TextTruth: An Unsupervised Approach to Discover Trustworthy Information from Multi-Sourced Text Data

Hengtong Zhang¹, Yaliang Li², Fenglong Ma¹, Jing Gao¹, Lu Su¹
¹SUNY Buffalo, Buffalo, NY USA
²Tencent Medical AI Lab, Palo Alto, CA USA
{hengtong, fenglong, jing, lusu}@buffalo.edu, yaliangli@tencent.com

ABSTRACT

Truth discovery has attracted increasingly more attention due to its ability to distill trustworthy information from noisy multi-sourced data without any supervision. However, most existing truth discovery methods are designed for structured data, and cannot meet the strong need to extract trustworthy information from raw text data as text data has its unique characteristics. The major challenges of inferring true information on text data stem from the multifactorial property of text answers (i.e., an answer may contain multiple key factors) and the diversity of word usages (i.e., different words may have the same semantic meaning). To tackle these challenges, in this paper, we propose a novel truth discovery method, named “TextTruth”, which jointly groups the keywords extracted from the answers of a specific question into multiple interpretable factors, and infers the trustworthiness of both answer factors and answer providers. After that, the answers to each question can be ranked based on the estimated trustworthiness of factors. The proposed method works in an unsupervised manner, and thus can be applied to various application scenarios that involve text data. Experiments on three real-world datasets show that the proposed TextTruth model can accurately select trustworthy answers, even when these answers are formed by multiple factors.

CCS CONCEPTS
• Information systems → Data mining.

KEYWORDS
Truth discovery; unsupervised learning; text mining

ACM Reference Format:

1 INTRODUCTION

In the big data era, tremendous data can be accessed on various online platforms, such as Amazon Mechanical Turk, Stack Exchange and Yahoo Answers. However, such multi-sourced data are usually contributed by non-expert online users, thus there may exist errors or even conflicts in the data. Therefore, how to automatically infer trustworthy information (i.e., the truths) from such noisy and conflicting data is a challenging problem.

To address this challenge, truth discovery methods have been proposed [4, 5, 8, 12–15, 19–21, 26, 27, 29, 36, 38, 43], which aim to estimate trustworthy information from conflicting data by considering user reliability degrees. Truth discovery approaches follow two fundamental principles: (1) If a user provides much trustworthy information or true answers, his/her reliability is high; (2) If an answer is supported by many reliable users, this answer is more likely to be true. Though yielding reasonably good performance, most existing truth discovery methods are designed for structured data, and are difficult to be directly applied to text data, which are unstructured and noisy. This significantly narrows the application domain of these truth discovery methods, as a large ratio of the multi-sourced data are text. Actually, there are several unique characteristics of natural language that hinder the existing truth discovery methods from being successfully applied to text data.

Figure 1 gives an illustration of these two characteristics of text data. First, the answer to a factoid question ¹ may be multifactorial, and it is usually hard for a given text answer to cover all the factors. For the question ‘What are the symptoms of flu?’, the correct answer should contain the following factors: fever, chills, cough, nasal symptom, ache, and fatigue. Even if the answer provided by a user covers two factors, such as cough and chills, the existing truth discovery methods may determine this answer to be totally wrong and assign a low reliability degree to this user. This is because these methods treat the whole answer as an integrated unit. However, if we take the fine-grained answer factors into consideration, the answer provided by this user is partially correct, which implies that we should give some credits to the user by increasing his/her reliability degree. Thus, how to identify partially correct answers and model factors of text answers is critical for the task of truth discovery on text data.

The second characteristic of text data is the diversity of word usages. Answers provided by online users may convey a very similar meaning with different keywords. For example, users may use words such as tired or exhausted to describe the symptom of fatigue. However, existing truth discovery approaches may treat them as

¹Note: This paper merely focuses on finding trustworthy answers for factoid questions: Factoid questions are defined as questions that can be answered with simple facts expressed in short text answers [11].
totally different answers. Thus, it is of great importance to model the diversity among answers in the text data when inferring trustworthy information.

In order to tackle the aforementioned challenges for inferring trustworthy information from text data, in this paper, we propose a model named “TextTruth”, which takes the keywords in each answer as inputs and outputs a ranking for the answer candidates based on their trustworthiness. Specifically, we first transform the keywords in text answers into pre-trained computable vector representations. Due to the fact that an answer may contain multiple factors, the “answer-level” or coarse-grained representations may not be able to capture the partially correct answers. Thus, we need to convert the whole answer into fine-grained factors. Then, we model the diversity of answers by clustering the keywords with similar semantic meanings. By doing so, we can estimate the trustworthiness of each answer factor instead of the whole answer and infer the correctness of each factor in the answer.

