowledge. In *Proc. 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2016)*.

Nicolas Marie, Fabien Gandon, Myriam Ribière, and Florentin Rodio. 2013. Discovery hub: On-the-fly linked data exploratory search. In *Proc. 9th International Conference on Semantic Systems (I-SEMANTICS 2013)*, pages 17–24.

Matthew Merzbacher. 2002. Automatic generation of trivia questions. In *Proc. 13th International Symposium on Foundations of Intelligent Systems (ISMIS 2002)*, pages 123–130.

Makoto Nakatsuji, Yasuhiro Fujiwara, Akimichi Tanaka, Toshio Uchiyama, Ko Fujimura, and Toru Ishida. 2010. Classical music for rock fans?: Novel recommendations for expanding user interests. In *Proc. 19th ACM International Conference on Information and Knowledge Management (CIKM 2010)*, pages 949–958. ACM.

Heather L. O'Brien. 2011. Exploring user engagement in online news interactions. In *Proc. Annual Meeting on Association for Information Science and Technology (ASIST 2011)*, pages 1–10. Wiley Subscription Services, Inc., A Wiley Company.

Kensuke Onuma, Hanghang Tong, and Christos Faloutsos. 2009. Tangent: A novel, 'surprise me', recommendation algorithm. In *Proc. 15th International Conference on Computational Linguistics (KDD 2009)*, pages 657–666. ACM.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. In *Proc. SIGMOD International Conference on Management of Data (SIGMOD 2000)*, pages 427–438. ACM.

D Seyler, M Yahya, and K Berberich. 2015. Generating quiz questions from knowledge base. In *Proc. 24st International World Wide Web Conference (WWW 2015)*. ACM.

Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297. Springer US.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proc. 16th International World Wide Web Conference (WWW 2007)*. ACM.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: a probabilistic taxonomy for text understanding. In *Proc. SIGMOD International Conference on Management of Data (SIGMOD 2012)*. ACM.