A Survey of Truth Discovery in Information Extraction

Xuan Wang, Yu Zhang and Yinyin Chen
1 University of Illinois at Urbana-Champaign, Champaign, IL USA
1{xwang, yuz9, ychen409}@illinois.edu

ABSTRACT

In the information era, the data is generated at a dramatic speed and can be collected from various sources. However, the multi-source data of the same object often conflict with each other. Truth discovery is proposed to aggregate the information from different sources and infer the true fact from the data. Information extraction is a task to extract information, such as entities and relations, from unstructured text data. The application of truth discovery in information extraction gains great attention recently as each fact can often be extracted by various systems from various sources. In this survey, we focus on the truth discovery methods, information extraction sub-tasks and application of truth discovery in information extraction.

1. INTRODUCTION

In the information era, the data is generated at a dramatic speed. These data are collected and analyzed by individuals, companies, and governments. However, the collected information may conflict with each other due to errors, missing records, out-of-date data, etc. How to aggregate the information from different sources to get the truth fact becomes a key problem.

A straightforward approach is to conduct majority voting for multi-source data. However, this approach may not always work in the cases that the true facts are only reported by a few high quality sources. The challenge here is that source reliability is usually unknown a priori in practice and has to be inferred from the data. Truth discovery methods are proposed to address this challenge. The two principles of truth discovery: (1) the sources that provide true information more often will be assigned higher reliability degrees, (2) the information that is supported by reliable sources will be regarded as truths.

Most truth discovery approaches take the inputs as structured data. Recently, truth discovery in information extraction (IE) gains great attention. The main purpose of information extraction is to extract entities and relations from text. The sub-tasks of IE include named entity recognition (NER), relation extraction (RE) and open-domain information extraction (OpenIE). IE deals with unstructured text data that are more noisy but also brings more information (e.g., uncertainty, evidence). For application of truth discovery in IE, the inaccurate information can come from two layers: the text corpora and the information extractor, which brings new challenges to source weight estimation.

In this survey, we first introduce the problem of truth discovery and discuss the current methods in this field in Section 2. Then we introduce the major sub-tasks in information extraction in Section 3. Last, we discuss some representative work of incorporating truth discovery in information extraction in Section 4.

2. TRUTH DISCOVERY

In this section, we define the problem of truth discovery, discuss the current methods in this field, and compare them from different perspectives.

2.1 Problem Setup

In the information era, the data is generated in a dramatic speed. These data are collected and analyzed by individuals, companies, and governments. However, the collected information of the same object may conflict with each other. The main task of truth discovery is to integrates noisy information from different data sources to find the truth for object by evaluating the reliability of each source.

To clearly state the problem, we first introduce some definitions and notations following [17].

- An object/fact \( f \) is a thing of interest, a source \( s \) describes the place where the information about objects can be collected from, and a claimed value \( o_s^f \) represents the information provided by source \( s \) about object \( f \).
- An observation \( o_s^f \), also known as a record/claim, is a 3-tuple that consists of an object \( f \), a source \( s \), and its provided value \( o \).
- The identified truth for an object \( t_f \) is the information selected as the most trustworthy one from all possible candidate values about this object.
- Source weight \( w_s \) reflects the probability of source \( s \) providing trustworthy information. A higher \( w_s \) indicates that source \( s \) is more reliable and the information from this source is more likely to be accurate.

The formal definition the truth discovery as defined by [17] is as follows:

**Definition 1.** For a set of objects \( \mathcal{F} \) that we are interested in, related information can be collected from a set of sources \( \mathcal{S} \). Our goal is to find the truth \( t_f \) for each object \( f \in \mathcal{F} \) by resolving the conflicts among the information
from different sources \( \{ o_f^s \}_{s \in S} \). Meanwhile, truth discovery methods estimate source weights \( \{ w_s \}_{s \in S} \) that will be used to infer truths.

The general principles of truth discovery adopted by truth discovery approaches are:

- A source providing more trustworthy information is more reliable.
- A more reliable source tends to provide more trustworthy information.

Accordingly, several assumptions [17] are often made on source reliability:

- Source consistency assumption: A source is likely to provide true information with the same probability for all the objects.
- Source independence assumption: The true information is more likely to be identical or similar among different sources, and the false information provided by different sources is more likely to be different.

### 2.2 Iterative Methods

In this section, we discuss several truth discovery methods [8, 16, 34, 54], which are designed as iterative procedures. These algorithms iteratively do the following steps until state stage is achieved:

- Conditioned on the current weights, for each claim \( o_f \), calculate the confidence/vote \( \text{vote}(o_f) \) by weighted aggregation of the observations from sources, and infer the truth \( t_f \).
- Calculate the weight for each source based on the claims given by the source and corresponding current truth.

#### 2.2.1 TruthFinder

TruthFinder [54] is a straightforward implementation in web site information. The weights of a source/web site \( s \) is calculated by the average vote of claims provided by \( s \).

\[
{w_s} = \frac{\sum_{f \in F_s} \text{vote}(o_f)}{|F_s|}
\]

where \( F_s \) is the set of the facts that is provided by source \( s \). Assuming the sources are consistent and independent, the probability of the claim \( o \) is wrong is the product of the probabilities that the sources providing \( o \) make mistakes. Therefore, confidence/vote of the claim \( v_o \) is computed by

\[
\text{vote}(o) = 1 - \prod_{s \in S_o} (1 - w_s)
\]

where \( S_o \) is the set of sources providing \( o \).