Compared with existing truth discovery methods, the advantages of the proposed TextTruth are two-fold: First, by evaluating the trustworthiness of each answer factor, the proposed model can naturally handle the partial correctness phenomenon of text answers. Second, by modeling answer keywords in the form of vector representations, we can make the factors within the answers computable such that the ubiquitous usage diversity issue on text data is addressed.

Experiments on three real-world datasets demonstrate that the proposed TextTruth model can improve the performance of finding trustworthy answers in text data compared with the state-of-the-art truth discovery approaches. The answers can thus be ranked based on the trustworthiness of their factors. The proposed TextTruth model can jointly learn semantic clusters (i.e., factors) for answer keywords and infer the reliability of each user as well as the trustworthiness of each answer factor. The answers can thus be ranked based on the trustworthiness of their factors.

We empirically show that the proposed model outperforms the state-of-the-art truth discovery methods for the task of answer ranking on three real-world datasets.

The rest of the paper is organized as follows: Section 2 is a survey of related work. In Section 3, we formally define the problem discussed in this paper. Then we describe the proposed TextTruth model, and provide a method for parameter estimation in Section 4. In Section 5, we conduct a series of experiments and case studies on real-world datasets. We conclude the paper in Section 6.

2 RELATED WORK

We survey the related work from three aspects: truth discovery, community question answering and answer selection.

Truth Discovery: The research topic of truth discovery, which aims to identify trustworthy information from conflicting multi-source data, has become a hot topic in recent years. A large variety of methods have been proposed to handle various scenarios such as: different data types [5, 13, 42, 44], source dependencies [5, 15, 43], fine-grained source reliability [20], entity/object dependency [22] and long-tail data [12, 39]. Among them, there are two truth discovery scenarios that are related to the problem studied in this paper. Firstly, as previously discussed, there may exist multiple factors in a text answer. Such setting could be related to the problem of multi-truth discovery [35, 44]. However, there are some significant differences. In [35, 44], the input from each user is structured categorical data. Hence, the methods proposed in these two papers cannot be directly extended to unstructured text data, where answers may be partially correct and contain diverse word expressions. Secondly, there is also some existing work that focuses on unstructured text inputs. For example, [6] specifies a confidence-aware source reliability estimation approach, which takes the SVO triples extracted from webpages as inputs. However, the ultimate goal of that paper is to reduce conflicting information in the process of knowledge
In this paper, we consider a general truth discovery scenario for factoid text questions and answers. Before introducing the problem formulation, we first define some basic terminologies that will be used in the rest of the paper:

Definition 3.1 (Question). A question \( q \) contains \( N_q \) words and can be answered by users.

Definition 3.2 (Answer). An answer given by user \( u \) to question \( q \) is denoted as \( a_{qu} \).

Definition 3.3 (Answer Keyword). Answer keywords are domain-specific content words / phrases in answers. The \( m \)-th answer keyword of the answer given by user \( u \) to question \( q \) is denoted as \( x_{qum} \).

Definition 3.4 (Answer Factor). Answer factors are the key points of the answers, which are represented as clusters of answer keywords. The \( k \)-th answer factor in the answers to question \( q \) is denoted as \( c_{qk} \).

For each question, there can be different answers provided by different users. These answers may consist of complex sentences with multiple factors and can be partially correct. This setting can support a broad range of text data. Formally, the problem discussed in this paper can be defined as:

Definition 3.5 (Problem Definition). Given a set of users \( \{ u \}_{1}^{U} \), a set of questions \( \{ q \}_{1}^{Q} \) and a set of answers \( \{ a_{qu} \}_{q, u = 1, 1}^{Q, U} \), where \( U \) denotes the number of users and \( Q \) stands for the number of questions. The goal of this paper is to extract highly-trustworthy answers and highly-trustworthy key factors in answers for each question.

4 METHODOLOGY

In this section, we first offer an overview of the proposed TextTruth model, and then explain in detail each component of it.