In each iteration, TruthFinder first uses the source weight/trustworthiness to compute the fact confidence/vote, and then recomputes the source trustworthiness from the fact confidence. Source quality finding and truth finding mutually enhance each other.

In most of the cases, the source independence assumption does not hold. Based on the TruthFinder, AccuSim [8] takes the source dependence into account, and give more credits to the independent sources. AccuCopy [16] further considers the copying relations among sources.

#### 2.2.2 Investment

In Investment algorithm [34], sources uniformly invest their reliability in claims and obtain corresponding returns. Essentially, it modifies the calculation of source weights \( w_s \) and information vote \( \text{vote}(o) \). Specifically, the vote of a claimed value is computed by a non-linear function \( G \),

\[
\text{vote}(o) = G \left( \sum_{s \in S} \frac{w_s}{|F_s|} \right)
\]

where the \( G \) is suggested to be \( G(x) = x^g \) with \( g = 1.2 \) empirically. In this step, the sources invest their reliability among claimed values, and in the next weights computation step, they obtain the credits back from the identified truths.

The weights of a source \( s \) is given by

\[
w_s = \sum_{f \in F_s} \left( \text{vote}(o_f) \frac{w_s}{|F_s|} \right) \frac{\sum_{s' \in S_f} w_{s'} / |F_{s'}|}{\sum_{s' \in S_f} w_{s'} / |F_{s'}|}
\]

Each source gets credit back proportional to the investment of the claim they made. As the vote function \( G \) is a super linear function, the sources providing more trustworthy information will get more credits back. In return, the trustworthy information will get higher vote and contribute more to the source weight.

### 2.3 Graphical Model Based Methods

Probabilistic graphical models are widely applied in this problem, such as Latent Truth Model(LTM) [50] and Gaussian Truth Model(GTM) [59]. The common Bayesian framework can be summarized as follows:

- Each claimed values \( o_f^s \) is generated based on the corresponding truth \( t_f \) and source weight \( w_s \),
  \[
o_f^s \sim f(t_f, w_s).
\]
- Source weight \( w_s \) is the precision (the reciprocal of variance), i.e., if source weight \( w_s \) is high, the variance is relative small and thus the claimed value \( o_f^s \) will be close to the truth \( t_f \).

And the inference of these models are usually done by MCMC and EM algorithms.

#### 2.3.1 Latent Truth Model

Latent Truth Model(LTM) [50] is designed to solve the multiple truths problem in truth discovery, i.e., for the same object \( o \), multiple claims are possibly true. It considers two types of errors, false positive and false negative. Figure 1 shows the graphical structure of conditional dependence of our model.

Specifically,

- For each fact \( f \),
  - the false positive rate \( \phi_0^f \) follows a beta distribution with parameter \( \alpha_0, \alpha_0 \),
    \[
    \phi_0^f \sim \text{Beta}(\alpha_0, \alpha_0);
    \]
  - the true positive rate \( \phi_1^f \) follows a beta distribution with parameter \( \alpha_1, \alpha_1 \),
    \[
    \phi_1^f \sim \text{Beta}(\alpha_1, \alpha_1);
    \]
- For each source \( s \),
  - the false positive rate \( \phi_0^s \) follows a beta distribution with parameter \( \alpha_{0,1}, \alpha_{0,0} \),
    \[
    \phi_0^s \sim \text{Beta}(\alpha_{0,1}, \alpha_{0,0});
    \]
  - the true positive rate \( \phi_1^s \) follows a beta distribution with parameter \( \alpha_{1,1}, \alpha_{1,0} \),
    \[
    \phi_1^s \sim \text{Beta}(\alpha_{1,1}, \alpha_{1,0}).
    \]
its prior truth probability \( \theta_f \) follows a beta distribution with parameter \( \beta_1, \beta_0 \),
\[
\theta_f \sim \text{Beta}(\beta_1, \beta_0);
\]
- its truth label \( t_f \) follows a Bernoulli distribution with probability \( \theta_f \),
\[
t_f \sim \text{Bernoulli}(\theta_f).
\]

- For each claim \( c \) of fact \( f \) from source \( s \), the observation of \( c, o_c \), is from a Bernoulli distribution,
- If \( f \) is false, the probability is the false positive rate of the source \( s \),
\[
o_c \sim \text{Bernoulli}(\phi_{c}^0);
\]
- If \( f \) is true, the probability is the true positive rate of the source \( s \),
\[
o_c \sim \text{Bernoulli}(\phi_{c}^1);
\]