4.1 Overview

When applying truth discovery methods to find the trustworthy answers to complex natural language questions, semantic correlations among answers should be taken into consideration, so that user reliability can be accurately estimated. However, learning accurate vector representations for the whole answers is difficult especially when the context corpus of these answer paragraphs is not sufficiently large. Moreover, due to the complexity of natural language, the meaning of an answer is too complicated to be represented by a single vector. To tackle such challenges, we rely on more fine-grained semantic units (i.e., answer factors) in each answer to determine the trustworthiness of each answer.

In this paper, for each question, we first extract the keywords in each answer and learn their vector representations. Then we cluster these word/phrase-level keywords into semantic clusters (i.e., factors). These factors represent all the possible key points in the answers to a question and can be used to determine the trustworthiness of an answer. For the keywords within each cluster, as they share very similar semantic meanings, their trustworthiness should be almost the same. In addition, users may have different reliabilities, which can be reflected in the answers they provided.

Based on the above ideas, we propose a two-step method to estimate the trustworthiness of each answer. In the first step, we specify a probabilistic model to model the generation of keywords with user reliabilities taken into consideration in Section 4.2. The generative model, which consists of three major components, jointly learns the answer factors and their truth label. The generative model first generates a mixture of answer factors and their semantic parameters. After that, the model generates two-fold user reliability variables, which model the comprehensiveness and accuracy of answer factors provided by a specific user. These two variables...
We develop a probabilistic model to jointly learn the answer factor vectors. The observation of the probabilistic model is denoted as \( \{ q, u \} \). Here, \( \pi \) is a Dirichlet distribution commonly used to generate mixture models. Formally, the mixture distribution \( \pi_q \) is generated as:

\[
\pi_q \sim \text{Dirichlet}(\beta).
\]

(1)

Here, \( \beta \) is a \( K_q \)-dimensional vector, where \( K_q \) denotes the number of factors in the answers to question \( q \).

For the \( k \)-th answer factor under question \( q \), we model its trustworthiness via a binary truth label \( t_{qk} \). Specifically, the model first generates the prior truth probability \( \gamma_{qk} \) which determines the prior distribution of how likely each factor is to be true, from a Beta distribution with hyper-parameters \( \alpha_q \) and \( \beta_q \):

\[
\gamma_{qk} \sim \text{Beta}(\alpha_q, \beta_q).
\]

(2)

Then the truth label \( t_{qk} \) is generated from a Bernoulli distribution with parameter \( \gamma_{qk} \):

\[
t_{qk} \sim \text{Bernoulli}(\gamma_{qk}).
\]

(3)

Finally, to model the semantic characteristic of each answer factor, we define the centroid parameter \( \mu_{qk} \) and concentration parameter \( \kappa_{qk} \) of the VMF distributions from its conjugate prior distribution \( \Phi(\mu_{qk}, \kappa_{qk}; m_0, R_0, c) \) [25], i.e.:

\[
\Phi(\mu_{qk}, \kappa_{qk}; m_0, R_0, c) \propto \left( C_D(\kappa_{qk}) \right)^{\alpha_q} \exp(\kappa_{qk} R_0 m_0^T \mu_{qk}).
\]

(4)

Here, \( C_D(\kappa) = \Gamma_{(D/2-1)}(\kappa)/(\Gamma_D(\kappa)) \) is the modified Bessel function of the first kind. In practice, there may be few answers that are totally irrelevant to the question. Since the answer factors in irrelevant answers are usually supported by very few users, they will not be regarded as trustworthy.

II. User Reliability Modeling: The reliability of each user is inferred according to the answers they provide. As aforementioned, the answer of a user \( u \) may merely cover part of the trustworthy answer factors, and at the same time may consist of untrustworthy answer factors. For instance, some users may only provide the factors that they are very confident of. On the contrary, other users may cover a broad collection of answer factors with different trustworthinesses in their answers. This naturally motivates us to use a two-fold score like [44] to model the reliability of a user.

Suppose we know all the answer factors and their truth labels in advance, for all the questions and their answers, we use \( TP_u \) and \( FP_u \) to denote the number of trustworthy and untrustworthy answer factors that are covered by the answers from user \( u \) (i.e., the number of true positive and false positive factors), respectively. Similarly, we use \( TN_u \) and \( FN_u \) to denote the number of trustworthy and untrustworthy answer factors that are not covered by the answers from user \( u \) (i.e., the number of false negative and true negative factors), respectively. Based on these statistics, we can intuitively use the false positive rate (defined as: \( \frac{FP_u}{TP_u+FN_u} \)), and the true positive rate (defined as: \( \frac{TP_u}{TP_u+FP_u} \)) to fully characterize \( u \)'s reliability.

Let’s resume the discussion of the proposed model. During the generative process, the answer factors and their truth labels are not known in advance. Inspired by [44], we also define two-fold user reliability variables \( \phi^0_u \) and \( \phi^1_u \) to model the false positive rate and the true positive rate of factors that are covered by the answers of user \( u \). Specifically, for each user \( u \), we generate \( \phi^0_u \) and \( \phi^1_u \) from two Beta distributions with hyper-parameters \( (\alpha_{0,1}, \alpha_{0,0}) \) and \( (\alpha_{1,1}, \alpha_{1,0}) \), respectively. Here, \( \alpha_{0,1} \) and \( \alpha_{0,0} \) are the prior false positive count and true negative count, respectively. Similarly, \( \alpha_{1,1} \) and \( \alpha_{1,0} \) are the prior true positive count and false negative count, respectively.
and $\alpha_{t,0}$ stand for the prior true positive count and the false negative count of each source, respectively. Formally:

$$
\begin{align*}
\phi_u^0 & \sim \text{Beta}(\alpha_{0,1}, \alpha_{0,0}) \quad \text{(False Positive Rate)} \\
\phi_u^1 & \sim \text{Beta}(\alpha_{1,1}, \alpha_{1,0}) \quad \text{(True Positive Rate)}. 
\end{align*}
$$

III. Observation Modeling: As aforementioned, we use the vector representations of keywords as observations. For the $m$-th word representation from user $u$ for question $q$, we specify the following generation process.

Firstly, we define a binary indicator $y_{u,q,k}$, which denotes whether the $k$-th factor of the answers to question $q$ should be covered by user $u$, based on the reliability of $u$. For question $q$, if its truth label $t_{qk} = 1$, the probability of user $u$ covering the $k$-th factor in its answer follows a Bernoulli distribution with reliability parameter $\phi_u^1$. Otherwise, if its truth label $t_{qk} = 0$, the probability follows a Bernoulli distribution with reliability parameter $\phi_u^0$. Formally, this process can be written as:

$$
y_{u,q,k} \sim \text{Bernoulli}(\phi_u^{y_{u,q,k}}) \quad \text{if} \quad t_{qk} = 0,
$$

$$
y_{u,q,k} \sim \text{Bernoulli}(\phi_u^1) \quad \text{if} \quad t_{qk} = 1.
$$

To this point, we have determined the set of answer factors that should be covered by the answer $a_{qu}$, with the reliability of $u$ taken into consideration.

Then, for the $m$-th keyword in the answer $a_{qu}$, its factor label $z_{qum}$ is drawn from a probability density function defined as:

$$
P(z_{qum} = k | \pi_q, y_{u,q,k}) = \left\{ \begin{array}{ll} 
\pi_k & \text{if } y_{u,q,k} = 1, \\
0 & \text{if } y_{u,q,k} = 0.
\end{array} \right.
$$

The density function jointly considers the answer factor mixture distribution and the set of binary indicators $y_{u,q,k}$. This means that both semantics and user reliabilities are used to determine the factor label of a specific answer keyword.

With the factor labels determined, the model samples keywords vectors that describe the semantic meaning of its corresponding factor. Note that this procedure should not involve the reliability of a user. The vector representation of a keyword (i.e. $v_{qum}$) is randomly sampled from a vMF distribution with parameter $\mu_{qk}, \kappa_{qk}$:

$$
v_{qum} \sim \text{vMF}(\mu_{qk}, \kappa_{qk}).
$$

Specifically, for a D-dimensional unit semantic vector $v$ that follows vMF distribution, its probability density function is given by:

$$
p(v_{qum}|\mu_{qk}, \kappa_{qk}) = C_D(\kappa_{qk}) \exp(\kappa_{qk} \mu_{qk}^T v_{qum}).
$$

The vMF distribution has two parameters: the mean direction $\mu_{qk}$ and the concentration parameter $\kappa_{qk}$ ($\kappa_{qk} > 0$). The distribution of $v_{qum}$ on the unit sphere concentrates around the mean direction $\mu_{qk}$, and is more concentrated if $\kappa_{qk}$ is larger. In our scenario, the mean vector $\mu$ acts as a semantic focus on the unit sphere, and produces relevant semantic embeddings around it. The superiority of the vMF distribution over other continuous distributions (e.g., Gaussian) for modeling textual embeddings has also been shown in the field of clustering [1] and topic modeling [9].