The full likelihood of observation \( o_t \), latent variables \( t \), and unknown parameters \( \theta, t, \phi^1 \) and \( \phi^0 \) given the hyper-parameters \( \alpha, \alpha_1, \beta \) is
\[
p(o_t, t, s, \theta, \phi^1, \phi^0 | \alpha, \alpha_1, \beta) = \prod_{s \in S} p(\phi^0_s | \alpha_0) p(\phi^1_s | \alpha_1) \times \prod_{f \in F} \left( p(\theta_f | \beta) \theta_f^t (1 - \theta_f)^{(1-t_f)} \prod_{c \in C_s} p(o_c | \phi^f_c) \right)
\]

The maximum a posterior (MAP) estimate for truth \( t \) is
\[
t_{MAP} = \arg \max_t \int \int p(o_t, t, s, \theta, \phi^1, \phi^0 | \alpha_0, \alpha_1, \beta) d\theta dt d\phi^1 d\phi^0
\]

The collapsed Gibbs sampling can be utilized because of the conjugacy of exponential families of the truth probability \( \theta \), source true discovery rate \( \phi^1 \) and false discovery rate \( \phi^0 \), in which they can be integrated out in the sampling process. In other words we can just iteratively sample the truth of facts \( t \) and avoid sampling these other quantities, which yields efficiency,
\[
p(t_f = i | t_{-f}, o_t, s) \propto \beta_1 \prod_{c \in C_s} \frac{n_{c,-i,0} - \alpha_{1,0}}{n_{c,-i,1} + n_{c,-i,0} + \alpha_{1,0}}
\]

where
\[
n_{c,-i,j} = |\{c \in C_s | s_{c'} = s_c, tf_{c'} = i, o_{c'} = j\}|
\]
i.e., the number of \( s_c \)'s claim whose observation is \( j \), and referred fact is not \( f \) and its truth is \( i \).

Accordingly, the MAP of source quality can be estimated as follows:
\[
sensitivity(s) = \phi^1_s = \frac{E(n_{s,i,1}) + \alpha_{1,1}}{E(n_{s,1,0}) + E(n_{s,1,1}) + \alpha_{1,0} + \alpha_{1,1}}
\]
\[
specificity(s) = 1 - \phi^0_s = \frac{E(n_{s,0,0}) + \alpha_{0,0}}{E(n_{s,0,0}) + E(n_{s,0,1}) + \alpha_{0,0} + \alpha_{0,1}}
\]

where \( E(n_{s,i,j}) = \sum_{c \in C_s, s_{c'} = \alpha, o_{c'} = j} p(tf_c = i) \) is the expected quality counts of source \( s \).

### 2.3.2 Gaussian Truth Model

Gaussian Truth Model(GTM) [59] is designed to adapt to numerical truth. It leverages two Gaussian generative processes in a principled way to simulate the generation of numerical truth and claims. Figure 2 is the graphical representation of GTM. Specifically,

- For each source \( s \), its quality \( \sigma_s^2 \) is generated from a inverse Gamma distribution with parameter \( (\alpha, \beta) \),
\[
\sigma_s^2 \sim \text{Inver-Gamma}(\alpha, \beta).
\]
- For each fact \( f \), the truth \( t_f \) is generated from Gaussian distribution with mean \( \mu_0 \) and variance \( \sigma_0^2 \).
\[
t_f \sim \mathcal{N}(\mu_0, \sigma_0^2) \sim \exp(-\frac{(t_f - \mu_0)^2}{2\sigma_0^2}).
\]
- For each claim \( c \) of fact \( f \), the observation \( o_c \) follows a Gaussian distribution with mean \( t_f \) and variance \( \sigma^2_{oc} \).
\[
o_c \sim \mathcal{N}(t_f, \sigma^2_{oc}) \sim \sigma^{-1}_{oc} \exp(-\frac{(o_c - t_f)^2}{2\sigma^2_{oc}}).
\]

The full likelihood of observed data \( o \) and unknown parameters \( t, \sigma^2 \) given the hyper-parameters \( \mu_0, \sigma^0_0, \alpha, \beta \) can be
It is not hard to see that more accurate sources (sources with \( \sigma^2 \)) can be achieved.

Such iterative computation can guarantee the log-likelihood will always increase in each iteration, so that a local optimal can be achieved.

2. Repeatedly estimate the truth and source weights following the general principles:
   - sources with higher weights contributes more in estimating the truth;
   - sources providing more truth receive higher weights.

3. Stop until certain criterion is achieved.

2.5 Advanced Methods

We briefly describe several representative advanced truth discovery work [9, 36, 50, 55]

- Multi-source sensing model (MSS) [36] is a graphical model based method considering the source dependency. It groups the highly dependent source together and assesses their reliability at the group level to decrease the risk of overusing the information from dependent sources.
- Regular EM [50] focused on the application of social sensing, using a maximum likelihood estimation approach. The optimal solution is obtained by solving an expectation maximization problem.
- CopyCEF [9] is a Bayesian probabilistic method for dynamic source discovery and copying relationship detecting. It develops an HMM model to capture whether a source is a copier of another source and the moment it starts to copy. It introduces a quality measure of data sources, CEF-measure, to measure the coverage, exactness, and freshness.
- Semi-Supervised Truth Discovery(SSTF) [55] is a semi-supervised approach that finds truth with the help of a small amount of ground truth data.