The overall generative process is summarized in Algorithm 1.

---

### Algorithm 1: Generative Process of TextTruth

```plaintext
for each question $q$ do
  Draw mixture $\pi_q \sim \text{Dirichlet}(\beta)$;
  for each answer factor $k$ do
    Draw centroid and concentration:
    $\mu_{qk}, \kappa_{qk} \sim \mathcal{F}(m_q, R_0, c)$;
    Draw truth parameter: $\gamma_{qk} \sim \text{Beta}(\alpha_0, \alpha_1)$;
    Draw a truth label: $t_{qk} \sim \text{Bernoulli}(\gamma_{qk})$;
  end
  for each user $u$ do
    for each answer $a_{qu}$ do
      for each answer factor $k$ do
        Draw binary label: $y_{u,q,k} \sim \text{Bernoulli}(\phi_u^{t_{qk}})$;
      end
      for each keyword $m$ do
        Draw a answer factor label: $P(z_{qum} = k | \pi, y_{u,q,k})$;
        Draw keyword embedding:
        $v_{qum} \sim \text{vMF}(\mu_{qz_{qum}}, \kappa_{qz_{qum}})$;
      end
    end
  end
end
```

---

### 4.3 Trustworthy-Aware Answer Scoring

Intuitively, the trustworthiness of an answer should be evaluated by the volume of correct information it provides. Hence, we propose a straightforward scoring mechanism to evaluate the trustworthiness score of each answer. Given the inferred truth labels for each answer factor of question $q$, we score the answers according to the number of answer keywords in the answer $a_{qu}$ that are related to the factor with truth label $t_{qk} = 1$, i.e.:

$$
\text{score}_{qu} = \sum_{k=1}^{K_q} N_{u,qk} \mathbb{I}(t_{qk} = 1),
$$

where $K_q$ is the number of answer factors for question $q$. $N_{u,qk}$ denotes the number of keywords that are provided by user $u$ and are clustered into factor $k$. If $t_{qk} = 1$, and $\mathbb{I}(t_{qk} = 1) = 1$ if $t_{qk} = 1$, and $\mathbb{I}(t_{qk} = 1) = 0$ if $t_{qk} = 0$. Note that there are many alternative ways of designing scoring functions.

### 4.4 Model Fitting

In this section, we present the approach to estimating the latent variables and the user reliability parameters.

#### Latent Variable Estimation: We use MCMC method to infer the latent variables $t, z, y$, and $k$. As one can see, the values of $y$ and $z$ have a large impact on the final results, and they may be sensitive to the initialization. Therefore, we make an approximation in latent variable estimation to make the process stable. The detailed steps are specified in the following paragraphs.

First, using conjugate distributions, we are able to analytically integrate out the model parameters and only sample the cluster
assignment variable $z$. This is done as follows:

$$P(z_{qum} = k \mid z_{q,-um}, \beta, m_0, R_0, c)$$

where $z_{q,-um}$ denotes the answer keywords under the $k$-th factor of question $q$, except current answer $v_{qum}$.

The second term in Eq. (11) is similar to the form of $\text{vMF Mixture Model}$, which can be written as:

$$P(v_{qum} \mid z_{q,-um}, \beta) \propto N_{qk} - um + \beta_k,$$  \hspace{1cm} (12)

where $N_{qk} - um$ denotes the number of answer keywords under the $k$-th factor of question $q$ except current answer $v_{qum}$.

The conditional distribution of $\kappa_{qk}$ is again not of a standard form, so we use a step of Metropolis Hasting sampling (with log-normal proposal distribution) to sample $\kappa_{qk}$. To this point, we get the full expression of Eq. (11). In the circumstance when the model fitting efficiency becomes a concern, the sampling process specified by Eq. (11) can be approximated via the method specified in [30], which also produces satisfactory results.

Here, we make an approximation by removing the impact of $y$ in terms of determining the value $z$. For the answer provided by user $u$ for questions $q$, $y_{u,q}$ is determined via:

$$y_{u,q} = \begin{cases} 0 & \text{If $\beta$ } m \text{ satisfies } z_{qum} = k, \\ 1 & \text{Otherwise}. \end{cases}$$ \hspace{1cm} (15)

Finally, we move on to sample the truth label for each answer factor under each question $q_k$ via the following posterior distribution:

$$P(t_{qk} = x \mid q_k, z_q, y_q, \alpha_{0,0}, \alpha_{0,1}, \alpha_{1,0}, \alpha_{1,1}, \alpha_{0}'(a), \alpha_{1}'(a))$$

where $U_q$ is the set of users that provide answer for question $q$. Here, $x \in \{0, 1\}$. $n_{u,0,0}$, $n_{u,0,1}$, $n_{u,1,0}$ and $n_{u,1,1}$ denote the number of true negative, false positive, false negative and true positive factors provided by user $u$, respectively.