3. INFORMATION EXTRACTION

The main purpose of Information Extraction (IE) [44] is to extract tuples in the form of \( (h, r, t) \) or \( (e_h, r, e_t) \). It is a step towards automating knowledge acquisition from text. The sub-task of IE can be divided into three categories: named entity recognition (NER), relation extraction (RE) and open-domain information extraction (OpenIE).

3.1 Named Entity Recognition

Entity recognition can be viewed as the first step of tuple extraction. It aims to discover entities (or token spans) of certain types from a given corpus. Existing studies leverage various labels of human supervision to do NER, from fully annotated documents (supervised), seed entities (weakly supervised), to knowledge bases (distantly supervised). In recent years, researchers considered using neural network models to automatically generate quality features. For example, Lample et al. [15], Ma and Hovy [24] and Liu et al. [22] all adopted a BiLSTM-CRF structure.
Since the Long-short Term Memory (LSTM) unit is commonly used in NER and RE, we briefly introduce it here. The formulas to update an LSTM unit at time $t$ are:

$$
i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i)$$

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f)$$

$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o)$$

$$g_t = \tanh(W^g x_t + U^g h_{t-1} + b^g)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

$$h_t = o_t \odot \tanh(c_t)$$

where $\sigma(\cdot)$ and $\tanh(\cdot)$ denote element-wise sigmoid and hyperbolic tangent functions, respectively, and $\odot$ denotes element-wise multiplication. The $i_t$, $f_t$, and $o_t$ are referred to as input, forget, and output gates, respectively. At $t = 1$, $h_0$ and $c_0$ are initialized to zero vectors. The trainable parameters are $W$, $U$, and $b$ for $j \in \{i, f, o, g\}$.

In the first character layer, character embeddings are used as the input to model character sequences. The difference in this layer is that Ma and Hovy [24] utilize Convolutional Neural Network (CNN) as the encoder, while the other two use Bi-directional Long-short Term Memory (BiLSTM). Besides, Liu et al. [22] further incorporate language models into the structure. In the second word layer, all of the three neural network architectures feed the encoded character representation and word embeddings into a BiLSTM. Finally, the word representation will be fed into the third layer (i.e., CRF) to get prediction of the label.

No matter which structure to use, to obtain an effective model, the amount of data is significant. Wang et al. [52] further propose a multi-task learning framework to combine different partially labeled datasets together to train a bi-neral NER model.

### 3.1.2 Weakly Supervised Methods

Weakly-supervised methods utilize a small set of typed entities as seeds and extract more entities of target types, which can largely reduce the amount of required labeled data. Its formal definition is:

**Definition 2.** Given a text corpus $D$, some target types $T$, where each entity type $t$ is characterized by a few seed instances $\{(e_i, t)\}_{i=1}^N$, weakly-supervised entity recognition aims to extract more instances $\{(e_i, t)\}_{i=1}^M$ from the corpus.

Pattern-based bootstrapping [12, 46] derives patterns from contexts of seed entities and uses them to incrementally extract new entities and new patterns unrestricted by specific domains, but often suffers low recall and semantic drift [20]. Iterative bootstrapping, such as probabilistic method [33] and label propagation [48] softly assign multiple types to an entity name and iteratively update its type distribution, yet cannot decide the exact type for each entity mentioned based on its local context.

### 3.1.3 Distantly Supervised Methods

Distantly supervised methods [18, 31, 40–42] avoid expensive human labels by leveraging type information of entity mentions which are confidently mapped to entries in Knowledge Bases. Its formal definition is:

**Definition 3.** Given a corpus $D$, a target type set $T$ and a knowledge base $\Psi$, distantly supervised entity recognition aims to: (1) extract candidate entity mentions $M$ from $D$; (2) generate seed mentions $M_0$ with $\Psi$; and (3) for each unlinkable candidate mention $m \in M_0$, estimate its type indicator vector $y_m$ to predict its type.

At the third step, linked mentions are used to type those unlinked ones in different ways, including training a contextual classifier [31], learning a sequence labeling model [18] and serving as labels in graph-based semi-supervised learning [40] (ClusType). Ren et al. [41, 42] further propose a novel optimization problem which is formulated to jointly embed entity mentions and types to the same space (PLE and AFET). They model noisy type set with a partial-label rank and type correlation with adaptive margin rank loss.

### 3.2 Relation Extraction

Relation Extraction can be viewed as the downstream step of NER. Given the corpus, RE aims to predict the relation between two detected entities. Many efforts [6, 29, 43] have been done on sentence-level relation extraction, where the algorithm can only use one single sentence to predict the relations of two entities within the sentence. Instead of looking at individual sentences, corpus-level relation extraction [3, 27, 44, 57] identifies relation instances from text corpora using evidences from multiple sentences.

RE can be either fully supervised or weakly supervised. Similar to weakly supervised NER, the definition of weakly supervised RE is:

**Definition 4.** Given a text corpus $D$, and some target relations $R$, where each target relation $r$ is characterized by a few seed instances $\{(e_{k_1}, e_{k_2}, r)\}_{k=1}^N$, weakly-supervised relation extraction aims to extract more instances $\{(e_{k_1}, e_{k_2}, r)\}_{k=1}^M$ from the corpus.