**User Reliability Estimation:** With $t$, $y$, and $z$ determined, we are able to obtain the closed-form solution for $\phi_u^0$ and $\phi_u^1$ by setting the partial derivatives of the negative log-likelihood respective to $\phi_u^0$ and $\phi_u^1$ to zero:

$$\phi_u^0 = \frac{\alpha_{0,1} + n_{u,0,1}}{\alpha_{0,0} + \alpha_{0,1} + n_{u,0,1} + n_{u,0,0}},$$ \hspace{1cm} (17)

where $n_{u,0,0}$, $n_{u,0,1}$, $n_{u,1,0}$ and $n_{u,1,1}$ are user reliability statistics, which denote the number of true negative, false positive, false negative and true positive factors provided by user $u$, respectively. Moreover, these statistics also allow us to calculate other user reliability metrics, e.g., precision score of a user:

$$\text{prec}_u = \frac{\alpha_{1,1} + n_{u,1,1}}{\alpha_{0,1} + \alpha_{1,1} + n_{u,0,1} + n_{u,1,1}}.$$ \hspace{1cm} (19)

This score is also used in the experiment section to validate the estimated user reliability.

### 5 EXPERIMENTS

In this section, we empirically validate the performance of the proposed method from the following aspects: Firstly, we compare the performance of the proposed method with the state-of-the-art truth discovery methods as well as a couple of retrieval based schemes to demonstrate the advantage of utilizing fine-grained semantic units of answers for better answer trustworthiness estimation. After that, we provide a case study to show that the results produced by the proposed method are highly interpretable. Finally, we validate the estimated user reliabilities with groundtruth to further prove that the proposed method can make a good estimation of user reliabilities.

#### 5.1 Datasets

**SuperUser Dataset & ServerFault Dataset:** These two datasets are collected from the community question answering (CQA) websites SuperUser.com and ServerFault.com, respectively. These two websites are mainly focused on the questions about general daily computer usages and server administration, respectively. The task on these datasets is to extract the most trustworthy answer to each question. We use the answers’ votes from SuperUser.com and ServerFault.com as the groundtruths for evaluation.

**Student Exam Dataset [24]:** This dataset is collected from introductory computer science assignments with answers provided by a class of undergraduate students in the University of North Texas. 30 students submit answers to these assignments. For each assignment, the students’ answers are collected via an online learning environment. The task on this dataset is to extract Top-K (K is set to 1-10 in this paper) trustworthy student answers for each question. The groundtruth answers are given by the instructors. All the answers are independently graded by two human judges, using an integer scale from 0 (completely incorrect) to 5 (perfect answer).

The statistics of these three datasets are shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1: Data Statistics.</th>
<th>SuperUser</th>
<th>ServerFault</th>
<th>Student Exam</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Questions</td>
<td>3379</td>
<td>7621</td>
<td>80</td>
</tr>
<tr>
<td># of Users</td>
<td>1036</td>
<td>1920</td>
<td>30</td>
</tr>
<tr>
<td># of Answers</td>
<td>16014</td>
<td>40373</td>
<td>2273</td>
</tr>
</tbody>
</table>

**Pre-Processing:** For all the datasets, we discard all code blocks, HTML tags, and stop words in the text. Answer keywords are extracted using entity dictionary and Stanford POS-Tagger\(^3\). To

\(^3\)https://nlp.stanford.edu/software/tagger.shtml
train word vector representations, we utilize all the crawled texts as the corpus. Skip-gram architecture in package gensim\(^4\) is used to learn the vector representation of every answer keyword. The dimensionality of word vectors is set to 100, context window size is set to 5, and the minimum occurrence count is set to 20. For more details on the embedding algorithm, please refer to [23].

5.2 Experiment Protocols

5.2.1 Comparison Methods. We compare the proposed TextTruth model against several state-of-the-art truth discovery and retrieval-based answer selection approaches.