There are broadly two types of approaches for corpus-level relation extraction: the pattern-based approach and the distributional approach.

#### 3.2.1 Pattern-based Methods

Pattern-based approaches predict the relation of an entity pair from multiple sentences mentioning both entities. To do that, traditional approaches [32, 45, 53] extract textual patterns (e.g., tokens between a pair of entities) and new relation instances in a bootstrapping manner. However, many relations could be expressed in a variety of ways. Due to such diversity, these approaches often have difficulty matching the learned patterns to unseen contexts, leading to the problem of semantic drift and inferior performance. Recent approaches address the problem by encoding textual patterns with neural networks. For example, CNN+ATT [19] leverages convolutional neural networks to encode and classify each sentence, and then consolidates the results of different sentences using an attention mechanism. PCNN+ATT [19] further involves the position embedding for each word and entity. PathCNN [58] considers some other sentences mentioning only one of the two entities in addition to the sentences mentioning both.

#### 3.2.2 Distributional Methods

Distributional approaches resort to the corpus level co-occurrence statistics of entities. The basic idea is to learn low-dimensional representations of entities to preserve such statistics, so that entities with similar semantic meanings
tend to have similar representations. With entity representations, a relation classifier can be learned using the labeled relation instances, which takes entity representations as features and predicts the relation of a pair of entities. To learn entity representations, some approaches [26, 35] only consider the given text corpus. Despite the unsupervised property, their performance is usually limited due to the lack of supervision. To learn more effective representations for relation extraction, some other approaches jointly learn entity representations and relation classifiers using the labeled instances. For example, TransE [4] leverages distributional approach for knowledge base completion, but only using the given seed instances for training. RK [51] further utilizes both the text corpus and the given relation instances to learn entity representations.

3.2.3 Hybrid Methods
There are also handful studies trying to integrate the distributional and pattern-based approaches. Typically, they jointly train a distributional model and a pattern model. For example, LexNET [47] uses a recurrent layer to encode local textual patterns, and then uses the encoding vector together with entity representations for prediction. DPE [37] jointly models the distributional information in text corpora, the given relation instances and the textual patterns. CONV [49] integrates the distributional and pattern-based methods by jointly optimizing the given seed instances and the instances extracted by textual patterns. REPPEL [38] consists of a pattern module and a distributional module. The pattern module aims at learning a set of reliable textual patterns for relation extraction; while the distributional module tries to learn a relation classifier on entity representations for prediction.

3.3 Open-domain Information Extraction
Named Entity Recognition and Relation Extraction can help us determine entity types and relation labels appeared in the training set or knowledge bases. However, their power is confined to the given entity type set $T$ or relation type set $R$, which makes it hard to transfer these models to new types or new domains. Starting from [2], Open domain information extraction (OpenIE), which directly extracts tuples from the corpus without given types, has been extensively studied in literature.

**Definition 5.** Given a corpus $D$, open-domain information extraction aims to extract entity phrases $E$, relation phrases $R$ (without given entity types or relation types) and tuples $\{(e_h, e_t, r_t)| i = 1, ..., n\}$, $(e_h, e_t \in E, r_t \in R)$, where each entity pair is extracted within one sentence.

Most of the existing work follows two lines, which are pattern-based methods and clause-based methods.

3.3.1 Pattern-based Methods
Pattern based information extraction can be as early as Hearst patterns [13] like “NP0 such as [NP1, NP2, ...]” for hyponymy relation extraction. Carlson et al. [5] and Mitchell et al. [28] introduced Never-Ending Language Learning (NELL) based on free text predicate patterns. ReVerb [10] identified relational phrase via part-of-speech-based regular expression. Besides part-of-speech tags, recent works start to use more linguistic features, like dependency parsing, to induct long distance relationships, such as.

PATTY [32] and OLLIE [45]. MetaPAD [14] generates quality meta patterns by context-aware segmentation, groups synonymous meta patterns, and adjusts entity-type levels for appropriate granularity in the pattern groups. Using (meta) pattern matching in the corpus, tuples with entities and relation phrases can be extracted.

3.3.2 Clause-based methods

4. TRUTH DISCOVERY IN INFORMATION EXTRACTION
For most truth discovery approaches, they assume the inputs are structured data. Information extraction deals with unstructured text data that are more noisy but also brings more information (e.g., uncertainty, evidence). For application of truth discovery in information extraction, the inaccurate information can come from two layers: the text corpora and the information extractor, which brings new challenges to source weight estimation. In this section, we discuss several representative work of incorporating truth discovery in information extraction.

4.1 Attention Mechanism
Attention mechanism is originally inspired by visual attention studies in neuroscience: many animals focus on specific parts of their visual object to give adequate responses. This principal is recently adopted in deep learning for natural language processing (NLP). An attention model will focus on specific part of the input an give it a larger weight in computing the final output. This general idea is similar to truth discovery that considers reliability scores of each source to infer the final output. The inaccurate information of the attention model comes from the text corpora.