**Bag-of-Word (BOW) Similarity:** The bag-of-word vectors of questions and their answers are extracted. Answers are ranked according to the similarity values between the question vector and its corresponding answer vectors.

**Topic Similarity:** We utilize Latent Dirichlet Allocation (i.e., LDA [2]) to extract a 100-dimension topic representation for each question and its corresponding answers. Similar to BOW, answers are ranked according to the cosine similarity to the question.

**CRH \([13]\) + Topic Dist.:** CRH is an optimization based truth discovery framework which can handle both categorical and continuous data. The goal of the optimization problem is to minimize the weighted loss of the aggregation results. In the experiment, we use the topic distributions as the representations of the whole answers to be fed to CRH.

**CRH \([13]\) + Word Vec.:** This baseline approach is similar to CRH + Topic Dist. except that the inputs are changed to the average word vectors of answers. These word vector representations are learned as in [23].

**CATD \([12]\) + Topic Dist.:** CATD is another optimization based truth discovery framework which considers the long-tail phenomena in the data. The optimization objective is similar to that of CRH. However, the upper bounds of user reliability are used for weight loss calculation. Similar to CRH + Topic Dist., we use the topic distributions as the representations of the whole answers to be fed to CATD.

**CATD \([12]\) + Word Vec.:** This baseline approach is similar to CATD + Topic Dist. except that the inputs are changed to the average word vectors of answers. The word vector representations are the same as those in CRH + Word Vector.

For each baseline approach, we implement it and set its parameters according to the method recommended by the original papers.

5.2.2 Evaluation Metrics. Due to the differences in dataset characteristics, evaluation metrics for three datasets are slightly different. On CQA datasets, we report the precision of returned best answers from each method for each question. On student test dataset, we report the average score of returned top-K (K is set to 1-10 in this paper) trustworthy answers from each method for each question.

5.3 Performance and Analysis

The results are shown in Figure 3 and Table 2. For student exam dataset, we only show the results on exam 1-3 data. The results on rest exams follow the same tendency. As one can see, the proposed method TextTruth consistently outperforms all the baseline methods. By outperforming various retrieval-based approaches and state-of-the-art truth discover approaches, the proposed TextTruth demonstrates its great advantages on natural language data.

Table 2: Results on ServerFault Dataset & SuperUser Dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>ServerFault</th>
<th>SuperUser</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOW Similarity</td>
<td>0.2077</td>
<td>0.1944</td>
</tr>
<tr>
<td>Topic Similarity</td>
<td>0.2462</td>
<td>0.2462</td>
</tr>
<tr>
<td>CATD + Topic Dist.</td>
<td>0.2311</td>
<td>0.2308</td>
</tr>
<tr>
<td>CATD + Word Vec.</td>
<td>0.1821</td>
<td>0.2234</td>
</tr>
<tr>
<td>CRH + Topic Dist.</td>
<td>0.2453</td>
<td>0.2453</td>
</tr>
<tr>
<td>CRH + Word Vec.</td>
<td>0.1847</td>
<td>0.2234</td>
</tr>
<tr>
<td>TextTruth</td>
<td>0.3985</td>
<td>0.4019</td>
</tr>
</tbody>
</table>

The reasons why the proposed TextTruth surpasses all the baseline methods are as follows. First, retrieval-based approaches (i.e., BOW Similarity and Topic Similarity) rank the answers merely based on the semantic similarity between the question and answers. However, a question itself does not necessarily cover all the semantics that should be covered in ideal answers. Therefore, retrieval-based methods only discover relevant answers instead of trustworthy answers. On the other hand, although existing truth discovery methods can capture user reliability for answer ranking, the performance is not very satisfactory. This is because these truth discovery approaches treat the answers as an integrated semantic unit, and ignore the fact that the semantic meaning of each answer may be complicated. Therefore, single vector representations fail to capture the innate correlations among these answers. To make things worse, CRH and CATD regard the weighted aggregation of these single vector representations as the “true” semantic representation to evaluate user reliabilities. However, answers from different users may involve distinct aspects of answers. Therefore, aggregating semantic representation of answers with distinct aspects only produces an inaccurate representation, which cannot be used to correctly estimate the reliabilities of users. The inaccurate user reliability estimation would further lead to incorrect aggregated results.