A formal definition of the attention model is defined as follows:

**Definition 6.** An attention model is a method that takes $n$ arguments $x_1, ..., x_n$ and a context $c$. It returns a vector $z$, which is a weighted arithmetic mean of $x_i, i \in \{1, ..., n\}$, and the weights $w_i, i \in \{1, ..., n\}$ are chosen according the relevance of each $x_i$ given the context $c$.

In a deep neural network model for NLP, the arguments $x_i$ could be the hidden states $h_i$ of the model, the context $c$ could be the input sentence and the returned vector $z$ could be a representation of the output sentence.

4.1.1 Att-ChemdNER
Att-ChemdNER [23] is an attention-based BiLSTM-CRF model for document-level chemical named entity recognition. The base model is a BiLSTM-CRF model, which is similar to that of Lample et al. [15]. A detailed explanation of the BiLSTM-CRF model is given in Section 3.1.1. However, this base model works on the sentence level and suffers
from the tagging inconsistency problem. Att-ChemdNER leverages document-level global information obtained by attention mechanism to enforce tagging consistency across multiple instances of the same token in a document. For an input document $D = (X_1, ..., X_m)$ consisting of $m$ sentences, each sentence is expressed as $X = (x_1, ..., x_n)$ where $n$ is the number of words in a sentence. We define $N$ as the number of words in the document. Like the BiLSTM-CRF model, the character and word embeddings are first given as input into a BiLSTM layer. In the attention model, a character and word embeddings are first given as input into a BiLSTM layer. In the attention model, the character and word embeddings are first given as input into a BiLSTM layer. In the attention layer, a new attention layer is added on top of the BiLSTM layer to capture the word attentions at the document level. In the attention layer, an attention matrix $A$ is introduced to calculate the similarity between the current word and all words in the document. Each attention weight $\alpha_{i,j}$ in the attention matrix is defined as:

$$
\alpha_{i,j} = \frac{\exp(score(x_i, x_j))}{\sum_k \exp(score(x_i, x_k))}.
$$

Here the score is referred to as an alignment function which they defined four alternatives (manhattan distance, euclidean distance, cosine distance and perceptron). Then a document level global vector $g_t$ is computed as a weighted sum of each BiLSTM output $h_i$:

$$
g_t = \sum_{i=1}^{N} \alpha_{i,j} h_j.
$$

Next, the document-level global vector and the BiLSTM output of the target word will be concatenated as a vector $[g_t; h_j]$ and fed into the CRF layer for the output label prediction.

### 4.1.2 CNN/PCNN+ATT

CNN/PCNN+ATT [19] is a neural relation extraction model with selective attention over instances. The base model is a CNN/PCNN model, which takes the word/word+position embeddings as input into a CNN model. Similar to the above Att-ChemdNER model, the attention model is employed to enforce document-level consistency of relations between each instance pair.

Suppose there is a set $S = \{x_1, ..., x_n\}$ containing $n$ sentences of the entity pair (head, tail). The selective attention $\alpha_i$ is defined as:

$$
\alpha_{i,j} = \frac{\exp(e_i)}{\sum_k \exp(e_k)},
$$

where $e_i$ is referred as a query-based function which scores how well the input sentence $x_i$ and the predict relation $r$ matches. Then the set vector $s$ is computed as the weighted sum of the sentence vectors:

$$
s = \sum_{i=1}^{N} \alpha_i x_i.
$$

Here the $\alpha_i$ is the weight of each sentence vector $x_i$. The vector $s$ is used as the representation of the entity pair (head, tail) for relation prediction.

### 4.2 Heterogeneous Supervision

Most existing methods for relation extraction are fully supervised methods that rely on human annotated training data, which are costly and time-consuming. To overcome this limitation, REHESSION [21] utilizes annotated data from different sources, e.g., knowledge base and domain heuristics. These annotations are referred to as heterogeneous supervision, and are often conflict with each other. REHESSION employs a probabilistic graphical model to infer the truth label from heterogeneous supervision. This general idea is also similar to truth discovery with the inaccurate information comes from the text corpora.

For a corpus $D$ with detected entities, we refer to its relation mentions as $C = \{(e_{1,1}, e_{1,2}, d), d \in D\}$. The corpus is annotated by heterogeneous supervision in the form of labeling function $\Lambda = \{\lambda_1, ..., \lambda_M\}$. The annotations generated by $\Lambda$ is referred as $O = \{o_{c,i} | \lambda_i$ generate annotation $o_{c,i}$ for $c \in C\}$. We define relation mentions annotated by $\Lambda$ as $C_\Lambda$ and relation mentions without annotation as $C_o$. The goal is to annotate each relation mention with a relation type from the relation type set $R = \{r_1, ..., r_K\}$.

The relation mentions $c \in C$ are characterized by abundant lexical features $f_c$. Each relation mention $c$ is first represented as bag-of-features. After learning text feature embeddings $v_c$, a direct mapping $g$ is learned to map text feature representations to relation mention representations:

$$
z_c = g(f_c) = \tanh(W \cdot \frac{1}{f_{c}} \sum_{f_i \in f_c} v_{i}).
$$

Here $W$ is a $n_x \times n_v$ matrix where $n_x$ is the dimension of relation mention embeddings and $n_v$ is the dimension of text feature embeddings.

The truth label discovery is modeled as a probabilistic graphical model shown in Figure 3. Here $z_c$ is the learned representation of $c \in C_i$. Assume the indicator of whether $c$ belongs to $S_i$, i.e., $s_{c,i} = \delta, c \in S_i$ is first generated based on context representation:

$$
p(s_{c,i} = 1 | z_c, 1_i) = p(c \in S_i) = \sigma(z_c^T 1_i).
$$

Then the correctness of annotation $o_{c,i}$, $\rho_{c,i} = \delta(o_{c,i} = o^*_{c,i})$ could be generated.

![Figure 3: The probabilistic graphical model of truth label discovery in REHESSION.](image)

With the inferred truth label $o^*_{c,i}$, the relation extraction model can be trained as a multi-class classifier. Specifically, relation type is modeled as minimizing the KL-divergence between $p(r_i | z_c)$ and $p(r_i | o^*_{c,i})$.

The final model is a joint optimization problem of three components: modeling relation mention, truth label discovery and modeling label type. An gradient descendent algorithm is used for parameter updating.
4.3 Multi-dimensional Truth Discovery

The attention mechanism and heterogeneous supervision only consider that the inaccurate information comes from the text corpora. However, in information extraction, the inaccurate information can come from both the text corpora and the information extractor. Yu et al. [56] propose a more general framework to integrate IE and truth discovery. They present an unsupervised multi-dimensional truth finding model (MTM) which incorporates signals from multiple sources, multiple systems and multiple pieces of evidence by knowledge graph construction through multi-layer deep linguistic analysis.

The task of Slot Filling Validation (SFV) is taken to demonstrate the effectiveness of the MTM model.

DEFINITION 7. The Slot Filling task is to collect from a large-scale multi-source corpus the values (slot fillers) for certain attributes (slot types) of a query entity. The combination of query entity, slot type and slot filler is called a claim. Along with each claim, each system also provide detailed context sentences as evidence which supports the claim. A response is a claim-evidence pair and is correct if and only if the claim is true and the evidence supports it.

The general principles of truth discovery under the SF setting are:

- A trusted source always supports true claims by giving convincing evidence.
- A good system tends to extract trustworthy responses from trusted sources.

The MTM model constructs a heterogeneous network for source, system and response. Consider a set of responses \( R = \{r_1, ..., r_m\} \) extracted from a set of sources \( S = \{s_1, ..., s_n\} \) and provided by a set of systems \( T = \{t_1, ..., t_l\} \). Let edge weight matrices be \( W^r_{m \times n} = \{w^r_{ij}\} \) and \( W^w_{m \times t} = \{w^w_{ij}\} \). A link \( w^r_{ij} = 1 \) is generated between \( r_i \) and \( s_j \) when response \( r_i \) is extracted from source \( s_j \), and a link \( w^w_{ij} = 1 \) is generated between \( r_i \) and \( t_k \) when response \( r_i \) is provided by system \( t_k \).

The credibility of the source, system and response are first initialized and then propagated through the heterogeneous network until converge, i.e. the change of response credibility is below a minimum threshold. During credibility propagation, two heuristics are explored in MTM:

- HEURISTIC 1: A response is more likely to be true if derived from many trustworthy sources. A source is more likely to be trustworthy if many responses derived from it are true.
- HEURISTIC 2: A response is more likely to be true if it is extracted by many trustworthy systems. A system is more likely to be trustworthy if many responses generated by it are true.

For each source \( s_i \), we initialize its credibility score \( c^t(s_i) \) uniformly as \( \frac{1}{m} \), where \( n \) is the number of sources. For each system \( t_i \), we initialize its credibility score \( c^r(t_i) \) based on their interactions on the predicted responses with the TextRank [25] algorithm. For each response (claim-evidence pair), we initialize its credibility score \( c^r(r) \) based on rich linguistic indicators. A propagation method is designed for the heterogeneous network with the three object types. Let \( W^r \) and \( W^w \) be the transpose of \( W^r_{m \times n} \) and \( W^w_{m \times t} \). We can compute the transition probability of \( s_i \) reaches \( r_j \) in the next iteration as a normalized weight \( p^r_{ij} = \frac{w^r_{ij}}{\sum_{k \in S} w^r_{ik}} \) such that \( \sum_{r_j \in R} p^r_{ij} = 1 \). Similarly, we can compute \( p^w_{jk} \) and \( p_{lj} \). Starting with the initialized source, the updated score considers both the initial score \( c^t(s) \) and the propagation from connected responses:

\[
    c(s_i) = (1 - \lambda_{sr})c^t(s_i) + \lambda_{rs} \sum_{r_j \in R} p^r_{ij} c(r_j). \quad (1)
\]

Similarly, the updated score of each system considers both the initial score \( c^r(t) \) and the propagation from connected responses:

\[
    c(t_k) = (1 - \lambda_{rt})c^r(t_k) + \lambda_{rt} \sum_{r_j \in R} p^r_{jk} c(r_j). \quad (2)
\]

The updated score of each response is influence by both the connected sources and the connected systems:

\[
    c(r_j) = (1 - \lambda_{sr} - \lambda_{rt})c^r(r_j) + \lambda_{sr} \sum_{s_i \in S} p^w_{ij} c(s_i) + \lambda_{rt} \sum_{t_k \in T} p_{lj} c(t_k). \quad (3)
\]

Here \( \lambda_{sr} \), \( \lambda_{rt} \), \( \lambda_{sr} \) and \( \lambda_{rt} \in [0, 1] \) are parameters that control the preference for the propagation over initial scores for each type. Experiments on a challenging SFV task shows that this unsupervised truth finding framework can even outperform a supervised learning approach by a large margin (~10% increase in F1).