In contrast to existing approaches, the proposed TextTruth regards each answer as a collection of fine-grained semantic units (i.e., factors), which are represented by separated keyword vector representations. Based on these semantic units, TextTruth discovers the innate factors of each answer by grouping keywords into factors, and evaluates the trustworthiness of each answer on the top of these factors. As mentioned in the above paragraph, the major reason why existing truth discovery methods cannot produce satisfactory results is that these methods cannot aggregate the semantic representation of answers with distinct aspects effectively. Instead, the proposed TextTruth evaluate the users’ reliabilities according to whether their answers contain keywords from the factors that are regarded to be correct (or incorrect). Therefore, the trustworthiness of each answer is better evaluated, which leads to the best result.

5.4 Case Study

To better evidence the analysis above, we give a case study on a question in the exam dataset. The question is related to the data structure. The result of the case study is shown in Table 3. In Table 3
A tree is a finite set of one or more nodes with a specially designated node called the root and the remaining nodes are partitioned into disjoint sets where each of these sets is a tree. Moreover, we can observe that the top-ranked words in blue color are keywords that are estimated to be trustworthy, while words in red color are keywords that are estimated to be untrustworthy or unrelated. The groundtruth answer is provided by the instructors.

As one can see, the proposed method can automatically select keywords that are meaningful to the questions, such as “node”, “tree” and “root”. Moreover, we can observe that the top-ranked answers have more true keywords than low-ranked untrustworthy answers. These phenomena again demonstrate that the results of the proposed model are both effective and interpretable. The case study also demonstrates why existing approaches fail to produce satisfactory results. First, the question itself merely consists of one keyword ‘tree’. Therefore, retrieval-based methods, rank ‘Untrustworthy Answer’ over ‘Top Answer 2’, because it contains exactly the same keyword that exists in the question. This indicates that we cannot rely merely on relevance to find trustworthy answers. Second, we can see that the correct keywords involve multiple aspects (i.e., factors). These factors shape a comprehensive description of a tree. Such phenomenon is very common in natural language questions and answers, but cannot be successfully handled by the existing methods. That is why the proposed method can produce better results than the state-of-the-art truth discovery methods.

### 5.5 User Reliability Validation

The quantitative results and the case study shown above have demonstrated that the proposed method can outperform other baseline methods. In this section, we further exhibit the estimated user reliabilities by the proposed approach. As there are no direct user reliability values on the CQA dataset, we only investigate the estimated user reliabilities on the student exam dataset. Specifically, we use the average score of a student’s answer to each question as the groundtruth reliability. Intuitively, the learned two-fold user reliability parameters (i.e., $\phi_0$ and $\phi_1$) are not directly proportional to the true user reliability; we use the metric $\text{prec}$ defined in Eq. (19) for user reliability validation. Due to space limitation, we only show one example, which comes from the mid-result of TextTruth on exam 10, in Figure 4. In Figure 4, each point denotes a user. The Y-axis is the user reliability groundtruth and the X-axis is the estimated user precision score. As one can see, the estimated user reliability score (X) typically increases when the groundtruth user...
As an emerging topic, truth discovery has shown its effectiveness in a wide range of applications with structured data. However, existing methods all suffer on unstructured text data, due to the semantic ambiguity of natural languages and the complexity of text answers. To tackle these challenges, in this paper, we propose a probabilistic model named TextTruth that takes vector representations of key factors extracted from answers as inputs and outputs the ranking of answers based on the trustworthiness of key factors within each answer. Specifically, the model jointly learns the clustering label and truth label for each answer factor cluster through modeling the generative process of answer factors’ embedding representations. Experimental results on three real-world datasets prove the effectiveness of the proposed TextTruth model. Furthermore, case studies illustrate that the learned labels are interpretable.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and suggestions. This work was sponsored in part by US National Science Foundation under grants IIS-1319973, IIS-1553411, CNS-1652503 and CNS-1737590. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency.

REFERENCES


[17] Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, and Jiawei Han. 2015. Plotcrowd: Fine grained truth discovery for crowdsourced data aggregation. *In Proc. of SIGKDD*. 745–754.

[18] Yaliang Li, Qi Li, Jing Gao, Lu Su, Bo Zhao, and Jiawei Han. 2015. On the discovery of evolving truth. *In Proc. of SIGKDD*. 675–684.

[19] Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. 2015. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. *In Proc. of SIGKDD*. 745–754.


[22] Yaliang Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. 2015. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. *In Proc. of SIGKDD*. 745–754.

[23] Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. 2015. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. *In Proc. of SIGKDD*. 745–754.