4.4 Truth Existence in Truth Discovery

Previous truth discovery models infer the most accurate answer from conflicting sources. However, in some cases, there exists questions for which the true answers are excluded from the candidate answers provided by all sources. Zhi et al. [60] propose a unsupervised truth existence model (TEM) which simultaneously infers truth as well as source quality without any a priori training involving ground truth answers. The SFV task is also taken to demonstrate the effectiveness of the TEM model.

Let \( E \) denote an empty answer when a source keeps silent to a question. Let \( Q = \{q_1, ..., q_M\} \) be the set of questions where \( M \) is the total number of questions. Let \( S = \{s_1, ..., s_N\} \) be the set of sources where \( N \) is the total number of sources. Let \( D_i = \{d_{i1}, ..., d_{IN}\} \) be the set of distinct non-empty candidate answers to question \( q_i \) provided by all sources in \( S \). Let \( A_i = \{a_{i1}, ..., a_{iN}\} \) be the set of observed answers to question \( q_i \) provided by all sources in \( S \). Each answer \( a_{ij} \) can take an empty answer \( E \) or a non-empty answer in \( D_i \). Let \( T = \{t_1, ..., t_M\} \) be the set of truths where each \( t_i \) associated with \( q_i \) can either be an empty answer \( E \) or a non-empty answer in \( D_i \), we define a has-truth question as the question whose truth is in the non-empty candidate answer set \( D_i \), and a no-truth question as the question whose truth is not in \( D_i \). The truth of a no-truth questions is the empty answer \( E \).

TEM is modeled as a probabilistic graphical model shown in Figure 4. For each source \( s_i \in S \), the source quality is defined by three measures: silent rate, false spoken rate and true spoken rate. We define silent rate, false spoken rate and true spoken rate by probabilities, denoted by \( \phi_j^{(1)}, \phi_j^{(2)} \) and
Silent rate is the probability that \( s_j \) keeps silent when a question has truth, i.e., \( \phi_j^{(3)} = P(a_{ij} = E \mid t_i = d_{in}, t_i \neq E) \).
False spoken rate is the probability it makes mistakes on either has-truth or no-truth questions, i.e., \( \phi_j^{(2)} = P(a_{ij} \neq d_{in} \mid t_i = d_{in}, t_i \neq E) = P(a_{ij} \neq E \mid t_i = E) \). True spoken rate is the probability to provide a trustworthy answer when a question has truth, i.e., \( \phi_j^{(1)} = P(a_{ij} = d_{in} \mid t_i = d_{in}, t_i \neq E) \).

The relationship between these three source quality measures is that \( \phi_j^{(1)} + \phi_j^{(2)} + \phi_j^{(3)} = 1 \). For each source \( s_j \in S \), we generate the source quality vector \( \phi_j = (\phi_j^{(1)}, \phi_j^{(2)}, \phi_j^{(3)}) \) from a Dirichlet distribution with hyper-parameter \( \alpha = (\alpha_1, \alpha_2, \alpha_3) \), i.e., \( \phi_j \sim \text{Dirichlet}(\alpha) \).

For each question \( q_i \in Q \), we define the prior of truth as \( \eta_i = (\eta_{i0}, \eta_{i1}, ..., \eta_{iN}) \), where \( \eta_{in}, n = 1, ..., N_i \) is the probability of truth \( t_i \) being one of the non-empty candidate answer \( d_{in} \) and \( \eta_{i0} \) is the probability of \( q_i \) being a no-truth question.

Figure 4: The probabilistic graphical model of TEM.

The observed answers to \( q_i \) provided by \( N \) sources in \( S \) are denoted by \( A_i \). The probability of observing \( A_i \) given source quality \( \phi s \) is:

\[
P(A_i \mid \phi s, \eta) = \sum_{n=1}^{N_i} \eta_{in} P(A_i \mid t_i = d_{in}, \phi s) + \eta_{i0} P(A_i \mid t_i = E, \phi s).
\]

TEM is formulated as a Maximum Likelihood Estimation (MLE) problem and the Expectation-Maximization (EM) algorithm is used to jointly maximize the source quality and truth.

5. DISCUSSION
In this survey, we discussed the current methods in truth discovery including iterative methods and graphical model based methods. We also compared these two methods from different perspectives. The application of truth discovery in information extraction gains great attention recently as each fact can often be extracted by various systems from various sources. We discussed some representative work of applying truth discovery in information extraction including entity recognition, relation extraction and slot filling. More efforts are in high demand to explore truth discovery in extracting reliable information from unstructured text data.

6. REFERENCES


